



Sicherheit der Apple-Plattformen

Mai 2022



Inhaltsverzeichnis

Sicherheit der Apple-Plattformen	5
Einführung in die Sicherheit der Apple-Plattformen	5
Hardware-sicherheit und Biometrie	8
Hardware-sicherheit – Übersicht	8
Sicherheit bei Apple-SoCs	10
Secure Enclave	11
Face ID und Touch ID	22
Hardwareabschaltung des Mikrofons	32
Express-Karten mit Energiereserve	33
Systemsicherheit	34
Systemsicherheit – Übersicht	34
Sicheres Starten	35
Sicherheit des Signed System Volume in iOS, iPadOS und macOS	64
Sichere Softwareaktualisierungen	66
Integrität des Betriebssystems	68
Zusätzliche macOS-Funktionen für die Systemsicherheit	71
Systemsicherheit bei watchOS	84
Generieren von Zufallszahlen	89
Apple Security Research Device	90
Verschlüsselung und Datensicherheit	92
Verschlüsselung und Datensicherheit – Übersicht	92
Gerätecodes und Passwörter	93
Datensicherheit	96
FileVault	112
Schutz persönlicher und personenbezogener Daten durch Apple	116
Digitales Signieren und Verschlüsseln	120

Sicherheit bei Apps	122
Sicherheit bei Apps – Übersicht	122
Sicherheit bei Apps für iOS und iPadOS	124
Sicherheit bei Apps für macOS	130
Sicherheitsfunktionen in der App „Notizen“	136
Sicherheitsfunktionen in der App „Kurzbefehle“	138
Sicherheit bei Diensten	139
Sicherheit bei Diensten – Übersicht	139
Apple-ID und verwaltete Apple-ID	139
iCloud	142
Code- und Passwortverwaltung	154
Apple Pay	166
Verwenden der Apple Wallet	182
iMessage	194
Sicherheit bei Apple Messages for Business	198
FaceTime-Sicherheit	199
Wo ist?	200
Continuity (Integration)	204
Netzwerksicherheit	208
Netzwerksicherheit – Übersicht	208
TLS-Sicherheit	208
IPv6-Sicherheit	211
VPN-Sicherheit (Virtual Private Network)	212
WLAN-Sicherheit	213
Bluetooth-Sicherheit	217
Sicherheit der Ultrabreitbandtechnologie von iOS	219
Sicherheit der Gesamtauthentifizierung (Single Sign-On)	219
AirDrop-Sicherheit	221
Sicherheit bei WLAN-Passwortfreigabe auf iPhone und iPad	223
Firewall-Sicherheit bei macOS	223
Sicherheit bei Developer Kits	224
Sicherheit bei Developer Kits – Übersicht	224
Sicherheit beim HomeKit	224
Sicherheit beim SiriKit für iOS, iPadOS und watchOS	231
Sicherheit beim DriverKit für macOS	232
Sicherheit beim ReplayKit für iOS und iPadOS	233
Sicherheit beim ARKit für iOS und iPadOS	235

Sichere Geräteverwaltung	236
Sichere Geräteverwaltung – Übersicht	236
Sicherheit beim Kopplungsmodell für iPhone und iPad	237
Mobilgeräteverwaltung (Mobile Device Management)	238
Sicherheit bei Apple Configurator	248
Sicherheit bei der Bildschirmzeit	249
Glossar	251
Dokumentrevisionen	257
Dokumentrevisionen	257
Copyright	264

Sicherheit der Apple-Plattformen

Einführung in die Sicherheit der Apple-Plattformen

Apple implementiert die Sicherheitsfunktionalität direkt im Kern seiner Plattformen. Dank der Erfahrung, die Apple aus der Entwicklung des weltweit modernsten Betriebssystems für Mobilgeräte mitbringt, konnten spezielle Sicherheitsarchitekturen konzipiert werden, die ganz genau auf die spezifischen Anforderungen von Mobilgeräten, Smartwatches, Desktop-Computern und Home-Geräten abgestimmt sind.

Bei jedem Apple-Gerät sind *Hardware*, *Software* und *Dienste* in einer Weise kombiniert, die ein hohes Maß an Sicherheit im Betrieb und ein transparentes Benutzererlebnis mit dem Ziel gewährleistet, persönliche Daten zuverlässig zu schützen. So bilden zum Beispiel von Apple entwickelte Chips und Sicherheitshardware das Fundament für kritische Sicherheitsfunktionen. Darüber hinaus sorgen Softwareschutzmechanismen für den sicheren Betrieb des Betriebssystems und der Apps anderer Anbieter. Schließlich unterstützen Dienste einen Mechanismus für die sichere und zeitnahe Aktualisierung von Softwareprodukten, bilden das Fundament für ein sicheres App-Ökosystem und sichern Kommunikations- und Zahlungsvorgänge ab. Daher bieten Apple-Geräte nicht nur Schutz für das Gerät an sich und die darauf befindlichen Daten, sondern für das gesamte Ökosystem – inklusive aller Aktivitäten, die ein Benutzer lokal, in Netzwerken oder mithilfe wichtiger Internetdienste ausführt.

Beim Design unserer Produkte legen wir auf einfache, intuitive Bedienung und auf Leistungsfähigkeit ebenso großen Wert wie auf Sicherheit. Zentrale Sicherheitsfunktionen wie die hardwarebasierte Verschlüsselung können nicht versehentlich deaktiviert werden. Andere Funktionen wie Face ID und Touch ID optimieren die Benutzererfahrung: Mit ihnen kann ein Benutzer noch einfacher und intuitiver zur Sicherheit und zum Schutz seines Geräts beitragen. Da viele dieser Funktionen standardmäßig aktiviert sind, müssen Benutzer oder IT-Abteilungen keine aufwendige Konfiguration vornehmen.

Dieses Dokument enthält detaillierte Informationen darüber, wie Sicherheitstechnologien und -funktionen in Apple-Plattformen implementiert sind. Organisationen erhalten darin auch Tipps und Hinweise dazu, wie sie die Sicherheitstechnologien und -funktionen der Apple-Plattformen mit ihren eigenen Richtlinien und Verfahren kombinieren können, um die für sie geltenden Sicherheitsanforderungen zu erfüllen.

Das Dokument ist in die folgenden Themenbereiche unterteilt:

- **Hardwaresicherheit und Biometrie:** Hier werden die Chips und Hardwarekomponenten beschrieben, die das Fundament für die Sicherheit von Apple-Geräten bilden. Dazu gehören Apple Chips, die Secure Enclave, kryptografische Engines, Face ID und Touch ID.
- **Systemsicherheit:** Unter diesen Themenbereich fallen integrierte Hard- und Softwarefunktionen, die das sichere Starten (Booten) und Aktualisieren und den sicheren Betrieb der Apple-Betriebssysteme gewährleisten.
- **Verschlüsselung und Datensicherheit:** Hier werden die Architektur und das Design beschrieben, mit denen Benutzerdaten geschützt werden, wenn ein Gerät verloren geht oder gestohlen wird oder wenn unbefugte Dritte oder Prozesse versuchen, ein Gerät zu verwenden oder es zu modifizieren.
- **Sicherheit bei Apps:** Unter dieses Themengebiet fallen Softwareprodukte und Dienste, die ein sicheres Ökosystem für Apps bereitstellen, in dem Apps sicher ausgeführt werden können, ohne dass die Integrität der Plattform beeinträchtigt wird.
- **Sicherheit bei Diensten:** Hier werden die Apple-Dienste beschrieben, die bei der Identifizierung, bei der Passwortverwaltung, bei Zahlungs- und Kommunikationsvorgängen und beim Auffinden eines verloren gegangenen Geräts zum Einsatz kommen.
- **Netzwerksicherheit:** Hier werden Netzwerkprotokolle nach Industriestandards beschrieben, die eine sichere Authentifizierung und die Verschlüsselung von Daten bei der Übertragung ermöglichen.
- **Sicherheit bei Kits für Entwickler:** In diesem Teil werden die „Kit“-Frameworks für die sichere, private Verwaltung von Daten der eigenen Gesundheit (Health) und des eigenen Zuhauses (Home) sowie Erweiterungen der Funktionalitäten von Apple-Geräten und Apple-Diensten beschrieben, die für Apps anderer Anbieter zur Verfügung stehen.
- **Sichere Geräteverwaltung:** Hier werden Methoden vorgestellt, mit denen Apple-Geräte verwaltet werden können und die Verwendung eines Geräts durch unbefugte Dritte verhindert und ein gestohlenen oder verlorenes Gerät per Fernbefehl gelöscht werden kann.

Der Sicherheit verpflichtet

Mit Nachdruck bemüht sich Apple darum, die Kundendaten zu schützen und bietet daher modernste Datenschutz- und Sicherheitstechnologien für den Schutz der Privatsphäre und persönlichen Daten von Benutzern sowie umfassende Methoden zum Schutz organisationseigener Daten in Unternehmensumgebungen. Apple zahlt im Rahmen des Programms „Apple Security Bounty“ Prämien an Entwickler aus, die durch ihre Arbeit dazu beitragen, Schwachstellen und Sicherheitslücken aufzudecken. Näheres zum Programm und den Prämienkategorien ist unter <https://developer.apple.com/security-bounty/> verfügbar.

Apple verfügt über ein dediziertes Sicherheitsteam, das Unterstützung für alle Apple-Produkte bietet. Das Team stellt auf die Sicherheit ausgelegte Auditing- und Testszenarien sowohl für Produkte bereit, die sich noch in der Entwicklung befinden, als auch für Produkte, die bereits veröffentlicht und ausgeliefert wurden. Zum Aufgabengebiet des Apple-Teams gehören außerdem die Bereitstellung von Sicherheitswerkzeugen, die Veranstaltung von Trainings und Schulungen und die aktive Überwachung und Verfolgung von Bedrohungen und von Berichten zu neuen Sicherheitslücken. Apple ist Mitglied von FIRST (Forum of Incident Response und Security Teams – weitere Informationen sind hier verfügbar <https://www.first.org>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt).

Apple wird auch künftig die Grenzen dessen, was im Bereich Sicherheit und Datenschutz und Privatsphäre machbar und möglich ist, immer weiter verschieben. Für noch mehr Effizienz und Sicherheit verwenden Apple-Geräte selbst designte Chips in der gesamten Produktlinie von der Apple Watch über das iPhone und iPad bis hin zu Macs mit T2 Security Chip und Apple Chips. Apple Chips bilden beispielsweise das Fundament für sicheres Starten, Face ID und Touch ID sowie die Datensicherheit. Darüber hinaus können die Sicherheitsfunktionen von Geräten mit Apple Chips dabei helfen, häufig verwendete Typen von Cyberattacken zu verhindern. Zu diesen Sicherheitsfunktionen gehören der Kernel-Integritätsschutz (Kernel Integrity Protection), Pointer Authentication Codes und schnelle Berechtigungseinschränkungen (Fast Permission Restrictions). So kann selbst in Fällen, in denen Schadcode bereits ausgeführt wurde, der potenzielle Schaden drastisch reduziert werden.

Damit die umfassenden Sicherheitsfunktionen unserer Plattformen optimal genutzt werden, sollten Organisationen ihre IT- und Sicherheitsrichtlinien dahingehend überprüfen, dass alle Ebenen der Sicherheitstechnologien dieser Plattformen zum Einsatz kommen.

Weitere Informationen, wie Probleme an Apple berichtet und Sicherheitsmitteilungen abonniert werden können, enthält der Artikel: [Sicherheits- oder Datenschutzwachstelle melden](#).

Apple ist der Überzeugung, dass Datenschutz ein Grundrecht ist, und hat zahlreiche Optionen und Kontrollen integriert, die es den Benutzern erlauben zu entscheiden, wie und wann Apps persönliche Daten nutzen und welche Daten genutzt werden dürfen. Weitere Informationen über die Strategien von Apple bezüglich Datensicherheit und Datenschutz, die entsprechenden Kontroll- und Steuerungsmechanismen auf Apple-Geräten sowie die Datenschutzrichtlinie von Apple sind auf der Website <https://www.apple.com/de/privacy> zu finden.

Hinweis: Sofern nichts anderes vermerkt ist, gelten die Informationen in diesem Dokument für die folgenden Betriebssystemversionen: iOS 15.4, iPadOS 15.4, macOS 12.3, tvOS 15.4 und watchOS 8.5.

Hardwaresicherheit und Biometrie

Hardwaresicherheit – Übersicht

Für die Sicherheit von Software ist entscheidend, dass sie auf Hardware mit eingebauter Sicherheit ausgeführt wird. Dies ist der Grund, weshalb bei Apple-Geräten (mit iOS, iPadOS, macOS, tvOS und watchOS) die Sicherheitsfunktionen bereits im Chip verankert sind. Dazu zählen eine CPU, die die Rechenleistung für die Sicherheitsfunktionen des Systems bereitstellt, und für Sicherheitsfunktionen dedizierte Chips. Sicherheitsfokussierte Hardware verfolgt das Prinzip, begrenzte und diskret definierte Funktionen zu unterstützen, um Angriffsflächen zu minimieren. Derartige Komponenten beinhalten ein Boot-ROM, das einen Hardware-Vertrauensanker für den sicheren Startvorgang, dedizierte AES-Engines für effiziente und sichere Verschlüsselung und Entschlüsselung sowie eine Secure Enclave bildet. Die *Secure Enclave* ist ein System on Chip (SoC) und kommt bei allen iPhone-, iPad-, Apple Watch-, Apple TV- und HomePod-Geräten sowie bei allen Mac-Computern mit Apple Chips und Mac-Computern mit Apple T2 Security Chip zum Einsatz. Die Secure Enclave folgt denselben Designprinzipien wie das SoC – sie enthält ihr eigenes, diskretes Boot-ROM und eine AES-Engine. Die Secure Enclave stellt auch das Fundament für die sichere Erstellung und Aufbewahrung der nötigen Schlüssel für die Verschlüsselung gespeicherter Daten dar und schützt und bewertet die biometrischen Daten, die von Face ID und Touch ID genutzt werden.

Die Speicherverschlüsselung muss schnell und effizient sein. Dabei dürfen jedoch die Daten (oder das *Verschlüsselungsmaterial*) für die Etablierung der kryptografischen Verschlüsselungsbeziehungen nicht offengelegt werden. Die AES-Hardware-Engine löst dieses Problem, indem sie beim *Schreiben und Lesen von Dateien* schnelle Inline-Verschlüsselung und -Entschlüsselung anwendet. Ein spezieller Kanal der Secure Enclave stellt der AES-Engine die nötigen Verschlüsselungsmaterialien zur Verfügung, ohne dass diese Informationen dabei gegenüber dem Anwendungsprozessor (oder der CPU) oder dem Betriebssystem im Allgemeinen offengelegt werden. Auf diese Weise wird sichergestellt, dass die Datensicherheits- und FileVault-Technologien von Apple die benutzereigenen Dateien schützen, ohne langlebige Verschlüsselungsschlüssel offenzulegen.

Apple hat einen sicheren Startvorgang (Booten) konzipiert, um die unterste Ebene der Software vor Manipulationen zu schützen und nur das Laden von vertrauenswürdiger Betriebssystemsoftware von Apple beim Start zu erlauben. Der Grundstock für das sichere Starten ist ein unveränderlicher Code namens „Boot-ROM“, der bereits bei der Herstellung des Apple-SoC generiert und als *Hardware-Vertrauensanker* bezeichnet wird. Bei Mac-Computern mit einem T2-Chip beginnt die Vertrauenskette für das sichere Starten von macOS mit dem T2-Chip. (Sowohl der T2-Chip als auch die Secure Enclave führen auch eigene sichere Startprozesse aus und nutzen dabei einen jeweils eigenen, separaten Boot-ROM – vollkommen analog zum sicheren Startvorgang von Chips aus der A-Reihe und der M1-Familie.)

Die Secure Enclave verarbeitet außerdem Gesichts- und Fingerabdruckdaten der Face ID- und Touch ID-Sensoren auf Apple-Geräten. Das ermöglicht die sichere Authentifizierung, wobei aber die biometrischen Daten des Benutzers privat und vertraulich behandelt werden. Dies gibt dem Benutzer außerdem die Möglichkeit, längere und komplexere Codes und Passwörter zu verwenden und sich dennoch in vielen Situationen schnell authentifizieren zu können, wenn er zum Beispiel auf Inhalte zugreifen oder Produkte erwerben möchte.

Sicherheit bei Apple-SoCs

Die von Apple entwickelten Chips bieten eine gemeinsame Architektur für alle Apple-Produkte und finden nicht nur in Mac-Computern, sondern auch in iPhone-, iPad-, Apple TV- und Apple Watch-Geräten Verwendung. Schon seit mehr als einem Jahrzehnt konzipiert und optimiert das für Apple Chips zuständige Designteam die Apple-SoCs (Systems on Chip). Das Ergebnis ist eine skalierbare Architektur, die für alle Geräte entwickelt wurde und im Hinblick auf Sicherheitsaspekte industrieweit führend ist. Diese gemeinsame Basis für Sicherheitsfunktionen ist nur für Unternehmen erreichbar, die eigene Chips passend zur Software entwickeln.

Die Apple Chips wurden mit speziellem Augenmerk auf die folgenden, für die Systemsicherheit relevanten Funktionen und Merkmale konzipiert und gefertigt:

Funktion	A10	A11, S3	A12, S4	A13, S5	A14, A15, S6, S7	M1-Familie
Kernel-Integritätsschutz	✓	✓	✓	✓	✓	✓
Schnelle Berechtigungseinschränkungen		✓	✓	✓	✓	✓
Integritätsschutz des System-Coprozessors			✓	✓	✓	✓
PACs (Pointer Authentication Codes)			✓	✓	✓	✓
Page Protection Layer (PPL)		✓	✓	✓	✓	Siehe Anmerkung unten.

Hinweis: Die PPL (Page Protection Layer) setzt voraus, dass die Plattform *ausschließlich* signierten und vertrauenswürdigen Code ausführt. Dieses Sicherheitsmodell ist in macOS nicht anwendbar.

Apple Chips aktivieren außerdem explizit die nachfolgenden Datensicherheitsfunktionen.

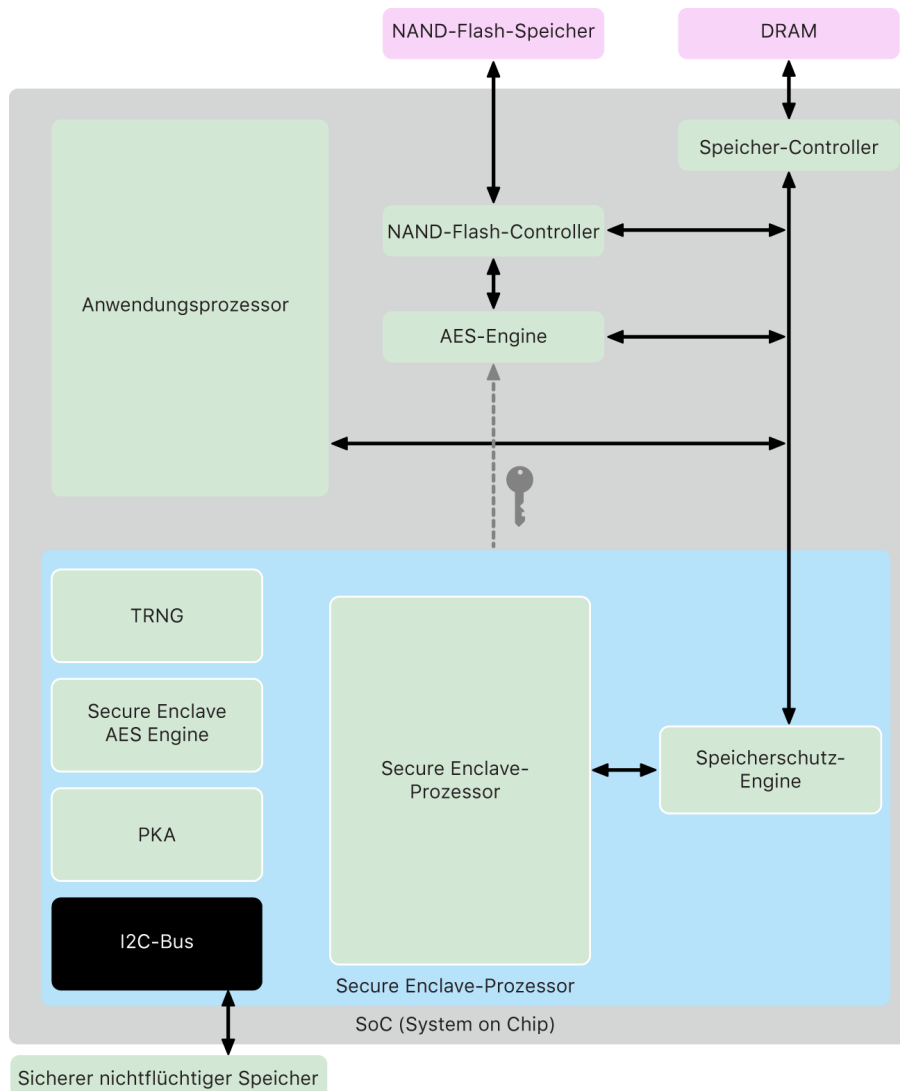
Funktion	A10	A11, S3	A12, S4	A13, S5	A14, A15, S6, S7, M1-Familie
Sealed Key Protection (SKP)	✓	✓	✓	✓	✓
recoveryOS – Datenschutz bei allen Datensicherheitsklassen	✓	✓	✓	✓	✓
Alternative Startmodi des DFU-Modus, Diagnose und Aktualisierung – Datenschutz bei Klasse A, B und C			✓	✓	✓

Secure Enclave

Die Secure Enclave ist ein dediziertes sicheres Subsystem in den aktuellen Versionen von iPhone, iPad, iPod touch, Mac, Apple TV, Apple Watch und HomePod.

Übersicht

Die Secure Enclave ist ein dediziertes sicheres Subsystem, das bei Apple in die [SoCs \(System on Chip\)](#) integriert ist. Die Secure Enclave ist vom Hauptprozessor isoliert, was eine zusätzliche Sicherheitsebene mit sich bringt, und ist darauf ausgelegt, die Sicherheit sensibler Benutzerdaten selbst dann zu gewährleisten, wenn der Kernel des Anwendungsprozessors kompromittiert werden sollte. Sie folgt denselben Designprinzipien wie das SoC insgesamt – ein Boot-ROM, das einen Hardware-Vertrauensanker etabliert, eine AES-Engine für effiziente und sichere kryptografische Operationen und geschützter Speicher. Die Secure Enclave umfasst keinen Speicher, verfügt aber über einen Mechanismus zum sicheren Speichern von Informationen auf dem angeschlossenen Speicher, der unabhängig von dem vom Anwendungsprozessor und dem Betriebssystem verwendeten NAND-Flashspeicher ist.



Die Secure Enclave ist eine Hardwarefunktion der meisten Versionen von iPhone, iPad, Mac, Apple TV, Apple Watch und HomePod. Im Einzelnen sind dies folgende Geräte:

- iPhone 5s oder neuer
- iPad Air oder neuer
- MacBook Pro-Computer mit Touch Bar (2016 und 2017) und Apple T1-Chip
- Intel-basierte Mac-Computer mit Apple T2 Security Chip
- Mac-Computer mit Apple Chips
- Apple TV HD (oder neuer)
- Apple Watch Series 1 oder neuer
- HomePod und HomePod mini

Secure Enclave-Prozessor

Der Prozessor der Secure Enclave stellt die hauptsächliche Rechenleistung für die Secure Enclave bereit. Um die strikte Isolation zu gewährleisten, wird der Prozessor der Secure Enclave ausschließlich von der Secure Enclave genutzt. Dies hilft bei der Abwehr von Seitenkanalattacken, die den Ansatz verfolgen, dass die Schadsoftware denselben Ausführungskern wie die Zielsoftware verwendet, die im Fokus des Angriffs steht.

Der Prozessor der Secure Enclave läuft auf einer von Apple angepassten Version des L4-Mikrokernels. Er ist so konzipiert, dass er bei einer niedrigeren Taktrate effizient arbeitet, was wiederum seinem Schutz vor Clock- und Power-Angriffen dient. Der Secure Enclave-Prozessor verfügt ab A11 und S4 über eine Memory Protection Engine und einen verschlüsselten Speicher mit Anti-Replay-Funktionen, sicheres Starten, einen dedizierten Generator für Zufallszahlen und eine eigene AES-Engine.

Engine für den Speicherschutz (Memory Protection Engine)

Die Secure Enclave wird aus einer dedizierten Region des DRAM-Gerätespeichers betrieben. Der Speicher der Secure Enclave ist durch mehrere Sicherheitsebenen vom Anwendungsprozessor isoliert.

Beim Starten des Geräts generiert der Boot-ROM der Secure Enclave einen zufälligen temporären Speichersicherheitsschlüssel für die Memory Protection Engine (Speicherschutz-Engine). Bei jedem Schreibvorgang, den die Secure Enclave in der für sie dedizierten Speicherregion vornimmt, verschlüsselt die Memory Protection Engine den Speicherblock mittels AES im Mac XEX-Modus (xor-encrypt-xor) und berechnet ein CMAC-Authentifizierungs-Tag (Cipher-based Message Authentication Code) für den Speicher. Die Memory Protection Engine speichert diesen Authentifizierungs-Tag zusammen mit dem verschlüsselten Speicher. Wenn die Secure Enclave den Speicher liest, prüft die Memory Protection Engine den Authentifizierungs-Tag. Stimmt das Authentifizierungs-Tag überein, entschlüsselt die Memory Protection Engine den Speicherblock. Stimmt das Tag nicht überein, meldet die Memory Protection Engine einen Fehler an die Secure Enclave. Nach einem Speicherauthentifizierungsfehler akzeptiert die Secure Enclave Anfragen erst wieder, nachdem das System neu gebootet wurde.

Ab den SoCs der Typen Apple A11 und S4 übernimmt die Memory Protection Engine auch den Schutz des Secure-Enclave-Speichers vor Replay-Angriffen. Um ein Replay sicherheitskritischer Daten zu verhindern, speichert die Memory Protection Engine eine eindeutige und einmal genutzte Nummer, die als *Nonce* bezeichnet wird, für den Speicherblock zusammen mit dem Authentifizierungs-Tag. Die Nonce ist ein zusätzliches Optimierungselement für das CMAC-Authentifizierungs-Tag. Die Nonces aller Speicherblöcke werden mithilfe einer Integritätsstruktur geschützt, die ihren Ursprung im dedizierten SRAM innerhalb der Secure Enclave hat. Bei Schreibvorgängen *aktualisiert* die Memory Protection Engine die Nonce und jede Ebene der Integritätsstruktur bis hin zum SRAM. Bei Lesevorgängen *prüft* die Memory Protection Engine die Nonce und jede Ebene der Integritätsstruktur bis hin zum SRAM. Fehlende Übereinstimmungen bei Nonces werden auf ähnliche Weise behandelt wie Abweichungen bei Authentifizierungs-Tags.

Bei A14, A15, der M1-Familie und neueren SoCs von Apple unterstützt die Memory Protection Engine zwei temporäre Speichersicherheitsschlüssel. Der erste wird für private Daten verwendet, die an die Secure Enclave gesendet werden. Der zweite wird für Daten verwendet, die mit der Secure Neural Engine geteilt werden.

Die Memory Protection Engine arbeitet inline und in einer für die Secure Enclave transparenten Weise. Die Secure Enclave liest und beschreibt den Speicher, als handele es sich bei ihm um einen regulären unverschlüsselten DRAM, während ein Beobachter außerhalb der Secure Enclave nur die verschlüsselte und authentifizierte Version des Speichers zu sehen bekommt. Dies führt zu einem starken und robusten Speicherschutz ohne Abstriche an der Leistung oder der Komplexität der Software.

Boot-ROM der Secure Enclave

Die Secure Enclave umfasst einen eigenen dedizierten Boot-ROM. Ähnlich wie der Boot-ROM des Anwendungsprozessors ist auch der Boot-ROM der Secure Enclave unveränderlicher Code, der den Hardware-Vertrauensanker für die Secure Enclave darstellt.

Beim Starten des Systems weist iBoot der Secure Enclave eine dedizierte Speicherregion zu. Vor der Nutzung des Speichers initialisiert der Boot-ROM der Secure Enclave die Memory Protection Engine, um den kryptografischen Schutz für den geschützten Speicher der Secure Enclave bereitzustellen.

Im Anschluss daran sendet der Anwendungsprozessor das [sepOS](#)-Image an den Boot-ROM der Secure Enclave. Nachdem das sepOS-Image in den geschützten Speicher der Secure Enclave geladen wurde, prüft der Boot-ROM der Secure Enclave den kryptografischen Hash-Wert und die Signatur des Image. Auf diese Weise wird geprüft und sichergestellt, dass das sepOS-Image für die Ausführung auf dem Gerät autorisiert ist. Wenn das sepOS-Image für die Ausführung auf dem Gerät korrekt signiert ist, übergibt der Boot-ROM der Secure Enclave die Steuerung an sepOS. Wenn die Signatur ungültig ist, wehrt der Boot-ROM der Secure Enclave bis zum nächsten Zurücksetzen des Chips jeden weiteren Versuch ab, die Secure Enclave zu verwenden.

Bei Apple A10 und neueren SoCs verschließt der Boot-ROM der Secure Enclave einen Hash-Wert von sepOS in einem für diesen Zweck dedizierten Register. Der Public Key Accelerator verwendet diesen Hash-Wert für an das Betriebssystem gebundene Schlüssel („OS-bound keys“).

Boot-Monitor der Secure Enclave

Bei Apple A13 und neueren SoCs umfasst die Secure Enclave einen Boot-Monitor, um die Integrität des Hash-Werts des gestarteten sepOS-Image in noch stärkerem Maße zu gewährleisten.

Beim Starten des Systems wird durch die [SCIP \(System Coprocessor Integrity Protection\)](#)-Konfiguration des Secure Enclave-Prozessors verhindert, dass ein anderer Code als der Boot-ROM der Secure Enclave ausgeführt wird. Der Boot-Monitor verhindert, dass die Secure Enclave die SCIP-Konfiguration direkt modifiziert. Damit das geladene sepOS-Image ausführbar wird, sendet der Boot-ROM der Secure Enclave eine Anfrage an den Boot-Monitor, in der die Adresse und die Größe des geladenen sepOS enthalten sind. Beim Erhalt dieser Anfrage setzt der Boot-Monitor den Prozessor der Secure Enclave zurück, erstellt einen Hash-Wert für das geladene sepOS, aktualisiert die SCIP-Einstellungen so, dass die Ausführung des geladenen sepOS erlaubt wird, und startet die Ausführung unter Verwendung des neu geladenen Codes. Im weiteren Verlauf des Startvorgangs wird derselbe Vorgang immer dann verwendet, wenn neuer Code ausführbar gemacht wird. Jedes Mal aktualisiert der Boot-Monitor einen laufenden Hash-Wert des Startprozesses. Der Boot-Monitor bettet auch kritische Sicherheitsparameter in diesen laufenden Hash-Wert ein.

Nach Abschluss des Startvorgangs finalisiert der Boot-Monitor den laufenden Hash und sendet ihn an den Public Key Accelerator (PKA), um den betriebssystemgebundenen Schlüssel zu verwenden. Dieser Prozess ist so konzipiert, dass die Bindung von Betriebssystemschlüsseln nicht umgangen werden kann, selbst wenn es zu einer Schwachstelle im Boot-ROM der Secure Enclave kommen sollte.

True Random Number Generator

Der True Random Number Generator (TRNG) wird verwendet, um sichere Zufallsdaten zu generieren. Die Secure Enclave nutzt den TRNG, wann immer sie einen zufälligen kryptografischen Schlüssel, einen Zufallswert für einen Schlüssel oder eine sonstige Entropie generiert. Der TRNG basiert auf mehreren Ringoszillatoren; das Post-Processing erfolgt mittels CTR_DRBG (ein auf Blockchiffren im Zählermodus basierender Algorithmus).

Kryptografische Stammschlüssel

Die Secure Enclave umfasst eine Unique ID (UID) als kryptografischen Stammschlüssel. Die UID ist für jedes einzelne Gerät eindeutig und steht in keiner Beziehung zu irgendeiner anderen ID oder Kennung des Geräts.

Eine nach dem Zufallsprinzip generierte UID wird zur Fertigungszeit fest mit dem SoC verschmolzen. Bei SoCs ab A9 wird die UID zur Fertigungszeit durch den TRNG der Secure Enclave generiert und mit einem Softwareprozess in die Sicherungen geschrieben, der vollständig in der Secure Enclave ausgeführt wird. Dieser Prozess stellt sicher, dass die UID während der Herstellung nicht außerhalb des Geräts sichtbar ist, sodass weder Apple noch ein Apple-Lieferant Zugriff auf sie hat oder sie speichern kann.

sepOS verwendet die UID, um gerätespezifische Secrets (Geheimnisse) zu schützen. Durch die UID können Daten kryptografisch an ein bestimmtes Gerät gebunden werden. So enthält beispielsweise die Schlüsselhierarchie, die das Dateisystem schützt, die UID. Dies bewirkt, dass nicht auf die Dateien zugegriffen werden kann, wenn die Speicherchips physisch von einem Gerät in ein anderes verschoben werden. Andere gerätespezifische und geschützte Secrets sind zum Beispiel die Face ID- und die Touch ID-Daten. Auf Mac-Computern wird dieser Verschlüsselungsstandard nur auf vollständig internen Speicher angewendet, der mit der AES-Engine verbunden ist. Beispielsweise werden weder externe, über USB verbundene Speichergeräte, noch PCIe-basierte Speichergeräte, die zum Mac Pro (2019) hinzugefügt werden, auf diese Weise verschlüsselt.

Die Secure Enclave hat außerdem eine Gerätegruppe-ID (GID), die für alle Geräte gleich ist, die einen bestimmten SoC verwenden. (Beispielsweise haben alle Geräte mit einem Apple A15 SoC dieselbe GID.)

Auf die UID und die GID kann nicht über [Joint Test Action Group \(JTAG\)](#) oder andere Debugging-Schnittstellen zugegriffen werden.

AES-Engine der Secure Enclave

Die AES-Engine der Secure Enclave ist ein Hardwareblock, der für die symmetrische Kryptografie basierend auf der AES-Chiffre verwendet wird. Die AES-Engine ist darauf ausgelegt, der Offenlegung von Informationen durch zeitliche Planung oder Static Power Analysis (SPA) entgegenzuwirken. Seit der A9-SoCs umfasst die AES-Engine auch Gegenmaßnahmen für Dynamic Power Analysis (DPA).

Die AES-Engine unterstützt Hardware- und Softwareschlüssel. Hardwareschlüssel werden aus der UID oder der GID der Secure Enclave hergeleitet. Diese Schlüssel verbleiben in der AES-Engine und sind nicht einmal für die sepOS-Software sichtbar. Die Software kann Funktionen zur Ver- und Entschlüsselung mittels Hardwareschlüssel anfordern, die Schlüssel aber nicht extrahieren.

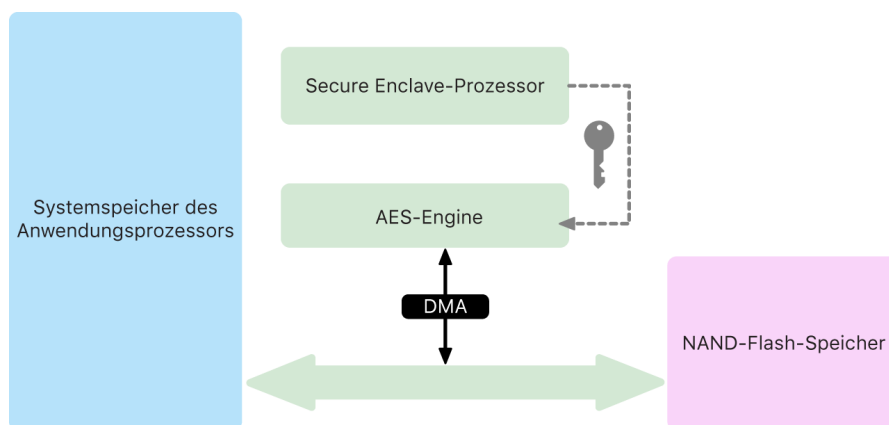
Bei Apple A10 und neueren SoCs umfasst die AES-Engine sperrbare Seed-Bits, sodass aus der UID oder GID abgeleitete Schlüssel diversifiziert werden können. Dadurch kann der Zugriff auf Daten abhängig vom Betriebsmodus des Geräts erlaubt oder verweigert werden. Auf diese Weise werden beispielsweise sperrbare Seed-Bits verwendet, um beim Starten im DFU-Modus den Zugriff auf passwortgeschützte Daten zu verwehren. Weitere Informationen sind unter [Gerätecodes und Passwörter](#) zu finden.

AES-Engine

Jedes Apple-Gerät mit einer Secure Enclave verfügt über eine dedizierte kryptografische AES-256-Engine (die „AES-Engine“), die sich im DMA-Pfad (Direct Memory Access) zwischen dem NAND (nicht flüchtigen) Flash-Speicher und dem Hauptsystemspeicher befindet, was eine höchst effiziente Dateiverschlüsselung ermöglicht. Bei A9-Prozessoren oder Prozessoren einer neueren A-Reihe befindet sich das Subsystem des Flash-Speichers auf einem isolierten Bus, der nur über die DMA Crypto Engine auf den Speicher zugreifen kann, in dem die Benutzerdaten enthalten sind.

Beim Booten generiert das sepOS mithilfe von TRNG einen temporären Verpackungsschlüssel. Die Secure Enclave übermittelt diesen Schlüssel über dedizierte Drähte an die AES-Engine, sodass ein Zugriff durch irgendwelche Software außerhalb der Secure Enclave ausgeschlossen ist. Danach kann das sepOS den temporären Verpackungsschlüssel verwenden, um Dateischlüssel zu verpacken, die vom Dateisystemtreiber des Anwendungsprozessors verwendet werden. Wenn der Dateisystemtreiber eine Datei liest oder schreibt, sendet er den verpackten Schlüssel an die AES-Engine, die ihn entpackt. Die AES-Engine legt den unverpackten Schlüssel nie gegenüber Software offen.

Hinweis: Die AES-Engine ist eine Komponente, die sowohl von der Secure Enclave als auch der AES-Engine der Secure Enclave vollständig getrennt ist. Allerdings ist ihr Betrieb, wie im Folgenden beschrieben, eng mit der Secure Enclave verknüpft.



Public Key Accelerator

Der Public Key Accelerator (PKA) ist ein Hardwareblock, der für Operationen der asymmetrischen Kryptografie verwendet wird. Der PKA unterstützt Signier- und Verschlüsselungsalgorithmen auf Basis von RSA und ECC (Elliptic Curve Cryptography, Verschlüsselung mit elliptischen Kurven). Der PKA ist darauf ausgelegt, der Offenlegung von Informationen durch zeitliche Planung oder Seitenkanalattacken wie SPA oder DPA entgegenzuwirken.

Der PKA unterstützt Hardware- und Softwareschlüssel. Hardwareschlüssel werden aus der UID oder der GID der Secure Enclave hergeleitet. Diese Schlüssel verbleiben im PKA und sind nicht einmal für die sepOS-Software sichtbar.

Ab SoCs vom Typ A13 ist die mathematische Richtigkeit der PKA-Verschlüsselungsimplementierungen durch formale Verifizierungstechniken nachweisbar.

Bei Apple A10 und neueren SoCs unterstützt der PKA an das Betriebssysteme gebundene Schlüssel, die auch als **Sealed Key Protection (SKP)** bezeichnet werden. Diese Schlüssel werden generiert, indem die UID des jeweiligen Geräts mit dem Hash-Wert der auf dem Gerät ausgeführten sepOS-Software kombiniert wird. Der Hash-Wert wird vom Boot-ROM der Secure Enclave bzw. bei Apple A13 und neueren SoCs vom Boot-Monitor der Secure Enclave bereitgestellt. Diese Schlüssel werden auch für die Verifizierung der sepOS-Version herangezogen, wenn Anfragen an bestimmte Apple-Dienste gesendet werden. Außerdem werden sie verwendet, um die Sicherheit von Daten zu verbessern, die per Code (Passcode) geschützt sind, indem sie dazu beitragen, den Zugriff auf Verschlüsselungsmaterialien zu verhindern, wenn kritische Systemänderungen ohne Benutzerauthentifizierung vorgenommen werden.

Sicherer nicht flüchtiger Speicher

Die Secure Enclave verfügt über eine dedizierte Einheit mit einem sicheren nicht flüchtigen Speicher. Der sichere nicht flüchtige Speicher ist über einen dedizierten I2C-Bus mit der Secure Enclave verbunden; das bedeutet, dass nur die Secure Enclave auf ihn zugreifen kann. Alle für die Verschlüsselung von Benutzerdaten eingesetzten Schlüssel haben ihren Ursprung in einer Entropie, die im nicht flüchtigen Speicher der Secure Enclave gespeichert ist.

Bei Geräten mit SoCs der Typen A12 und S4 (oder neuer) ist die Secure Enclave mit einer Secure Storage-Komponente gekoppelt, die als Entropiespeicher dient. Die Secure Storage-Komponente ist selbst mit einem unveränderlichem ROM-Code, einem Hardwaregenerator für Zufallszahlen, einem gerätespezifischen kryptografischen Schlüssel, kryptografischen Engines und einem Mechanismus zum Schutz vor physischer Manipulation ausgestattet. Die Secure Enclave und die Secure Storage-Komponente kommunizieren über ein verschlüsseltes und authentifiziertes Protokoll, das den ausschließlichen Zugriff auf die Entropie sicherstellt.

Geräte, die im Herbst 2020 oder später ausgeliefert wurden, sind mit einer Secure Storage-Komponente der zweiten Generation ausgestattet. Diese Secure Storage-Komponente der zweiten Generation stellt zusätzlich Lockboxen für Zähler bereit. Jede dieser Lockboxen enthält einen Salt- und einen Prüfwert für den Code mit jeweils 128 Bit sowie einen 8-Bit-Zähler und einen 8-Bit-Maximalwert. Für den Zugriff auf diese Lockboxen wird ein verschlüsseltes und authentifiziertes Protokoll verwendet.

Zähler-Lockboxen enthalten die Entropie, die für die Entsperrung von codegeschützten Benutzerdaten erforderlich ist. Für den Zugriff auf die Benutzerdaten muss die gekoppelte Secure Enclave den richtigen Entropiewert für den Code aus dem Code des Benutzers und der UID der Secure Enclave herleiten. Der Benutzercode kann nicht durch Entsperrvorgänge bereitgestellt werden, die aus einer anderen Quelle als der gekoppelten Secure Enclave stammen. Wenn die maximale Anzahl der Versuche überschritten wird (z. B. 10 Versuche auf einem iPhone), werden die mit dem Code geschützten Informationen durch die Secure Storage-Komponente vollständig gelöscht.

Zum Erstellen einer solchen Zähler-Lockbox sendet die Secure Enclave den Entropiewert für den Code und den Maximalwert für den Zähler an die Secure Storage-Komponente. Der Secure Storage-Komponente generiert den Salt-Wert mit seinem Generator für Zufallszahlen. Danach leitet er einen Prüfwert für den Code und einen Lockbox-Entropiewert aus dem Entropiewert für den Code, dem eindeutigen kryptografischen Schlüssel der Secure Storage-Komponente und dem Salt-Wert ab. Die Secure Storage-Komponente initialisiert die Zähler-Lockbox mit dem Wert 0, dem bereitgestellten Maximalwert für den Zähler, dem abgeleiteten Prüfwert für den Code und dem Salt-Wert. Im Anschluss übergibt die Secure Storage-Komponente den generierten Lockbox-Entropiewert an die Secure Enclave.

Um zu einem späteren Zeitpunkt den Lockbox-Entropiewert aus der Zähler-Lockbox abzurufen, sendet die Secure Enclave den Entropiewert für den Code an die Secure Storage-Komponente. Die Secure Storage-Komponente erhöht zunächst den Zähler für die Lockbox. Wenn der erhöhte Zählerwert den Maximalwert übersteigt, löscht die Secure Storage-Komponente die Zähler-Lockbox vollständig. Solange die Maximalanzahl an Versuchen nicht überschritten wird, versucht die Secure Storage-Komponente, den Prüfwert für den Code und den Lockbox-Entropiewert mit demselben Algorithmus abzuleiten, der für die Erzeugung der Zähler-Lockbox verwendet wurde. Wenn der abgeleitete Prüfwert für den Code mit dem hinterlegten Prüfwert für den Code übereinstimmt, gibt die Secure Storage-Komponente den Lockbox-Entropiewert zurück an die Secure Enclave und setzt den Zähler wieder auf 0 zurück.

Die Schlüssel, die für den Zugriff auf passwortgeschützte Daten verwendet werden, haben ihren Ursprung in der in Zähler-Lockboxen hinterlegten Entropie. Weitere Informationen sind unter [Datensicherheit – Übersicht](#) zu finden.

Der sichere nicht flüchtige Speicher wird für alle Anti-Replay-Dienste in der Secure Enclave verwendet. Anti-Replay-Dienste in der Secure Enclave werden zum Widerrufen von Daten zu Ereignissen verwendet, die Anti-Replay-Grenzen markieren. Dazu gehören unter anderem:

- Codeänderungen
- Aktivieren oder Deaktivieren von Face ID oder Touch ID
- Hinzufügen oder Entfernen von Gesichtsdaten in Face ID oder Fingerabdruckdaten in Touch ID
- Zurücksetzen von Face ID oder Touch ID
- Hinzufügen oder Entfernen einer Karte in Apple Pay
- Löschen aller Inhalte und Einstellungen

In Architekturen ohne Secure Storage-Komponente wird das EEPROM (electrically erasable programmable read-only memory) eingesetzt, um der Secure Enclave sichere Speicherdienste bereitzustellen. Das EEPROM ist genau wie die Secure Storage-Komponente ausschließlich über die Secure Enclave erreichbar und mit ihr verbunden. Es enthält allerdings weder dedizierte Hardwaresicherheitsfunktionen noch kann es exklusiven Zugriff auf die Entropie (über ihre physischen Verbindungscharakteristiken hinaus) oder die Zähler-Lockbox-Funktionen bieten.

Secure Neural Engine

Bei Geräten mit Face ID konvertiert die Secure Neural Engine (sichere neuronale Engine) 2D-Bilder und Tiefendarstellungen in die mathematische Darstellung des Gesichts eines Benutzers.

Bei SoCs der Typen A11 bis A13 ist die Secure Neural Engine in die Secure Enclave integriert. Im Interesse einer hohen Leistung verwendet die Secure Neural Engine den direkten Zugriff (DMA, Direct Memory Access). Eine IOMMU-Einheit (zur Eingabe-Ausgabe-Speicherverwaltung), die der Steuerung des sepOS-Kernels untersteht, beschränkt diesen direkten Zugriff auf autorisierte Speicherregionen.

Ab A14 und der M1-Familie ist die Secure Neural Engine als sicherer Modus in die neuronale Engine des Anwendungsprozessors implementiert. Ein dedizierter Controller zur Wahrung der Hardwaresicherheit bewirkt das Umschalten zwischen Anfragen des Anwendungsprozessors und solchen der Secure Enclave, wobei bei jedem Wechsel der Status der neuronalen Engine zurückgesetzt wird, um die Sicherheit der Face ID-Daten auf Dauer zu wahren. Eine dedizierte Engine übernimmt die Speicherverschlüsselung, die Authentifizierung und die Zugriffssteuerung. Gleichzeitig kommen ein separater kryptografischer Schlüssel und ein Speicherbereich zum Einsatz, um die Secure Neural Engine auf autorisierte Speicherregionen zu beschränken.

Strom- und Taktmonitore

Die gesamte Elektronik ist für den Betrieb innerhalb eines bestimmten Netzspannungs- und Netzfrequenzbereichs ausgelegt. Beim Betrieb außerhalb des vorgegebenen Bereichs kann es zu Fehlfunktionen der Elektronik mit dem Ergebnis kommen, dass Sicherheitsvorkehrungen umgangen werden können. Die Secure Enclave ist mit auf das Monitoring ausgelegten Schaltkreisen ausgestattet, die sicherstellen, dass Netzspannung und Netzfrequenz in einem sicheren Bereich verbleiben. Diese für das Monitoring verwendeten Schaltkreise sind für einen sehr viel größeren Spannungs- und Frequenzbereich ausgelegt als die übrigen Bestandteile der Secure Enclave. Beim Erkennen eines illegalen Betriebspunkts werden die Taktgeber der Secure Enclave automatisch gestoppt; sie können bis zum nächsten Zurücksetzen des SoCs nicht neu gestartet werden.

Secure Enclave – Funktionsübersicht

Hinweis: A12-, A13-, S4- und S5-Produkte, die erstmals im Herbst 2020 ausgeliefert werden, verfügen über eine Secure Storage-Komponente der zweiten Generation, während frühere, auf diesen SoCs basierende Produkte eine Secure Storage-Komponente der ersten Generation besitzen.

SoC	Engine für den Speicherschutz (Memory Protection Engine)	Sicherer Speicher	AES-Engine	PKA
A8	Verschlüsselung und Authentifizierung	EEPROM	Ja	Nein
A9	Verschlüsselung und Authentifizierung	EEPROM	DPA-Schutz	Ja
A10	Verschlüsselung und Authentifizierung	EEPROM	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel
A11	Verschlüsselung, Authentifizierung und Replay-Verhinderung	EEPROM	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel
A12 (vor Herbst 2020 herausgegebene Apple-Geräte)	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 1. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel
A12 (nach Herbst 2020 herausgegebene Apple-Geräte)	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 2. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel
A13 (vor Herbst 2020 herausgegebene Apple-Geräte)	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 1. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel und Boot-Monitor
A13 (nach Herbst 2020 herausgegebene Apple-Geräte)	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 2. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel und Boot-Monitor
A14, A15	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 2. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel und Boot-Monitor
S3	Verschlüsselung und Authentifizierung	EEPROM	DPA-Schutz und sperrbare Seed-Bits	Ja
S4	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 1. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel

SoC	Engine für den Speicherschutz (Memory Protection Engine)	Sicherer Speicher	AES-Engine	PKA
S5 (vor Herbst 2020 herausgegebene Apple-Geräte)	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 1. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel
S5 (nach Herbst 2020 herausgegebene Apple-Geräte)	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 2. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel
S6, S7	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 2. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel
T2	Verschlüsselung und Authentifizierung	EEPROM	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel
M1-Familie	Verschlüsselung, Authentifizierung und Replay-Verhinderung	Secure Storage-Komponente der 2. Generation	DPA-Schutz und sperrbare Seed-Bits	An Betriebssystem gebundene Schlüssel und Boot-Monitor

Face ID und Touch ID

Sicherheit von Face ID und Touch ID

Codes (Gerätecodes) und Passwörter sind essenziell für die Sicherheit von Apple-Geräten. Gleichzeitig müssen Benutzer in der Lage sein, komfortabel auf ihre Geräte zuzugreifen – was oftmals mehr als 100-mal am Tag vorkommen kann. Die biometrische Authentifizierung bietet eine Möglichkeit, die Sicherheit eines starken Gerätecodes beizubehalten – bzw. Code oder Passwort sogar zu stärken, da diese häufig nicht manuell eingegeben werden müssen – und Geräte gleichzeitig schnell und bequem mit einem Fingerdruck oder Blick zu entsperren. Face ID und Touch ID sind kein Ersatz für einen Code oder ein Passwort, sie erleichtern und beschleunigen aber in vielen Situationen den Zugriff auf das Gerät.

Die biometrische Sicherheitsarchitektur von Apple basiert auf einer strikten Trennung der Kompetenzen zwischen dem biometrischen Sensor und der Secure Enclave sowie der sicheren Verbindung zwischen diesen beiden Komponenten. Der Sensor erfasst das biometrische Bild und übermittelt dieses sicher an die Secure Enclave. Bei der Registrierung verarbeitet, verschlüsselt und sichert die Secure Enclave die entsprechenden als Vorlage dienenden Daten von Face ID und Touch ID. Bei der Suche nach Übereinstimmungen vergleicht die Secure Enclave die eingehenden Daten des biometrischen Sensors mit den gesicherten Vorlagen, um zu ermitteln, ob das Gerät entsperrt oder die Mitteilung ausgegeben werden soll, dass eine Übereinstimmung gültig ist (bei Apple Pay, in Apps und bei anderen Verwendungen von Face ID und Touch ID). Die Architektur unterstützt Geräte, die sowohl über einen Sensor als auch eine Secure Enclave verfügen (wie beispielsweise das iPhone, das iPad und einige Mac-Computer). Sie unterstützt auch die Möglichkeit, den Sensor physisch zu trennen und als Peripheriegerät zu behandeln, das dann in einem Mac mit Apple Chips mit der Secure Enclave auf sichere Weise gekoppelt wird.

Sicherheit bei Face ID

Mit einem einfachen Blick auf das Gerät entsperrt Face ID unterstützte Apple-Geräte sicher. Das TrueDepth-Kamerasystem ermöglicht eine intuitive und sichere Authentifizierung, für die auf der Basis modernster Technologien die Geometrie des Gesichts des Benutzers präzise erfasst wird. Face ID verwendet neuronale Netzwerke, um die Blickrichtung zu erkennen, den Abgleich auszuführen und Manipulation zu verhindern, damit das Gerät per Blickkontakt entsperrt werden kann. Dies kann bei unterstützten Geräten auch beim Tragen einer Maske durchgeführt werden. Face ID passt sich automatisch an Änderungen im Aussehen des Benutzers an und schützt dessen Privatsphäre und die Sicherheit seiner biometrischen Daten.

Face ID wurde entwickelt, um anhand der Blickrichtung zu erkennen, dass eine Interaktion mit dem Gerät erfolgen soll, um eine zuverlässige Authentifizierung mit einer niedrigen Fehlerrate beim Abgleich zu bieten und digitaler oder physischer Manipulation entgegenzuwirken.

Die TrueDepth-Kamera sucht automatisch nach dem Gesicht, wenn (durch Anheben oder durch Tippen auf den Bildschirm) der Ruhezustand von mit Face ID ausgestatteten Apple-Geräten beendet wird. Sie sucht ebenfalls nach dem Gesicht, wenn diese Geräte versuchen, den Benutzer zu authentifizieren, um eine erhaltene Mitteilung oder Benachrichtigung anzuzeigen, oder wenn eine unterstützte App eine Face ID-Authentifizierung anfordert. Wenn ein Gesicht erkannt wird, bestätigt Face ID die Absicht zum Entsperren des Geräts, indem geprüft wird, ob die Augen geöffnet sind und die Aufmerksamkeit auf das Gerät gerichtet ist. Für eine bessere Nutzbarkeit wird die Aufmerksamkeitsprüfung der Face ID ausgeschaltet, wenn VoiceOver aktiviert ist, und kann, falls erforderlich, auch separat deaktiviert werden. Beim Verwenden von Face ID mit einer Maske ist die Erkennung des Blicks immer erforderlich.

Nachdem die TrueDepth-Kamera ein auf das Gerät gerichtetes Gesicht bestätigt, projiziert und liest sie über Tausende Infrarotpunkte, um eine Tiefendarstellung und ein 2D-Infrarotbild des Gesichts zu erstellen. Diese Daten werden verwendet, um eine Reihe von 2D-Bildern und Tiefendarstellungen zu erstellen, die digital signiert und an die Secure Enclave gesendet werden. Als Gegenmaßnahme gegen digitale und physische Manipulation erstellt die TrueDepth-Kamera eine zufällige Reihenfolge der 2D-Bilder und Tiefendarstellungen und projiziert ein gerätespezifisches Zufallsmuster. Ein Bestandteil der Secure Neural Engine – innerhalb der Secure Enclave geschützt – wandelt diese Daten in eine mathematische Darstellung um und vergleicht diese Daten mit den registrierten Gesichtsdaten. Bei den registrierten Gesichtsdaten handelt es sich ebenfalls um eine mathematische Darstellung des Gesichts eines Benutzers in einer Vielzahl von Posen.

Sicherheit bei Touch ID

Touch ID ist das Sensorsystem für Fingerabdrücke, mit dem ein sicherer Zugriff auf unterstützte Apple-Geräte schneller und einfacher möglich ist. Diese Technologie liest Fingerabdruckdaten aus jedem beliebigen Winkel und erfasst den Fingerabdruck des Benutzers nach und nach immer genauer. Dabei erweitert der Sensor die Fingerabdruckdarstellung, wenn bei jeder Nutzung zusätzliche überlappende Knoten identifiziert werden.

Apple-Geräte mit einem Touch ID-Sensor können per Fingerabdruck entsperrt werden. Mit Touch ID entfällt nicht die Notwendigkeit für einen Gerätecode oder ein Benutzerpasswort. Diese sind nach dem Starten oder dem Neustarten des Geräts bzw. (bei einem Mac) nach dem Abmelden erforderlich. In bestimmten Apps kann auch Touch ID anstelle des Gerätecodes oder des Benutzerpassworts verwendet werden – zum Beispiel zum Entsperren passwortgeschützter Notizen in der App „Notizen“, zum Entsperren von Websites, die durch den Schlüsselbund geschützt sind, und zum Entsperren von Passwörtern unterstützter Apps. In manchen Szenarien ist jedoch immer ein Gerätecode oder ein Benutzerpasswort erforderlich (zum Beispiel, wenn es darum geht, einen vorhandenen Gerätecode oder ein vorhandenes Benutzerpasswort zu ändern, einen vorhandenen und registrierten Fingerabdruck zu entfernen oder einen neuen Fingerabdruck zu registrieren).

Wenn der Fingerabdrucksensor einen aufgelegten Finger bemerkt, initiiert er den Fingerabdruckscanner, der wiederum den Abdruck des aufgelegten Fingers scannt und den Scan an die Secure Enclave sendet. Der Kanal zum Sichern dieser Verbindung variiert, je nachdem, ob der Touch ID-Sensor im Gerät mit Secure Enclave integriert ist oder sich in einem separaten Peripheriegerät befindet.

Solange der gescannte Fingerabdruck für die Analyse vektorisiert wird, wird das Rasterbild vorübergehend im verschlüsselten Speicher innerhalb der Secure Enclave gespeichert. Danach wird es verworfen. Bei der Analyse wird der Verlauf der subkutanen **Papillarleisten** abgebildet. Dabei handelt es sich um ein verlustbehaftetes Verfahren, bei dem Details, die zur Rekonstruktion des Fingerabdrucks des Benutzers benötigt würden, nicht gespeichert werden. Bei der Registrierung wird das Abbild der Knoten, das sich daraus ergibt, in einem verschlüsselten Format gesichert und kann nur von der Secure Enclave als Vorlage gelesen werden, um zukünftige Übereinstimmungen zu überprüfen. Es sind keine Informationen zur Identität enthalten. Diese Daten werden ausschließlich auf dem Gerät gesichert. Sie werden nicht an Apple gesendet und sind nicht Teil einer Gerätesicherung (Backups).

Integrierte Sicherheit für den Touch ID-Kanal

Die Kommunikation zwischen der Secure Enclave und dem integrierten Touch ID-Sensor erfolgt über einen SPI-Bus (Serial Peripheral Interface). Der Prozessor leitet die Daten an die Secure Enclave weiter, kann sie aber nicht lesen. Sie werden mit einem Sitzungsschlüssel verschlüsselt und authentifiziert, der mithilfe eines gemeinsamen Schlüssels ausgehandelt wird, welcher für jeden Touch ID-Sensor und die zugehörige Secure Enclave bei der Fabrikation festgelegt wird. Der gemeinsame Schlüssel für jeden Touch ID-Sensor ist stark, zufällig (randomisiert) und verschieden. Beim Austausch des Sitzungsschlüssels wird AES **Key Wrapping** verwendet, wobei beide Seiten einen zufälligen Schlüssel bereitstellen, aus denen der Sitzungsschlüssel erstellt wird; die Transportverschlüsselung, die die Authentifizierung und Vertraulichkeit gewährleistet, erfolgt per AES-CCM.

Magic Keyboard mit Touch ID

Das Magic Keyboard mit Touch ID (und das Magic Keyboard mit Touch ID und Ziffernblock) bietet einen Touch ID-Sensor in einer externen Tastatur, die mit allen Mac-Geräten mit Apple Chips verwendet werden kann. Das Magic Keyboard mit Touch ID übernimmt die Rolle des biometrischen Sensors. Die Tastatur sichert keine biometrischen Vorlagen, überprüft keine biometrischen Übereinstimmungen und setzt auch keine Sicherheitsrichtlinien durch (z. B. die Eingabe des Passworts, wenn das Gerät 48 Stunden nicht entsperrt wurde). Der Touch ID-Sensor im Magic Keyboard mit Touch ID muss vor der Verwendung mit der Secure Enclave auf dem Mac auf sichere Weise gekoppelt sein. Erst dann führt die Secure Enclave die Registrierung, Suche nach Übereinstimmungen und die Durchsetzung von Sicherheitsrichtlinien auf dieselbe Art und Weise wie bei einem integrierten Touch ID-Sensor durch. Apple führt bei einem Magic Keyboard mit Touch ID, das im Lieferumfang des Mac enthalten ist, den Kopplungsprozess bereits in der Fabrik durch. Die Kopplung kann auch durch den Benutzer durchgeführt werden (sofern erforderlich). Ein Magic Keyboard mit Touch ID kann nur mit einem Mac sicher gekoppelt werden. Der Mac kann hingegen bis zu fünf sichere Kopplungen mit verschiedenen Magic Keyboards mit Touch ID verwalten.

Das Magic Keyboard mit Touch ID ist mit integrierten Touch ID-Sensoren kompatibel. Wenn ein Finger, der auf einem Mac mit integriertem Touch ID-Sensor registriert wurde, auf einem Magic Keyboard mit Touch ID verwendet wird, führt die Secure Enclave im Mac die Übereinstimmungsprüfung erfolgreich durch – und umgekehrt.

Um die sichere Kopplung und somit die Kommunikation zwischen der Secure Enclave im Mac und dem Magic Keyboard mit Touch ID zu unterstützen, ist die Tastatur mit einem PKA-Hardwareblock (Public Key Accelerator), z. B. für Bescheinigungen, und hardwarebasierten Schlüsseln ausgestattet, um notwendige kryptografische Prozesse durchzuführen.

Sichere Kopplung

Bevor ein Magic Keyboard mit Touch ID für Touch ID-Aufgaben verwendet werden kann, muss die Tastatur mit dem Mac sicher gekoppelt werden. Zum Koppeln tauschen die Secure Enclave auf dem Mac und der PKA-Block im Magic Keyboard mit Touch ID öffentliche Schlüssel aus, die ihren Ursprung in der vertrauenswürdigen Apple CA haben. Darüber hinaus verwenden sie hardwarebasierte Bescheinigungsschlüssel und temporäre ECDH-Schlüssel, um auf sichere Weise ihre Identität zu bestätigen. Auf dem Mac werden diese Daten von der Secure Enclave geschützt. Auf dem Magic Keyboard mit Touch ID schützt der PKA-Block diese Daten. Nach der sicheren Kopplung sind alle gesendeten Touch ID-Daten zwischen dem Mac und dem Magic Keyboard mit Touch ID per AES-GCM mit einer Schlüssellänge von 256 Bits verschlüsselt und mit temporären ECDH-Schlüsseln geschützt, die eine NIST P-256-Kurve verwenden, die auf den gesicherten Identitäten basiert. (Normale Tastatureingaben werden wie bei allen Bluetooth-Tastaturen mit den in Bluetooth integrierten Sicherheitsmaßnahmen übertragen.)

Gesicherte Absicht zum Koppeln

Um manche Touch ID-Aufgaben zum ersten Mal ausführen zu können (z. B. einen neuen Fingerabdruck registrieren), müssen Benutzer physisch ihre Absicht bestätigen, dass sie ein Magic Keyboard mit Touch ID mit ihrem Mac verwenden wollen. Die physische Absicht kann einerseits bestätigt werden, indem zweimal die Taste „Ein/Aus“ des Mac gedrückt wird, wenn eine Aufforderung durch die Benutzeroberfläche angezeigt wird. Und andererseits kann die Bestätigung durch eine erfolgreiche Übereinstimmung des gescannten Fingerabdrucks erfolgen, der zuvor auf dem Mac registriert wurde. Weitere Informationen sind unter [Gesicherte Absicht und Verbindungen zur Secure Enclave](#) zu finden.

Apple Pay-Transaktionen können mit einer Touch ID-Übereinstimmung oder durch Eingabe des macOS-Benutzerpassworts und zweimaliges Drücken der Touch ID-Taste auf dem Magic Keyboard mit Touch ID bestätigt werden. Letzteres erlaubt es dem Benutzer, die Absicht sogar ohne Touch ID-Übereinstimmung physisch zu bestätigen.

Kanalsicherheit des Magic Keyboard mit Touch ID

Um einen sicheren Kommunikationskanal zwischen dem Touch ID-Sensor im Magic Keyboard mit Touch ID und der Secure Enclave auf dem gekoppelten Mac sicherzustellen, muss Folgendes erfüllt sein:

- Eine sichere Kopplung zwischen dem PKA-Block des Magic Keyboard mit Touch ID und der Secure Enclave (wie oben beschrieben)
- Ein sicherer Kanal zwischen dem Sensor des Magic Keyboard mit Touch ID und dem zugehörigen PKA-Block

Der sichere Kanal zwischen dem Sensor des Magic Keyboard mit Touch ID und dem zugehörigen PKA-Block wird mit einem individuellen, zwischen den beiden Komponenten geteilten Schlüssel in der Fabrik hergestellt. (Hier handelt es sich um dieselbe Technik, mit der bei Mac-Computern mit integrierter Touch ID der sichere Kanal zwischen der Secure Enclave auf dem Mac und dem integrierten Sensor hergestellt wird.)

Face ID, Touch ID, Codes und Passwörter

Um Face ID oder Touch ID zu verwenden, muss das Gerät so eingerichtet werden, dass zum Entsperren ein Code benötigt wird. Wenn Face ID oder Touch ID eine erfolgreiche Übereinstimmung erkennen, wird das Gerät des Benutzers entsperrt, ohne dass nach dem Code für das Gerät oder dem Passwort des Benutzers gefragt wird. Das vereinfacht die Verwendung längerer, komplexerer Codes und Passwörter, da Benutzer seltener zur manuellen Eingabe gezwungen sind. Face ID und Touch ID sind kein Ersatz für den Code oder das Passwort des Benutzers. Sie ermöglichen lediglich einen erleichterten Zugriff auf das Gerät innerhalb ausgeklügelter Grenzen und Zeitbeschränkungen. Dies ist wichtig, da ein sicherer Code und ein sicheres Passwort die Grundlage dafür bilden, wie ein iPhone-, iPad-, Mac- oder Apple Watch-Gerät die Benutzerdaten kryptografisch schützt.

Erforderlichkeit eines Gerätecodes oder eines Benutzerpassworts

Benutzer können ihren Gerätecode oder ihr Passwort jederzeit anstelle von Face ID oder Touch ID verwenden. Es gibt jedoch Situationen, in denen die biometrischen Methoden nicht erlaubt sind. Für folgende sicherheitsrelevante Vorgänge ist die Eingabe eines Passworts oder Gerätecodes immer erforderlich:

- Aktualisieren der Software
- Löschen des Geräts
- Anzeigen oder Ändern von Codeeinstellungen
- Installieren von Konfigurationsprofilen
- Entsperren des Bereichs „Privatsphäre & Sicherheit“ in den Mac-Systemeinstellungen
- Entsperren des Bereichs „Benutzer & Gruppen“ in den Mac-Systemeinstellungen (sofern FileVault aktiviert ist)

Ein Code oder ein Passwort ist ebenfalls erforderlich, wenn sich das Gerät in einem der folgenden Zustände befindet:

- Das Gerät wurde unmittelbar zuvor erst eingeschaltet oder neu gestartet.
- Der Benutzer hat sich von seinem Mac-Account abgemeldet (oder hat sich noch nicht angemeldet).

- Der Benutzer hat das Gerät seit über 48 Stunden nicht entsperrt.
- Der Benutzer hat Gerätecode oder Passwort seit 156 Stunden (6,5 Tagen) und eine biometrische Methode in den letzten 4 Stunden nicht zum Entsperren des Geräts verwendet.
- Das Gerät wurde per Fernzugriff gesperrt.
- Der Benutzer hat den Abschaltvorgang oder einen Notruf beendet (durch gleichzeitiges Drücken und Halten einer Lautstärketaste und des Ein-/Ausschalters und anschließendes Drücken auf „Abbrechen“).
- Es wurden fünf erfolglose Versuche des biometrischen Abgleichs unternommen (auch wenn es aus Gründen der Bedienbarkeit vonseiten des Geräts möglich sein kann, nach einer geringen Anzahl von Fehlversuchen anstelle der biometrischen Authentifizierung den Code oder das Passwort einzugeben).

Wenn Face ID mit einer Maske auf einem iPhone aktiviert ist, ist die Funktion für die nächsten 6,5 Stunden nach einer der folgenden Benutzeraktionen verfügbar:

- Erfolgreicher Face ID-Abgleichungsversuch (mit oder ohne Maske)
- Bestätigung des Gerätecodes
- Entsperren des Geräts mit der Apple Watch

Jede dieser Aktionen erweitert den Zeitraum nach der Durchführung um weitere 6,5 Stunden.

Wenn auf einem iPhone oder einem iPad Face ID oder Touch ID aktiviert ist, wird das Gerät sofort gesperrt, wenn der Ein-/Ausschalter gedrückt wird. Außerdem wird das Gerät grundsätzlich gesperrt, wenn es in den Ruhezustand wechselt. Face ID und Touch ID benötigen einen erfolgreichen Abgleich (oder optional die Verwendung des Codes), um den Ruhezustand zu beenden.

Die Wahrscheinlichkeit, dass eine beliebige andere Person in der Lage ist, das iPhone oder das iPad zu entsperren, liegt bei der Face ID bei 1:1.000.000. Dies gilt auch in den Fällen, in denen Face ID mit Maske aktiviert ist. Bei iPhone, iPad oder Mac-Modellen mit Touch ID oder bei den Geräten, die mit einem Magic Keyboard gekoppelt sind, liegt die Wahrscheinlichkeit bei ca. 1:50.000. Diese Wahrscheinlichkeit steigt bei mehreren registrierten Fingerabdrücken (auf bis zu 1:10.000 bei fünf Fingerabdrücken) oder Erscheinungsbildern (auf bis zu 1:500.000 bei zwei Erscheinungsbildern). Als zusätzlichen Schutz lassen Face ID und Touch ID nur fünf Fehlversuche zu, bevor ein Code oder ein Passwort eingegeben werden muss, um Zugriff auf das Gerät zu erhalten. In den folgenden Fällen ist die Wahrscheinlichkeit für einen falschen Abgleich mit Face ID höher:

- Bei Zwillingen und Geschwistern, die dem Benutzer ähnlich sehen
- Bei Kindern unter 13 Jahren (weil deren individuelle Gesichtszüge möglicherweise noch nicht voll ausgeprägt sind)

In diesen beiden Fällen liegt die Wahrscheinlichkeit höher, wenn Face ID mit einer Maske verwendet wird. Wenn im Hinblick auf einen möglicherweise falschen Abgleich Bedenken bestehen, empfiehlt Apple die Verwendung eines Codes für die Authentifizierung.

Sicherheit beim Gesichtsabgleich

Der Gesichtsabgleich in der Secure Enclave wird mit neuronalen Netzwerken durchgeführt, die speziell für diesen Zweck trainiert wurden. Apple hat die neuronalen Netzwerke für den Gesichtsabgleich auf der Basis von über einer Milliarde Bildern entwickelt (einschließlich Infrarot- und Tiefenbildern), die mit dem Einverständnis der Studienteilnehmer erfasst wurden. Apple hat in einem zweiten Schritt mit Teilnehmern auf der ganzen Welt zusammengearbeitet, um eine repräsentative Personengruppe abzudecken, die sich in Geschlecht, Alter, Herkunft und anderen Faktoren unterscheidet. Die Studien wurden bei Bedarf verbessert, um einen hohen Grad an Genauigkeit für unterschiedliche Benutzer sicherzustellen. Face ID ist darauf ausgelegt, auch in Situationen zu funktionieren, in denen Benutzer Hüte, Halstücher, Brillen, Kontaktlinsen und Sonnenbrillen tragen. Face ID unterstützt auch das Entsperren mit Maske auf iPhone-Geräten ab iPhone 12 und iOS 15.4 (oder neuer). Außerdem funktioniert es drinnen, draußen und sogar bei völliger Dunkelheit. Ein zusätzliches neuronales Netzwerk, das trainiert wurde, um Manipulationen zu erkennen, schützt vor Versuchen, das Gerät mit Fotos oder Masken zu entsperren. Face ID-Daten (einschließlich mathematischer Darstellungen des Gesichts) werden verschlüsselt und sind nur für die Secure Enclave verfügbar. Diese Daten werden ausschließlich auf dem Gerät gesichert. Sie werden nicht an Apple gesendet und sind nicht Teil einer Gerätesicherung (Backups). Die folgenden Face ID-Daten werden bei normalem Betrieb gesichert und nur zur Verwendung durch die Secure Enclave verschlüsselt:

- Die mathematische Darstellung des Gesichts eines Benutzers, die bei der Registrierung berechnet wurde
- Die mathematische Darstellung des Gesichts eines Benutzers, die bei einem Versuch berechnet wurde, das Gerät zu entsperren, wenn Face ID diese Darstellung als hilfreich für die Verbesserung eines zukünftigen Abgleichs einstuft

Bilder des Gesichts, die im normalen Betrieb erfasst werden, werden nicht gesichert, sondern verworfen, sobald die Berechnung der mathematischen Darstellung abgeschlossen wurde, die für die Registrierung oder den Vergleich mit den registrierten Face ID-Daten benötigt wird.

Optimieren des Face ID-Abgleichs

Zur Leistungsverbesserung beim Entsperren des Geräts und zur Kompensation natürlicher Veränderungen in Gesicht und Aussehen verbessert Face ID die gespeicherte mathematische Darstellung im Laufe der Zeit. Bei einem erfolgreichen Abgleich verwendet Face ID die neu berechnete mathematische Darstellung (sofern sie eine ausreichende Qualität hat) möglicherweise für eine begrenzte Anzahl weiterer Abgleichvorgänge, bevor die Daten verworfen werden. Wenn umgekehrt Face ID ein Gesicht nicht erkennt, die Abgleichqualität aber über einem bestimmten Schwellenwert liegt, und der Benutzer unmittelbar nach dem Fehlversuch seinen Code eingibt, erstellt Face ID im Gegenzug eine weitere Aufnahme und optimiert die bereits registrierten Face ID-Daten mit der neu berechneten mathematischen Darstellung. Diese neuen Face ID-Daten werden verworfen, wenn der Benutzer damit nicht mehr abgeglichen werden kann oder eine vorgegebene Anzahl von Abgleichvorgängen erreicht wird. Die neuen Daten werden auch beim Auswählen der Option zum Zurücksetzen von Face ID verworfen. Mit diesen Verbesserungsprozessen kann Face ID sich an deutliche Änderungen der Gesichtsbehaarung oder des Make-ups eines Benutzers anpassen und gleichzeitig Falschauthentifizierungen minimieren.

Verwendung von Face ID und Touch ID

Entsperren eines Geräts oder Benutzer-Accounts

Wenn Face ID oder Touch ID deaktiviert ist und ein Gerät gesperrt wird, werden die Schlüssel für die höchste Klasse der Datensicherheit, die in der Secure Enclave gespeichert sind, verworfen. Die Dateien und die Objekte des Schlüsselbunds der betreffenden Klasse sind erst wieder verfügbar, wenn der Benutzer das Gerät oder den Account durch Eingabe seines Codes oder Passworts wieder entsperrt.

Ist Face ID oder Touch ID aktiviert, werden die Schlüssel nicht verworfen, wenn das Gerät gesperrt wird. Sie werden vielmehr mit einem Schlüssel verpackt, der an das Face ID- oder Touch ID-Teilsystem innerhalb der Secure Enclave übergeben wird. Wenn der Benutzer versucht, das Gerät oder den Account zu entsperren, und das Gerät einen erfolgreichen Abgleich bestätigt, stellt das Gerät den Schlüssel zum Entpacken der Datensicherheitsschlüssel bereit: Das Gerät bzw. der Account wird daraufhin entsperrt. Dieses Verfahren bietet zusätzlichen Schutz, da die Teilsysteme „Datensicherheit“ und Face ID oder Touch ID zusammenarbeiten müssen, damit das Gerät entsperrt werden kann.

Wenn das Gerät neu gestartet wird, gehen die Schlüssel verloren, die zum Entsperren des Geräts oder Accounts per Face ID oder Touch ID erforderlich sind. Sie werden von der Secure Enclave verworfen, wenn irgendeine Bedingung eintritt, die die Eingabe eines Codes oder Passworts erfordert.

Absichern von Käufen mit Apple Pay

Benutzer können Face ID und Touch ID in Kombination mit Apple Pay auch verwenden, um einfach und sicher in Ladengeschäften, Apps und im Web einzukaufen:

- *Per Face ID in Ladengeschäften:* Um eine Zahlung in einem Ladengeschäft per Face ID zu autorisieren, muss der Benutzer zunächst seine Zahlungsabsicht mit einem Doppelklick auf die Seitentaste bestätigen. Dieser Doppelklick erfasst die Zahlungsabsicht des Benutzers mit einer physischen Geste, die direkt mit der Secure Enclave verknüpft und resistent gegen Betrug während eines böswilligen Prozesses ist. Danach muss er eine Authentifizierung mit Face ID vornehmen, bevor er das Gerät vor das kontaktlose Zahlungsterminal hält. Nach der Authentifizierung per Face ID kann der Benutzer eine andere Apple Pay-Zahlungsmethode wählen. Dies erfordert eine neuerliche Authentifizierung, der Benutzer muss dann aber die Seitentaste nicht erneut doppelklicken.
- *Per Face ID in Apps und im Internet:* Für eine Zahlung innerhalb von Apps und im Web muss der Benutzer seine Zahlungsabsicht mit Doppelklicken der Seitentaste bestätigen. Danach erfolgt die Authentifizierung per Face ID, um die Zahlung zu autorisieren. Wenn die Apple Pay-Transaktion nicht innerhalb von 60 Sekunden nach dem Doppelklicken der Seitentaste abgeschlossen wird, muss die Zahlungsabsicht ein weiteres Mal durch Doppelklicken bestätigt werden.
- *Per Touch ID:* Bei Verwendung von Touch ID wird die Zahlungsabsicht mit der Geste zum Aktivieren des Touch ID-Sensors in Kombination mit dem erfolgreichen Abgleich des Fingerabdrucks des Benutzers bestätigt.

Verwenden der vom System bereitgestellten APIs

Apps anderer Anbieter können vom System bereitgestellte APIs verwenden, um den Benutzer aufzufordern, sich per Face ID oder Touch ID oder durch Eingabe eines Codes oder Passworts zu authentifizieren. Apps, die Touch ID unterstützen, unterstützen ohne jegliche Änderungen automatisch auch Face ID. Bei Verwendung von Face ID oder Touch ID wird die App nur darüber informiert, ob die Authentifizierung erfolgreich war. Die App kann aber nicht auf Face ID, Touch ID oder die Daten zugreifen, die für den registrierten Benutzer hinterlegt sind.

Schützen von Schlüsselbundobjekten

Objekte im Schlüsselbund können auch mithilfe von Face ID oder Touch ID geschützt werden, indem sie nur bei einem erfolgreichen Abgleich oder der erfolgreichen Eingabe des Gerätecodes oder des Account-Passworts durch die Secure Enclave freigegeben werden. App-Entwicklern stehen APIs zur Verfügung, mit denen geprüft werden kann, ob der Benutzer einen Code oder ein Passwort festgelegt hat, bevor Face ID oder Touch ID, ein Gerätecode oder ein Passwort angefordert wird, um Objekte im [Schlüsselbund](#) zu entsperren. App-Entwickler haben folgende Möglichkeiten:

- Sie können vorgeben, dass API-Operationen für die Authentifizierung weder auf das Passwort einer App noch auf den Code für das Gerät zurückgreifen. Sie können abfragen, ob ein Benutzer registriert ist, sodass Face ID oder Touch ID als zweiter Faktor in sicherheitskritischen Apps verwendet werden kann.
- Sie können ECC-Schlüssel (Elliptic Curve Cryptography, elliptische Kurvenkryptografie) in der Secure Enclave generieren und verwenden, die durch Face ID oder Touch ID geschützt werden können. Operationen mit diesen Schlüsseln erfolgen stets in der Secure Enclave, nachdem diese die Nutzung autorisiert hat.

Tätigen und Bestätigen von Käufen

Face ID oder Touch ID können auch so konfiguriert werden, dass damit Käufe im iTunes Store, im App Store, in Apple Books usw. bestätigt werden können, ohne dass Benutzer das Passwort zu ihrer Apple-ID eingeben müssen. Wird ein Kauf getätigt, verifiziert die Secure Enclave, dass die biometrische Authentifizierung erfolgte, und stellt erst dann die ECC-Schlüssel bereit, mit der die Store-Anfrage signiert wird.

Gesicherte Absicht und Verbindungen zur Secure Enclave

Mit der gesicherten Absicht ist es möglich, eine Absicht des Benutzers ohne Interaktion mit dem Betriebssystem oder des Anwendungsprozessors zu bestätigen. Hierfür ist eine physische Verbindung von einer physischen Taste zur Secure Enclave notwendig, die in folgenden Geräten verfügbar ist:

- iPhone X oder neuer
- Apple Watch Series 1 oder neuer
- iPad Pro (alle Modelle)
- iPad Air (2020)
- Mac-Computer mit Apple Chips

Mit dieser Verbindung können Benutzer die Durchführung von Vorgängen auf eine Art und Weise bestätigen, die nicht durch eine mit Root- oder Kernel-Berechtigungen ausgeführte Software kompromittiert werden kann.

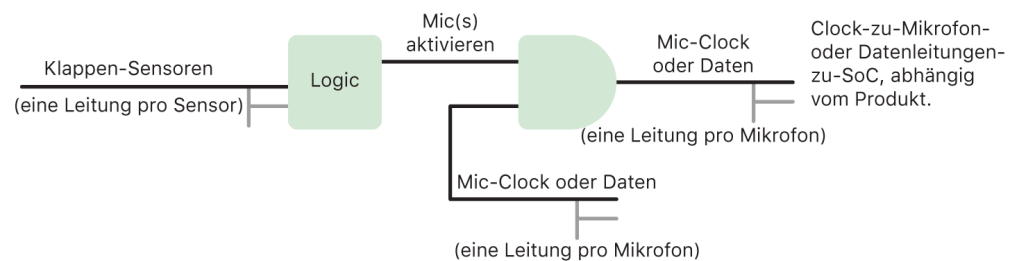
Diese Funktion wird verwendet, um die Absicht des Benutzers bei Transaktionen per Apple Pay und beim Abschluss des Kopplungsvorgangs eines Magic Keyboard mit Touch ID mit einem Mac mit Apple Chips zu bestätigen. Bei Aufforderung durch die Benutzeroberfläche kann durch zweimaliges Drücken der entsprechenden Taste (für Face ID) oder durch Scannen des Fingerabdrucks (für Touch ID) die Absicht des Benutzers bestätigt werden. Weitere Informationen sind unter [Absichern von Käufen mit Apple Pay](#) zu finden. Ein ähnlicher Mechanismus – der auf der Secure Enclave und der T2-Firmware basiert – wird auf MacBook-Modellen mit Apple T2 Security Chip (ohne Touch Bar) unterstützt.

Hardwareabschaltung des Mikrofons

Alle Mac-Notebooks mit Apple Chips oder mit einem Apple T2 Security Chip unterstützen einen Mechanismus für die Hardwareabschaltung, der bewirkt, dass das Mikrofon ausgeschaltet wird, sobald der Mobilcomputer zugeklappt bzw. das Gerät mit dem Deckel der Hülle zugedeckt wird. Auf allen 13" MacBook Pro- und MacBook Air-Notebooks mit T2-Chip, allen Mac-Notebooks mit T2-Chip ab 2019 und allen Mac-Notebooks mit Apple Chips ist diese Abschaltvorrichtung ausschließlich in der Hardware verankert. Die Abschaltvorrichtung bewirkt, dass jeglicher Software – selbst Software mit der Root- oder Kernel-Berechtigung von macOS und selbst Software auf dem T2-Chip oder sonstiger Firmware – verwehrt bleibt, das Mikrofon einzuschalten, solange der Mobilcomputer zugeklappt ist. (Die Kamera wird hardwareseitig nicht abgeschaltet, da ihr Blickfeld bei einem zugeklappten Mobilcomputer bzw. bei einem Gerät mit geschlossener Hülle ohnehin verdeckt ist.)

Ab 2020 verfügen auch iPad-Modelle über die Hardwareabschaltung für das Mikrofon. Wenn ein MFI-konformes Case (einschließlich der Cases, die von Apple angeboten werden) mit dem iPad verwendet und geschlossen wird, wird die Mikrofonverbindung hardwareseitig getrennt. Dies unterbindet die Verfügbarkeit von Audiodaten des Mikrofons für jegliche Software – selbst für Software mit Root- oder Kernel-Berechtigung in iPadOS oder für sonstige Firmware des Geräts.

Die in diesem Abschnitt angesprochenen Schutzvorkehrungen sind entsprechend dem Schaltplan unten direkt mit der Hardwarelogik integriert:



Bei jedem Gerät mit einer Hardwareabschaltung für das Mikrofon erkennen ein oder mehrere Sensoren im Displayteil, wenn der Deckel oder die Hülle physisch geschlossen wird; Anhaltspunkt hierfür ist eine durch die Interaktion bedingte physische Eigenschaft (z. B. der Halleffekt oder der Winkel des Scharniers). Im Falle eines Sensors, der kalibriert werden muss, werden bei der Fertigung des Geräts Parameter festgelegt; der Kalibrierungsprozess umfasst eine nicht aufhebbare Hardware Sperre für jegliche spätere Änderungen an den sensitiven Parametern des Sensors. Diese Sensoren geben ein direktes Hardwaresignal ab, das über ein einfaches Set nicht programmierbarer Hardwarelogik weitergeleitet wird. Diese Logik bewirkt das Entprellen, die Hysterese und/oder die Verzögerung um bis zu 500 ms, ehe das Mikrofon deaktiviert wird. Abhängig vom Produkt erfolgt die Implementierung dieses Signals, indem die Leitungen deaktiviert werden, die für den Datentransport zwischen dem Mikrofon und dem SoC (System on Chip) verwendet werden, oder indem eine der Eingangsleitungen zum Mikrofonmodul deaktiviert wird (z. B. die Taktleitung oder eine ähnliche effektive Steuerungsoption), sodass dieses aktiv bleiben kann.

Express-Karten mit Energiereserve

Wenn iOS nicht läuft, weil das iPhone aufgeladen werden muss, enthält die Batterie möglicherweise noch genügend Energie, um Express-Kartentransaktionen auszuführen. Unterstützte iPhone-Geräte unterstützen diese Funktion automatisch für:

- Als Express-ÖPNV-Karten ausgewiesene Zahlungs- oder ÖPNV-Karten
- Studierendenausweise, bei denen der Expressmodus aktiviert ist
- Autoschlüssel, bei denen der Expressmodus aktiviert ist
- Hausschlüssel, bei denen der Expressmodus aktiviert ist
- Hotel- oder Unternehmens-Zugangskarten, bei denen der Expressmodus aktiviert ist

Durch das Drücken der Seitentaste (bzw. durch das Drücken der Home-Taste auf iPhone SE-Geräten der 2. Generation) werden das Symbol für den Batterietiefstand und ein Text angezeigt, der darauf hinweist, dass Express-Karten zur Verwendung verfügbar sind. Der NFC-Controller führt Express-Kartentransaktionen unter den gleichen Bedingungen durch wie bei laufendem iOS. Allerdings wird nur durch eine haptische Benachrichtigung auf Transaktionen hingewiesen (es wird keine sichtbare Mitteilung oder Benachrichtigung angezeigt). Bei iPhone SE-Geräten der 2. Generation können einige Sekunden vergehen, bis eine abgeschlossene Transaktion als solche auf dem Bildschirm angezeigt wird. Diese Funktion ist nicht verfügbar, wenn ein von einem Standardbenutzer ausgelöster Abschaltvorgang ausgeführt wird.

Systemsecurity

Systemsecurity – Übersicht

Basierend auf den einzigartigen Funktionen der Hardware von Apple ist die Systemsecurity so konzipiert, dass der Zugriff auf Systemressourcen in Apple-Geräten überwacht wird, ohne die Benutzerfreundlichkeit zu beeinträchtigen. Die Systemsecurity umfasst den Startvorgang, Softwareaktualisierungen und den Schutz von Systemressourcen wie CPU, Speicher, Festplatte, Software und gespeicherte Daten.

Die neuesten Versionen von Apple-Betriebssystemen bieten höchste Sicherheit. Ein entscheidender Aspekt für die Apple-Sicherheit ist das *sichere Starten (Secure Boot)*; dieser Ansatz verhindert, dass das System während des Systemstarts durch Schadsoftware infiziert wird. Der sichere Startvorgang beginnt in der Hardware und erstellt mithilfe der Software eine „Chain of Trust“, bei der jeder Schritt überprüft, ob der nächste Schritt ordnungsgemäß funktioniert, bevor die Steuerung übergeben wird. Dieses Sicherheitsmodell unterstützt nicht nur den standardmäßigen Startvorgang von Apple-Geräten, sondern auch die verschiedenen Modi für die Wiederherstellung und zeitnahe Aktualisierung auf Apple-Geräten. Subkomponenten wie der T2-Chip und die Secure Enclave führen auch eigene sichere Startprozesse aus, um sicherzustellen, dass nur vertrauenswürdiger Code von Apple ausgeführt wird. Das Aktualisierungssystem wurde entwickelt, um Downgrade-Angriffe zu verhindern. Dies bedeutet, dass Geräte sich nicht zum Stehlen von Benutzerdaten auf eine ältere Version des Betriebssystems (von der der Angreifer weiß, wie er sie manipulieren kann) herabstufen lassen.

Darüber hinaus verfügen Apple-Geräte über Mechanismen für den Start- und Laufzeitschutz und wahren damit ihre eigene Integrität im laufenden Betrieb. Von Apple entwickelte Chips auf iPhone-, iPad-, Apple Watch-, Apple TV- und HomePod-Geräten sowie Mac Computer mit Apple Chips bieten eine gemeinsame Architektur, die die Integrität des Betriebssystems schützt. macOS bietet darüber hinaus erweiterte und konfigurierbare Sicherheitsfunktionen, die das abweichende Rechenmodell von macOS unterstützen, sowie Funktionen, die auf allen Mac-Computern mit Apple Chips Hardwareplattformen unterstützt werden.

Sicheres Starten

Startvorgang bei iOS- und iPadOS-Geräten

Jeder einzelne Schritt des Startvorgangs enthält von Apple kryptografisch signierte Komponenten, um die Integrität zu überprüfen, sodass der Startvorgang nur nach Verifizierung der „Chain of Trust“ fortgeführt wird. Zu diesen Komponenten gehören Bootloader, Kernel, Kernel-Erweiterungen und Baseband-Firmware für Mobilfunkgeräte. Dieser sichere Startvorgang sorgt dafür, dass die unteren Software-Ebenen nicht unbefugt manipuliert werden können.

Wenn ein iOS- oder iPadOS-Gerät eingeschaltet wird, führt dessen Anwendungsprozessor sofort Code aus dem schreibgeschützten Festspeicher aus, der **Boot-ROM** genannt wird. Dieser unveränderliche Code, *Hardware-Vertrauensanker* genannt, wird bei der Fabrikation des Chips festgelegt und ist implizit vertrauenswürdig. Der Code im Boot-ROM enthält den öffentlichen Schlüssel der Zertifizierungsstelle (CA) des Apple-Root-Zertifikats. Anhand dieses Root- oder Stammzertifikats wird überprüft, ob der **iBoot**-Bootloader von Apple signiert wurde, bevor die Genehmigung erteilt wird, diesen Code zu laden. Das ist der erste Schritt in der Chain of Trust, bei der bei jedem Schritt überprüft wird, ob der jeweils nächste Schritt von Apple signiert wurde. Bei Beendigung seiner Aufgaben prüft und führt iBoot den iOS- oder iPadOS-Kernel aus. Bei Geräten mit einem Prozessor der A9- oder einer früheren A-Reihe lädt und prüft der Boot-ROM einen zusätzlichen **Low-Level-Bootloader (LLB)**, der wiederum iBoot lädt und überprüft.

Ein Fehler beim Laden oder Überprüfen der folgenden Angaben wird abhängig von der Hardware unterschiedlich gehandhabt:

- *Boot-ROM kann LLB nicht laden (ältere Geräte):* **DFU-Modus (Device Firmware Upgrade)**
- *LLB oder iBoot:* **Wiederherstellungsmodus**

In beiden Fällen muss das Gerät per USB mit dem Finder (macOS 10.15 oder neuer) oder iTunes (macOS 10.14 oder älter) verbunden und auf die Werkseinstellungen zurückgesetzt werden.

Das **Boot Progress Register (BPR)** wird von der Secure Enclave verwendet, um den Zugriff auf Benutzerdaten in verschiedenen Modi einzuschränken. Es wird aktualisiert, bevor die folgenden Modi aktiviert werden:

- *DFU-Modus:* Vom Boot-ROM auf Geräten mit einem Apple A12 oder einem neueren SoC (System on Chip) festgelegt
- *Wiederherstellungsmodus:* Von iBoot auf Geräten mit Apple-Prozessoren der Reihe Apple A10, Apple S2 und neueren SoCs festgelegt

Bei Geräten mit Zugriff auf das Mobilfunknetz führt das Baseband-Subsystem einen ähnlichen sicheren Startvorgang aus, für den signierte Software und Schlüssel verwendet werden, die vom Baseband-Prozessor überprüft werden.

Die Secure Enclave führt außerdem einen sicheren Startvorgang aus, um sicherzustellen, dass ihre Software (sepOS) von Apple verifiziert und signiert ist.

Speichersichere iBoot-Implementierung

Bei iOS 14 und iPadOS 14 wurde die Toolchain des C-Compilers, der zum Erstellen des Bootloaders iBoot verwendet wird, von Apple unter dem speziellen Aspekt der Sicherheit modifiziert. Mit der modifizierten Toolchain wird Code implementiert, der bewirkt, dass für C-Programme typische Risiken hinsichtlich der Arbeitsspeicher- und Typsicherheit vermieden werden. Das hilft beispielsweise, die Schwachstellen in folgenden Klassen zu vermeiden:

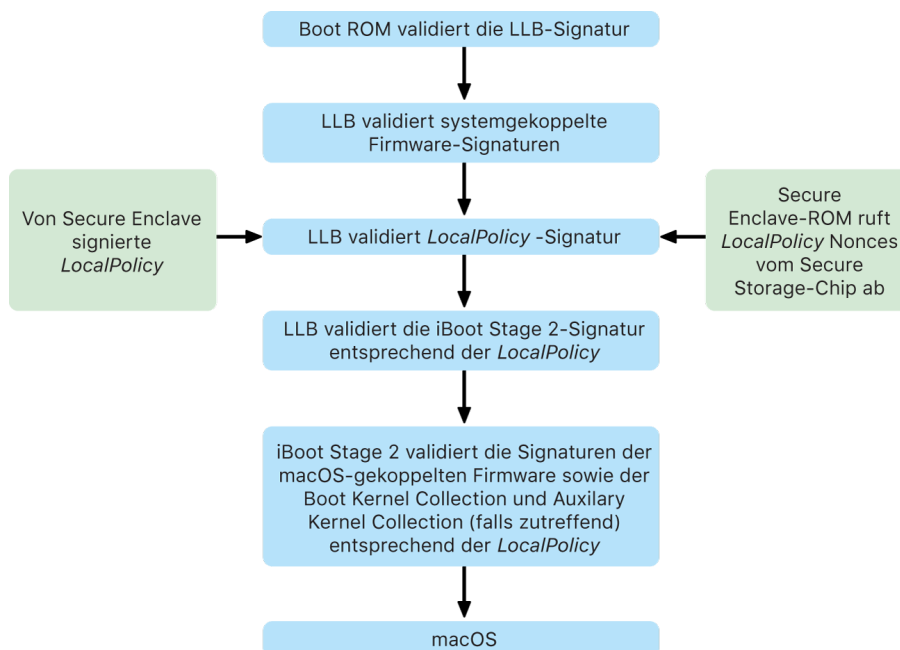
- Pufferüberläufe: Es wird sichergestellt, dass alle Zeiger über Begrenzungsinformationen verfügen, die beim Zugriff auf den Speicher überprüft werden.
- Heap-Missbrauch: Heap-Daten werden von den zugehörigen Metadaten getrennt und Fehlerbedingungen wie beispielsweise doppelte Freigaben werden präzise erkannt.
- „Type“-Verwechslung: Es wird sichergestellt, dass alle Zeiger Informationen zum Laufzeittyp umfassen, die bei „Pointer Cast“-Operationen überprüft werden.
- „Type“-Verwechslung: Durch Separieren aller dynamischen Speicherzuordnungen nach statischem Typ werden Verwechslungen infolge der Fehlerbedingung „Nutzung nach Freigabe“ (use-after-free) vermieden.

Diese Technologie steht bei iPhone-Geräten mit Apple A13 Bionic-Chip (oder neuer) sowie bei iPad-Geräten mit A14 Bionic-Chip zur Verfügung.

Mac-Computer mit Apple Chips

Startvorgang von einem Mac mit Apple Chips

Der Startvorgang beim Einschalten eines Mac mit Apple Chips ist dem Startvorgang eines iPhone- oder iPad-Geräts sehr ähnlich.



Im ersten Schritt in der Chain of Trust führt der Chip Code aus dem **Boot-ROM** aus. Der sichere macOS-Startvorgang auf einem Mac mit Apple Chips verifiziert nicht nur den Code des Betriebssystems selbst, sondern auch die Sicherheitsrichtlinien und sogar Kernel-Erweiterungen (unterstützt, aber nicht empfohlen), die von autorisierten Benutzern konfiguriert wurden.

Beim Starten von LLB (Low-Level-Bootstrap) werden die Signaturen überprüft und mit dem System gekoppelte Firmware-Objekte für Intra-SoC-Kerne wie beispielsweise die Controller für Festspeicher, Display, Systemverwaltung und Thunderbolt geladen. LLB ist auch verantwortlich für das Laden der LocalPolicy-Datei (der lokalen Richtlinie), bei der es sich um eine Datei handelt, die von der Secure Enclave signiert wird. Die LocalPolicy-Datei beschreibt die Konfiguration, die der Benutzer im Hinblick auf die Sicherheitsrichtlinien für den Systemstart und die Laufzeit ausgewählt hat. Diese lokale Richtlinie (LocalPolicy) weist dasselbe Datenstrukturformat auf wie alle anderen Startobjekte, wird aber nicht (wie Softwareaktualisierungen) von einem zentralen Apple-Server, sondern lokal mithilfe eines privaten Schlüssels signiert, der nur innerhalb der Secure Enclave des jeweiligen Computers verfügbar ist.

Um das Replay einer früheren LocalPolicy-Datei zu verhindern, muss LLB eine **Nonce** aus der an die Secure Enclave angeschlossenen Secure Storage-Komponente abrufen. Für diesen Zweck verwendet iBoot den Boot-ROM der Secure Enclave; iBoot stellt sicher, dass die Nonce in der LocalPolicy-Datei mit der Nonce in der Secure Storage-Komponente übereinstimmt. Auf diese Weise wird verhindert, dass eine ältere lokale Richtlinie, die möglicherweise für ein niedrigeres Sicherheitsniveau konfiguriert wurde, erneut auf das System angewendet wird, nachdem dessen Sicherheitsniveau per Upgrade erhöht wurde. Dies hat zum Ergebnis, dass der sichere Startvorgang auf einem Mac mit Apple Chips nicht nur vor Rollbacks von Betriebssystemversionen, sondern auch vor der Herabstufung von Sicherheitsrichtlinien schützt.

Die lokale Richtlinie erfasst, für welches Sicherheitsniveau das Betriebssystem konfiguriert ist – „Volle Sicherheit“, „Reduzierte Sicherheit“ oder „Permissive Sicherheit“.

- *Volle Sicherheit:* Das System verhält sich wie iOS und iPadOS und lässt nur das Starten von Software zu, die zum Zeitpunkt ihrer Installation als die neueste Version klassifiziert war.
- *Reduzierte Sicherheit:* LLB wird angewiesen, „globalen“ Signaturen zu vertrauen, die an das Betriebssystem gekoppelt sind. Dies ermöglicht es dem System, ältere Versionen von macOS auszuführen. Ältere Versionen von macOS weisen unweigerlich nicht behobene Sicherheitslücken auf; aus diesem Grund ist die Rede von „*reduzierter Sicherheit*“. Dies ist auch die Richtlinienebene, die erforderlich ist, um das Booten von Kernel-Erweiterungen zu unterstützen.
- *Permissive Sicherheit:* Das Verhalten des Systems ähnelt insofern dem Verhalten bei reduzierter Sicherheit, als für **iBoot** und darüber hinaus die Prüfung globaler Signaturen verwendet wird. Anders als bei der reduzierten Sicherheit wird iBoot aber angewiesen zu akzeptieren, dass bestimmte Startobjekte durch die Secure Enclave mit demselben Schlüssel signiert werden, mit dem die LocalPolicy-Datei signiert ist. Diese Richtlinienebene unterstützt das Erstellen, Signieren und Starten eigener XNU-Kernel durch Benutzer.

Wenn sich für LLB aus der lokalen Richtlinie ergibt, dass das ausgewählte Betriebssystem im Modus „Volle Sicherheit“ ausgeführt wird, evaluiert LLB die personalisierte Signatur für iBoot. Bei der Ausführung im Modus „Reduzierte Sicherheit“ oder „Permissive Sicherheit“ wird die globale Signatur evaluiert. Jeglicher Fehler bei der Signaturprüfung führt dazu, dass recoveryOS gestartet wird und Optionen für die Reparatur bereitgestellt werden.

Nach der Übergabe der Steuerung von LLB an iBoot werden an macOS gekoppelte Firmware-Objekte wie die Secure Neural Engine, der „Always On“-Prozessor und andere Firmware gestartet. iBoot überprüft auch die Informationen bezüglich der LocalPolicy-Datei, die von LLB an iBoot übergeben wird. Wenn aus der lokalen Richtlinie hervorgeht, dass eine Auxiliary Kernel Collection (AuxKC) vorhanden sein müsste, sucht iBoot im Dateisystem nach diesem Cache und stellt sicher, dass er in der Secure Enclave mit demselben Schlüssel wie in der LocalPolicy-Datei signiert wurde und dass der Hash-Wert mit dem in der LocalPolicy-Datei enthaltenen Hash-Wert übereinstimmt. Wenn die Prüfung des AuxKC erfolgreich ist, platziert iBoot diesen Cache zusammen mit der Boot Kernel-Sammlung im Arbeitsspeicher, bevor die gesamte Speicherregion mit Boot Kernel-Sammlung und AuxKC per [System Coprocessor Integrity Protection \(SCIP\)](#) gesperrt wird. Wenn aus der Richtlinie hervorgeht, dass ein AuxKC vorhanden sein müsste, er aber nicht gefunden wird, setzt das System das Starten von macOS ohne diesen Cache fort. iBoot ist auch verantwortlich für die Prüfung des Root-Hash-Werts des Signed System Volume (SSV), damit die Integrität des Dateisystems, das der Kernel aktiviert, in vollem Umfang gewährleistet ist.

Startmodi eines Mac mit Apple Chips

Mac-Computer mit Apple Chips verfügen über die unten genannten Startmodi.

Modus	Taste(n)	Beschreibung
macOS	Im ausgeschalteten Zustand muss die Taste „Ein/Aus“ gedrückt und wieder freigegeben werden.	<ol style="list-style-type: none">1. Boot-ROM übergibt die Steuerung an LLB.2. LLB lädt mit dem System gekoppelte Firmware-Objekte und die LocalPolicy-Datei für die ausgewählte macOS-Version.3. LLB verankert eine Angabe im Boot Progress Register (BRP), die besagt, dass macOS gestartet wird, und übergibt die Steuerung an iBoot.4. iBoot lädt die mit macOS gekoppelten Firmware-Objekte, den statischen Cache für die Vertrauensstellung („Static Trust Cache“), die Gerätestruktur („Device Tree“) und die Boot Kernel-Sammlung.5. Sofern die lokale Richtlinie dies zulässt, lädt iBoot die Auxiliary Kernel Collection (AuxKC) mit den Kernel-Erweiterungen (KEXT) von Drittanbietern.6. iBoot prüft den Hash-Wert der Stammsignatur für das Signed System Volume (SSV), sofern diese Prüfung nicht durch die lokale Richtlinie deaktiviert wurde.

Modus	Taste(n)	Beschreibung
Gekoppeltes recoveryOS	Im ausgeschalteten Zustand muss die Taste „Ein/Aus“ („Power Button“) gedrückt gehalten werden.	<ol style="list-style-type: none"> 1. Boot-ROM übergibt die Steuerung an LLB. 2. LLB lädt mit dem System gekoppelte Firmware-Objekte und die LocalPolicy-Datei für die recoveryOS-Software. 3. LLB verankert eine Angabe im Boot Progress Register (BRP), die besagt, dass die gekoppelte recoveryOS-Software gestartet wird, und übergibt die Steuerung an iBoot für das gekoppelte recoveryOS. 4. iBoot lädt die mit macOS gekoppelten Firmware-Objekte, den Cache für die Vertrauensstellung („Trust Cache“), die Gerätestruktur („Device Tree“) und die Boot Kernel-Sammlung. 5. Wenn das Starten des gekoppelten recoveryOS fehlschlägt, wird der Versuch unternommen, in das Fallback-recoveryOS zu starten. <p><i>Hinweis:</i> Ein Herabstufen der Sicherheit ist laut der LocalPolicy-Datei für das gekoppelte recoveryOS nicht zulässig.</p>
recoveryOS als Fallback	Im ausgeschalteten Zustand muss die Taste „Ein/Aus“ („Power Button“) zweimal kurz hintereinander gedrückt und niedergehalten werden.	<ol style="list-style-type: none"> 1. Boot-ROM übergibt die Steuerung an LLB. 2. LLB lädt mit dem System gekoppelte Firmware-Objekte und die LocalPolicy-Datei für die recoveryOS-Software. 3. LLB verankert eine Angabe im Boot Progress Register (BRP), die besagt, dass die gekoppelte recoveryOS-Software gestartet wird, und übergibt die Steuerung an iBoot für recoveryOS. 4. iBoot lädt die mit macOS gekoppelten Firmware-Objekte, den Cache für die Vertrauensstellung („Trust Cache“), die Gerätestruktur („Device Tree“) und die Boot Kernel-Sammlung. <p><i>Hinweis:</i> Ein Herabstufen der Sicherheit ist laut der LocalPolicy-Datei für das gekoppelte recoveryOS nicht zulässig.</p>
Sicherer Modus (Safe mode)	Nachdem recoveryOS in der oben beschriebenen Weise gestartet wurde, muss die Umschalttaste gedrückt und das gewünschte Startvolume ausgewählt werden.	<ol style="list-style-type: none"> 1. recoveryOS wird in der oben beschriebenen Weise gestartet. 2. Durch die Auswahl eines Volumes bei gedrückter Umschalttaste wird das Programm BootPicker angewiesen, die betreffende macOS-Software für das Starten in normaler Weise vorzumerken. Darüber hinaus wird die Variable nvram definiert, die iBoot anweist, den AuxKC beim nächsten Startvorgang nicht zu laden. 3. Das System wird mit der Software auf dem ausgewählten Volume neu gestartet; der AuxKC wird von iBoot aber nicht geladen.

Einschränkungen für das gekoppelte recoveryOS

In macOS 12.0.1 (oder neuer) installiert jede neue macOS-Installation eine gekoppelte Version von recoveryOS in der zugehörigen APFS-Volumengruppe. Dieses Design ist den Benutzern von Intel-basierten Mac-Computern bekannt, aber auf einem Mac mit Apple Chips bietet es zusätzliche Sicherheits- und Kompatibilitätsgarantien. Da jede macOS-Installation nun ein spezielles gekoppeltes recoveryOS besitzt, ist sichergestellt, dass nur dieses speziell gekoppelte recoveryOS eine Herabstufung der Sicherheit durchführen kann. Auf diese Weise sind die Installationen neuerer macOS-Versionen vor Manipulationen durch ältere macOS-Versionen – und umgekehrt – geschützt.

Die Kopplungseinschränkungen werden folgendermaßen umgesetzt:

- Alle Installationen von macOS 11 sind mit dem recoveryOS gekoppelt. Wenn eine macOS 11-Installation als standardmäßiges Startsystem ausgewählt wird, kann das entsprechende recoveryOS beim Hochfahren eines Mac mit Apple Chips gestartet werden, indem der Ein-/Ausschalter gedrückt wird. Das recoveryOS kann die Herabstufung der Sicherheitseinstellungen für eine beliebige macOS 11-Installation, aber nicht für eine macOS 12.0.1-Installation durchführen.
- Wenn eine macOS 12.0.1-Installation (oder neuer) als standardmäßiges Startsystem ausgewählt wird, kann das zugehörige gekoppelte recoveryOS beim Hochfahren des Mac gestartet werden, indem der Ein-/Ausschalter gedrückt wird. Das gekoppelte recoveryOS kann die Herabstufung der Sicherheitseinstellungen für die gekoppelte macOS-Installation, aber nicht für eine andere macOS-Installation durchführen.

Das gekoppelte recoveryOS einer macOS-Installation kann nur gestartet werden, wenn die Installation als standardmäßiges Startsystem ausgewählt wurde. Dies ist in der Systemeinstellung „Startvolume“ möglich oder durch das Auswählen eines Volumes bei gedrückter Option-Taste im recoveryOS.

Hinweis: Das Fallback-recoveryOS kann keine Herabstufungen für macOS-Installationen durchführen.

Sicherheitsrichtlinie „Startup Disk“ bei Mac-Computern mit Apple Chips

Übersicht

Im Gegensatz zu einem Intel-basierten Mac werden bei einem Mac mit Apple Chips die Sicherheitsrichtlinien auf der Ebene einzelner Betriebssysteme festgelegt. Das bedeutet, dass auf demselben Mac mehrere installierte Instanzen von macOS mit unterschiedlichen Versionen und Sicherheitsrichtlinien unterstützt werden. Aus diesem Grund wurde das Startersicherheitsdienstprogramm um eine *Option zur Auswahl des gewünschten Betriebssystems* ergänzt.

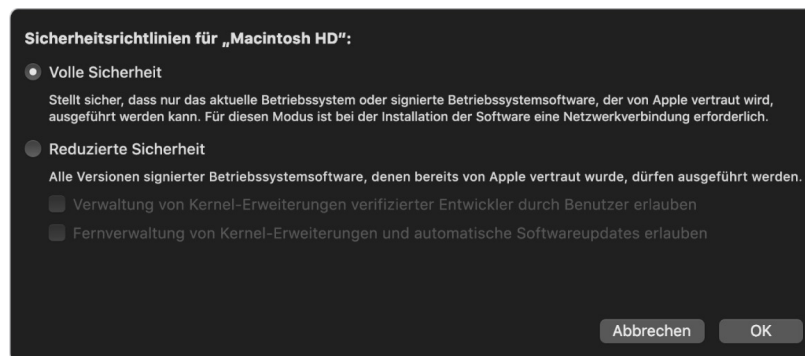


Auf einem Mac mit Apple Chips gibt das Systemsicherheitsdienstprogramm den generellen, vom Benutzer konfigurierten Sicherheitsstatus von macOS an, wie etwa das Starten einer Kernel-Erweiterung oder die Konfiguration des Systemintegritätsschutzes (System Integrity Protection, SIP). Wenn das Ändern der Sicherheitseinstellung zur Folge hat, dass die Sicherheit deutlich herabgesetzt und das System anfälliger für Aktionen wird, die es kompromittieren können, müssen Benutzer den recoveryOS-Prozess starten, indem sie die Taste „Ein/Aus“ („Power Button“) gedrückt halten, um diese Änderung vorzunehmen. (Das entsprechende Signal kann dann nicht durch Schadsoftware ausgelöst werden, sondern nur von einer Person, die physischen Zugang zum Gerät hat.) Aus diesem Grund werden auf einem Mac mit Apple Chips keine Firmware-Passwörter vorausgesetzt (oder unterstützt) – alle kritischen Änderungen werden bereits von der Benutzerauthentifizierung abgesichert. Weitere Informationen zu SIP sind unter [Systemintegritätsschutz](#) zu finden.

Die Einstellungen „Volle Sicherheit“ und „Reduzierte Sicherheit“ können mit dem Startersicherheitsdienstprogramm von recoveryOS festgelegt werden. Die Einstellung „Permissive Sicherheit“ kann dagegen nur per Befehlszeilenprogramm durch Benutzer festgelegt werden, die sich des Risikos bewusst sind, dass mit dieser Einstellung ihr Mac viel an Sicherheit einbüßt.

Richtlinie „Volle Sicherheit“

„Volle Sicherheit“ ist die Standardeinstellung, die ein Verhalten wie bei iOS und iPadOS zur Folge hat. Wenn Software heruntergeladen und für die Installation vorbereitet wird, verwendet macOS nicht die zur Software gehörige globale Signatur, sondern fordert bei dem Apple-Signierungsserver, der auch für iOS und iPadOS verwendet wird, eine neue „personalisierte“ Signatur an. Eine Signatur gilt als personalisiert, wenn sie die [Exclusive Chip Identification \(ECID\)](#) – eine eindeutige ID, die in diesem Fall für die Apple-CPU spezifisch ist – als Teil der Signierungsanfrage umfasst. Die vom Signierungsserver zurückgegebene Signatur ist dann eindeutig und nur für diese spezifische Apple-CPU nutzbar. Wenn die Richtlinie „Volle Sicherheit“ aktiv ist, stellen der Boot-ROM und LLB sicher, dass eine bestimmte Signatur nicht nur von Apple signiert, sondern für den spezifischen Mac signiert wird. Auf diese Weise wird die jeweilige Version von macOS an diesen Mac gebunden.



Die Verwendung eines Online-Signierungsservers bietet auch mehr Schutz vor Rollback-Angriffen als typische Ansätze mit globalen Signaturen. In einem globalen Signierungssystem könnte die Sicherheitsepoche mehrfach fortgeführt worden sein, ohne dass dies einem System, das nicht die neueste Firmware kennt, bekannt ist. Beispielsweise akzeptiert aktuell ein Computer in der Annahme, in Sicherheitsepoche 1 zu sein, Software von Sicherheitsepoche 2, auch wenn 5 die tatsächliche aktuelle Sicherheitsepoche ist. Mit dem Online-Signierungssystem bei Apple Chips kann der Signierungsserver das Erstellen von Signaturen für jegliche Software ablehnen, die sich nicht in der neuesten Sicherheitsepoche befindet.

Hinzu kommt, dass ein Angreifer, der nach einer Änderung der Sicherheitsepoche eine Schwachstelle entdeckt, nicht einfach die anfällige Software einer früheren Epoche von System A nehmen und auf System B anwenden kann, um es anzugreifen. Dadurch dass die anfällige Software einer älteren Epoche für System A personalisiert wurde, ist sie nicht übertragbar und kann nicht für einen Angriff auf System B verwendet werden. Das Zusammenwirken all dieser Mechanismen bietet eine stärkere Gewähr dafür, dass Angreifer nicht gezielt anfällige Software auf einem Mac platzieren können, um die Schutzvorkehrungen der neuesten Software zu umgehen. Ein Benutzer, der über den Namen und das Passwort eines Administrators für den Mac verfügt, kann jederzeit die Sicherheitsrichtlinie auswählen, die sich für seine Zwecke am besten eignet.

Richtlinie „Reduzierte Sicherheit“

Die reduzierte Sicherheit ähnelt im Verhalten der „mittleren“ Sicherheit bei Intel-basierten Mac-Computern mit einem T2-Chip; hier generiert der jeweilige Lieferant (in diesem Fall Apple) eine digitale Signatur für den Code, die belegt, dass der Code von dem betreffenden Lieferanten stammt. Angreifer haben so keine Möglichkeit, nicht signierten Code einzuschleusen. Diese Signatur wird von Apple als „globale“ Signatur bezeichnet, weil sie auf einem beliebigen Mac für beliebig lange Zeit verwendet werden kann, sofern auf diesem Mac die reduzierte Sicherheit konfiguriert ist. Die reduzierte Sicherheit selbst bietet keinen Schutz vor Rollback-Angriffen (obwohl nicht autorisierte Änderungen am Betriebssystem dazu führen können, dass nicht mehr auf Benutzerdaten zugegriffen werden kann). Weitere Informationen sind unter [Kernel-Erweiterungen auf einem Mac mit Apple Chips](#) zu finden.



Außer Benutzern zu ermöglichen, auch ältere Versionen von macOS ausführen zu können, ist die reduzierte Sicherheit Voraussetzung für andere Aktionen, die ein Sicherheitsrisiko für das System mit sich bringen, beispielsweise das Hinzufügen von Kernel-Erweiterungen (KEXTs) von Drittanbietern. Kernel-Erweiterungen haben dieselben Berechtigungen wie der Kernel, weshalb Schwachstellen und Sicherheitslücken in Kernel-Erweiterungen von Drittanbietern dazu führen können, dass das gesamte Betriebssystem kompromittiert wird. Aus diesem Grund sind Entwickler angehalten, Systemerweiterungen zum Einsatz zu bringen, bevor bei künftigen Mac-Computern mit Apple Chips die Unterstützung für Kernel-Erweiterungen durch macOS entfernt wird. Auch wenn Kernel-Erweiterungen von Drittanbietern aktiviert werden, können sie nicht „on demand“ in den Kernel geladen werden. Die Kernel-Erweiterungen werden vielmehr in einer Auxiliary Kernel Collection (AuxKC) zusammengeführt, deren Hash in der LocalPolicy-Datei gesichert wird. Deshalb ist ein Neustart erforderlich. Weitere Informationen zur AuxKC-Generierung sind unter [Kernel-Erweiterungen bei macOS](#) zu finden.

Richtlinie „Permissive Sicherheit“

Der Modus „Permissive Sicherheit“ ist für Benutzer gedacht, die sich des Risikos bewusst sind, dass dieser Modus die Sicherheit ihres Mac-Computers deutlich beeinträchtigt. Dieser Modus unterscheidet sich vom Modus „Ohne Sicherheit“ bei Intel-basierten Mac-Computern mit T2-Chip. Bei der permissiven Sicherheit erfolgt die Signaturprüfung zwar weiterhin über die gesamte Startkette, durch das Herabstufen der Richtlinie auf „Permissiv“ wird iBoot aber angewiesen, lokal in der Secure Enclave signierte Startobjekte zu akzeptieren, beispielsweise eine vom Benutzer generierte Boot Kernel-Sammlung, die aus einem angepassten XNU-Kernel erstellt wird. Auf diese Weise bietet der Modus „Permissive Sicherheit“ architekturbasiert die Möglichkeit, einen beliebigen Kernel für ein „Betriebssystem ohne jegliche Vertrauenswürdigkeit“ auszuführen. Wenn eine angepasste Boot Kernel-Sammlung oder ein Betriebssystem ohne jegliche Vertrauenswürdigkeit geladen wird, stehen bestimmte Entschlüsselungsschlüssel nicht mehr zur Verfügung. Dadurch wird verhindert, dass ein nicht vertrauenswürdige Betriebssystem auf Daten vertrauenswürdiger Betriebssysteme zugreift.

Wichtig: Apple bietet keinen Support für angepasste XNU-Kernel und stellt keine bereit.



Die permissive Sicherheit unterscheidet sich auf eine weitere Weise vom Modus „Ohne Sicherheit“ auf Intel-basierten Mac-Computern mit T2-Chip. Sie ist eine Voraussetzung für einige Herabstufungen der Sicherheit, die in der Vergangenheit unabhängig voneinander gesteuert werden konnten. Das wichtigste Beispiel ist, dass Benutzer zum Deaktivieren des Systemintegritätsschutzes (System Integrity Protection, SIP) auf einem Mac mit Apple Chips explizit bestätigen müssen, dass sie das System in den Modus „Permissive Sicherheit“ versetzen möchten. Dies wird gefordert, da das Deaktivieren von SIP das System immer schon in einen Zustand versetzt hat, in dem der Kernel sehr viel einfacher kompromittiert werden kann. Wird SIP auf einem Mac mit Apple Chips deaktiviert, wird insbesondere auch das Erzwingen der KEXT-Signatur(en) im Zuge der AuxKC-Generierung deaktiviert, was dazu führt, dass beliebige KEXT-Erweiterungen in den Kernel-Arbeitspeicher geladen werden können. Auf Mac-Computern mit Apple Chips wurde SIP außerdem insofern verbessert, als der Richtlinienpeicher aus dem NVRAM in die LocalPolicy-Datei verlagert wurde. So wird erreicht, dass das Deaktivieren von SIP durch einen Benutzer authentifiziert werden muss, der Zugriff auf den Signierschlüssel der lokalen Richtlinie hat; diesen Zugriff erhält er aber nur während des recoveryOS-Prozesses, den er durch Gedrückthalten der Taste „Ein/Aus“ („Power Button“) initiiert. Dadurch wird es für einen auf die Software abzielenden Angreifer und selbst für einen Angreifer, der physisch anwesend ist, deutlich schwerer, SIP zu deaktivieren.

Im Startersicherheitsdienstprogramm ist eine Herabstufung in den Modus „Permissive Sicherheit“ nicht möglich. Benutzer können nur Befehlszeilendienstprogramme wie `csrutil` (zur Deaktivierung der SIP) in Terminal ausführen, um die Herabstufung vorzunehmen. Nach der Herabstufung durch den Benutzer wird dieser Modus im Startersicherheitsdienstprogramm angezeigt, sodass Benutzer leicht einen Sicherheitsmodus auswählen können, der mehr Sicherheit bietet.

Hinweis: Bei einem Mac mit Apple Chips wird keine bestimmte Richtlinie für das Starten von Medien vorausgesetzt (oder unterstützt), da technisch gesehen alle Startvorgänge lokal erfolgen. Will ein Benutzer den Startvorgang von einem externen Speichermedium ausführen, muss die entsprechende Version des Betriebssystems zunächst durch einen authentifizierten Neustart über recoveryOS personalisiert werden. Auf diese Weise wird eine LocalPolicy-Datei auf dem internen Speichermedium erstellt, die für den vertrauenswürdigen Startvorgang des Betriebssystems auf dem internen Speichermedium verwendet wird. Das bedeutet, dass die Konfiguration für das Starten von Betriebssystemen auf externen Speichermedien immer explizit für das jeweilige Betriebssystem aktiviert wird und bereits eine Autorisierung durch den Benutzer erfordert. Deshalb ist keine weitere Sicherheitskonfiguration erforderlich.

Erstellung und Verwaltung von Signaturschlüsseln für die lokale Richtlinie (LocalPolicy)

Erstellung

Wenn macOS das erste Mal werksseitig installiert oder eine kabelgebundene Löschung/Installation durchgeführt wird, führt der Mac Code von der temporären RAM-Disk für die Wiederherstellung aus, um den Standardzustand zu initialisieren. Im Zuge dieses Prozesses erstellt die Wiederherstellungsumgebung ein neues Paar privater und öffentlicher Schlüssel, die in der Secure Enclave gespeichert werden. Der private Schlüssel wird als *OIK (Owner Identity Key)* bezeichnet. Wenn ein OIK bereits vorhanden ist, wird er im Rahmen dieses Prozesses zerstört. Die Wiederherstellungsumgebung initialisiert den für die Aktivierungssperre verwendeten Schlüssel. Dieser wird als *UIK (User Identity Key)* bezeichnet. Zu diesem Prozess, der nur auf Mac-Computern mit Apple Chips ausgeführt wird, gehört die Abfrage der UIK-Zertifizierung für die Aktivierungssperre. Hierbei handelt es sich um eine Reihe von vorgesehenen Beschränkungen, die zum Validierungszeitpunkt auf die LocalPolicy angewendet werden. Wenn das Gerät keinen für die Aktivierungssperre zertifizierten UIK erhält (z. B. weil das Gerät aktuell mit einem „Wo ist?“-Account verknüpft und als verloren markiert wurde), kann mit dem Erstellen einer lokalen Richtlinie nicht fortgefahren werden. Wenn dem Gerät ein *ucrt (user identity certificate)* ausgestellt wird, enthält dieses „ucrt“ vom Server erzwungene und vom Benutzer festgelegte Richtlinieneinschränkungen in einer X.509 v3-Erweiterung.

Wenn eine Aktivierungssperre/ucrt erfolgreich abgerufen wurde, werden die erhaltenen Daten serverseitig in einer Datenbank gespeichert und auch an das Gerät zurückgesendet. Nachdem das Gerät ein ucrt erhalten hat, wird eine Zertifizierungsanfrage für den öffentlichen Schlüssel, der dem OIK entspricht, an den BAA-Server (Basic Attestation Authority) gesendet. Die BAA verifiziert die OIK-Zertifizierungsanfrage mit dem öffentlichen Schlüssel des in der Datenbank gespeicherten ucrt. Wenn die BAA die Zertifizierung verifizieren kann, stellt sie für den öffentlichen Schlüssel ein Zertifikat aus und gibt das von der BAA signierte OIC (*Owner Identity Certificate*) zurück, das die im ucrt gespeicherten Einschränkungen enthält. Das OIC wird an die Secure Enclave zurückgesendet. Von da an wird das OIC immer zur Image4-Datei hinzugefügt, wenn die Secure Enclave eine neue lokale Richtlinie (LocalPolicy) signiert. LLB vertraut dem BAA-Rootzertifikat grundsätzlich, weshalb auch dem OIC vertraut wird. Dies hat wiederum zur Folge, dass der gesamten LocalPolicy-Signatur vertraut wird.

Einschränkungen der Remote-Richtlinie (RemotePolicy)

Alle Image4-Dateien – nicht nur die lokalen Richtlinien – umfassen Einschränkungen für die Auswertung des Image4-Manifests. Diese sind im untergeordneten Zertifikat mittels spezieller Objektkennungen (OIDFs, Object Identifiers) kodiert. Die Image4-Prüfbibliothek ruft im Zuge der Signaturprüfung die spezielle OID für die Zertifikateinschränkung ab und wertet die darin enthaltenen Einschränkungen auf mechanischem Weg aus. Die Einschränkungen weisen folgende Form auf:

- X muss vorhanden sein
- X darf nicht vorhanden sein
- X muss einen bestimmten Wert aufweisen

So enthalten beispielsweise bei „personalisierten“ Signaturen die Zertifikateinschränkungen die Anweisung „ECID muss vorhanden sein“, während sie bei „globalen“ Signaturen die Anweisung „ECID darf nicht vorhanden sein“ enthalten. Durch diese Einschränkungen wird sichergestellt, dass alle Image4-Dateien, die mit einem bestimmten Schlüssel signiert sind, bestimmten Anforderungen genügen müssen, damit die Generierung eines fehlerhaft signierten Image4-Manifests vermieden wird.

Im Kontext einzelner LocalPolicy-Dateien werden diese Image4-Zertifikateinschränkungen als „*RemotePolicy*“ (Remote-Richtlinie) bezeichnet. Es sind unterschiedliche Remote-Richtlinien für die lokalen Richtlinien von verschiedenen Startumgebungen möglich. Die Remote-Richtlinie wird dazu verwendet, die lokale Richtlinie für die recoveryOS-Software dahingehend zu beschränken, dass sich beim Starten der recoveryOS-Software die recoveryOS-Software stets nur so verhalten kann, als gelte die Festlegung „Volle Sicherheit“. Dies erhöht das Vertrauen in die Integrität der Startumgebung der recoveryOS-Software als einem Ort, an dem die Richtlinie geändert werden kann. Die Remote-Richtlinie schränkt die lokale Richtlinie dahingehend ein, dass sie die ECID des Mac, auf der sie generiert wurde, und den spezifischen „rpnh“-Wert (Remote Policy Nonce Hash) enthält, der in der Secure Storage-Komponente auf dem betreffenden Mac gespeichert ist. Der „rpnh“-Wert – und somit auch die Remote-Richtlinie – wird nur geändert, wenn Aktionen bezüglich der Funktion „Wo ist?“ und der Aktivierungssperre erfolgen, zum Beispiel die Registrierung, das Aufheben der Registrierung oder das Sperren oder Löschen per Fernzugriff. Die Einschränkungen für die Remote-Richtlinie werden zum Ausstellungszeitpunkt des UIK-Zertifikats (User Identity Key) ermittelt und festgelegt und das ausgestellte ucr (User Identity Certificate) damit signiert. Einige Einschränkungen der Remote-Richtlinie wie ECID, ChipID und BoardID werden vom Server festgelegt. Dadurch soll verhindert werden, dass ein Gerät LocalPolicy-Dateien für andere Geräte signieren kann. Andere Einschränkungen der Remote-Richtlinie können vom Gerät festgelegt werden, um zu verhindern, dass die Sicherheit der lokalen Richtlinie herabgestuft wird, ohne dass dafür die notwendige lokale Authentifizierung für den Zugriff auf den aktuellen OIK und die Remote-Authentifizierung für den Account, der die Aktivierungssperre aktiviert hat, durchgeführt wurde.

Inhalt einer LocalPolicy-Datei für Mac-Computer mit Apple Chips

Die lokale Richtlinie (LocalPolicy-Datei) ist eine von der Secure Enclave signierte Image4-Datei. Image4 ist ein in ASN.1 (Abstract Syntax Notation One) DER-codiertes Datenstrukturformat, mit dem für Apple-Plattformen Informationen über Objekte beschrieben werden, die für den sicheren Startvorgang relevant sind. Bei einem auf Image4 basierten Modell für den sicheren Startvorgang werden Sicherheitsrichtlinien zum Zeitpunkt der Softwareinstallation angefordert. Zu diesem Zweck wird eine Signierungsanfrage an einen zentralen Apple-Signierungsserver gesendet. Wenn die Richtlinie akzeptabel ist, sendet der Signierungsserver eine signierte Image4-Datei zurück, die eine Vielzahl von jeweils vier Zeichen langen Codesequenzen (4CC) enthält. Diese signierten Image4-Dateien und 4CCs werden während des Systemstarts von Software wie dem Boot-ROM oder LLB evaluiert.

Übergabe des Eigentümerstatus zwischen Betriebssystemen

Der Zugriff auf den OIK (Owner Identity Key) wird als „Eigentümerstatus“ bezeichnet. Der Eigentümerstatus ist notwendig, um es Benutzern zu erlauben, die LocalPolicy-Datei nach dem Ändern der Richtlinie oder Software erneut zu signieren. Der OIK wird mit derselben Schlüsselhierarchie geschützt, die unter [Sealed Key Protection \(SKP\)](#) beschrieben ist. Dabei wird der OIK mit demselben KEK (Key Encryption Key) wie der VEK (Volume Encryption Key) geschützt. Das bedeutet, dass er im Normalfall sowohl mit den Benutzerpasswörtern als auch den Spezifikationen des Betriebssystems und der Richtlinie geschützt ist. Es gibt nur einen OIK für alle Betriebssysteme auf dem Mac. Aus diesem Grund ist beim Installieren eines zweiten Betriebssystems die ausdrückliche Zustimmung des Benutzers auf dem ersten Betriebssystem erforderlich, um den Eigentümerstatus an den Benutzer auf dem zweiten Betriebssystem zu übergeben. Allerdings existieren für das zweite Betriebssystem noch keine Benutzer, wenn das Installationsprogramm auf dem ersten Betriebssystem ausgeführt wird. Benutzer werden in Betriebssystemen normalerweise erst generiert, wenn das Betriebssystem gestartet und der Systemassistent ausgeführt wurde. Daher sind zwei neue Schritte erforderlich, wenn ein zweites Betriebssystem auf einem Mac mit Apple Chips installiert wird:

- LocalPolicy-Datei für das zweite Betriebssystem erstellen
- Benutzerinstallation („Install User“) für die Übergabe des Eigentümerstatus vorbereiten

Wenn ein Installationsassistent gestartet und eine Installation auf einem zweiten leeren Volume durchgeführt wird, wird der Benutzer gefragt, ob er einen Benutzer vom aktuellen Volume auf das zweite Volume als erster Benutzer kopieren möchte. Wenn der Benutzer dies bestätigt, wird die Benutzerinstallation (eigentlich ein vom Passwort und den Hardwarechlüsseln des ausgewählten Benutzers abgeleiteter KEK) für die Verschlüsselung des OIK verwendet, wenn dieser an das zweite Betriebssystem übergeben wird. Im Anschluss fragt der Installationsassistent des zweiten Betriebssystems nach dem Passwort dieses Benutzers, um den Zugriff auf den OIK in der Secure Enclave durch das neue Betriebssystem zu erlauben. Wenn sich Benutzer gegen das Kopieren eines Benutzers entscheiden, wird die Benutzerinstallation trotzdem auf dieselbe Weise ausgeführt, allerdings wird ein leeres Passwort anstatt eines Benutzerpassworts verwendet. Der zweite Prozess ist für bestimmte Systemadministrationsszenarien konzipiert. Benutzer, die Installationen auf verschiedenen Volumes verwenden und die Übergabe des Eigentumsstatus auf die sicherste Art und Weise durchführen möchten, sollten immer einen Benutzer vom ersten Betriebssystem auf das zweite Betriebssystem kopieren.

LocalPolicy-Datei auf einem Mac mit Apple Chips

Bei einem Mac mit Apple Chips wurde die Steuerung für die lokale Sicherheitsrichtlinie an eine Anwendung übergeben, die in der Secure Enclave ausgeführt wird. Diese Software kann ausgehend von den Anmeldeinformationen des Benutzers und dem Startmodus der primären CPU feststellen, wer die Sicherheitsrichtlinie ändern kann und aus welcher Startumgebung heraus dies möglich ist. Dadurch kann verhindert werden, dass Schadsoftware die Steuerung für die Sicherheitsrichtlinie gegen den Benutzer einsetzen kann, indem der Benutzer herabgestuft wird, um danach zusätzliche Rechte und Berechtigungen zu erhalten.

Eigenschaften des LocalPolicy-Manifests

Die LocalPolicy-Datei umfasst einige architekturelle 4CCs, die in nahezu allen Image4-Dateien zu finden sind – beispielsweise Board- oder Modellkennung (BORD), Angabe eines bestimmten Apple-Chips (CHIP) oder [Exclusive Chip Identification \(ECID\)](#). Die im Folgenden beschriebenen 4CCs beschränken sich allerdings auf die Sicherheitsrichtlinien, die vom Benutzer konfiguriert werden können.

Hinweis: Apple bezeichnet mit dem Begriff *1TR (Paired One True recoveryOS)* den Prozess, mit dem durch das Drücken der physischen Taste „Ein/Aus“ („Power Button“) das Starten der gekoppelten recoveryOS-Software initiiert wird. Hierin unterscheidet sich dieser Prozess von einem normalen recoveryOS-Start, der über den NVRAM oder durch zweimaliges Drücken und anschließendes Halten bewerkstelligt werden kann bzw. der ausgelöst werden kann, wenn beim Starten ein Fehler auftritt. Das Drücken einer physischen Taste auf eine bestimmte Art und Weise stärkt das Vertrauen darin, dass die Startumgebung für einen Angreifer, der in macOS eingedrungen ist und allein auf die Software abzielt, nicht zugänglich ist.

LocalPolicy Nonce Hash (lpth)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* „lpth“ wird als Maßnahme gegen das Anti-Replay der LocalPolicy-Datei eingesetzt. Hierbei handelt es sich um einen SHA-384-Hash-Wert der Nonce der lokalen Richtlinie (LPN, Local Policy Nonce), der in der Secure Storage-Komponente gespeichert ist und über den Boot-ROM der Secure Enclave oder die Secure Enclave zugänglich ist. Die eigentliche Nonce wird nur gegenüber sepOS, niemals gegenüber dem Anwendungsprozessor offengelegt. Ein Angreifer, der LLB dahingehend manipulieren möchte, dass eine frühere von ihm abgegriffene LocalPolicy-Datei gültig ist, müsste somit einen Wert in der Secure Storage-Komponente platzieren, dessen Hash-Wert identisch mit dem „lpth“-Wert aus der lokalen Richtlinie ist, deren Replay ausgelöst werden soll. Im Normalfall ist nur eine einzige LPN im System gültig. Eine Ausnahme dazu besteht nur während einer Softwareaktualisierung. Während dieses Vorgangs sind zwei LPNs gleichzeitig gültig, damit es im Fall eines Fehlers bei der Aktualisierung möglich ist, wieder die alte Software zu starten. Wenn die LocalPolicy-Datei irgendeines Betriebssystems geändert wird, werden alle Richtlinien mit dem neuen „lpth“-Wert signiert, der mit dem neuen LPN in der Secure Storage-Komponente korrespondiert. Diese Änderung erfolgt, wenn der Benutzer Änderungen an den Sicherheitseinstellungen vornimmt oder neue Betriebssysteme mit einer jeweils neuen LocalPolicy-Datei erstellt.

Remote Policy Nonce Hash (rpth)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* „rpth“ zeigt das gleiche Verhalten wie „lpth“; dieser Wert wird aber nur aktualisiert, wenn die Remote-Richtlinie aktualisiert wird. Das ist zum Beispiel der Fall, wenn der Status der Registrierung für „Wo ist?“ geändert wird. Diese Änderung erfolgt implizit, wenn der Benutzer auf seinem Mac den Status von „Wo ist?“ ändert.

recoveryOS Nonce Hash (ronh)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* „ronh“ zeigt das gleiche Verhalten wie „lpth“; dieser Wert ist aber ausschließlich in der lokalen Richtlinie für das recoveryOS des Systems enthalten. Er wird aktualisiert, wenn recoveryOS des Systems (zum Beispiel im Zuge einer Softwareaktualisierung) aktualisiert wird. Die Verwendung dieser separaten Nonce (neben lpth und rpth) hat folgenden Grund: Wenn ein Gerät mit „Wo ist?“ in den Status „Deaktiviert“ versetzt wird, können vorhandene Betriebssysteme deaktiviert werden (indem ihre LPN und RPN aus der Secure Storage-Komponente entfernt werden), während zugleich das recoveryOS des Systems seine Startfähigkeit behält. Auf diese Weise können die Betriebssysteme wieder aktiviert werden, wenn der Benutzer des Systems den Nachweis erbringt, dass sich das Gerät wieder unter seiner Kontrolle und Steuerung befindet, indem er das iCloud-Passwort eingibt, das für den „Wo ist?“-Account hinterlegt ist. Diese Änderung erfolgt, wenn ein Benutzer das recoveryOS des Systems aktualisiert oder neue Betriebssysteme erstellt.

Next Stage Image4 Manifest Hash (nsih)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* Das Feld „nsih“ steht für den SHA-384-Hash-Wert der Image4-Manifestdatenstruktur, die das gestartete macOS-Betriebssystem beschreibt. Das Image4-Manifest für macOS enthält Kennzahlen für alle Startobjekte (zum Beispiel für iBoot, den statischen Cache für die Vertrauensstellung („Static Trust Cache“), die Gerätestruktur („Device Tree“), die Boot Kernel-Sammlung und den Root-Hash-Wert des Signed System Volume (SSV)). Wenn LLB instruiert wird, ein bestimmtes macOS-Betriebssystem zu starten, stellt LLB sicher, dass der Hash-Wert des Image4-Manifests für macOS, das an iBoot angehängt ist, mit dem Wert übereinstimmt, der im Feld „nsih“ der LocalPolicy-Datei erfasst ist. Auf diesem Weg erfasst nsih die Absicht des Benutzers, für welches Betriebssystem er eine lokale Richtlinie (LocalPolicy-Datei) erstellt hat. Der nsih-Wert wird implizit geändert, wenn eine Softwareaktualisierung durchgeführt wird.

Auxiliary Kernel Collection (AuxKC) Policy Hash (auxp)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* macOS
- *Beschreibung:* „auxp“ ist ein SHA-384-Hash-Wert der Richtlinie für die Liste der vom Benutzer autorisierten Kernel-Erweiterungen (UAKL, User-authorized KEXT list). Dieser Wert wird zum Zeitpunkt der AuxKC-Generierung verwendet, um sicherzustellen, dass nur Kernel-Erweiterungen (KEXTs) in die AuxKC übernommen werden, die vom Benutzer autorisiert wurden. „smb2“ ist erforderlich, um dieses Feld festlegen zu können. Der „auxp“-Wert wird implizit geändert, wenn der Benutzer die UAKL-Liste ändert, indem er in „Systemeinstellungen“ > „Sicherheit“ eine Kernel-Erweiterung genehmigt.

Auxiliary Kernel Collection (AuxKC) Image4 Manifest Hash (auxi)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* macOS
- *Beschreibung:* Nachdem das System verifiziert hat, dass der Hash-Wert der UAKL mit dem Wert im Feld „auxp“ der lokalen Richtlinie übereinstimmt, veranlasst es, dass die AuxKC vom Prozessor der Secure Enclave signiert wird, der für das Signieren der LocalPolicy-Datei zuständig ist. Danach wird ein SHA-384-Hash-Wert der Image4-Manifestsignatur der AuxKC in der lokalen Richtlinie platziert um die Möglichkeit einer Verwechslung und des fälschlichen Abgleichs einer zu einem früheren Zeitpunkt signierten AuxKC mit einem Betriebssystem auszuschließen. Wenn iBoot das Feld „auxi“ in der lokalen Richtlinie vorfindet, versucht iBoot, die AuxKC vom Speichermedium zu laden und deren Signatur zu validieren. Außerdem wird geprüft, ob der Hash-Wert des an den AuxKC angeschlossenen Image4-Manifests mit dem Wert im Feld „auxi“ übereinstimmt. Wenn die AuxKC aus irgendeinem Grund nicht geladen werden kann, setzt das System den Startvorgang ohne dieses Startobjekt fort, d. h. ohne dass irgendwelche Kernel-Erweiterung von Drittanbietern geladen werden. Das Feld „auxp“ muss vorhanden sein, damit das Feld „auxi“ in der LocalPolicy-Datei festgelegt werden kann. Der „auxi“-Wert wird implizit geändert, wenn der Benutzer die UAKL-Liste ändert, indem er in „Systemeinstellungen“ > „Sicherheit“ eine Kernel-Erweiterung genehmigt.

Auxiliary Kernel Collection (AuxKC) Receipt Hash (auxr)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* macOS
- *Beschreibung:* „auxr“ ist ein SHA-384-Hash-Wert des AuxKC-Empfangsbelegs; aus ihm geht hervor, welcher Satz von Kernel-Erweiterungen genau in die AuxKC aufgenommen wurde. Bei dem AuxKC-Beleg kann es sich um eine Untermenge der UAKL-Liste handeln, da Kernel-Erweiterungen, die bekanntermaßen für Angriffe genutzt werden können, aus der AuxKC ausgeschlossen werden können, selbst wenn sie vom Benutzer autorisiert wurden. Außerdem können bestimmte Kernel-Erweiterungen, die dafür verwendet werden können, die Begrenzung des Benutzer-Kernels zu umgehen, eine funktionale Einschränkung zur Folge haben, sodass beispielsweise Apple Pay nicht verwendet und 4K/HDR-Inhalte nicht wiedergegeben werden können. Benutzer, die von entsprechenden Möglichkeiten Gebrauch machen möchten, sollten sich für eine restriktivere Form der Aufnahme in die AuxKC entscheiden. Das Feld „auxp“ muss vorhanden sein, damit das Feld „auxr“ in der LocalPolicy-Datei festgelegt werden kann. Der „auxr“-Wert wird implizit geändert, wenn der Benutzer über „Systemeinstellungen“ > „Sicherheit“ eine neue AuxKC erstellt.

CustomOS Image4 Manifest Hash (coih)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* 1TR
- *Beschreibung:* Der coih ist ein SHA-384-Hash des CustomOS-Image4-Manifest. Die Payload für dieses Manifest wird von iBoot (anstelle des XNU-Kernels) verwendet, um die Steuerung zu übergeben. Benutzer ändern den coih-Wert implizit, wenn sie das `kmutil configure-boot`-Befehlszeilenprogramm in 1TR verwenden.

APFS volume group UUID (vuid)

- *Typ:* OctetString (16)
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* „vuid“ ist die Volumegruppe, die der Kernel als Root (Stamm) verwenden soll. Dieses Feld dient primär der Information; es wird nicht für sicherheitsrelevante Einschränkungen verwendet. Die Festlegung der „vuid“ erfolgt implizit, wenn der Benutzer eine neue Betriebssysteminstallation erstellt.

Key encryption key (KEK) Group UUID (kuid)

- *Typ:* OctetString (16)
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* „kuid“ ist das gestartete Volume. Der Key Encryption Key wird üblicherweise für Zwecke des Datenschutzes und der Datensicherheit verwendet. Bei jeder LocalPolicy-Datei wird er verwendet, um den Signierschlüssel der lokalen Richtlinie zu schützen. Die Festlegung der „kuid“ erfolgt implizit, wenn der Benutzer eine neue Betriebssysteminstallation erstellt.

Paired recoveryOS Trusted Boot Policy Measurement (prot)

- *Typ:* OctetString (48)
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* Das Trusted Boot Policy Measurement der gekoppelten recoveryOS-Software ist eine spezielle iterative SHA-384-Hash-Berechnung auf der Basis des Image4-Manifests einer lokalen Richtlinie und unter Ausschluss von Nonces, um eine konsistente Messung im Zeitablauf zu erhalten (da Nonces wie „lph“ häufig aktualisiert werden). Das Feld „prot“ ist nur in lokalen Richtlinien für macOS enthalten und enthält eine Angabe, welche LocalPolicy-Datei für recoveryOS mit welcher LocalPolicy-Datei für macOS korrespondiert.

Has Secure Enclave Signed recoveryOS Local Policy (hrlp)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* „hrlp“ gibt an, ob der oben beschriebene „prot“-Wert eine Messung für die LocalPolicy-Datei für eine durch die Secure Enclave signierte recoveryOS-Software ist oder nicht. Ist dies nicht der Fall, wird die LocalPolicy-Datei von recoveryOS vom Apple Online-Signierungsserver signiert, der Objekte wie macOS Image4-Dateien signiert.

Local Operating System Version (love)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR, recoveryOS, macOS
- *Beschreibung:* „love“ zeigt die Version des Betriebssystems an, für das die lokale Richtlinie (LocalPolicy) erstellt wurde. Die Version wird von dem nächsten Statusmanifest während der Erstellung der LocalPolicy abgerufen und verwendet, um die Kopplungseinschränkungen für das recoveryOS umzusetzen.

Secure Multi-Boot (smb0)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR, recoveryOS
- *Beschreibung:* Wenn „smb0“ vorhanden ist und den Wert „True“ enthält, lässt LLB es zu, dass das nsih-Manifest (Next stage Image4) global signiert wird und keine personalisierte Signatur erforderlich ist. Dieses Feld kann vom Benutzer mit dem Start sicherheitsdienstprogramm oder mit `bputil` geändert werden, um die Herabstufung auf „Reduzierte Sicherheit“ vorzunehmen.

Secure Multi-Boot (smb1)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR
- *Beschreibung:* Wenn „smb1“ vorhanden ist und den Wert „True“ enthält, lässt iBoot es zu, dass Objekte wie beispielsweise eine angepasste Kernel-Sammlung in der Secure Enclave mit demselben Schlüssel wie die LocalPolicy-Datei signiert wird. Das Feld „smb0“ kann nur vorhanden sein, wenn das Feld „smb1“ vorhanden ist. Dieses Feld kann vom Benutzer mit dem Befehlszeilenprogramm `csrutil` oder `bputil` geändert werden, um die Herabstufung auf „Permissive Sicherheit“ vorzunehmen.

Secure Multi-Boot (smb2)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR
- *Beschreibung:* Wenn „smb2“ vorhanden ist und den Wert „True“ enthält, lässt iBoot es zu, dass die Auxiliary Kernel Collection in der Secure Enclave mit demselben Schlüssel wie die LocalPolicy-Datei signiert wird. Das Feld „smb2“ kann nur vorhanden sein, wenn das Feld „smb0“ vorhanden ist. Dieses Feld kann vom Benutzer mit dem Start sicherheitsdienstprogramm oder `bputil` geändert werden, um die Herabstufung auf „Reduzierte Sicherheit“ vorzunehmen und Kernel-Erweiterungen von Drittanbietern zu aktivieren.

Secure Multi-Boot (smb3)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR
- *Beschreibung:* Wenn „smb3“ vorhanden ist und den Wert „True“ enthält, bedeutet dies, dass sich ein Benutzer auf dem System für die Registrierung in einer MDM-Lösung (Mobile Device Management) entschieden hat. Das Vorhandensein dieses Felds bewirkt, dass die Prozessoranwendung der Secure Enclave, die die LocalPolicy-Datei steuert, die Authentifizierung über die MDM-Lösung anstelle der lokalen Benutzerauthentifizierung akzeptiert. Dieses Feld kann vom Benutzer mit dem Start sicherheitsdienstprogramm oder mit `bputil` geändert werden, um die verwaltete Steuerung für Kernel-Erweiterungen von Drittanbietern und von Softwareaktualisierungen zu aktivieren. (In macOS 11.2 oder neuer kann die MDM-Lösung auch eine Aktualisierung auf die neueste macOS-Version veranlassen, wenn als Sicherheitsmodus „Volle Sicherheit“ ausgewählt ist.)

Secure Multi-Boot (smb4)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* macOS
- *Beschreibung:* Wenn „smb4“ vorhanden ist und den Wert „True“ enthält, bedeutet dies, dass das Gerät über [Apple School Manager](#), [Apple Business Manager](#) oder Apple Business Essentials in einer MDM-Lösung registriert wurde und von ihr gesteuert wird. Das Vorhandensein dieses Felds bewirkt, dass die Anwendung der Secure Enclave, die die LocalPolicy-Datei steuert, die Authentifizierung über die MDM-Lösung anstelle der lokalen Benutzerauthentifizierung akzeptiert. Dieses Feld wird durch die MDM-Lösung geändert, wenn sie erkennt, dass die Seriennummer des Geräts in einem dieser drei Dienste erscheint.

System Integrity Protection (sip0)

- *Typ:* Unsignierte 64-Bit-Ganzzahl
- *Veränderbare Umgebungen:* 1TR
- *Beschreibung:* „sip0“ umfasst die Bits der SIP-Richtlinie (System Integrity Protection, Systemintegritätsschutz), die bisher im NVRAM gespeichert wurden. Neue SIP-Richtlinienbits werden hier hinzugefügt (und nicht in die nachfolgend beschriebenen Felder der lokalen Richtlinie), wenn sie ausschließlich von macOS und nicht auch von LLB verwendet werden. Dieses Feld kann von einem Benutzer mit `csrutil` aus 1TR geändert werden, um SIP zu deaktivieren und die Herabstufung auf „Permissive Sicherheit“ vorzunehmen.

System Integrity Protection (sip1)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR
- *Beschreibung:* Wenn „sip1“ vorhanden ist und den Wert „True“ enthält, lässt iBoot Fehler beim Prüfen des Hash-Werts des SSV zu. Dieses Feld kann von einem Benutzer mit `csrutil` oder `bputil` aus 1TR geändert werden.

System Integrity Protection (sip2)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR
- *Beschreibung:* Wenn sip2 vorhanden ist und den Wert „True“ enthält, sperrt iBoot nicht das *CTRR-Hardwareregister (Configurable Text Read-only Region)*, das Kernelspeicher als nicht beschreibbar markiert. Dieses Feld kann von einem Benutzer mit `csrutil` oder `bputil` aus 1TR geändert werden.

System Integrity Protection (sip3)

- *Typ:* Boolescher Wert
- *Veränderbare Umgebungen:* 1TR
- *Beschreibung:* Wenn „sip3“ vorhanden ist und den Wert „True“ enthält, erzwingt iBoot nicht die Verwendung der eigenen integrierten Positivliste für die NVRAM-Variable boot-args, die ansonsten die Optionen filtern würde, die an den Kernel übergeben werden. Dieses Feld kann von einem Benutzer mit csrutil oder bputil aus 1TR geändert werden.

Zertifikate und Remote-Richtlinien (RemotePolicy)

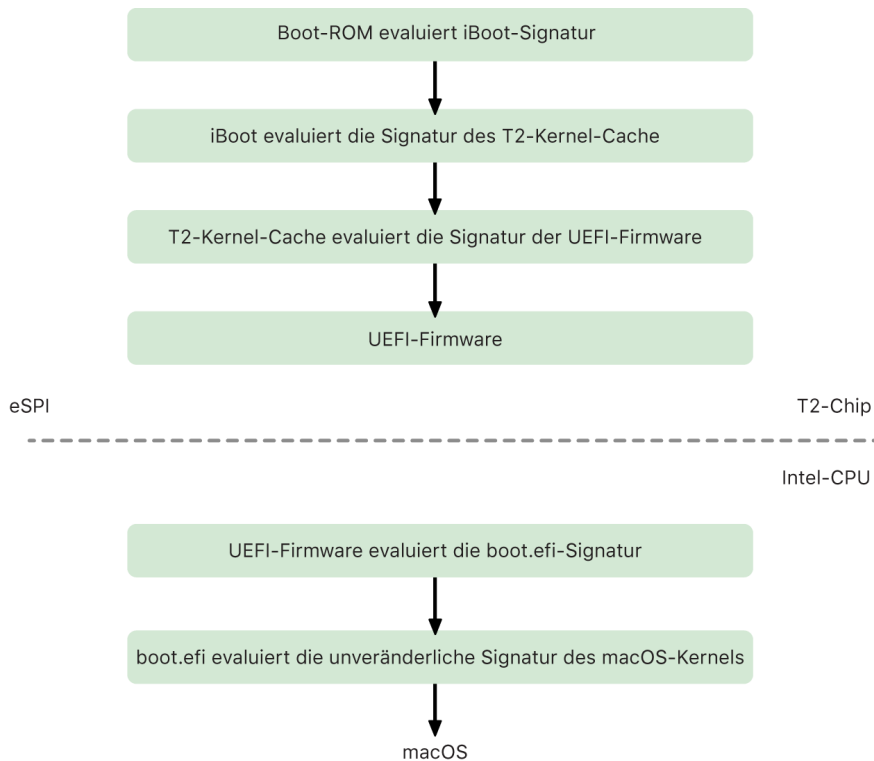
Wie unter [Erstellung und Verwaltung von Signaturschlüsseln für die lokale Richtlinie \(LocalPolicy\)](#) beschrieben ist, enthält die Image4-Datei der lokalen Richtlinie auch das OIC (Owner Identity Certificate) und die eingebettete Remote-Richtlinie (RemotePolicy).

Intel-basierte Mac-Computer

Startvorgang bei Intel-basierten Mac-Computern

Intel-basierte Mac-Computer mit Apple T2 Security Chip

Wenn ein Intel-basierter Mac-Computer mit Apple T2 Security Chip eingeschaltet wird, führt der Chip genau wie bei iPhone- und iPad-Geräten sowie Mac-Computern mit Apple Chips einen sicheren Startvorgang von seinem **Boot ROM** aus. Dadurch wird der iBoot-Bootloader verifiziert. Es ist der erste Schritt in der Chain of Trust. iBoot überprüft den Kernel und den Kernel-Erweiterungscode auf dem T2-Chip, der wiederum die Intel UEFI-Firmware überprüft. Die UEFI-Firmware und die zugehörige Signatur sind zunächst nur für den T2-Chip verfügbar.



Nach der Überprüfung wird das UEFI-Firmware-Image in einem Teil des T2-Chip-Speichers abgebildet. Dieser Speicher wird der Intel-CPU über das eSPI (enhanced Serial Peripheral Interface) zur Verfügung gestellt. Wenn die Intel-CPU erstmals gestartet wird, ruft sie die UEFI-Firmware über das eSPI von der auf Integrität geprüften und im Speicher abgebildeten Kopie der Firmware ab, die sich auf dem T2-Chip befindet.

Die „Chain of Trust“-Evaluierung wird auf der Intel-CPU fortgesetzt, wobei die UEFI-Firmware die Signatur für boot.efi, den bootloader von macOS, überprüft. Die Intel-spezifischen macOS-Signaturen für sicheres Starten werden in dem Image4-Format gespeichert, das auch für das sichere Starten von iOS, iPadOS und T2-Chip verwendet wird. Bei dem Code, der die Syntaxprüfung für die Image4-Dateien ausführt, handelt es sich um den gleichen gehärteten Code aktueller iOS- und iPadOS-Implementierungen für sicheres Starten. Boot.efi überprüft wiederum die Signatur einer neuen Datei namens „immutablekernel“. Wenn das sichere Starten aktiviert ist, repräsentiert „immutablekernel“ den kompletten Satz der Apple-Kernel-Erweiterungen, die zum Starten von macOS erforderlich sind. Die Richtlinie für sicheres Starten endet mit der Übergabe an „immutablekernel“ und im Anschluss daran werden die macOS-Sicherheitsrichtlinien (wie beispielsweise Systemintegritätsschutz und signierte Kernel-Erweiterungen) wirksam.

Wenn bei diesem Vorgang Fehler oder Störungen auftreten, wird der Mac in den [Wiederherstellungsmodus](#) für den Apple T2 Security Chip oder den [DFU \(Device Firmware Upgrade\)-Modus](#) für den Apple T2 Security Chip versetzt.

Microsoft Windows auf Intel-basierten Mac-Computern mit T2-Chip

Standardmäßig vertrauen Intel-basierte Mac-Computer, die das sichere Starten unterstützen, nur von Apple signierten Inhalten. Zur Steigerung der Sicherheit von Boot Camp-Installationen wird aber auch das sichere Starten von Windows unterstützt. Die [UEFI-Firmware \(Unified Extensible Firmware Interface\)](#) umfasst eine Kopie des Zertifikats „Microsoft Windows Production CA 2011“, das zur Authentifizierung der Microsoft-Bootloader eingesetzt wird.

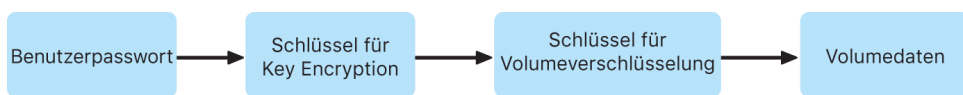
Hinweis: Derzeit wird dem Zertifikat „Microsoft Corporation UEFI CA 2011“ kein Vertrauen erteilt, da dies die Überprüfung von Code, der von Microsoft-Partnern signiert wurde, erlauben würde. Diese UEFI-Zertifizierungsstelle (CA) wird häufig verwendet, um die Authentizität der Bootloader für andere Betriebssysteme (z. B. für Linux-Varianten) zu überprüfen.

Die Unterstützung für das sichere Starten von Windows ist nicht standardmäßig aktiviert. Sie wird vielmehr mit dem Boot Camp-Assistenten (BCA) aktiviert. Wenn ein Benutzer BCA ausführt, wird macOS so neu konfiguriert, dass es signierten Erstanbieter-Code von Microsoft während des Starts akzeptiert. Wenn macOS nach Beendigung des Boot Camp-Assistenten beim sicheren Starten die Vertrauensevaluierung von Apple für Erstanbieter nicht besteht, versucht die UEFI-Firmware, das Vertrauen des Objekts der UEFI-Formatierung für das sichere Starten entsprechend zu evaluieren. Wenn die Vertrauensevaluierung erfolgreich verläuft, setzt der Mac den Vorgang fort und startet Windows. Andernfalls wird der Mac in den macOS-Wiederherstellungsmodus versetzt und der Benutzer informiert, dass die Vertrauensevaluierung fehlschlug.

Intel-basierte Mac-Computer ohne T2-Chip

Intel-basierte Mac-Computer ohne T2-Chip bieten keine Unterstützung für sicheres Starten. Infolgedessen lädt die [UEFI-Firmware \(Unified Extensible Firmware Interface\)](#) das macOS-Startprogramm (boot.efi) ohne Überprüfung des Dateisystems; das Startprogramm lädt den Kernel (prelinkedkernel) ohne Überprüfung des Dateisystems. Um die Integrität des Startvorgangs zu schützen, sollten Benutzer daher alle folgenden Sicherheitsvorkehrungen aktivieren:

- *System Integrity Protection (SIP)*: Diese Option ist standardmäßig aktiviert und schützt das Startprogramm und den Kernel vor böswilligen Schreibvorgängen innerhalb eines laufenden macOS.
- *FileVault*: Zum Aktivieren dieser Option gibt es zwei Methoden: entweder durch den Benutzer oder durch einen [MDM-Administrator \(Mobile Device Management\)](#). Dies dient dem Schutz vor einem persönlich anwesenden Angreifer, der den Festplattenmodus nutzt, um das Startprogramm zu überschreiben.
- *Firmware-Passwort*: Zum Aktivieren dieser Option gibt es zwei Methoden: entweder durch den Benutzer oder durch einen MDM-Administrator. Dieses Passwort verhindert, dass ein persönlich anwesender Angreifer alternative Startmodi auslöst, über die das Startprogramm überschrieben werden kann. Zu diesen Startmodi gehören beispielsweise recoveryOS, der Einzelbenutzermodus und der Festplattenmodus. Außerdem wird so das Starten von alternativen Medien verhindert, über die ein Angreifer Code zum Überschreiben des Startprogramms ausführen könnte.



Startmodi von Intel-basierten Mac-Computern mit Apple T2 Security Chip

Intel-basierte Mac-Computer mit Apple T2 Security Chip verfügen über eine Reihe von Startmodi, die beim Starten durch Drücken von Tastenkombinationen ausgelöst und von der UEFI-Firmware oder dem Startprogramm erkannt werden. Einige Startmodi, etwa der Einzelbenutzermodus, funktionieren nur, wenn die Sicherheitsrichtlinie im Startersicherheitsdienstprogramm auf „Ohne Sicherheit“ geändert wurde.

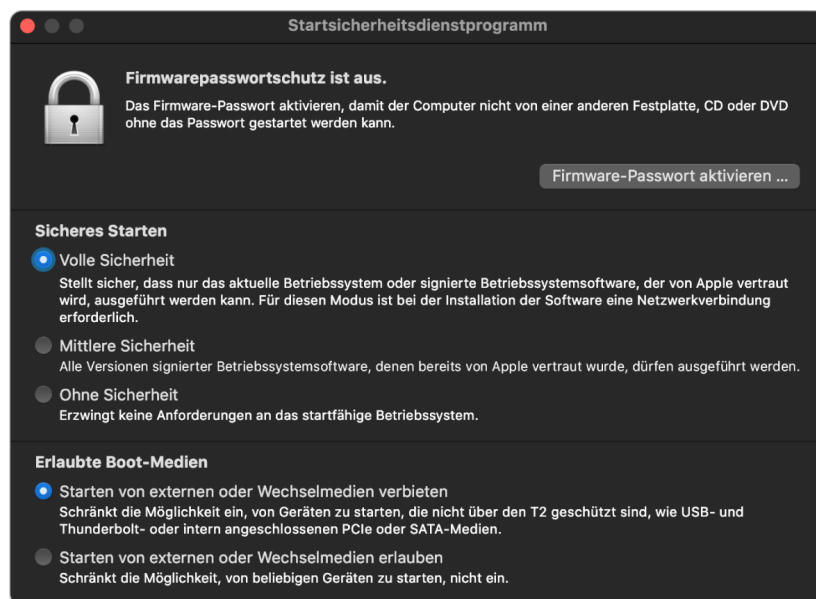
Modus	Taste(n)	Beschreibung
macOS boot	Keine	Die UEFI-Firmware übergibt die Steuerung an das macOS-Startprogramm (ein UEFI-Programm), das diese wiederum an den macOS-Kernel weitergibt. Beim Standardstartvorgang eines Mac mit aktiviertem FileVault zeigt das macOS-Startprogramm das Anmeldefenster an, das das Passwort zum Entschlüsseln des Speichers verwendet.
Startmanager	Wahl (↵)	Die UEFI-Firmware startet die integrierte UEFI-Anwendung, die dem Benutzer das Auswahlfenster für das Startgerät anzeigt.
Festplattenmodus (Target Disk Mode, TDM)	T	Die UEFI-Firmware startet das integrierte UEFI-Programm, das das interne Speichergerät als reines, blockbasiertes Speichergerät via FireWire, Thunderbolt, USB oder einer Kombination dieser drei Technologien (abhängig vom Mac-Modell) offenlegt.
Einzelbenutzermodus	Befehl (⌘)-S	Der macOS-Kernel übergibt die Markierung <code>-s</code> im Argumentvektor von <code>launchd</code> ; <code>launchd</code> erstellt dann die Einzelbenutzer-Shell im „tty“ der App „Konsole“. <i>Hinweis:</i> Wenn der Benutzer die Shell verlässt, setzt macOS den Startvorgang fort und zeigt das Anmeldefenster an.
recoveryOS	Befehl (⌘)-R	Die UEFI-Firmware lädt ein minimales macOS von einer signierten Image-Datei (.dmg) auf dem internen Speichergerät.

Modus	Taste(n)	Beschreibung
Internet recoveryOS	Wahl (⌘)-Befehl (⌘)-R	Die signierte Image-Datei wird mittels HTTP aus dem Internet geladen.
Diagnose	D	Die UEFI-Firmware lädt eine minimale UEFI-Diagnoseumgebung von einer signierten Image-Datei (.dmg) auf dem internen Speichergerät.
Internetdiagnose	Wahl (⌘)-D	Die signierte Image-Datei wird mittels HTTP aus dem Internet geladen.
Windows-Start	Keine	Wenn Windows mit Boot Camp installiert wurde, übergibt die UEFI-Firmware die Steuerung an das Windows-Startprogramm, das diese wiederum an den Windows-Kernel übergibt.

Startsicherheitsdienstprogramm auf Mac-Computern mit Apple T2 Security Chip

Übersicht

Auf Intel-basierten Mac-Computern mit Apple T2 Security Chip handhabt das Startsicherheitsdienstprogramm eine größere Anzahl von Einstellungen für Sicherheitsrichtlinien. Der Zugriff auf das Dienstprogramm erfolgt durch das Starten im recoveryOS-Modus und die Auswahl der Option „Startsicherheitsdienstprogramm“ im Menü „Dienstprogramme“. Dies schützt die unterstützten Sicherheitseinstellungen vor Manipulationsversuchen durch einen Angreifer.



Kritische Richtlinienänderungen erfordern jetzt eine Authentifizierung, selbst im [Wiederherstellungsmodus](#). Beim erstmaligen Öffnen des Start sicherheitsdienstprogramms wird der Benutzer aufgefordert, ein Administratorpasswort der primären macOS-Installation zu verwenden, das dem aktuell gestarteten recoveryOS zugeordnet ist. Sofern es noch keinen Administrator gibt, muss einer erstellt werden, damit die Richtlinie geändert werden kann. Der T2-Chip macht es erforderlich, dass der Mac aktuell im recoveryOS-Modus gestartet wird und dass eine Authentifizierung mit durch die Secure Enclave gesicherten Anmeldedaten erfolgt, damit eine solche Richtlinienänderung vorgenommen werden kann. Für die Änderungen von Sicherheitsrichtlinien gelten zwei implizite Anforderungen. recoveryOS muss:

- von einem Speichergerät gestartet werden, das direkt mit dem T2-Chip verbunden ist, weil Partitionen auf anderen Geräten über keine Secure Enclave-basierten Anmeldedaten verfügen, die an das interne Speichergerät gebunden sind.
- auf einem [APFS](#)-basierten Volume erfolgen, da nur zur Authentifizierung bei der Wiederherstellung verwendete Anmeldedaten unterstützt werden, die an die Secure Enclave auf dem Preboot-APFS-Volume eines Laufwerks gesendet wurden. Es ist nicht möglich, mit HFS+ formatierte Volumes für das sichere Starten zu verwenden.

Diese Richtlinie wird nur im Start sicherheitsdienstprogramm auf Intel-basierten Mac-Computern angezeigt, die mit einem T2-Chip ausgestattet sind. Während in den meisten Fällen keine Änderungen an der Richtlinie für das sichere Starten erforderlich sind, haben letztendlich die Benutzer die Kontrolle über die Einstellungen ihrer Geräte und können sich entscheiden, die Funktionalität für das sichere Starten auf ihrem Mac ihren Anforderungen entsprechend zu deaktivieren oder herabzustufen.

Änderungen an der Richtlinie für das sichere Starten, die innerhalb dieser App vorgenommen werden, gelten nur für die Evaluierung der Chain of Trust, die auf dem Intel-Prozessor überprüft wird. Die Option für das sichere Starten des T2-Chips ist immer in Kraft.

Die Richtlinie für das sichere Starten kann mit einer dieser drei Einstellungen konfiguriert werden: „Volle Sicherheit“, „Mittlere Sicherheit“ und „Ohne Sicherheit“. Bei Auswahl von „Ohne Sicherheit“ wird die Evaluierung für sicheres Starten auf dem Intel-Prozessor vollständig deaktiviert und dem Benutzer erlaubt so zu starten, wie er es möchte.

Richtlinie für das Starten mit der Option „Volle Sicherheit“

„Volle Sicherheit“ ist die Standardrichtlinie für das Starten. Sie verhält sich ähnlich wie bei iOS, iPadOS und Mac-Computern mit Apple Chips. Zu dem Zeitpunkt, an dem die Software heruntergeladen und zur Installation vorbereitet wird, wird sie mit einer Signatur personalisiert, die die [ECID \(Exclusive Chip Identification\)](#) enthält (eine eindeutige ID, die in diesem Fall für den T2-Chip spezifisch ist). Die vom Signierungsserver zurückgegebene Signatur ist dann eindeutig und nur für diesen bestimmten T2-Chip nutzbar. Die [UEFI-Firmware \(Unified Extensible Firmware Interface\)](#) soll sicherstellen, dass wenn die Richtlinie „Volle Sicherheit“ aktiv ist, eine bestimmte Signatur nicht nur von Apple signiert, sondern für den spezifischen Mac signiert wird. Auf diese Weise wird die jeweilige Version von macOS an diesen Mac gebunden. Dies trägt, wie unter „Volle Sicherheit“ beschrieben, zum Verhindern von Rollback-Angriffen auf Mac-Computern mit Apple Chips bei.

Richtlinie für das Starten mit der Option „Mittlere Sicherheit“

Die Startrichtlinie „Mittlere Sicherheit“ ist dem herkömmlichen sicheren Start via UEFI ähnlich, bei der ein Hersteller (in diesem Fall Apple) eine digitale Signatur für den Code generiert, um sicherzustellen, dass dieser Code vom Hersteller stammt. Angreifer haben so keine Möglichkeit, nicht signierten Code einzuschleusen. Diese Signatur wird von Apple als „globale“ Signatur bezeichnet, weil sie auf einem beliebigen Mac für beliebig lange Zeit verwendet werden kann, sofern auf diesem Mac die mittlere Sicherheit konfiguriert ist. Weder iOS oder iPadOS noch der T2-Chip selbst unterstützen globale Signaturen. Mit dieser Einstellung wird nicht versucht, Rollback-Angriffe zu verhindern.

Richtlinie für das Starten von Medien

Die Richtlinie für das Starten von Medien wird nur auf Intel-basierten Mac-Computern mit T2-Chip angezeigt und ist ganz und gar unabhängig von der Richtlinie für das sichere Starten. Selbst wenn ein Benutzer das sichere Starten deaktiviert, ändert sich dadurch das Standardverhalten nicht, d. h., das Starten des Mac-Computers von anderen Medien als dem Speichergerät, das direkt mit dem T2-Chip verbunden ist, wird unterbunden. (Bei Mac-Computern mit Apple Chips wird keine bestimmte Richtlinie für das Starten von Medien vorausgesetzt. Weitere Informationen sind unter [Sicherheitsrichtlinie „Startup Disk“](#) zu finden.)

Schutz per Firmware-Passwort bei Intel-basierten Mac-Computern

Auf Intel-basierten Mac-Computern mit Apple T2 Security Chip unterstützt macOS die Verwendung eines Firmware-Passworts, um unbeabsichtigte Änderungen von Firmware-Einstellungen auf dem jeweiligen Mac zu verhindern. Mit dem Firmware-Passwort wird verhindert, dass alternative Boot- oder Startmodi ausgewählt werden – zum Beispiel das Starten im recoveryOS-, Einzelbenutzer- oder Festplattenmodus oder das Starten von einem nicht autorisierten Volume.

Hinweis: Bei Mac-Computern mit Apple Chips ist kein Firmware-Passwort erforderlich, da die kritische Firmware-Funktionalität, die damit beschränkt wurde, in die recoveryOS-Software verlagert wurde (wenn FileVault aktiviert ist) und recoveryOS eine Benutzerauthentifizierung voraussetzt, bevor der Zugang zu kritischen recoveryOS-Funktionen gewährt wird.

Der elementarste Modus für ein Firmware-Passwort lässt sich auf Intel-basierten Mac-Computern *ohne* T2-Chip über das Dienstprogramm für das im recoveryOS-Modus genutzte Firmware-Passwort und auf Intel-basierten Mac-Computern *mit* einem T2-Chip über das Startersicherheitsdienstprogramm erzielen. Erweiterte Optionen (etwa die Fähigkeit, bei jedem Start ein Passwort zu verlangen) sind über das Befehlszeilenprogramm `firmwarepasswd` in macOS verfügbar.

Das Festlegen eines Firmware-Passworts ist besonders wichtig, um das Risiko von Angriffen auf Intel-basierten Mac-Computern ohne T2-Chip durch physisch anwesende Angreifer zu reduzieren. Das Firmware-Passwort kann Angreifer daran hindern, im recoveryOS-Modus zu starten, von dem aus sie den Systemintegritätsschutz (System Integrity Protection, SIP) deaktivieren können. Da das Starten alternativer Medien eingeschränkt wird, haben Angreifer keine Möglichkeit, privilegierten Code aus einem anderen Betriebssystem auszuführen, um die Firmware der Peripherie anzugreifen.

Für Benutzer, die ihr Passwort vergessen haben, gibt es einen Mechanismus zum Zurücksetzen des Firmware-Passworts. Die Benutzer müssen dazu beim Starten eine Tastenkombination drücken, damit ihnen eine modellspezifische Zeichenfolge angezeigt wird, die sie an AppleCare weitergeben können. AppleCare versieht eine Ressource, deren Signatur vom [URI \(Uniform Resource Identifier\)](#) überprüft wird, mit einer digitalen Signatur. Wenn die Signatur validiert wird und der Inhalt für den jeweiligen Mac bestimmt ist, entfernt die UEFI-Firmware das Firmware-Passwort.

Für Benutzer, die erreichen wollen, dass ausschließlich sie selbst ihr Firmware-Passwort per Software entfernen können, wurde in macOS 10.15 die Option `-disable-reset-capability` zum Befehlszeilenprogramm `firmwarepasswd` hinzugefügt. Vor dem Aktivieren dieser Option müssen Benutzer zustimmen, dass sie, falls das Passwort vergessen wird und entfernt werden muss, die Kosten für den unvermeidlichen Austausch der Logik-Platine übernehmen werden. Organisationen, die ihre Mac-Computer vor Eingriffen durch externe Angreifer und Mitarbeiter schützen möchten, müssen auf Systemen, deren Eigentümer sie sind, ein Firmware-Passwort einrichten. Dies kann auf folgende Weisen direkt auf dem jeweiligen Gerät geschehen:

- Manuell zum Zeitpunkt der Bereitstellung des Geräts mithilfe des Befehlszeilenprogramms `firmwarepasswd`
- Mithilfe von Verwaltungswerkzeugen anderer Anbieter, die das Befehlszeilenprogramm `firmwarepasswd` nutzen
- Mithilfe einer Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM)

recoveryOS und Diagnoseumgebungen bei Intel-basierten Mac-Computern

recoveryOS

Das recoveryOS ist völlig separat vom Haupt-macOS und sein gesamter Inhalt ist in einer Image-Datei namens „BaseSystem.dmg“ gespeichert. Die ebenfalls zugeordnete Datei „BaseSystem.chunklist“ wird verwendet, um die Integrität der Datei „BaseSystem.dmg“ zu überprüfen. Diese „chunklist“ besteht aus einer Reihe von Hash-Werten für 10-MB-Blöcke der Datei „BaseSystem.dmg“. Die [UEFI-Firmware \(Unified Extensible Firmware Interface\)](#) evaluiert zuerst die Signatur der chunklist-Datei und dann den Hash-Wert für jeweils einen Block (oder „Chunk“) der Datei „BaseSystem.dmg“. Dadurch soll sichergestellt werden, dass dieser Hash dem signierten Inhalt in der chunklist-Datei entspricht. Ergibt sich bei einem dieser Hash-Werte keine Übereinstimmung, wird das Starten der lokalen recoveryOS-Software abgebrochen. Danach versucht die UEFI-Firmware, von der recoveryOS-Version im Internet zu starten.

Wenn die Überprüfung erfolgreich abgeschlossen wird, aktiviert die UEFI-Firmware die Image-Datei „BaseSystem.dmg“ als RAM und startet die darin enthaltene Datei „boot.efi“. Da der gesamte Inhalt des Betriebssystems (von dem diese Elemente lediglich eine Untergruppe bilden) bereits auf seine Integrität überprüft wurde, ist es weder erforderlich, dass die UEFI-Firmware boot.efi prüft, noch dass boot.efi den Kernel prüft.

Apple-Diagnose

Das Verfahren zum Starten der lokalen Diagnoseumgebung ist größtenteils identisch mit dem beim Starten des recoveryOS. Es werden separate Dateien „AppleDiagnostics.dmg“ und „AppleDiagnostics.chunklist“ verwendet, wobei diese jedoch auf die gleiche Weise überprüft werden wie die BaseSystem-Dateien. Anstatt boot.efi zu starten, startet die UEFI-Firmware innerhalb der Image-Datei (.dmg-Datei) eine Datei namens „diags.efi“. Diese Datei ist wiederum verantwortlich dafür, eine Reihe anderer UEFI-Treiber aufzurufen, die mit der Hardware interagieren und nach Fehlern in der Hardware suchen können.

recoveryOS aus dem Internet und Diagnoseumgebungen

Wenn beim Starten der lokalen Wiederherstellungs- oder Diagnoseumgebungen ein Fehler auftritt, versucht die UEFI-Firmware, die Image-Dateien stattdessen aus dem Internet zu laden. (Darüber hinaus kann ein Benutzer explizit fordern, dass die Image-Dateien aus dem Internet abgerufen werden. Dazu muss er beim Starten eine bestimmte Tastenkombination gedrückt halten.) Die Integritätsüberprüfung der Image-Dateien und Chunklists, die vom OS-Wiederherstellungsserver geladen werden, erfolgt auf die gleiche Weise wie bei Image-Dateien, die von einem Speichergerät abgerufen werden.

Auch wenn die Verbindung zum OS-Wiederherstellungsserver mittels HTTP erfolgt, werden alle geladenen Inhalte wie oben beschrieben weiterhin auf ihre Integrität überprüft. Daher bieten sie keine Angriffsfläche für Manipulationsversuche durch einen Angreifer, der die Kontrolle über das Netzwerk erlangt hat. Sollte ein einzelner Block die Integritätsprüfung nicht bestehen, wird er vom OS-Wiederherstellungsserver maximal 11-mal neu angefordert, bevor der Vorgang abgebrochen und eine Fehlermeldung angezeigt wird.

Als 2011 die Wiederherstellung aus dem Internet und die Diagnosemodi für Mac-Computer eingeführt wurden, kam man zu der Entscheidung, dass es besser sei, für die UEFI-Firmware das einfachere HTTP-Transportprotokoll zu verwenden und die Authentifizierung von Inhalten mithilfe des Chunklist-Mechanismus vorzunehmen, anstatt die kompliziertere HTTPS-Funktionalität zu verwenden, durch die die potenzielle Angriffsfläche der Firmware vergrößert würde.

Sicherheit des Signed System Volume in iOS, iPadOS und macOS

Mit macOS 10.15 wurde von Apple das R/O-Systemvolumen eingeführt. Hierbei handelt es sich um ein dediziertes, isoliertes Volumen für Systeminhalte. Bei macOS 11 (oder neuer) kamen erstmals starke kryptografische Schutzmechanismen für Systeminhalte auf einem *Signed System Volume (SSV)* hinzu. Das SSV umfasst einen Kernel-Mechanismus, der die Integrität des Systeminhalts zum Zeitpunkt von dessen Ausführung prüft und Daten – Code ebenso wie jegliche andere Daten – zurückweist, die keine von Apple ausgegebene, kryptografische Signatur aufweisen. Beim Starten von iOS 15 oder iPadOS 15 erlangt das Systemvolumen auf einem iOS- oder iPadOS-Gerät auch den kryptografischen Schutz eines signierten Systemvolumens.

Das SSV kann nicht nur die Manipulation an der im Betriebssystem integrierten Apple-Software verhindern, sondern sorgt auch für verlässlichere und sichere Softwareaktualisierungen von macOS. Darüber hinaus nutzt das SSV [APFS-Snapshots \(Apple File System\)](#), sodass in dem Fall, dass eine Aktualisierung nicht zu Ende geführt werden kann, die vorherige Systemversion wiederhergestellt werden kann, ohne dass eine Neuinstallation erforderlich wird.

Seit der Einführung sorgt APFS für die Integrität der Systemmetadaten auf der Basis nicht-kryptografischer Prüfsummen, die für das interne Speichergerät ermittelt werden. Das SSV stärkt den Integritätsmechanismus dadurch, dass kryptografische Hash-Werte hinzukommen und der Mechanismus ausgeweitet wird, sodass jedes Byte der Dateidaten erfasst wird. Für Daten vom internen Speichergerät (einschließlich der Metadaten des Dateisystems) wird im Dateipfad ein kryptografischer Hash-Wert generiert, der mit einem erwarteten Wert in den Metadaten des Dateisystems abgeglichen wird. Weichen die Werte voneinander ab, geht das System davon aus, dass die Daten manipuliert wurden, und übergibt keine Daten an die anfordernde Software.

Jeder SHA-256-Hash-Wert des SSV wird im Metadatenhauptbaum des Dateisystems gespeichert, für den seinerseits ein eigener Hash-Wert generiert wird. Da jeder Knoten im Baum rekursiv die Integrität der Hash-Werte seiner Kinder überprüft, vergleichbar zu einem binären Hash-Baum (Merkle), umfasst daher der Hash-Wert des Stammknotens – er wird als „*Seal*“ (Siegel) bezeichnet – jedes Byte der Daten auf dem SSV. Dies bedeutet, dass die kryptografische Signatur das gesamte Systemvolumen erfasst und abdeckt.

Beim Installieren und Aktualisieren von macOS wird dieser Seal-Wert aus dem Dateisystem auf dem Gerät neu berechnet. Das dabei ermittelte Ergebnis wird überprüft, indem es mit der Kennzahl abgeglichen wird, die von Apple signiert wurde. Bei Mac-Computern mit Apple Chips prüft der Bootloader den Seal-Wert, bevor es die Steuerung an den Kernel übergibt. Bei Intel-basierten Mac-Computern mit Apple T2 Security Chip leitet der Bootloader die Kennzahl und die Signatur an den Kernel weiter, der dann den Seal-Wert prüft, unmittelbar bevor er das Root-Dateisystem lädt. In beiden Fällen wird bei einem Scheitern der Prüfung der Startvorgang angehalten und der Benutzer aufgefordert, macOS neu zu installieren. Diese Prozedur wird bei jedem Startvorgang wiederholt, es sei denn, der Benutzer hat einen Modus für eine niedrigere Sicherheitsebene gestartet und in einem separaten Vorgang entschieden, das Signed System Volume zu deaktivieren.

Während iOS- oder iPadOS-Softwareupdates wird das Systemvolumen auf ähnliche Weise vorbereitet und neu berechnet. Die iOS- und iPadOS-Bootloader überprüfen, dass das Siegel intakt ist und mit dem von Apple signierten Wert übereinstimmt, bevor dem Gerät erlaubt wird, den Kernel zu starten. Wenn es beim Starten keine Übereinstimmung gibt, wird der Benutzer aufgefordert, die Systemsoftware auf dem Gerät zu aktualisieren. Benutzern ist es nicht erlaubt, den Schutz eines signierten Systemvolumens in iOS oder iPadOS zu deaktivieren.

SSV und Codesignierung

Die Code-Signierung ist immer noch vorhanden und wird vom Kernel durchgesetzt. Das SSV (Signed System Volume) übernimmt den Schutz, sobald irgendwelche Bytes vom internen Speichergerät gelesen werden. Im Gegensatz dazu übernimmt die Codesignierung den Schutz, wenn Mach-Objekte dem Speicher als ausführbare Objekte zugewiesen werden. Sowohl das SSV als auch die Codesignierung tragen zum Schutz von ausführbarem Code in allen Read- und Execute-Pfaden bei.

SSV und FileVault

Bei macOS 11 stellt das SSV einen gleichwertigen Schutz dar, solange sich das Dateisystem im Ruhezustand befindet. Aus diesem Grund muss das Systemvolumen nicht mehr verschlüsselt werden. Jegliche Modifikation, die am Dateisystem vorgenommen wird, solange es nicht in Gebrauch ist, wird vom Dateisystem erkannt, wenn es geladen wird. Wenn FileVault vom Benutzer aktiviert wurde, wird der Inhalt des Benutzers nach wie vor mithilfe des vom Benutzer bereitgestellten Secret auf dem Datenvolumen verschlüsselt.

Wenn sich der Benutzer dafür entscheidet, das SSV zu deaktivieren, wird das System anfällig für Manipulationen, solange es sich im Ruhezustand befindet. Derartige Manipulationen können einen Angreifer in die Lage versetzen, beim nächsten Start des Systems verschlüsselte Benutzerdaten zu extrahieren. Aus diesem Grund verweigert das System einem Benutzer das Deaktivieren des SSV, wenn FileVault aktiviert ist. Der Schutz während des Nichtgebrauchs muss auf beiden Volumens konsistent aktiviert oder deaktiviert werden.

Bei macOS 10.15 (und früheren Versionen) schützte FileVault die Software des Betriebssystems, während sich dieses im Ruhezustand befindet, indem die Benutzer- und Systeminhalte mit einem Schlüssel verschlüsselt wurden, der durch ein vom Benutzer bereitgestelltes Secret geschützt war. Auf diese Weise kann ein Angreifer mit physischem Zugriff auf das Gerät das Dateisystem mit der Systemsoftware nicht modifizieren oder darauf zugreifen.

SSV und Mac-Computer mit Apple T2 Security Chip

Bei Mac-Computern mit einem Apple T2 Security Chip untersteht nur macOS selbst dem SSV-Schutz. Die Software, die auf dem T2-Chip gestartet wird und macOS prüft, wird durch den sicheren Startvorgang geschützt.

Sichere Softwareaktualisierungen

Sicherheit ist ein Prozess. Der verlässliche Start der ab Werk installierten Betriebssystemversion allein reicht nicht aus – es muss einen Mechanismus geben, der den schnellen und sicheren Empfang der aktuellsten Sicherheitsupdates ermöglicht. Apple veröffentlicht in regelmäßigen Abständen Softwareaktualisierungen, um neu erkannte Sicherheitslücken zu schließen. Benutzer von iOS- und iPadOS-Geräten erhalten Benachrichtigungen zu Aktualisierungen auf dem jeweiligen Gerät. Benutzer von Mac-Computern können verfügbare Aktualisierungen in den Systemeinstellungen finden. Die Aktualisierungen werden drahtlos bereitgestellt, um Sicherheitslücken so schnell wie möglich schließen zu können.

Aktualisierungsprozess

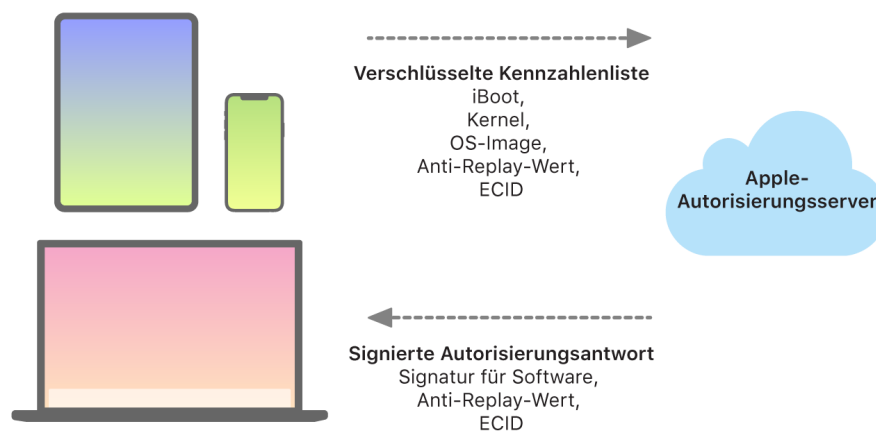
Der Aktualisierungsvorgang verwendet denselben hardwarebasierten Vertrauensanker, der auch vom sicheren Startvorgang benutzt wird, um sicherzustellen, dass nur von Apple signierter Code installiert wird. Mit der [Autorisierung der Systemsoftware](#) stellt der Aktualisierungsvorgang sicher, dass sich nur legitime Kopien von Betriebssystemversionen, die aktiv von Apple signiert werden, auf iOS- und iPadOS-Geräten oder auf Mac-Computern installieren lassen, auf denen im Start Sicherheitsdienstprogramm die Einstellung „Volle Sicherheit“ als Richtlinie für sicheres Starten konfiguriert ist. Auf Grundlage dieser Sicherheitsmaßnahmen ist Apple in der Lage, die Signierung älterer Betriebssystemversionen mit bekannten Schwachstellen zu stoppen, was zum Verhindern von Downgrade-Angriffen beiträgt.

Um die Sicherheit bei der Softwareaktualisierung weiter zu erhöhen, wird außerdem, wenn ein zu aktualisierendes Gerät physisch mit einem Mac verbunden ist, eine vollständige Kopie von iOS oder iPadOS heruntergeladen und installiert. OTA-Softwareaktualisierungen (Over The Air) laden aber anstelle des vollständigen Betriebssystems *nur die für die Aktualisierung benötigten Komponenten*, wodurch die Netzwerkeffizienz verbessert wird. Außerdem können Softwareaktualisierungen auf einem Mac mit macOS 10.13 (oder neuer), auf dem die Option für das Inhaltscaching aktiviert ist, zwischengespeichert werden, damit iOS- und iPadOS-Geräte die erforderliche Aktualisierung nicht erneut über das Internet laden müssen. (Sie müssen allerdings immer noch eine Verbindung zu Apple-Servern herstellen, um den Aktualisierungsprozess abzuschließen.)

Personalisierter Aktualisierungsprozess

Bei Aktualisierungen (bei Upgrades ebenso wie bei Updates) wird eine Verbindung zum Apple-Server für die Installationsautorisierung hergestellt, der eine Liste kryptografischer Kennzahlen für jeden Bestandteil des Installationspakets enthält, der installiert werden soll (z. B. iBoot, Kernel und Image des Betriebssystems), einen Anti-Replay-Zufallswert (Nonce) und die eindeutige **ECID (Exclusive Chip Identification)**.

Der Autorisierungsserver vergleicht die Liste der Kennzahlen mit den Versionen, deren Installation erlaubt ist, und fügt bei einem Treffer die ECID zu den Kennzahlen hinzu und signiert das Ergebnis. Der Server überträgt beim Aktualisierungsvorgang einen kompletten signierten Datensatz an das Gerät. Mit der hinzugefügten ECID wird die Autorisierung für das anfragende Gerät „personalisiert“. Indem nur bekannte Kennzahlen autorisiert und signiert werden, stellt der Server sicher, dass die Aktualisierung genau in der von Apple vorgesehenen und bereitgestellten Weise vorgenommen wird.



Die „Chain of Trust“-Evaluierung beim Start überprüft, ob die Signatur von Apple stammt und ob die Kennzahl des vom Speichergerät geladenen Objekts in Kombination mit der ECID des Geräts von der Signatur abgedeckt wird. Mit diesen Schritten wird sichergestellt, dass die Autorisierung gerätespezifisch ist und dass keine älteren Versionen eines Betriebssystems oder einer Firmware von einem Gerät auf ein anderes kopiert werden können. Die **Nonce** verhindert, dass ein Angreifer die Antwort des Servers sichern und sie dafür verwenden kann, ein Gerät zu manipulieren oder die Systemsoftware anderweitig zu verändern.

Die Personalisierung ist ein Grund dafür, dass eine Netzwerkverbindung mit Apple immer eine Aktualisierung aller Geräte mit Apple Chips einschließlich Intel-basierter Mac-Computer mit Apple T2 Security Chip erfordert.

Das Datenvolumen des Benutzers wird während einer Softwareaktualisierung nie aktiviert. Dadurch werden Lese- und Schreibvorgänge auf diesem Volume während der Aktualisierung verhindert.

Bei Geräten mit Secure Enclave nutzt diese Hardware auf ähnliche Weise die **Autorisierung der Systemsoftware**, um die Herkunft und Echtheit der Software zu überprüfen und die Installation eines älteren Betriebssystems zu verhindern.

Integrität des Betriebssystems

Bei der Entwicklung von Apple-Betriebssystemen steht die Sicherheit im Mittelpunkt. Das Design enthält einen Hardware-Vertrauensanker, der das sichere Starten ermöglicht, und einen sicheren Prozess für Softwareaktualisierungen. Apple-Betriebssysteme nutzen darüber hinaus speziell angefertigte Chip-basierte Hardwarefunktionen, die vor Exploits während des Systembetriebs schützen. Diese Funktionen schützen die Integrität von vertrauenswürdigen Code bei der Ausführung. Kurz gesagt tragen Apple-Betriebssysteme dazu bei, Angriffen, Missbrauchs- und Sabotagetechniken entgegenzuwirken – unabhängig davon, ob diese ihren Ursprung in Apps, im Internet oder in anderen Kanälen haben. Die hier aufgelisteten Schutzvorkehrungen sind auf Geräten mit unterstützten Apple-SoCs verfügbar – dazu gehören iOS, iPadOS, tvOS, watchOS und ab jetzt auch macOS auf Mac-Computern mit Apple Chips.

Funktion	A10	A11, S3	A12, S4	A13, S5	A14, A15, S6, S7	M1-Familie
Kernel-Integritätsschutz	✓	✓	✓	✓	✓	✓
Schnelle Berechtigungseinschränkungen		✓	✓	✓	✓	✓
Integritätsschutz des System-Coprozessors			✓	✓	✓	✓
PACs (Pointer Authentication Codes)			✓	✓	✓	✓
Page Protection Layer (PPL)		✓	✓	✓	✓	Siehe Anmerkung unten.

Hinweis: Die PPL (Page Protection Layer) setzt voraus, dass die Plattform *ausschließlich* signierten und vertrauenswürdigen Code ausführt. Dieses Sicherheitsmodell ist für macOS nicht anwendbar.

Kernel-Integritätsschutz

Nachdem die Kernel des Betriebssystems die Initialisierung abgeschlossen haben, wird der Kernel-Integritätsschutz (Kernel Integrity Protection, KIP) aktiviert, um Änderungen an Kernel und Treibercode zu verhindern. Der [Speichercontroller](#) stellt eine geschützte Region im physischen Speicher bereit, die von [iBoot](#) verwendet wird, um den Kernel und Kernel-Erweiterungen zu laden. Nach dem Startvorgang blockiert der Speichercontroller Schreibzugriffe auf die geschützte Region im physischen Speicher. Die MMU (Memory Management Unit) des Anwendungsprozessors wird so konfiguriert, dass das Mapping von privilegiertem Code vom physischen Speicher außerhalb der geschützten Speicherregion verhindert wird und dass schreibbare Mappings des physischen Speichers innerhalb der Kernel-Speicherregion verhindert werden.

Die zur Aktivierung des KIP verwendete Hardware wird nach der Beendigung des Startvorgangs gesperrt, um eine Neukonfiguration zu verhindern.

Schnelle Berechtigungseinschränkungen

Mit dem Apple A11 Bionic S3 SoC wurde eine neue Hardware-Primitive eingeführt. Diese Primitive (Fast Permission Restrictions) umfasst ein CPU-Register für das schnelle Einschränken von Berechtigungen pro Thread. Dank dieser schnellen Berechtigungseinschränkung (auch als APRR-Register bezeichnet) sind unterstützte Betriebssysteme in der Lage, Ausführungsberechtigungen aus dem Arbeitsspeicher zu entfernen – ohne den Mehraufwand eines Systemaufrufs und eines Durchlaufs oder Leerens der Seitentabelle. Dies bietet eine zusätzliche Abwehrebene gegen Angriffe aus dem Internet, insbesondere gegenüber Aktionen, die JIT-kompiliert (Just-in-time) sind, da Speicher nicht effektiv genutzt werden kann, während er gleichzeitig les- und beschreibbar ist.

Integritätsschutz des System-Coprozessors

Die Coprozessor-Firmware übernimmt viele kritische Systemaufgaben – zum Beispiel die Secure Enclave, den Bildsensorprozessor und den Bewegungscoprozessor. Aus diesem Grund kommt ihrer Sicherheit eine Schlüsselrolle bei der Sicherheit des Gesamtsystems zu. Apple verwendet einen Mechanismus namens System-Coprozessor-Integritätsschutz (*System Coprocessor Integrity Protection, SCIP*), um Modifikationen der Coprozessor-Firmware zu verhindern.

SCIP funktioniert auf ganz ähnliche Weise wie der Kernel-Integritätsschutz (KIP): Beim Starten lädt iBoot die Firmware jedes Coprozessors in eine geschützte Speicherregion, die eigens dafür reserviert und von der KIP-Region abgetrennt ist. iBoot konfiguriert die Speichereinheit jedes Prozessors, um Folgendes zu verhindern:

- Ausführbare Mappings außerhalb des entsprechenden Teils der geschützten Speicherregion
- Verhindern von schreibbaren Mappings innerhalb des entsprechenden Teils der geschützten Speicherregion

Beim Starten wird zum Konfigurieren der SCIP für die Secure Enclave das Betriebssystem der Secure Enclave verwendet. Nachdem der Startvorgang abgeschlossen wurde, wird die Hardware gesperrt, die zum Aktivieren von SCIP verwendet wurde. Dadurch soll eine Neukonfiguration verhindert werden.

PACs (Pointer Authentication Codes)

Pointer Authentication Codes (PACs) werden verwendet, um zu verhindern, dass Fehler ausgenutzt werden können, die den Speicher modifizieren. Systemsoftware und integrierte Apps verwenden PAC, um Änderungen an Funktionszeigern und Rückgabeadressen (Codezeigern) zu verhindern. PAC verwendet fünf geheime 128-Bit-Werte, um Kernel-Anweisungen und Daten zu signieren, und jeder Benutzerbereich verfügt über seine eigenen B-Schlüssel. Objekte werden wie folgt mit Salt- und Signaturdaten versehen.

Objekt	Schlüssel	Salt
Funktionsrückgabeadresse	IB	Speicheradresse
Funktionszeiger	IA	0
Funktion für Blockaufrufe	IA	Speicheradresse
Objective-C-Methodencache	IB	Speicheradresse + Klasse + Selektor
C++ V-Tabelleneinträge	IA	Speicheradresse + Hash (beschädigter Methodename)
Berechnetes Goto-Etikett	IA	Hash (Funktionsname)
Kernel-Thread-Status	GA	•
Benutzer-Thread-Statusregister	IA	Speicheradresse
C++ V-Tabellenzeiger	DA	0

Der Signaturwert wird in nicht benutzten Füll-Bits oben im 64-Bit-Zeiger gespeichert. Die Signatur wird vor der Verwendung überprüft und die Füllung wird wiederhergestellt, um eine funktionierende Speicheradresse sicherzustellen. Ein Scheitern der Überprüfung führt zum Abbruch. Diese Überprüfung erschwert viele Angriffe – zum Beispiel einen ROP-Angriff (Return Oriented Programming), der versucht, durch die Manipulation der auf dem Stack abgelegten Funktionsrückgabeadressen das Gerät zu verleiten, vorhandenen Code in böswilliger Absicht auszuführen.

Page Protection Layer (PPL)

Bei iOS, iPadOS und watchOS sorgt PPL (Page Protection Layer) für den Schutz von Code im Benutzerbereich (User Space), nachdem die Überprüfung der Code-Signatur abgeschlossen wurde. PPL baut auf Kernel-Integritätsschutz und schnellen Berechtigungseinschränkungen auf, um die Überschreibungen von Berechtigungen für Seitentabellen sorgfältig zu verwalten und so sicherzustellen, dass nur die PPL geschützten Seiten, die Benutzercode und Seitentabellen enthalten, ändern kann. Das System reduziert die Angriffsfläche drastisch, indem es das Erzwingen der systemweiten Codeintegrität unterstützt, selbst wenn der Kernel beeinträchtigt wurde. macOS bietet diesen Schutz nicht an, da PPL nur auf Systemen zum Einsatz kommt, auf denen jeglicher ausgeführter Code signiert sein muss.

Zusätzliche macOS-Funktionen für die Systemsicherheit

Zusätzliche macOS-Funktionen für die Systemsicherheit

macOS läuft auf einer breiten Palette von Hardware (z. B. Intel-basierte CPUs, Intel-basierte CPUs in Kombination mit dem Apple T2 Security Chip und Apple Chips SoCs) und unterstützt eine Reihe von allgemeinen Computeranwendungsfällen. Manche Benutzer verwenden ausschließlich die vorinstallierten Apps und die Apps aus dem App Store. Es gibt aber auch Kernel-Hacker, die praktisch alle Schutzvorkehrungen der Plattform deaktivieren müssen, damit sie ihren Code mit den höchsten Vertrauensebenen ausführen und testen können. Die meisten Benutzer fallen zwischen diese beiden Extreme und viele von ihnen nutzen Peripheriegeräte und Software, die ein unterschiedlich hohes Maß an Zugriffsberechtigungen erfordern. Apple hat die macOS-Plattform mit einem integrierten Ansatz für Hardware, Software und Dienste entwickelt, der Sicherheit durch Design bietet und das Konfigurieren, Implementieren und Verwalten erleichtert (und die Konfigurationsmöglichkeit beibehält, die von Benutzern erwartet wird). macOS umfasst die wichtigen Sicherheitstechnologien, die IT-Fachkräfte brauchen, um Unternehmensdaten zu schützen und in Umgebungen mit sicheren Unternehmensnetzwerken zu integrieren.

Die folgenden Funktionen unterstützen und sichern die vielfältigen Anforderungen von macOS-Benutzern. Zu ihnen gehören:

- Sicherheit des Signed System Volume
- Systemintegritätsschutz
- Caches für die Vertrauensstellung (Trust Caches)
- Schutz für Peripheriegeräte
- Unterstützung von Rosetta 2 (automatische Übersetzung) und Sicherheit bei Mac-Computern mit Apple Chips
- DMA-Unterstützung und Sicherheit
- Unterstützung und Sicherheit für Kernel-Erweiterungen (kext)
- OROM-Unterstützung und -Sicherheit
- Sicherheit für UEFI-Firmware bei Intel-basierten Mac-Computern

Systemintegritätsschutz

macOS verwendet Kernel-Erweiterungen, um das Schreiben kritischer Systemdateien mit einer Funktion namens *Systemintegritätsschutz* (*System Integrity Protection, SIP*) einzuschränken. Diese Funktion ist eine separate Erweiterung, die zusätzlich zum hardwarebasierten Kernel-Integritätsschutz (Kernel Integrity Protection, KIP) auf Mac-Computern mit Apple Chips verfügbar ist. Sie schützt den Kernel im Speicher vor Modifikationen. Zwingende Zugriffssteuerungstechnologien werden genutzt, um diese und andere Schutzmechanismen auf Kernel-Ebene bereitzustellen, darunter auch Sandboxing und Data Vault.

Zwingende Zugriffssteuerungen

macOS nutzt zwingende Zugriffssteuerungen, die vom Entwickler definierte Sicherheitseinschränkungen durchsetzen und nicht überschrieben werden können. Dieser Ansatz unterscheidet sich von der benutzerbestimmbaren Zugriffssteuerung, die Benutzern das Überschreiben von Sicherheitsrichtlinien nach ihrem Ermessen erlaubt.

Zwingende Zugriffssteuerungen sind für Benutzer nicht sichtbar, sind jedoch die Basistechnologie, die die Aktivierung wichtiger Funktionen wie Sandboxing, Kindersicherung, verwaltete Einstellungen, Erweiterungen und Systemintegritätsschutz ermöglicht.

Systemintegritätsschutz

Der *Systemintegritätsschutz (System Integrity Protection, SIP)* bietet Schutz auf Systemebene, der den Zugriff von Komponenten in bestimmten kritischen Bereichen des Dateisystems auf das reine Lesen beschränkt und so verhindert, dass sie durch schädlichen Code ausgeführt oder geändert werden. Beim Systemintegritätsschutz handelt es sich um eine computerspezifische Einstellung, die standardmäßig aktiviert wird, wenn ein Benutzer auf OS X 10.11 (oder neuer) aktualisiert. Wenn diese Einstellung auf Intel-basierten Mac-Computern deaktiviert wird, wird der Schutz für alle Partitionen auf dem physischen Speichergerät aufgehoben. macOS wendet diese Sicherheitsrichtlinie auf jeden Prozess an, der auf dem System läuft, unabhängig davon, ob der Prozess in einer Sandbox oder mit administrativen Rechten ausgeführt wird.

Caches für die Vertrauensstellung (Trust Caches)

Eines der Elemente des sicheren Startvorgangs ist der statische Cache für die Vertrauensstellung (Static Trust Cache). Dabei handelt es sich um einen vertrauenswürdigen Datensatz aller Mach-O-Binärdateien, die in das signierte Systemvolumen gemastert wurden. Jede Mach-O-Binärdatei wird von einem Codeverzeichnis-Hash repräsentiert. Diese Hash-Werte werden vor ihrer Eingliederung in den Trust Cache sortiert, um effiziente Suchvorgänge zu ermöglichen. Das Codeverzeichnis ist das Ergebnis des Signiervorgangs, der von `codesign(1)` durchgeführt wird. Die Durchsetzung des Trust Cache setzt voraus, dass SIP aktiviert bleibt. Um die Durchsetzung des Trust Cache auf einem Mac mit Apple Chips deaktivieren zu können, muss daher der sichere Startvorgang für den Modus „Permissive Sicherheit“ konfiguriert sein.

Wenn eine Binärdatei ausgeführt wird (ob beim Auslösen eines neuen Prozesses oder beim Mapping von ausführbarem Code in einen bestehenden Prozess), wird ihr Codeverzeichnis extrahiert und ein Hash-Wert für das Codeverzeichnis generiert. Wird der daraus resultierende Hash-Wert im Trust Cache gefunden, erhalten die ausführbaren Mappings, die für die Binärdatei erstellt wurden, Plattformberechtigungen – das bedeutet, dass sie über jegliche Berechtigungen verfügen und ohne weitere Authentizitätsprüfung hinsichtlich der Signatur ausgeführt werden können. Dies unterscheidet sich von dem Vorgang bei Intel-basierten Mac-Computern, bei dem Plattformberechtigungen auf Inhalte des Betriebssystems durch das Apple-Zertifikat übertragen werden, das die Binärdateien signiert. (Dieses Zertifikat schränkt nicht ein, welche Berechtigungen die Binärdatei haben kann.)

Plattformfremde Binärdateien (z. B. beglaubigter Code von Dritten) müssen über gültige Zertifikatketten verfügen, um ausführbar zu sein. Die Berechtigungen, über die sie verfügen können, werden von dem Signierungsprofil eingeschränkt, das dem Entwickler vom Apple Developer Program ausgestellt wurde.

Alle in macOS enthaltenen Binärdateien werden mit einer *Plattformkennung* signiert. Bei Mac-Computern mit Apple Chips weist diese Kennung darauf hin, dass die Binärdatei zwar von Apple signiert ist, der Hash-Wert ihres Codeverzeichnisses allerdings im Trust Cache vorliegen muss, um ausführbar zu sein. Bei Intel-basierten Mac-Computern wird die Plattformkennung verwendet, um Berechtigungen von Binärdateien aus einer älteren Version von macOS gezielt zu widerrufen. Dieses gezielte Widerrufen soll die Ausführung dieser Binärdateien auf neueren Versionen verhindern.

Der Static Trust Cache weist Binärdateien vollständig einer bestimmten Version von macOS zu. Dies verhindert, dass berechtigterweise von Apple signierte Binärdateien älterer Betriebssysteme in neuere Versionen eingeführt werden, um Vorteile für einen Angreifer zu schaffen.

Nicht im Betriebssystem enthaltene Binärdateien

Einige Binärdateien – z. B. Xcode und die Entwickler-Tools – werden nicht mit einer Plattformkennung versehen. Dennoch sind sie dazu berechtigt, auf einem Mac mit Apple Chips und solchen mit T2-Chip mit Plattformberechtigungen ausgeführt zu werden. Da diese Plattformsoftware unabhängig von macOS bereitgestellt wird, unterliegt sie nicht dem vom Static Trust Cache auferlegten Widerrufsverhalten.

Ladbare Trust Caches

Apple stellt bestimmte Softwarepakete in Kombination mit *ladbaren Trust Caches* bereit. Diese Caches haben dieselbe Datenstruktur wie der statische Trust Cache. Während es jedoch nur einen statischen Trust Cache gibt, dessen Inhalte nach den ersten Initialisierungsschritten des Kernel unter Garantie in schreibgeschützten Bereichen gesperrt werden, werden ladbare Trust Caches zur Laufzeit zum System hinzugefügt.

Diese Trust Caches werden entweder mit denselben Mechanismen authentifiziert, die auch die Boot-Firmware authentifizieren (Personalisierung durch den vertrauenswürdigen Apple Signierungsdienst), oder sie werden als global signierte Objekte authentifiziert (deren Signaturen sie nicht an ein bestimmtes Gerät binden).

Ein Beispiel für einen personalisierten Trust Cache ist der Cache, der mit der Image-Datei für die Ausführung von Felddiagnosen auf Mac-Computern mit Apple Chips bereitgestellt wird. Dieser Trust Cache wird zusammen mit der Image-Datei personalisiert und in den Kernel des jeweiligen Mac-Computers geladen, wenn er in einen Diagnosemodus gestartet wird. Der Trust Cache erlaubt der Software in der Image-Datei, mit Plattformberechtigungen ausgeführt zu werden.

Ein Beispiel für einen global signierten Trust Cache wird mit macOS Softwareaktualisierungen bereitgestellt. Dieser Trust Cache ermöglicht einem Teil des Codes der Softwareaktualisierung – dem *Update Brain* – mit Plattformberechtigungen ausgeführt zu werden. Das Update Brain führt alle Schritte zur Vorbereitung der Aktualisierung aus, die das Host-System nicht versionsübergreifend einheitlich bereitstellen kann.

Sicherheit von Peripherieprozessoren bei Mac-Computern

Alle modernen Computersysteme umfassen zahlreiche integrierte Peripherieprozessoren, die Aufgaben wie Netzwerkbetrieb, Grafikverarbeitung, Energieverwaltung und mehr übernehmen. Häufig dienen diese Peripherieprozessoren nur einem bestimmten Zweck und sind deutlich leistungsschwächer als die primäre CPU. Allerdings geraten integrierte Peripheriegeräte, die nicht genügend Sicherheit implementieren, ins Visier von Angreifern, die nach einfacheren Zielen suchen, um diese auszunutzen und das Betriebssystem dauerhaft zu infizieren. Ist die Firmware eines Peripherieprozessors erst einmal infiziert, könnte ein Angreifer die Software der primären CPU ansteuern oder auch direkt sensible Daten abgreifen (z. B. könnte ein Ethernetgerät den Inhalt von Paketen sehen, die nicht verschlüsselt sind).

Apple verfolgt das Ziel, die Anzahl erforderlicher Peripherieprozessoren (wann immer möglich) zu reduzieren und Designs, die Firmware erfordern, gänzlich zu vermeiden. Wenn allerdings separate Prozessoren mit eigener Firmware benötigt werden, wird durch entsprechende Maßnahmen sichergestellt, dass Angreifer sich auf diesem Prozessor nicht dauerhaft „einnisten“ können. Dies kann durch die Verifizierung des Prozessors erfolgen. Hierfür gibt es zwei Möglichkeiten:

- Durch Ausführen des Prozessors in einem Modus, in dem er beim Starten geprüfte Firmware aus der primären CPU herunterlädt
- Indem der Peripherieprozessor seinen eigenen sicheren Startvorgang implementiert, um bei jedem Starten des Mac-Computers die eigene Firmware zu überprüfen

Apple arbeitet mit Herstellern zusammen, um deren Implementierungen zu überprüfen und deren Designs mit gewünschten Eigenschaften zu erweitern, beispielsweise:

- Sicherstellen eines Minimums an kryptografischer Stärke
- Konsequentes Widerrufen von Firmware, die als schlecht bekannt ist
- Deaktivieren von Debugging-Schnittstellen
- Signieren der Firmware mit kryptografischen Schlüsseln, die in von Apple gesteuerten HSMs (Hardware Security Modules) gespeichert sind

In den letzten Jahren hat Apple mit einigen externen Anbietern zusammengearbeitet, damit die „Image4“-Datenstrukturen, der Bestätigungscode und die Signierungsinfrastruktur zum Einsatz kommen, die von Apple Chips genutzt werden.

Wenn weder der Betrieb ohne Speicher noch Speicher plus sicheres Starten eine Option ist, verlangt das Design, dass Firmware-Aktualisierungen kryptografisch signiert und überprüft werden, bevor der permanente Speicher aktualisiert werden kann.

Rosetta 2 auf Mac-Computern mit Apple Chips

Mac-Computer mit Apple Chips können Code ausführen, der für den x86_64 Befehlssatz (Instruction Set) kompiliert wurde. Dafür nutzen sie einen Übersetzungsmechanismus namens *Rosetta 2*. Es stehen zwei Übersetzungsarten zur Verfügung: Just-in-time und Ahead-of-time.

Just-in-time-Übersetzung

Bei der Just-in-time (JIT)-Übersetzungspipeline wird ein x86_64 Mach-Objekt in einer frühen Ausführungsphase der Image-Datei erkannt. Wird es erkannt, übergibt der Kernel die Steuerung an einen speziellen Rosetta Übersetzungs-Stub anstatt an den dynamischen Link-Editor `dyld(1)`. Dieser Stub übersetzt x86_64-Seiten während der Ausführung der Image-Datei. Dieser Übersetzungsvorgang vollzieht sich vollständig innerhalb des Prozesses. Der Kernel verifiziert weiterhin die Code-Hashes aller x86_64-Seiten, indem er sie mit der Codesignatur abgleicht, die mit der Binärdatei verknüpft ist. Wenn die Hash-Werte nicht übereinstimmen, erzwingt der Kernel die für diesen Prozess geeignete Wiederherstellungsrichtlinie.

Ahead-of-time-Übersetzung

Bei der Ahead-of-time (AOT)-Übersetzung, werden x86_64-Binärdateien zu den Zeitpunkten aus dem Speicher ausgelesen, die das System als optimal für die Reaktionsschnelligkeit dieses Codes beurteilt. Die übersetzten Artefakte werden als spezielle Mach-Objektdateien in den Speicher geschrieben. Diese Datei ist einer ausführbaren Image-Datei ähnlich, aber sie ist markiert, sodass angezeigt wird, dass sie das übersetzte Produkt einer anderen Image-Datei ist.

Bei diesem Modell bezieht das AOT-Artefakt alle Informationen über seine Identität aus der ursprünglichen ausführbaren x86_64-Image-Datei. Um diese Bindung durchzusetzen, signiert eine privilegierte Entität im Benutzerbereich das Übersetzungsartefakt mit einem gerätespezifischen Schlüssel, der von der Secure Enclave verwaltet wird. Dieser Schlüssel wird nur der privilegierten Entität im Benutzerbereich zur Verfügung gestellt, die mit einer eingeschränkten Berechtigung als solche identifiziert wird. Das Codeverzeichnis, das für das Übersetzungsartefakt erstellt wird, enthält den Codeverzeichnis-Hash der ursprünglichen ausführbaren x86_64-Image-Datei. Die Signatur des Übersetzungsartefakts selber wird als *Ergänzungssignatur* bezeichnet.

Die AOT-Pipeline startet ähnlich wie die JIT-Pipeline: Der Kernel übergibt die Steuerung an die Rosetta-Laufzeit anstatt an den dynamischen Link-Editor `dyld(1)`. Anschließend sendet die Rosetta-Laufzeit eine IPC-Anfrage (Interprocess Communication) an den Rosetta-Systemservice. Dieser fragt ab, ob eine AOT-Übersetzung für die aktuelle ausführbare Image-Datei verfügbar ist. Wird eine solche gefunden, stellt der Rosetta-Dienst ein Handle für diese Übersetzung bereit, wonach sie in den Prozess gemappt und ausgeführt wird. Während der Ausführung erzwingt der Kernel die Codeverzeichnis-Hashes des Übersetzungsartefakts, welches durch die Signatur authentifiziert wird, die im gerätespezifischen Signierungsschlüssel hinterlegt ist. Die Codeverzeichnis-Hashes der ursprünglichen x86_64-Image-Datei sind nicht an diesem Prozess beteiligt.

Übersetzte Artefakte werden im Data Vault gespeichert, auf den keine andere Entität außer dem Rosetta-Dienst Laufzeitzugriff erhält. Der Rosetta-Dienst verwaltet den Zugriff auf seinen Cache, indem er schreibgeschützte Dateideskriptoren an einzelne Übersetzungsartefakte verteilt. Auf diese Weise wird der Zugriff auf den AOT-Artefakt-Cache eingeschränkt. Der IPC dieses Diensts und sein Footprint sind absichtlich sehr begrenzt, um die Angriffsfläche klein zu halten.

Stimmt der Codeverzeichnis-Hash der ursprünglichen x86_x64-Image-Datei nicht mit demjenigen überein, der in die Signatur des AOT-Übersetzungsartefakt kodiert ist, wird dies als gleichbedeutend mit einer ungültigen Codesignatur angesehen; in diesem Fall werden entsprechende Durchsetzungsmaßnahmen ergriffen.

Fragt ein Remote-Prozess die Berechtigungen oder andere Identitätseigenschaften einer ausführbaren AOT-Übersetzung beim Kernel ab, werden die Identitätseigenschaften der ursprünglichen x86_x64-Image-Datei ausgegeben.

Inhalte des statischen Cache für die Vertrauensstellung (Static Trust Cache)

macOS 11 (oder neuer) wird mit „fat“ Mach-Binärdateien ausgeliefert, die Slices des x86_X64- und des arm64-Computercodes enthalten. Benutzer können das x86_x64-Slice einer Systemdatei auf einem Mac mit Apple Chips mit der Rosetta-Pipeline ausführen; z. B. um ein Plug-In zu laden, für das keine native arm64-Variante existiert. Um diesen Ansatz zu unterstützen, enthält der Static Trust Cache, der mit macOS ausgeliefert wird, allgemein drei Codeverzeichnis-Hashes pro Mach-Objektdatei:

- Codeverzeichnis-Hash des arm64-Slice
- Codeverzeichnis-Hash des x86_64-Slice
- Codeverzeichnis-Hash der AOT-Übersetzung des x86_64-Slice

Der AOT-Übersetzungsprozess von Rosetta ist deterministisch und reproduziert die gleiche Ausgabe für jede Eingabe – unabhängig davon, wann oder auf welchem Gerät die Übersetzung durchgeführt wurde.

Während der Erstellung von macOS wird jedes Mach-Objekt durch die AOT-Übersetzungspipeline von Rosetta geleitet, die mit der jeweiligen Version von macOS verknüpft ist. Der daraus resultierende Codeverzeichnis-Hash wird im Trust Cache erfasst. Die übersetzten Produkte selbst sind aus Effizienzgründen nicht im Betriebssystem enthalten und können auf Anfrage des Benutzers neu erstellt werden.

Wenn eine x86_x64-Image-Datei auf einem Mac mit Apple Chips ausgeführt wird und sich der Codeverzeichnis-Hash dieser Image-Datei im statischen Trust Cache befindet, wird erwartet, dass sich der Codeverzeichnis-Hash des resultierenden AOT-Artefakts *ebenfalls* im statischen Trust Cache befindet. Solche Produkte werden nicht mit dem gerätespezifischen Schlüssel signiert, da die Signierungsautorität im sicheren Startvorgang verankert ist.

Nicht signierter x86_x64-Code

Mac-Computer mit Apple Chips erlauben keine Ausführung von nativem arm64-Code ohne verknüpfte Codesignatur. Dabei kann es sich um eine ganz einfache Ad-hoc-Codesignatur (vgl. `codesign(1)`) handeln, die keinen Teil der Identität der geheimen Hälfte eines asymmetrischen Schlüsselpaars aufweist. (Bei ihr handelt es sich einfach um eine unauthentifizierte Kennzahl der Binärdatei.)

Übersetzter x86_x64-Code darf dagegen aus Gründen der Binärkompatibilität ohne jegliche Signaturinformationen mit Rosetta ausgeführt werden. Diesem Code wird keine spezifische Identität durch den gerätespezifischen Signierungsprozess der Secure Enclave offengelegt. Er wird mit genau denselben Einschränkungen ausgeführt wie ein nativer unsignierter Code auf einem Intel-basierten Mac.

DMA-Schutz bei Mac-Computern

Computer müssen den direkten Speicherzugriff (Direct Memory Access, DMA) von Peripheriegeräten unterstützen, um einen hohen Durchsatz bei Highspeed-Schnittstellen wie PCIe, FireWire, Thunderbolt und USB zu erzielen. Sie müssen also in der Lage sein, Lese- und Schreibvorgänge im RAM auszuführen, ohne dass die CPU kontinuierlich involviert ist. Seit 2012 haben Mac-Computer als DMA-Schutz zahlreiche Technologien implementiert und so eine der besten und umfassendsten DMA-Schutzvorkehrungen getroffen, die es je auf einem PC gegeben hat.

DMA-Schutz bei einem Mac mit Apple Chips

Die Apple-SoCs umfassen für jeden DMA-Agent im System eine IOMMU-Einheit ([Input/Output Memory Management Unit](#)), die PCIe- und Thunderbolt-Ports einschließt. Jede IOMMU-Einheit verfügt über eine eigene Tabelle für die Adressübersetzung, die für die Übersetzung von DMA-Anfragen herangezogen wird. Das bedeutet, dass Peripheriegeräte, die über PCIe oder Thunderbolt verbunden sind, nur auf Arbeitsspeicher zugreifen können, der explizit zur Verwendung durch sie bereitgestellt wird. Peripheriegeräte können somit nicht auf Speicherbereiche anderer Teile des Systems zugreifen, etwa auf den Kernel oder die Firmware, oder auf Arbeitsspeicher, der anderen Peripheriegeräten zugewiesen ist. Wenn eine IOMMU-Einheit den Versuch eines Peripheriegeräts erkennt, auf Arbeitsspeicher zuzugreifen, der nicht zur Verwendung durch dieses Peripheriegerät bereitgestellt wurde, signalisiert sie eine kritische Kernel-Situation.

DMA-Schutz bei Intel-basierten Mac-Computern

Intel-basierte Mac-Computer mit Intel Virtualization Technology for Directed I/O (Vt-d) initialisieren die IOMMU-Einheit und damit die Möglichkeit, die DMA-Neuzuweisung vorzunehmen und sie per Interrupt zu unterbrechen, in einer sehr frühen Phase des Startvorgangs, um verschiedenen Klassen sicherheitsrelevanter Schwachstellen entgegenzuwirken. Die Apple-Hardware für IOMMU verwendet zu Beginn des Vorgangs die Richtlinie „default-deny“, sodass DMA-Anfragen durch Peripheriegeräte sofort nach dem Einschalten des Systems automatisch abgeblockt werden. Nachdem die Software initialisiert wurde, lassen die IOMMU-Einheiten DMA-Anfragen von Peripheriegeräten nur zu, wenn diese sich auf Arbeitsspeicherbereiche beziehen, die explizit für die Verwendung durch sie bereitgestellt wurden.

Hinweis: Das Unterbrechen (Interrupt) der Neuzuweisung für PCIe erübrigt sich bei einem Mac mit Apple Chips, da jede IOMMU-Einheit die MSIs für die jeweils eigenen Peripheriegeräte verarbeitet.

Ab macOS 11 führen alle Mac-Computer mit Apple T2 Security Chip UEFI-Treiber aus, die DMA in einer restriktiven Ring-3-Umgebung ermöglichen, wenn die betreffenden Treiber mit externen Geräten gekoppelt werden. Diese Eigenschaft trägt dazu bei, die missbräuchliche Nutzung sicherheitsrelevanter Schwachstellen abzuwehren. Solche Schwachstellen können auftreten, wenn ein Gerät mit Schadsoftware während des Startvorgangs auf unvorhergesehene Weise mit einem UEFI-Treiber interagiert. Im Speziellen reduziert es die Auswirkungen von Schwachstellen bei der Handhabung von DMA-Puffern durch die Treiber.

Kernel-Erweiterungen bei macOS

Wenn Kernel-Erweiterungen von Drittanbietern aktiviert werden, können sie ab macOS 11 nicht „on demand“ in den Kernel geladen werden. Sie werden vielmehr in einer *Auxiliary Kernel Collection (AuxKC)* zusammengeführt, die im Zuge des Startvorgangs geladen wird. Bei einem Mac mit Apple Chips wird die Kennzahl der AuxKC in die LocalPolicy-Datei eingetragen und signiert (bei früherer Hardware residiert die AuxKC auf dem Datenvolume). Ein AuxKC-Neuaufbau setzt die Genehmigung durch den Benutzer und den Neustart von macOS voraus, damit die Änderungen in den Kernel geladen werden. Außerdem setzt er voraus, dass der sichere Startvorgang für den Modus „Reduzierte Sicherheit“ konfiguriert ist.

Wichtig: Kernel-Erweiterungen werden für macOS nicht mehr empfohlen. Kernel-Erweiterungen stellen ein Risiko für die Integrität und Zuverlässigkeit des Betriebssystems dar. Daher empfiehlt Apple den Benutzern, Lösungen ohne Erweiterungen des Kernels vorzuziehen.

Kernel-Erweiterungen auf einem Mac mit Apple Chips

Kernel-Erweiterungen müssen für einen Mac mit Apple Chips explizit aktiviert werden. Dazu muss beim Starten die Taste „Ein/Aus“ gedrückt werden, um in den Modus 1TR (One True Recovery) zu wechseln. Danach muss die Herabstufung auf „Reduzierte Sicherheit“ vorgenommen und die Option zum Aktivieren von Kernel-Erweiterungen aktiviert werden. Für diese Aktion muss außerdem ein Administratorpasswort eingegeben werden, um die Herabstufung zu autorisieren. Die Kombination aus 1TR-Modus und obligatorischer Passworтеingabe erschwert es Angreifern, die allein auf die Software abzielen, von einem Punkt innerhalb von macOS Kernel-Erweiterungen in macOS einzuschleusen, die sie dahingehend ausnutzen können, sich Kernel-Berechtigungen anzueignen.

Nachdem ein Benutzer das Laden von Kernel-Erweiterungen autorisiert hat, wird der oben beschriebene Arbeitsablauf „Laden vom Benutzer genehmigter Kernel-Erweiterungen“ verwendet, um die Installation der Kernel-Erweiterungen zu autorisieren. Die für den obigen Arbeitsablauf verwendete Autorisierung wird auch verwendet, um einen SHA-384-Hash-Wert der UAKL (User-Authorized Kext List) in der LocalPolicy-Datei zu erfassen. Der Daemon für die Kernel-Verwaltung (kmd) ist in der Folge dafür zuständig, dass nur solche Kernel-Erweiterungen, die in der UAKL enthalten sind, für die Aufnahme in die AuxKC validiert werden.

- Wenn SIP (System Integrity Protection) aktiviert ist, wird die Signatur jeder Kernel-Erweiterung verifiziert, bevor sie in die AuxKC aufgenommen wird.
- Wenn SIP deaktiviert ist, wird das Signieren von Kernel-Erweiterungen nicht erzwungen.

Dies ermöglicht den Arbeitsablauf „Permissive Sicherheit“, mit dem Entwickler und Kunden, die nicht am Apple Developer Program teilnehmen, Kernel-Erweiterungen testen können, ehe sie signiert werden.

Nachdem die AuxKC erstellt wurde, wird ihre Kennzahl an die Secure Enclave gesendet, um signiert und in die Image4-Datenstruktur einbezogen zu werden, die während des Startvorgangs durch iBoot evaluiert werden kann. Im Zuge des AuxKC-Aufbaus wird auch ein KEXT-Beleg generiert. Dieser Beleg umfasst eine Liste der Kernel-Erweiterungen, die tatsächlich in die AuxKC aufgenommen wurden, da es sich dabei um eine Untermenge der UAKL handeln kann, wenn verbotene Kernel-Erweiterungen erkannt werden. Ein SHA-384-Hash-Wert der Image4-Datenstruktur der AuxKC und des KEXT-Belegs werden in die LocalPolicy-Datei aufgenommen. Der Hash-Wert der Image4-Struktur der AuxKC wird von iBoot beim Starten für eine zusätzliche Verifizierung verwendet, um auszuschließen, dass eine von der Secure Enclave signierte ältere Image4-Datei der AuxKC mit einer neueren LocalPolicy-Datei gestartet wird. Der KEXT-Beleg wird von Subsystemen wie Apple Pay verwendet, um festzustellen, ob aktuell irgendwelche Kernel-Erweiterungen geladen sind, die die Vertrauenswürdigkeit von macOS beeinträchtigen könnten. Ist dies der Fall, können Apple Pay-Funktionen möglicherweise deaktiviert werden.

Alternativen zu Kernel-Erweiterungen (macOS 10.15 oder neuer)

macOS 10.15 erlaubt Entwicklern, das Funktionsspektrum von macOS zu erweitern. Entwickler können hierzu Systemerweiterungen installieren und verwalten, die statt auf Kernel-Ebene im User Space ausgeführt werden. Systemerweiterungen steigern die Stabilität und Sicherheit von macOS dadurch, dass sie im User Space laufen. Auch wenn Kernel-Erweiterungen grundsätzlich vollen Zugriff auf das gesamte Betriebssystem haben, werden Erweiterungen, die im User Space ausgeführt werden, nur die Berechtigungen erteilt, die für die Ausführung ihrer speziellen Funktion erforderlich sind.

Entwickler können Frameworks wie DriverKit, EndpointSecurity und NetworkExtension nutzen, um USB-Treiber und Treiber für Benutzeroberflächen, Tools für die Endgerätesicherheit (wie Prävention von Datenverlusten und andere Endgeräte-Agenten) sowie VPN- und Netzwerktools zu schreiben, ohne Kernel-Erweiterung schreiben zu müssen. Sicherheitsagenten anderer Anbieter sollten nur dann verwendet werden, wenn sie die Vorteile dieser APIs nutzen oder einen soliden Plan haben, wie sie diese übernehmen und auf Kernel-Erweiterungen verzichten können.

Laden vom Benutzer genehmigter Kernel-Erweiterungen

Zur Verbesserung der Sicherheit ist die Einwilligung des Benutzers erforderlich, um Kernel-Erweiterungen zu laden, die mit oder nach der Installation von macOS 10.13 installiert werden. Dieser Prozess wird auch als „Laden von Kernel-Erweiterungen nach Benutzereinwilligung“ bezeichnet. Für die Einwilligung in eine Kernel-Erweiterung ist eine Administratorautorisierung erforderlich. Kernel-Erweiterungen setzen keine Autorisierung voraus, wenn sie:

- auf einem Mac mit macOS 10.12 (oder älter) installiert wurden,
- zuvor bewilligte Erweiterungen ersetzen,
- durch Verwendung des Befehlszeilenprogramms `spctl`, das beim Starten über `recoveryOS` bereitgestellt wird, ohne Benutzereinwilligung geladen werden dürfen,
- mittels Mobile Device Management (MDM)-Konfiguration geladen werden dürfen.

Ab macOS 10.13.2 können Benutzer MDM verwenden, um eine Liste von Kernel-Erweiterungen festzulegen, die ohne Benutzereinwilligung geladen werden. Diese Option erfordert einen Mac mit macOS 10.13.2, der in [Apple School Manager](#) oder [Apple Business Manager](#) oder in einer MDM-Lösung durch den Benutzer registriert wird.

OROM-Sicherheit (Option ROM) in macOS

Hinweis: OROMs werden aktuell nicht auf Mac-Computern mit Apple Chips unterstützt.

OROM-Sicherheit bei Mac-Computern mit Apple T2 Security Chip

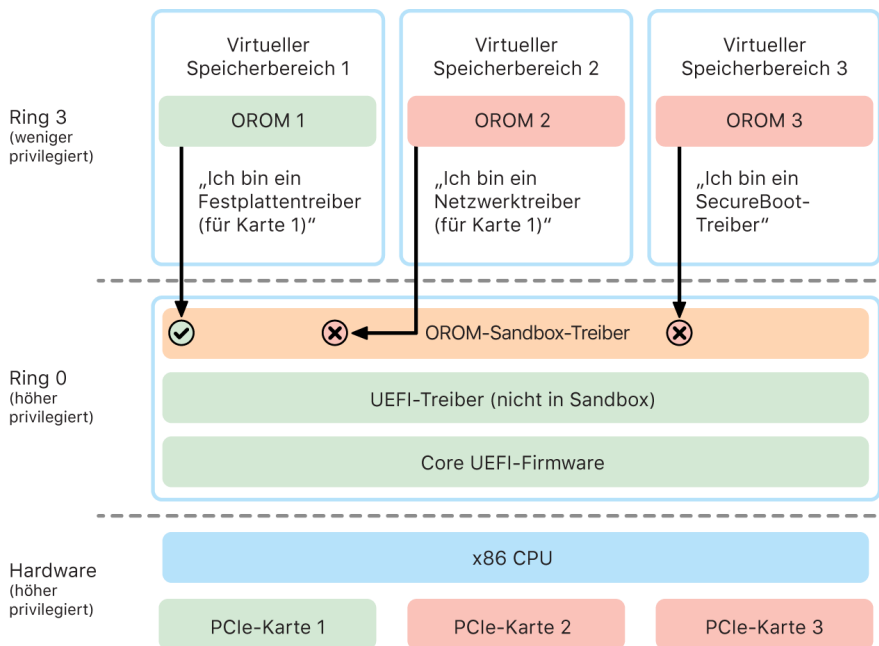
Sowohl bei Thunderbolt- als auch bei PCIe-Geräten kann ein „Option ROM“, kurz „OROM“, d. h. ein Erweiterungs-ROM physisch an das Gerät angeschlossen werden. (Hierbei handelt es sich in der Regel nicht um einen echten ROM, sondern um einen wiederbeschreibbaren Chip, der Firmware speichert.) Auf UEFI-basierten Systemen besteht diese Firmware normalerweise aus einem UEFI-Treiber, der von der UEFI-Firmware eingelesen und ausgeführt wird. Der ausgeführte Code hat die Aufgabe, die Hardware zu initialisieren und zu konfigurieren, von der er abgerufen wird, damit die Hardware für den Rest der Firmware nutzbar gemacht werden kann. Diese Funktion ist erforderlich, damit spezialisierte Hardware anderer Anbieter in frühesten Startphasen geladen und betrieben werden kann – zum Beispiel zum Starten von externen RAID-Arrays.

Da aber OROMs generell wiederbeschreibbar sind, kann ein Angreifer den OROM eines legitimen Peripheriegeräts überschreiben. Der Code des Angreifers wird frühzeitig im Startvorgang ausgeführt und kann die Ausführungsumgebung manipulieren und die Integrität nachfolgend geladener Software verletzen. Wenn der Angreifer sein eigenes schadhaftes Gerät in das System einschleust, ist er ebenfalls in der Lage, bösartigen Code auszuführen.

Bei macOS 10.12.3 wurde das Verhalten von Mac-Computern, die nach 2011 auf den Markt kamen, so geändert, dass beim Starten des Mac-Computers standardmäßig keine OROMs ausgeführt werden, sofern nicht eine bestimmte Tastenkombination gedrückt wird. Diese Kombination von Schlüsseln dient dem Schutz vor schädlichen OROMs, die versehentlich in die macOS-Startsequenz eingeführt werden. Das Standardverhalten des Dienstprogramms für Firmware-Passwörter wurde wie folgt geändert: Wenn der Benutzer nun ein Firmware-Passwort festlegt, können OROMs nicht ausgeführt werden, selbst wenn die Tastenkombination gedrückt wird. Dies ist eine Schutzmaßnahme gegen einen physisch anwesenden Angreifer, der in böser Absicht einen schädlichen OROM einschleust. Für Benutzer, die weiterhin OROMs ausführen müssen und ein Firmware-Passwort definiert haben, kann mit dem Befehlszeilenprogramm `firmwarepasswd` von macOS eine vom Standard abweichende Option konfiguriert werden.

Sicherheit bei OROM-Sandbox

Bei macOS 10.15 wurde die UEFI-Firmware aktualisiert. Sie enthält nun einen Mechanismus für das Sandboxing von OROMs und das Entziehen von Berechtigungen von OROMs. Die UEFI-Firmware führt üblicherweise den gesamten Code (inklusive OROMs) auf der höchsten Berechtigungsstufe für die CPU aus (als „Ring 0“ bezeichnet). Außerdem verwendet sie einen einzelnen, gemeinsam genutzten virtuellen Speicherbereich für den gesamten Code und alle Daten. Ring 0 ist die Berechtigungsstufe, auf der der macOS-Kernel ausgeführt wird, wohingegen Apps auf der niedrigeren Berechtigungsstufe Ring 3 ausgeführt werden. Die OROM-Sandbox entzieht OROMs die Berechtigung, indem sie wie der Kernel die virtuelle Speichertrennung nutzt und dann veranlasst, dass die OROMs in Ring 3 ausgeführt werden.



Die Sandbox sorgt zudem für eine deutliche Einschränkung sowohl der Schnittstellen, die die OROMs aufrufen können (ähnlich wie beim Filtern von Systemaufrufen in Kernels), als auch der Art des Geräts, als das sich ein OROM registrieren kann (ähnlich wie bei App-Genehmigungen). Dieses Design bietet den Vorteil, dass schädliche OROMs innerhalb von Ring 0 keine direkten Schreibvorgänge mehr ausführen können. Stattdessen sind sie auf die sehr enge und gut definierte Sandbox-Schnittstelle beschränkt. Diese beschränkte Schnittstelle reduziert die Angriffsfläche erheblich und zwingt Angreifer dazu, zuerst die Sandbox zu überwinden und ihre Berechtigungen aufzustoßen.

Sicherheit für UEFI-Firmware bei Intel-basierten Mac-Computern

Ein Intel-basierter Mac mit Apple T2 Security Chip bietet Sicherheit dank der UEFI-Firmware (Intel-Firmware).

Übersicht

Seit 2006 nutzen Mac-Computer mit einer Intel-basierten CPU eine Intel-Firmware, die auf dem Extensible Firmware Interface (EFI) Development Kit (EDK) Version 1 oder Version 2 basiert. EDK2-basierter Code ist mit der UEFI-Spezifikation (Unified Extensible Firmware Interface) konform. In diesem Abschnitt wird die Intel-Firmware als *UEFI-Firmware* bezeichnet. Die UEFI-Firmware war der erste Code, der auf dem Intel-Chip ausgeführt wurde.

Für einen Intel-basierten Mac ohne Apple T2 Security Chip ist der Vertrauensanker für die UEFI-Firmware der Chip, auf dem die Firmware gespeichert ist. Aktualisierungen der UEFI-Firmware werden von Apple digital signiert und durch die Firmware überprüft, bevor der Speicher aktualisiert wird. Zum Schutz vor Rollback-Angriffen müssen Aktualisierungen stets eine Version aufweisen, die neuer ist als die aktuell vorhandene Version. Ein Angreifer mit direktem Zugriff auf den Mac könnte jedoch Hardware an den Firmware-Speicherchip anschließen und den Chip mit schädlichen Inhalten aktualisieren. Wenn im frühen Startvorgang der UEFI-Firmware (bevor diese den Speicherchip mit einem Schreibschutz versieht) Sicherheitslücken entdeckt werden, könnte auch dies zu einer dauerhaften Infizierung der UEFI-Firmware führen. Hierbei handelt es sich um eine durch die Hardwarearchitektur bedingte Beschränkung, die in den meisten Intel-basierten PCs auftritt und die in allen Intel-basierten Mac-Computern ohne T2-Chip vorhanden ist.

Um physische Angriffe zu verhindern, die die UEFI-Firmware unterwandern, wurde die Architektur von Mac-Computern dahingehend erneuert, dass der Vertrauensanker in der UEFI-Firmware des T2-Chips liegt. Auf diesen Mac-Computern ist der Vertrauensanker für die UEFI-Firmware insbesondere die T2-Firmware, wie im Abschnitt [Startvorgang bei Intel-basierten Mac-Computern](#) beschrieben wird.

Unterkomponente Intel Management Engine (ME)

Bei der *ME-Firmware (Intel Management Engine)* handelt es sich um eine Unterkomponente, die innerhalb der UEFI-Firmware gespeichert ist. Die ME – ein separater Prozessor und separates Subsystem in Intel-Chips – wird in erster Linie für den Schutz von Audio- und Video-Urheberrechten auf Mac-Computern eingesetzt, die nur über Intel-basierte Grafikschnittstellen verfügen. Um die Angriffsfläche dieser Unterkomponente zu reduzieren, führt ein Intel-basierter Mac eine eigene ME-Firmware aus, bei der auf die meisten Komponenten verzichtet wird. Da die ME-Firmware des Mac kleiner als der von Intel bereitgestellte, standardmäßige Mindest-Build ist, fehlen zahlreiche Komponenten, die in der Vergangenheit Gegenstand öffentlicher Angriffe durch Sicherheitsexperten waren.

System Management Mode (SMM)

Intel-Prozessoren verfügen über einen speziellen Ausführungsmodus, der sich vom normalen Betrieb unterscheidet. Dieser Modus wird als *System Management Mode (SMM)* bezeichnet. Ursprünglich wurde er eingeführt, um zeitkritische Operationen wie beispielsweise die Energieverwaltung zu handhaben. Mac-Computer haben für die Ausführung solcher Aufgaben jedoch schon immer einen eigenen Mikrocontroller verwendet, der als *System Management Controller (SMC)* bezeichnet wird. Mittlerweile ist der SMC kein separater Mikrocontroller mehr, sondern ein integraler Bestandteil des T2-Chips.

Systemsicherheit bei watchOS

Die Apple Watch nutzt viele derselben hardwarebasierten Funktionen für die Plattformsicherheit, die auch iOS und iPadOS verwenden. Die Apple Watch übernimmt beispielsweise folgende Aufgaben:

- Durchführung des sicheren Startens und sicherer Softwareaktualisierungen
- Unterstützung der Integrität des Betriebssystems
- Schutz der Daten auf dem Gerät und bei der Kommunikation mit dem jeweils gekoppelten iPhone oder mit dem Internet

Die unterstützten Technologien umfassen jene, die unter „Systemsicherheit“ aufgelistet sind (z. B. KIP, SKP und SCIP), sowie Datensicherheit, den Schlüsselbund und Netzwerktechnologien.

Aktualisieren von watchOS

watchOS kann so konfiguriert werden, dass es nachts aktualisiert wird. Weitere Informationen dazu, wie der Apple Watch-Code gespeichert und während der Aktualisierung verwendet wird, sind im Abschnitt „[Keybags](#)“ zu finden.

Handgelenkerkennung

Wenn die Handgelenkerkennung aktiviert ist, wird das Gerät automatisch gesperrt, kurz nachdem der Benutzer es vom Handgelenk abgenommen hat. Wenn die Handgelenkerkennung deaktiviert ist, stellt das Kontrollzentrum eine Option zum Sperren der Apple Watch bereit. Wenn die Apple Watch gesperrt ist, kann Apple Pay nur verwendet werden, nachdem der Code auf der Apple Watch eingegeben wurde. Die Handgelenkerkennung wird mit der App „Watch“ auf dem iPhone ausgeschaltet. Diese Einstellung kann mithilfe einer MDM-Lösung (Mobile Device Management) erzwungen werden.

Aktivierungssperre

Wenn „Wo ist?“ auf dem iPhone aktiviert ist, kann die damit gekoppelte Apple Watch die Aktivierungssperre verwenden. Die Aktivierungssperre erschwert unbefugten Personen den Verkauf oder die Verwendung einer Apple Watch, die verloren oder gestohlen wurde. Die Aktivierungssperre bewirkt, dass die Apple-ID des Benutzers und dessen Passwort eingegeben werden müssen, um die Kopplung aufzuheben, die Daten zu löschen oder die Apple Watch neu zu aktivieren.

Sicheres Koppeln mit iPhone-Geräten

Eine Apple Watch kann nur mit jeweils einem iPhone gleichzeitig gekoppelt werden. Wenn die Koppelung der Apple Watch aufgehoben wird, kommuniziert das iPhone Anweisungen zum Löschen aller Inhalte und Daten von dieser Apple Watch.

Die Kopplung des Apple Watch-Geräts mit dem iPhone ist durch einen Out-of-Band-Prozess abgesichert, bei dem öffentliche Schlüssel ausgetauscht werden. Anschließend werden die BLE-Link Shared Secrets (Bluetooth Low Energy) ausgetauscht. Auf der Apple Watch wird ein animiertes Muster angezeigt, das von der Kamera des iPhone erfasst wird. Das Muster enthält ein codiertes Secret, das für die BLE 4.1-Out-of-Band-Kopplung verwendet wird. Als Ersatzfunktion zur Kopplung (falls erforderlich) wird die Eingabe eines BLE-Schlüssels verwendet.

Nachdem die BLE-Sitzung etabliert und unter Verwendung des höchsten Sicherheitsprotokolls verschlüsselt wurde, das laut Bluetooth Core Specification verfügbar ist, tauschen das iPhone und das Apple Watch-Gerät ihre Schlüssel auf einem der folgenden Wege aus:

- Ein auf Basis von Apple Identity Service (IDS) adaptierter Prozess, der im Abschnitt [Sicherheit bei iMessage – Übersicht](#) beschrieben wird.
- Ein Schlüsselaustausch unter Verwendung von IKEv2/IPsec. Der anfängliche Schlüsselaustausch wird entweder mithilfe des Schlüssels der Bluetooth-Sitzung (bei Kopplungsszenarien) oder mithilfe der IDS-Schlüssel (bei der Aktualisierung des Betriebssystems) authentifiziert. Jedes Gerät generiert dazu ein Paar randomisierter öffentlicher und privater 256-Bit-Ed25519-Schlüssel, deren öffentliche Schlüssel im Zuge des anfänglichen Schlüsselaustauschs ausgetauscht werden.

Hinweis: Der Mechanismus, der beim Austausch der Schlüssel und bei der Verschlüsselung zum Einsatz kommt, hängt von der Version der Betriebssysteme ab, die auf dem iPhone und dem Apple Watch-Gerät ausgeführt werden. iPhone-Geräte mit iOS 13 (oder neuer), die mit einem Apple Watch-Gerät mit watchOS 6 (oder neuer) gekoppelt werden, verwenden ausschließlich IKEv2/IPsec für den Schlüsselaustausch und die Verschlüsselung.

Nachdem die Schlüssel ausgetauscht wurden, geschieht Folgendes:

- Der Schlüssel der Bluetooth-Sitzung wird verworfen und die gesamte Kommunikation zwischen dem iPhone und dem Apple Watch-Gerät wird mithilfe einer der oben beschriebenen Methoden verschlüsselt – wobei verschlüsselte Bluetooth-, WLAN- und Mobilfunk-Links eine zweite Verschlüsselungsebene bereitstellen.
- (Nur IKEv2/IPsec) Die Schlüssel werden im Schlüsselbund des Systems gespeichert und für die Authentifizierung künftiger IKEv2/IPsec-Sitzungen zwischen den Geräten verwendet. Die weitere Kommunikation zwischen diesen Geräten ist verschlüsselt und die Integrität wird durch AES-256-GCM oder ChaCha20-Poly1305 (256-Bit-Schlüssel) auf iPhone-Geräten mit iOS 15 (oder neuer) geschützt, die mit einer Apple Watch Series 4 (oder neuer) mit watchOS 8 (oder neuer) gekoppelt sind.

Die Bluetooth Low Energy-Geräteadresse wird in 15-Minuten-Intervallen geändert, um das Risiko zu reduzieren, dass das Gerät durch Senden einer dauerhaften ID lokal verfolgt wird.

Zur Unterstützung von Apps, die Streaming-Daten benötigen, erfolgt die Verschlüsselung mithilfe der Methoden, die im Abschnitt [FaceTime-Sicherheit](#) beschrieben werden. Hierzu wird der vom gekoppelten iPhone bereitgestellte IDS-Dienst (Apple Identity Service) oder eine direkte Internetverbindung verwendet.

Die Apple Watch implementiert hardwareverschlüsselten Speicher sowie klassenbasierten Schutz für Dateien und Schlüsselbundobjekte. Darüber hinaus werden zugriffsgesteuerte Keybags für Schlüsselbundobjekte verwendet. Die für die Kommunikation zwischen dem Apple Watch- und dem iPhone-Gerät verwendeten Schlüssel sind durch einen klassenbasierten Schutz gesichert. Weitere Informationen sind unter [Keybags für Datensicherheit](#) zu finden.

Automatisches Entsperren mit der Apple Watch

Für noch mehr Komfort bei der Verwendung von mehreren Apple-Geräten können einige Geräte in bestimmten Situationen andere Geräte automatisch entsperren. Es gibt drei Möglichkeiten zum automatischen Entsperren:

- Eine Apple Watch kann mit einem iPhone entsperrt werden.
- Ein Mac kann mit einer Apple Watch entsperrt werden.
- Ein iPhone kann mit einer Apple Watch entsperrt werden, wenn ein Benutzer eine Mund-Nasen-Bedeckung trägt.

Alle drei Verwendungsmöglichkeiten haben dasselbe Fundament: ein gegenseitig authentifiziertes STS-Protokoll (Station-to-Station) mit Langzeitschlüsseln, die beim Aktivieren der Funktion ausgetauscht werden, und individuellen temporären Sitzungsschlüsseln, die für jede Anfrage ausgehandelt werden. Unabhängig vom Kommunikationskanal wird der STS-Tunnel direkt zwischen den Secure Enclaves der beiden Geräte ausgehandelt und das gesamte kryptografische Material verbleibt in dieser sicheren Domain (außer bei Mac-Computern ohne Secure Enclave, die den STS-Tunnel im Kernel auflösen).

Entsperren

Eine vollständige Sequenz zum Entsperren wird in zwei Phasen unterteilt. In einem ersten Schritt erstellt das zu entsperrende Gerät (das „Ziel“) ein kryptografisches Secret zum Entsperren und sendet dieses an das Gerät, das das Entsperren durchführt (der „Initiator“). In einem zweiten Schritt entsperrt der Initiator mit dem zuvor erstellten Secret das Gerät.

Um das automatische Entsperren zu ermöglichen, wird zwischen den Geräten eine BLE-Verbindung hergestellt. Im Anschluss wird ein vom Zielgerät zufällig erstelltes, 32-Byte langes Secret zum Entsperren über den STS-Tunnel an den Initiator gesendet. Beim nächsten Entsperren mit biometrischen Daten oder einem Code verpackt das Zielgerät den codebasierter Schlüssel (PDK) mit dem Secret zum Entsperren und löscht das Secret zum Entsperren aus dem Speicher.

Zum Entsperren stellt das Gerät eine neue BLE-Verbindung her und verwendet dann ein Peer-to-Peer-WLAN, um auf sichere Weise die Entfernung zwischen den Geräten zu schätzen. Wenn sich die Geräte im spezifizierten Bereich befinden und die erforderlichen Sicherheitsrichtlinien erfüllt sind, sendet der Initiator das Secret zum Entsperren über den STS-Tunnel an das Ziel. Das Ziel erstellt dann ein neues 32-Byte langes Secret zum Entsperren und sendet es zurück an den Initiator. Wenn das aktuelle Secret zum Entsperren, das der Initiator gesendet hat, den Entsperreintrag erfolgreich entschlüsselt, wird das Zielgerät entsperrt und der PDK mit neuen Secret zum Entsperren verpackt. In einem letzten Schritt werden das neue Secret zum Entsperren und der PDK aus dem Speicher des Zielgeräts gelöscht.

Sicherheitsrichtlinien für das automatische Entsperren mit der Apple Watch

Für noch mehr Komfort kann die Apple Watch direkt nach der Erstkonfiguration mit einem iPhone entsperrt werden, ohne dass der Benutzer dafür zuerst den Code auf der Apple Watch eingeben muss. Um dies zu ermöglichen, wird das zufällig erstellte Secret zum Entsperren (das nach dem Aktivieren der Funktion und dem ersten Entsperrvorgang erstellt wird) verwendet, um einen langlebigen Escrow-Eintrag zu erstellen, der im Apple Watch-Keybag gesichert wird. Das Secret für den Escrow-Eintrag wird im iPhone-Schlüsselbund gesichert und nach jedem Neustart der Apple Watch verwendet, um selbstständig eine neue Sitzung zu erstellen.

Sicherheitsrichtlinien für das automatische Entsperren mit dem iPhone

Beim automatischen Entsperren des iPhone mit der Apple Watch gelten zusätzliche Sicherheitsrichtlinien. Die Apple Watch kann nicht anstelle von Face ID für andere Aufgaben auf dem iPhone verwendet werden, etwa für Apple Pay oder die Autorisierung von Apps. Wenn die Apple Watch ein gekoppeltes iPhone erfolgreich entsperrt, wird auf der Apple Watch eine Mitteilung angezeigt und ein haptisches Signal ausgegeben. Wenn der Benutzer in der Mitteilung auf die Taste zum Sperren des iPhone tippt, sendet die Apple Watch dem iPhone einen Sperrbefehl via BLE. Nachdem der Sperrbefehl erhalten wurde, wird das iPhone gesperrt. Darüber hinaus werden Face ID und auch das Entsperren mit der Apple Watch deaktiviert. Das iPhone muss beim nächsten Mal mit dem Code des iPhone entsperrt werden.

Um ein gekoppeltes iPhone mit der Apple Watch erfolgreich zu entsperren (wenn die Funktion aktiviert ist), müssen die folgenden Kriterien erfüllt sein:

- Das iPhone muss mindestens einmal mit einer anderen Methode entsperrt worden sein, nachdem die zugehörige Apple Watch am Handgelenk getragen und entsperrt wurde.
- Die Sensoren müssen in der Lage sein, eine Mund-Nasen-Bedeckung zu erkennen.
- Die gemessene Entfernung darf 2–3 Meter nicht überschreiten.
- Die Apple Watch darf sich nicht im Modus für die Schlafenszeit befinden.
- Die Apple Watch oder das iPhone müssen kürzlich entsperrt worden sein oder die Apple Watch muss physische Bewegungen erkannt haben, die darauf schließen lassen, dass die Person aktiv ist (und z. B. nicht schläft).
- Das iPhone muss mindestens einmal innerhalb der letzten 6,5 Stunden entsperrt worden sein.
- Das iPhone muss sich in einem Zustand befinden, in dem Face ID das Gerät entsperren darf. (Weitere Informationen sind unter [Face ID, Touch ID, Codes und Passwörter](#) zu finden.)

Bestätigen in macOS mit der Apple Watch

Wenn das automatische Entsperren mit der Apple Watch aktiviert ist, kann die Apple Watch anstelle von oder in Kombination mit Touch ID verwendet werden, um Aufforderungen zur Autorisierung und Authentifizierung von Folgendem zu bestätigen:

- macOS- und Apple-Apps, die eine Autorisierung verlangen
- Drittanbieter-Apps, die eine Authentifizierung verlangen
- Gesicherte Safari-Passwörter
- Geschützte Notizen

Sichere Verwendung von WLAN, Mobilfunk, iCloud und Gmail

Wenn sich die Apple Watch außerhalb der Bluetooth-Reichweite befindet, kann alternativ eine WLAN- oder Mobilfunk-Verbindung verwendet werden. Die Apple Watch stellt automatisch eine Verbindung zu WLANs her, mit denen sich zuvor das gekoppelte iPhone verbunden hat und dessen Anmeldedaten mit der Apple Watch synchronisiert wurden, während sich beide Geräte in Reichweite befanden. Das Verhalten „Automatisch verbinden“ kann danach auf Netzwerkbasis im Bereich „Einstellungen“ > „WLAN“ auf dem Apple Watch-Gerät konfiguriert werden. WLANs, zu denen von keinem der Geräte zuvor eine Verbindung hergestellt wurde, können im Abschnitt „WLAN“ in der App „Einstellungen“ der Apple Watch manuell verbunden werden.

Befinden sich Apple Watch und iPhone außerhalb ihrer Reichweite, stellt die Apple Watch direkt eine Verbindung zu iCloud und Gmail-Servern her, um Mail abzurufen. In dieser Situation erfolgt keine Synchronisation der Mail-Daten mit dem gekoppelten iPhone über das Internet. Bei Gmail-Accounts muss sich der Benutzer im Abschnitt „Mail“ der App „Watch“ auf dem iPhone bei Google authentifizieren. Der von Google empfangene OAuth-Token wird in verschlüsseltem Format über Apple IDS (Identity Service) an die Apple Watch gesendet. Danach kann es zum Abrufen von Mail verwendet werden. Dieser OAuth-Token wird nicht vom gekoppelten iPhone für die Konnektivität mit dem Gmail-Server verwendet.

Generieren von Zufallszahlen

Wichtige Bausteine für eine zuverlässig geschützte Software sind die sogenannten kryptografisch sicheren Zufallszahlengeneratoren (CPRNGs – cryptographically secure pseudo-random number generators). Apple stellt hierfür einen vertrauenswürdigen Software-CPRNG bereit, der in den Kernels von iOS, iPadOS, macOS, tvOS und watchOS ausgeführt wird. Seine Aufgabe ist es, reine Entropie vom System zu aggregieren und sowohl im Kernel als auch im Benutzerbereich (User Space) sichere Zufallszahlen für Nutzer bereitzustellen.

Entropiequellen

Der Kernel-CPRNG wird während des Startvorgangs und der Nutzungsdauer des Geräts von mehreren Entropiequellen versorgt. Hierzu gehören (bedingt durch die Verfügbarkeit):

- Der TRNG der Hardware der Secure Enclave
- Während des Startvorgangs erfasste zeitbasierte Jitter
- Von Hardware-Interrupts erfasste Entropie
- Eine Seed-Datei, die für den Fortbestand der Entropie während der Startvorgänge eingesetzt wird
- Intel-Zufallsanweisungen – z. B. RDSEED und RDRAND (nur Intel-basierte Mac-Computer)

Kernel-CPRNG

Der Kernel-CPRNG ist ein von Fortuna abgeleitetes Design, das auf eine 256-Bit-Sicherheitsebene abzielt. Er stellt hochwertige Zufallszahlen für Nutzer des Benutzerbereichs (User Space) über die folgenden APIs bereit:

- Systemaufruf `getentropy(2)`
- Zufallsgerät (`/dev/random`)

Der Kernel-CPRNG akzeptiert die vom Benutzer bereitgestellte Entropie über Schreibvorgänge auf dem Zufallsgerät.

Apple Security Research Device

Das Apple Security Research Device ist ein iPhone für die Sicherheitsforschung, das mit speziellen Sicherungsschaltungen ausgestattet ist („specially fused“). Das iPhone ermöglicht es Sicherheitsforschern, Forschungsarbeiten an und mit iOS vorzunehmen, ohne sich über die Sicherheitsfunktionen der Plattform hinwegzusetzen oder sie zu deaktivieren. Das Gerät erlaubt es Forschern, Sideload-Inhalte zu verwenden und mit Berechtigungen auszuführen, die denen der Plattform gleichkommen, und so Forschungsarbeiten auf einer Plattform durchzuführen, die Produktionsgeräte genauer nachbildet.

Damit sichergestellt ist, dass die Benutzergeräte nicht durch die für die Sicherheitsforschung geltende Ausführungsrichtlinie beeinträchtigt werden, werden Richtlinienänderungen in einer Variante von iBoot und in der Boot Kernel-Sammlung implementiert. Sie lassen sich nicht auf der Benutzerhardware starten. Das für die Forschung konzipierte iBoot-Objekt prüft, ob ein neuer „Fusing“- oder Schaltungsstatus vorliegt, und startet eine Schleife, die eine kritische Situation signalisiert, wenn es auf Hardware ausgeführt wird, die nicht über für Forschung ausgestattete Sicherungsschaltungen verfügt.

Das cryptex-Subsystem erlaubt es einem Forscher, einen [personalisierten Cache für die Vertrauensstellung \(„Trust Cache“\)](#) und eine Image-Datei mit entsprechendem Inhalt zu laden. Damit sichergestellt ist, dass dieses Subsystem nicht auf Benutzergeräten ausgeführt werden kann, wurde eine Reihe tiefgreifender Schutz- und Abwehrmaßnahmen implementiert:

- launchd lädt nicht die launchd-Eigenschaftsliste von cryptexd, wenn ein normales Kundengerät erkannt wird.
- cryptexd bricht ab, wenn ein normales Kundengerät erkannt wird.
- AppleImage4 gibt die [Nonce](#) nicht weiter, die für die Überprüfung eines Forschungs-Cryptex auf einem normalen Kundengerät verwendet wird.
- Der Signierungsserver verweigert das Personalisieren einer für cryptex konzipierten Image-Datei, wenn das betreffende Gerät nicht in einer expliziten Positivliste enthalten ist.

Damit die Privatsphäre des Sicherheitsforschers gewahrt bleibt, werden im Zuge der Personalisierung nur die Kennzahlen (z. B. Hashes) der ausführbaren Objekte oder des Kernel-Caches und die Kennungen (IDs) des für die Sicherheitsforschung genutzten Geräts an Apple gesendet. Der auf dem Gerät geladene cryptex-Inhalt wird nicht an Apple übermittelt.

Das für die Sicherheitsforschung konzipierte Gerät weist diverse Unterschiede gegenüber anderen Geräten auf. Sie sollen vermeiden, dass Außenstehende in böswilliger Absicht versuchen, ein Forschungsgerät als Benutzergerät auszugeben und eine Person dahingehend zu täuschen, dass es für den alltäglichen Gebrauch nutzbar ist. Diese Unterschiede sind:

- Das für die Sicherheitsforschung konzipierte Gerät kann nur gestartet werden, solange seine Batterie geladen wird. Dies kann über ein Lightning-Kabel oder ein Qi-Ladegerät erfolgen. Wenn beim Starten des Geräts die Batterie aktuell nicht geladen wird, wird das Gerät in den Wiederherstellungsmodus versetzt. Wenn der Benutzer das Gerät neu startet, solange die Batterie geladen wird, wird der normale Startvorgang ausgeführt. Sobald XNU gestartet wurde, muss das Gerät für den weiteren Betrieb nicht länger geladen werden.
- Während des iBoot-Startvorgangs wird „*Security Research Device*“ unter dem Apple-Logo eingeblendet.
- Der XNU-Kernel wird im Modus „Verbose“ gestartet.
- Das Gerät wird seitlich mit einer „ingeätzten“ Nachricht versehen: „Property of Apple. Confidential and Proprietary. Call +1 877 595 1125.“

Die folgenden zusätzlichen Maßnahmen werden in der nach dem Startvorgang erscheinenden Software implementiert:

- Während das Gerät konfiguriert wird, wird „*Security Research Device*“ angezeigt.
- „*Security Research Device*“ wird auf dem Sperrbildschirm und in der App „Einstellungen“ angezeigt.

Das für die Sicherheitsforschung konzipierte Gerät erlaubt einem Forscher Tätigkeiten, die bei einem Benutzergerät nicht möglich sind. Sie können:

- Sideload-Inhalt als ausführbare Objekte auf das Gerät laden und mit beliebigen Berechtigungen auf derselben Berechtigungsebene wie die Komponenten des Apple-Betriebssystems ausführen
- Dienste während des Startvorgangs starten
- Inhalte über Neustarts hinweg beibehalten
- die Berechtigung `research.com.apple.license-to-operate` verwenden, um einen Prozess zum Debuggen eines beliebigen anderen Prozesses auf dem System zu erlauben, einschließlich Systemprozesse.

Der Namespace `research.` wird nur von der RESEARCH-Variante der AppleMobileFileIntegrity-Kernel-Erweiterung berücksichtigt. Jeder Prozess mit dieser Berechtigung wird auf einem Kundengerät während der Signaturüberprüfung beendet.

- eigene Kernel-Caches personalisieren und wiederherstellen

Verschlüsselung und Datensicherheit

Verschlüsselung und Datensicherheit – Übersicht

Der sichere Startvorgang, die Systemsicherheit und die Sicherheitsfunktionen der Apps tragen alle dazu bei, dass auf einem Gerät nur vertrauenswürdige Codes und Apps ausgeführt werden können. Apple-Geräte verfügen über zusätzliche Verschlüsselungsfunktionen, die den Schutz der Benutzerdaten gewährleisten, selbst wenn andere Bereiche oder Teile der Sicherheitsinfrastruktur manipuliert werden (wenn beispielsweise ein Gerät verloren geht oder wenn nicht vertrauenswürdiger Code darauf ausgeführt wird). Alle diese Funktionen haben entscheidende Vorteile für Benutzer und IT-Administratoren, da die persönlichen Daten und die Firmendaten zu jeder Zeit geschützt sind und die Möglichkeit gegeben ist, Geräte bei Diebstahl oder Verlust per Fernzugriff vollständig zu löschen.

iOS- und iPadOS-Geräte nutzen eine Methodik für die Dateiverschlüsselung, die als „*Datensicherheit*“ bezeichnet wird. Intel-basierte Mac-Computer nutzen im Gegensatz dazu „*FileVault*“ als Methodik für die Dateiverschlüsselung und den Schutz von Daten. Ein Mac mit Apple Chips verwendet ein hybrides Modell, das Datensicherheit mit zwei Einschränkungen unterstützt: Die geringste Sicherheitsstufe (D) wird nicht unterstützt und die Standardstufe (Klasse C) verwendet einen Volumenschlüssel und verhält sich genau wie FileVault auf einem Intel-basierten Mac. In allen Fällen ist der Ausgangspunkt für die Hierarchien der Schlüsselverwaltung im dedizierten Chip der Secure Enclave angelegt. Außerdem nutzen beide Modelle eine dedizierte AES-Engine, um die Line-Speed-Verschlüsselung zu unterstützen und sicherzustellen, dass langlebige Verschlüsselungsschlüssel niemals an den Kernel des Betriebssystems oder die CPU übergeben werden müssen (wo sie potenziell manipuliert werden könnten). (Ein Intel-basierter Mac mit T1-Chip oder ohne Secure Enclave setzt keinen dedizierten Chip zum Schutz seiner FileVault-Verschlüsselungsschlüssel ein.)

Neben Datensicherheit und FileVault zum Schutz vor unautorisierten Datenzugriffen verwendet Apple *Betriebssystemkernel* für weitere Schutz- und Sicherheitsmaßnahmen. Der Kernel verwendet Zugriffssteuerungen für das Sandboxing von Apps (womit eingeschränkt wird, auf welche Daten eine App zugreifen kann) und einen Mechanismus, der als [Data Vault](#) bezeichnet wird (der nicht die möglichen Anfragen einschränkt, die eine App stellen kann, sondern stattdessen den Zugriff auf App-Daten durch andere Apps beschränkt).

Gerätecodes und Passwörter

Zum Schutz der Benutzerdaten vor bösartigen Angriffen verwendet Apple in iOS und iPadOS Codes und in macOS Passwörter. Je länger ein Code oder Passwort ist, desto stärker ist der Code oder das Passwort – und desto besser werden Brute-Force-Angriffen abgewehrt. Um Angriffe noch besser abzuwehren, erzwingt Apple eine Zeitverzögerung (für iOS und iPadOS) und eine maximale Anzahl an Passwortversuchen (auf dem Mac).

Durch das Einrichten eines Gerätecodes oder -passworts in iOS und iPadOS wird automatisch die [Datensicherheit](#) aktiviert. Die Datensicherheit wird auch auf anderen Geräten mit einem Apple-SoC (System on Chip) aktiviert, etwa auf einem Mac mit Apple Chips, auf Apple TV und auf der Apple Watch. In macOS verwendet Apple das integrierte Volume-Verschlüsselungsprogramm *FileVault*.

Starke Codes und Passwörter erhöhen die Sicherheit

iOS und iPadOS unterstützen Gerätecodes aus sechs oder vier Ziffern sowie alphanumerische Gerätecodes beliebiger Länge. Zusätzlich zum Entsperren des Geräts stellt der Gerätecode oder das Gerätepasswort die Entropie für bestimmte Verschlüsselungscodes zur Verfügung. Das bedeutet, dass ein Angreifer, der ein Gerät in seinem Besitz hat, ohne den Gerätecode nicht auf Daten bestimmter Sicherheitsklassen zugreifen kann.

Der Gerätecode oder das Gerätepasswort ist mit der UID des Geräts verknüpft, sodass Brute-Force-Angriffe direkt auf dem Gerät durchgeführt werden müssen, dem der Angriff gilt. Ein Zähler für die Anzahl der Wiederholungen sorgt dafür, dass mehr Zeit für jeden einzelnen Versuch benötigt wird. Dieser Zähler wurde so kalibriert, dass für einen Versuch etwa 80 Millisekunden benötigt werden. Das bedeutet, dass es über fünfzehn Jahre dauern würde, alle sechsstelligen alphanumerischen Gerätecodes mit Kleinbuchstaben und Zahlen auszuprobieren.

Je stärker der Gerätecode eines Benutzers ist, desto sicherer ist auch der Verschlüsselungscode. Und durch die Verwendung von Face ID und Touch ID kann ein viel stärkerer Code erstellt werden, der sonst nicht praktikabel wäre. Durch den stärkeren Gerätecode wird die Wirksamkeit der Entropie effektiv erhöht, mit der die für die Datensicherheit verwendeten Verschlüsselungsschlüssel geschützt werden, ohne dass die Benutzerfreundlichkeit leidet, wenn Geräte mehrmals am Tag entsperrt werden müssen.

Wenn ein langes Passwort eingegeben wird, das nur aus Ziffern besteht, wird auf dem Sperrbildschirm ein numerisches Tastenfeld anstelle der vollständigen Tastatur angezeigt. Ein längerer Gerätecode, der nur aus Ziffern besteht, kann einfacher einzugeben sein als ein kürzerer alphanumerischer Gerätecode und bietet ähnlich hohe Sicherheit.

Unter „Einstellungen“ > „Touch ID & Code“ oder „Face ID & Code“ kann über die Option „Eigener alphanumerischer Code“ ein längerer, alphanumerischer Code festgelegt werden.

Abwehren von Brute-Force-Attacken durch zunehmende Zeitverzögerungen (iOS, iPadOS)

Um in iOS und iPadOS Brute-Force-Codeangriffe noch besser abzuwehren, werden zunehmende zeitliche Verzögerungen eingesetzt, wenn ein ungültiger Gerätecode auf dem Sperrbildschirm eingegeben wurde. Details hierzu können der Tabelle unten entnommen werden.

Versuche	Erzwungene Verzögerung
1–4	Keine
5	1 Minute
6	5 Minuten
7–8	15 Minuten
9	1 Stunde

Wenn die Option „Daten löschen“ aktiviert ist (unter „Einstellungen“ > „Touch ID & Code“) und der Code zehn-mal hintereinander falsch eingegeben wurde, werden alle Inhalte und Einstellungen auf dem Speichergerät entfernt. Aufeinanderfolgende Versuche mit demselben falschen Code werden auf die Höchstgrenze nicht angerechnet. Diese Einstellung ist auch als Verwaltungsrichtlinie über eine Lösung für die Mobilgeräteverwaltung ([Mobile Device Management, MDM](#)), die diese Funktion unterstützt, sowie über Exchange ActiveSync verfügbar, wobei die maximal zulässige Anzahl von Fehleingaben auch verringert werden kann.

Bei Geräten mit einer Secure Enclave werden die Verzögerungen durch die Secure Enclave erzwungen. Wird das Gerät während einer zeitlich fixierten Verzögerung neu gestartet, wird die Verzögerung dennoch durchgesetzt und der Timer auf den Anfang des aktuellen Verzögerungsintervalls zurückgesetzt.

Abwehren von Brute-Force-Attacken durch zunehmende Zeitverzögerungen (macOS)

Um Brute-Force-Angriffen beim Starten des Mac vorzubeugen, sind im Anmeldefenster und im Target Disk Mode (TDM) maximal 10 Versuche der Passworteingabe zulässig, wobei nach einer bestimmten Anzahl an fehlgeschlagenen Versuchen die Wartezeit bis zum nächsten Versuch verlängert wird. Diese Verzögerungen werden durch die Secure Enclave erzwungen. Wird der Mac während eines Verzögerungsintervalls neu gestartet, wird die Verzögerung weiterhin durchgesetzt und der Timer auf den Anfang des aktuellen Verzögerungsintervalls zurückgesetzt.

Die Tabelle unten zeigt die Verzögerungen zwischen Passwordeingabeversuchen auf einem Mac mit Apple Chips und einem Mac mit T2-Chip.

Versuche	Erzwungene Verzögerung
5	1 Minute
6	5 Minuten
7	15 Minuten
8	15 Minuten
9	1 Stunde
10	Deaktiviert

Als Vorkehrung gegen den unwiederbringlichen Verlust von Daten infolge von Malware-Angriffen auf das Passwort des Benutzers werden diese Limits nicht durchgesetzt, nachdem sich der Benutzer erfolgreich an seinem Mac angemeldet hat. Sie werden aber nach dem erneuten Starten wieder in Kraft gesetzt. Wenn die 10 zulässigen Versuche aufgebraucht sind, sind nach einem Neustart in recoveryOS 10 weitere Versuche möglich. Wenn auch diese Versuche aufgebraucht sind, sind 10 weitere Versuche für jeden FileVault-Wiederherstellungsmechanismus (iCloud-Wiederherstellung, FileVault-Wiederherstellungsschlüssel und institutioneller Schlüssel) möglich. Maximal sind damit 30 weitere Versuche zulässig. Wenn auch diese weiteren Versuche aufgebraucht sind, verarbeitet die Secure Enclave keinerlei Prozesse zum Entschlüsseln des Volumes oder zum Prüfen des Passworts mehr. Die Daten auf dem betreffenden Laufwerk oder Volume sind dann nicht wiederherstellbar.

Um den Schutz in einer Unternehmensumgebung zu gewährleisten, sollten die IT-Verantwortlichen mithilfe einer Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM) FileVault-Konfigurationsrichtlinien definieren und deren Einhaltung erzwingen. Organisationen haben mehrere Möglichkeiten, um verschlüsselte Volumes zu verwalten – zum Beispiel institutionelle Wiederherstellungsschlüssel, persönliche Wiederherstellungsschlüssel (die optional treuhänderisch in der MDM-Lösung gespeichert werden können) oder eine Kombination der beiden Varianten. Auch die Schlüsselrotation kann in der MDM-Lösung als Richtlinie konfiguriert werden.

Bei einem Mac mit einem Apple T2 Security Chip übernimmt das Passwort eine ähnliche Funktion. Der Unterschied besteht darin, dass der generierte Schlüssel nicht für die Datensicherheit, sondern für die FileVault-Verschlüsselung herangezogen wird. macOS bietet außerdem zusätzliche Optionen für die Wiederherstellung eines Passworts:

- iCloud-Wiederherstellung
- FileVault-Wiederherstellung
- Institutioneller FileVault-Schlüssel

Datensicherheit

Datensicherheit – Übersicht

Apple nutzt eine als „Datensicherheit“ bezeichnete Technologie, um die Daten zu schützen, die im Flash-Speicher von Geräten mit einem Apple-SoC abgelegt sind, z. B. iPhone-, iPad-, Apple Watch- und Apple TV-Geräte sowie Mac-Computer mit Apple Chips. Die Datensicherheit ermöglicht es einem Gerät, auf übliche Ereignisse wie eingehende Telefonanrufe zu reagieren, und erlaubt zugleich einen hohen Verschlüsselungsstandard für die Benutzerdaten. Für bestimmte Apps des Systems (z. B. „Nachrichten“, „Mail“, „Kalender“, „Kontakte“ und „Fotos“) und für Daten der App „Health“ wird die Datensicherheit standardmäßig aktiviert. Apps anderer Anbieter werden automatisch auf diese Weise geschützt.

Implementierung

Datensicherheit wird durch die Erzeugung und Verwaltung einer Hierarchie von Schlüsseln implementiert. Sie baut auf den Technologien zur Hardwareverschlüsselung auf, die in jedes Apple-Gerät integriert sind. Die Datensicherheit wird mit einem pro Datei erzeugten Schlüssel gesteuert, wobei jede Datei einer Klasse zugeordnet wird. Der Zugriff wird dadurch bestimmt, ob die Klassenschlüssel entsperrt wurden oder nicht. Das APFS (Apple File System) ermöglicht dem Dateisystem, die Schlüssel anhand der jeweiligen Länge weiter zu unterteilen (wobei Teile einer Datei verschiedene Schlüssel haben können).

Jedes Mal, wenn eine Datei auf dem Datenvolume erstellt wird, erzeugt die Datensicherheit einen neuen 256-Bit-Schlüssel (den „*pro Datei*“ erzeugten Schlüssel) und übergibt diesen an die Hardware-AES-Engine. Diese verwendet den Schlüssel zum Verschlüsseln der Datei, wenn sie in den Flash-Speicher geschrieben wird. Bei Geräten mit A14, A15 und der M1-Familie erfolgt die Verschlüsselung unter Verwendung von AES-256 im XTS-Modus. Dabei durchläuft der pro Datei erzeugte 256-Bit-Schlüssel eine „Key Derivation Function“ (NIST Special Publication 800-108), sodass sich jeweils die 256-Bit langen Tweak- und Cipher-Schlüssel ergeben. Die Hardware-Generationen mit A9 bis A13 sowie S5, S6 und S7 verwenden AES-128 im XTS-Modus, bei dem der 256-Bit-Schlüssel gesplittet wird, sodass sich die jeweils 128 Bit langen Tweak- und Cipher-Schlüssel ergeben.

Bei einem Mac mit Apple Chips verwendet die Datensicherheit die Klasse C (siehe [Datensicherheitsklassen](#)) als Standard, verwendet aber einen Volumeschlüssel statt eines „pro Länge“ oder „pro Datei“ erzeugten Schlüssels. Dadurch wird für die Nutzerdaten das FileVault-Sicherheitsmodell nachgebildet. Benutzer müssen dennoch FileVault aktivieren, wenn sie den vollen Schutz erhalten möchten, den die Verknüpfung der Schlüsselhierarchie mit ihrem Passwort ermöglicht. Entwickler können außerdem eine höhere Sicherheitsklasse aktivieren, die einen „pro Datei“ oder „pro Länge“ erzeugten Schlüssel verwendet.

Datensicherheit bei Apple-Geräten

Auf Apple-Geräten mit Datensicherheit ist jede Datei mit einem individuellen „pro Datei“ oder „pro Länge“ erzeugten Schlüssel geschützt. Der Schlüssel, der mit dem NIST AED-Schlüsselalgorithmus verpackt ist, wird zusätzlich mit einem von mehreren Klassenschlüsseln verpackt, je nachdem, wie auf die Datei zugegriffen werden soll. Der verpackte pro Datei erzeugte Schlüssel wird wiederum in den Metadaten der Datei gespeichert.

Geräte mit APFS-Format unterstützen möglicherweise auch das Klonen von Dateien (Kopien ohne Aufwand durch Copy-on-Write-Technologie). Wenn eine Datei geklont wird, wird jeder Hälfte des Klons ein neuer Schlüssel zum Akzeptieren von eingehenden Schreibanforderungen zugeordnet, damit neue Daten mit einem neuen Schlüssel auf die Datenträger geschrieben werden. Mit der Zeit setzt sich die Datei aus verschiedenen Längen (oder Fragmenten) zusammen, die jeweils auf eigene Schlüssel verweisen. Alle Teile, aus denen sich eine Datei zusammensetzt, werden jedoch durch denselben Klassenschlüssel geschützt.

Wird eine Datei geöffnet, werden ihre Metadaten mit dem Dateisystemschlüssel entschlüsselt, wodurch der pro Datei erzeugte Schlüssel und ein Vermerk, mit welcher Klasse sie geschützt ist, entpackt werden. Der pro Datei (oder pro Länge) erzeugte Schlüssel wird mit dem Klassenschlüssel entpackt und an die Hardware-AES-Engine gesendet, die die Datei beim Lesen aus dem Flash-Speicher entschlüsselt. Die gesamte Verarbeitung des verpackten Dateischlüssels erfolgt innerhalb der Secure Enclave. Der Dateischlüssel wird zu keinem Zeitpunkt gegenüber dem Anwendungsprozessor offengelegt. Beim Starten handelt die Secure Enclave einen temporären Schlüssel mit der AES-Engine aus. Schlüssel einer Datei, die innerhalb der Secure Enclave entschlüsselt werden, werden mit diesem temporären Schlüssel neu verschlüsselt und erst in dieser Form an den Anwendungsprozessor zurückgesendet.

Die Metadaten für alle Dateien des Dateisystems auf dem Datenvolume werden mit einem Zufallsschlüssel verschlüsselt, der erzeugt wird, wenn das Betriebssystem erstmals installiert oder das Gerät von einem Benutzer vollständig gelöscht wird. Dieser Schlüssel wird mithilfe eines Key-Wrapping-Schlüssels verschlüsselt und verpackt, der nur der Secure Enclave für die Langzeitspeicherung bekannt ist. Der Key-Wrapping-Schlüssel wird jedes Mal geändert, wenn der Benutzer sein Gerät vollständig löscht. Bei A9-SoCs (oder neuer) nutzt die Secure Enclave Entropie- und Anti-Replay-Systeme, um die Möglichkeit der Auslöschbarkeit zu schaffen und den Schutz des Key-Wrapping-Schlüssels und anderer Daten zu gewährleisten. Weitere Informationen sind unter [Sicherer nicht flüchtiger Speicher](#) zu finden.

Wie der pro Datei (oder pro Länge) erzeugte Schlüssel wird der Metadatenschlüssel gegenüber dem Anwendungsprozessor nie direkt offengelegt. Die Secure Enclave stellt vielmehr bei jedem Startvorgang eine temporäre Version bereit. Beim Speichern wird der verschlüsselte Dateisystemschlüssel zusätzlich mit einem „auslöschbaren Schlüssel“ verpackt, der im Effaceable Storage gespeichert ist, oder alternativ mit einem Key-Wrapping-Schlüssel, der durch den Anti-Replay-Mechanismus der Secure Enclave geschützt ist. Dieser Schlüssel sorgt nicht für zusätzliche Vertraulichkeit der Daten. Er ist vielmehr so konzipiert, dass er auf Anforderung schnell gelöscht werden kann (durch den Benutzer mit der Option „Inhalte & Einstellungen löschen“ oder durch einen Benutzer oder Administrator, der in einer Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM), in Microsoft Exchange ActiveSync oder iCloud einen Befehl zum Fernlöschen absendet). Wird der Schlüssel auf diese Weise gelöscht, werden alle Dateien kryptografisch unzugänglich gemacht.

Der Inhalt einer Datei kann mit einem oder mehreren pro Datei (oder pro Länge) erzeugten Schlüsseln verschlüsselt werden, die mit einem Klassenschlüssel verpackt und in den Metadaten der Datei gespeichert werden, die wiederum mit dem Dateisystemschlüssel verschlüsselt sind. Der Klassenschlüssel wird mit der UID der Hardware und – bei manchen Klassen – mit dem Code des Benutzers geschützt. Diese Hierarchie bietet gleichzeitig Flexibilität und Effizienz. Wird beispielsweise die Klasse einer Datei geändert, muss nur der pro Datei erzeugte Schlüssel neu verpackt werden. Bei Änderung des Codes wird nur der Klassenschlüssel neu verpackt.

Datensicherheitsklassen

Einer Datei, die auf einem Gerät neu erstellt wird, das Datensicherheit unterstützt, wird von der App, die sie erzeugt, eine Klasse zugewiesen. Jede Klasse verwendet unterschiedliche Richtlinien, die bestimmen, wann auf die Daten zugegriffen werden kann. Die grundlegenden Klassen und Richtlinien werden im Folgenden beschrieben. Mac-Computer mit Apple Chips unterstützen die „Klasse D: Kein Schutz“ nicht und etablieren eine Sicherheitsgrenze für die An- und Abmeldung (nicht das Sperren und Entsperren wie auf iPhone-, iPad- und iPod touch-Geräten).

Klasse	Datensicherheitstyp
Klasse A: Vollständiger Schutz	NSFileProtectionComplete
Klasse B: Geschützt, außer wenn offen	NSFileProtectionCompleteUnlessOpen
Klasse C: Geschützt bis zur ersten Benutzerauthentifizierung	NSFileProtectionCompleteUntilFirstUserAuthentication
<i>Hinweis:</i> macOS verwendet einen Volumeschlüssel für die Neuerstellung von FileVault-Schutzcharakteristiken.	
Klasse D: Kein Schutz	NSFileProtectionNone
<i>Hinweis:</i> Nicht von macOS unterstützt.	

Vollständiger Schutz

NSFileProtectionComplete: Der Klassenschlüssel wird mit einem Schlüssel geschützt, der aus dem Code oder Passwort des Benutzers und der UID des Geräts abgeleitet wird. Kurz nachdem ein Benutzer ein Gerät sperrt (10 Sekunden, wenn die Einstellung „Passwort erforderlich“ auf „Sofort“ festgelegt ist), wird der entschlüsselte Klassenschlüssel verworfen, damit auf alle Daten in dieser Klasse erst wieder zugegriffen werden kann, wenn der Benutzer erneut den Code eingibt oder das Gerät per Face ID oder Touch ID entsperrt (bzw. sich anmeldet).

In macOS wird der entschlüsselte Klassenschlüssel, kurz nachdem der letzte Benutzer abgemeldet wird, verworfen, sodass auf alle Daten in dieser Klasse erst wieder zugegriffen werden kann, wenn ein Benutzer erneut den Code eingibt oder sich per Touch ID am Gerät anmeldet.

Geschützt, außer wenn offen

NSFileProtectionCompleteUnlessOpen: Es ist möglich, dass Dateien geschrieben werden müssen, solange das Gerät gesperrt oder der Benutzer abgemeldet ist. Ein gutes Beispiel sind E-Mail-Anhänge, die im Hintergrund geladen werden. Dieses Verhalten wird durch die Verschlüsselung mittels [asymmetrischer elliptischer Kurven](#) (ECDH über Curve25519) erreicht. Der übliche [pro Datei](#) erzeugte Schlüssel wird durch einen Schlüssel geschützt, der gemäß NIST SP 800-56A per One-Pass-Diffie-Hellman-Schlüsselvereinbarung hergeleitet wird.

Der temporäre öffentliche Schlüssel für die Vereinbarung wird zusammen mit dem verpackten, pro Datei erzeugten Schlüssel gespeichert. Hierfür wird die KDF (Concatenation Key Derivation Function, anerkannte Alternative 1) verwendet, wie unter 5.8.1 der NIST SP 800-56A beschrieben. Die Algorithmus-ID wird weggelassen. Als temporäre bzw. statisch öffentliche Schlüssel werden PartyUInfo und PartyVInfo verwendet. Als Hash-Funktion wird SHA-256 verwendet. Sobald die Datei geschlossen wird, wird der pro Datei erzeugte Schlüssel aus dem Speicher gelöscht. Damit die Datei erneut geöffnet werden kann, wird das Shared Secret mit dem privaten Schlüssel der Klasse „Geschützt, außer wenn offen“ und dem temporären öffentlichen Schlüssel der Datei neu erstellt. Diese werden verwendet, um den pro Datei erzeugten Schlüssel zu entpacken, mit dem wiederum die Datei entschlüsselt wird.

In macOS ist der private Teil des Klassenschlüssels „NSFileProtectionCompleteUnlessOpen“ zugänglich, solange ein beliebiger Benutzer am System angemeldet oder authentifiziert ist.

Geschützt bis zur ersten Benutzerauthentifizierung

NSFileProtectionCompleteUntilFirstUserAuthentication: Diese Klasse verhält sich wie die Klasse „Vollständiger Schutz“, allerdings mit dem Unterschied, dass der entschlüsselte Klassenschlüssel nicht aus dem Speicher entfernt wird, wenn das Gerät gesperrt oder der Benutzer abgemeldet ist. Der Schutz in dieser Klasse ist mit der vollständigen Festplattenverschlüsselung auf Desktopcomputern vergleichbar. Daten werden so vor Angriffen geschützt, die einen Neustart beinhalten. Dies ist die Standardklasse für Daten aus Apps von Drittanbietern, denen keine andere [Datensicherheitsklasse](#) zugewiesen wird.

In macOS verwendet diese Klasse einen Volumeschlüssel, der sich wie FileVault verhält und verfügbar ist, solange das Volume aktiviert ist.

Kein Schutz

NSFileProtectionNone: Dieser Klassenschlüssel wird nur mit der UID geschützt und im [Efaceable Storage](#) gespeichert. Da alle Schlüssel zum Entschlüsseln von Dateien dieser Klasse auf dem Gerät gespeichert werden, bietet diese Verschlüsselung nur den Vorteil einer schnellen Fernlöschung. Wenn einer Datei keine Datensicherheitsklasse zugewiesen wird, wird sie dennoch in verschlüsselter Form gespeichert (wie dies mit allen Daten auf einem iOS- oder iPadOS-Gerät geschieht).

Dies wird in macOS nicht unterstützt.

Hinweis: Bei macOS-Volumes, die nicht mit einem gestarteten Betriebssystem korrespondieren, sind alle Datensicherheitsklassen zugänglich, solange das Volume aktiviert ist. Die standardmäßige Datensicherheitsklasse ist *NSFileProtectionCompleteUntilFirstUserAuthentication*. Die Funktionalität für pro Länge erzeugte Schlüssel ist sowohl für Rosetta 2-Apps als auch für native Apps verfügbar.

Keybags für Datensicherheit

Bei iOS und iPadOS, watchOS und tvOS werden sowohl für Dateien als auch für Schlüsselbunde die Schlüssel für die [Datensicherheitsklassen](#) in Keybags gesammelt und verwaltet. Diese Betriebssysteme verwenden die folgenden Keybags: User, Device, Backup, Escrow und iCloud Backup.

Keybag „User“

Im Keybag „User“ werden die verpackten Klassenschlüssel gespeichert, die im normalen Betrieb des Geräts verwendet werden. Wenn beispielsweise ein Gerätecode eingegeben wird, wird *NSFileProtectionComplete* aus dem Keybag „User“ geladen und entpackt. Hierbei handelt es sich um eine binäre Liste von Eigenschaften (.plist), die in der Klasse „Kein Schutz“ gespeichert wird.

Bei Geräten mit SoCs vor A9 wird der Inhalt der .plist-Datei mit einem Schlüssel verschlüsselt, der im Efaceable Storage aufbewahrt wird. Zur Aufrechterhaltung der Sicherheit wird dieser Schlüssel jedes Mal, wenn der Besitzer seinen Code ändert, gelöscht und neu generiert.

Bei Geräten mit einem A9-SoC oder einem neueren SoC enthält die .plist-Datei einen Schlüssel, der angibt, dass sich der Keybag in einem Schließfach befindet, der durch die Secure Enclave geschützt ist und der Kontrolle durch eine Anti-Replay-Nonce unterliegt.

Die Secure Enclave verwaltet den Keybag „User“. Über sie kann der Sperrstatus des Geräts abgerufen werden. Sie gibt die Rückmeldung, dass das Gerät entsperrt ist, nur unter der Voraussetzung, dass auf alle Klassenschlüssel im Keybag „User“ zugegriffen werden kann und dass sie erfolgreich entpackt wurden.

Keybag „Device“

Im Keybag „Device“ werden die verpackten Klassenschlüssel gespeichert, die für Vorgänge an und mit gerätespezifischen Daten verwendet werden. iPadOS-Geräte, die für die gemeinsame Nutzung durch mehrere Personen konfiguriert sind, benötigen in bestimmten Situationen Zugriff auf Anmeldeinformationen, noch bevor sich ein Benutzer angemeldet hat. Dies macht einen Keybag erforderlich, der nicht durch den Gerätecode eines Benutzers geschützt ist.

iOS und iPadOS bieten keine Unterstützung für die kryptografische Trennung von benutzerbasierten Dateisysteminhalten. Das bedeutet, dass das System zum Verpacken von [pro Datei](#) erzeugten Schlüsseln Klassenschlüssel aus dem Keybag „Device“ verwendet. Der Schlüsselbund verwendet im Gegensatz dazu Klassenschlüssel aus dem Keybag „User“, um Objekte im benutzereigenen Schlüsselbund zu schützen. Bei iOS- und iPadOS-Geräten, die für die Nutzung durch einen einzigen Benutzer konfiguriert sind (Standardkonfiguration), sind der Keybag „Device“ und der Keybag „User“ identisch und durch den Gerätecode des Benutzers geschützt.

Keybag „Backup“

Der Keybag „Backup“ wird erstellt, wenn im Finder (bei macOS 10.15 oder neuer) oder in iTunes (bei macOS 10.14 oder neuer) ein verschlüsseltes Backup erstellt und auf dem Computer gespeichert wird, auf dem das Gerät gesichert wird. Es wird ein neuer Keybag mit einem neuen Satz Schlüssel erstellt und die gesicherten Daten werden mit diesen neuen Schlüsseln erneut verschlüsselt. Nicht-migrierende Schlüsselbundobjekte bleiben (wie oben beschrieben) mit dem von der UID abgeleiteten Schlüssel verpackt, sodass sie auf dem Gerät, von dem sie ursprünglich gesichert wurden, wiederhergestellt werden können, von anderen Geräten aus aber nicht auf sie zugegriffen werden kann.

Der Keybag wird durch das festgelegte Passwort geschützt und durchläuft 10 Millionen Iterationen der Schlüsselherleitungsfunktion PBKDF2. Trotz dieses Zählers wird keine Verknüpfung mit einem bestimmten Gerät hergestellt, sodass theoretisch ein Brute-Force-Angriff von mehreren Computern gleichzeitig aus auf den Keybag „Backup“ ausgeführt werden könnte. Dieser Bedrohung kann mit einem hinreichend sicheren Passwort entgegengewirkt werden.

Wenn ein Benutzer ein Backup nicht verschlüsseln lässt, werden die Backup-Dateien unabhängig von ihrer Datensicherheitsklasse nicht verschlüsselt, der Schlüsselbund wird aber weiterhin mit einem von der UID abgeleiteten Schlüssel geschützt. Aus diesem Grund können Schlüsselbundobjekte nur auf ein neues Gerät migriert werden, wenn ein Backup-Passwort festgelegt wurde.

Keybag „Escrow“

Der Keybag „Escrow“ wird für die Synchronisierung mit dem Finder (macOS 10.15 oder neuer) oder iTunes (in macOS 10.14 oder älter) via USB oder die Mobilgeräteverwaltung (Mobile Device Management, MDM) verwendet. Mit diesem Keybag kann der Finder oder iTunes Daten sichern und synchronisieren, ohne dass der Benutzer einen Code eingeben muss. Außerdem ermöglicht der Keybag einer MDM-Lösung, den Code eines Benutzers per Fernzugriff zu löschen. Der Keybag wird auf dem Computer, der zum Synchronisieren vom Finder oder von iTunes verwendet wird, oder in der MDM-Lösung gespeichert, die das Gerät per Fernzugriff verwaltet.

Der Keybag „Escrow“ verbessert die Benutzerfreundlichkeit beim Synchronisieren von Geräten – ein Vorgang, bei dem potenziell auf Daten aller Klassen zugegriffen werden muss. Wenn ein mit einem Code gesperrtes Gerät erstmals mit dem Finder oder mit iTunes verbunden wird, muss der Benutzer einen Gerätecode eingeben. Das Gerät erstellt daraufhin einen Keybag „Escrow“, der mit einem neu erzeugten Schlüssel geschützt wird und dieselben Klassenschlüssel enthält, die auf dem Gerät verwendet werden. Der Keybag „Escrow“ und der Schlüssel, mit dem er geschützt wird, werden zwischen dem Gerät und dem Host/Server aufgeteilt, wobei den auf dem Gerät gespeicherten Daten die Klasse „Geschützt bis zur ersten Benutzerauthentifizierung“ zugewiesen wird. Aus diesem Grund muss der Code für das Gerät das erste Mal nach einem Neustart eingegeben werden, bevor der Benutzer ein Backup mit dem Finder oder mit iTunes erstellen kann.

Bei OTA-Softwareaktualisierungen (Over-The-Air) wird der Benutzer vor dem Starten der Aktualisierung nach seinem Code gefragt. Auf dieser Basis wird in einer sicheren Umgebung ein Einmal-Token erstellt, der nach dem Aktualisieren zum Entsperren des Keybag „User“ verwendet wird. Dieser Token kann nicht generiert werden, ohne dass der Code des Benutzers eingegeben wird. Wird der Code des Benutzers geändert, wird ein zuvor erstellter Token hinfällig.

Einmal nutzbare Token für das Entsperren können für die beaufsichtigte und auch die unbeaufsichtigte Installation einer Softwareaktualisierung verwendet werden. Diese Token werden mithilfe eines Codes verschlüsselt, der aus dem aktuellen Wert eines monotonen Zählers in der Secure Enclave, der UUID des Keybag und der UID der Secure Enclave gebildet wird.

Bei A9 SoCs (oder neuer) sind einmal nutzbare Token nicht mehr auf Zähler oder auslöschbaren Speicher angewiesen. Stattdessen werden sie von der Anti-Replay-Nonce geschützt, die von der Secure Enclave gesteuert wird.

Ein zum Entsperren genutzter Einmal-Token, der für eine beaufsichtigte Softwareaktualisierung verwendet wird, verliert nach 20 Minuten seine Gültigkeit. Bei iOS 13 und iPadOS 13.1 (oder neuer) wird der Token in einem Schließfach gespeichert, das durch die Secure Enclave geschützt ist. Bis iOS 13 wurde ein solcher Token aus der Secure Enclave exportiert und in den Effaceable Storage geschrieben oder durch den Anti-Replay-Mechanismus der Secure Enclave geschützt. Ein richtlinienspezifischer Timer veranlasst, dass der Wert des Zählers erhöht wird, wenn das Gerät nicht innerhalb von 20 Minuten neu gebootet wird.

Unbeaufsichtigte (automatische) Softwareaktualisierungen werden ausgeführt, wenn das System erkennt, dass eine Aktualisierung verfügbar ist, und eine der folgenden Situationen vorliegt:

- Bei iOS 12 (oder neuer) ist die Funktion für „automatische Aktualisierungen“ konfiguriert.
- Der Benutzer wählt „Später installieren“, wenn er über die verfügbare Aktualisierung informiert wird.

Wenn der Benutzer seinen Code eingibt, wird ein einmal nutzbarer Token zum Entsperren generiert, der bis zu 8 Stunden in der Secure Enclave verbleibt und gültig ist. Wird die Aktualisierung nicht vorgenommen, wird der zum Entsperren genutzte Einmal-Token bei jedem Sperrvorgang vernichtet und bei jedem nachfolgenden Entsperrvorgang neu erstellt. Durch jeden Entsperrvorgang wird das Zeitfenster von 8 Stunden neu gestartet. Nach 8 Stunden entwertet ein richtlinienspezifischer Timer den zum Entsperren genutzten Einmal-Token.

Keybag „iCloud Backup“

Der Keybag „iCloud Backup“ ist dem Keybag „Backup“ ähnlich. Alle Klassenschlüssel in diesem Keybag sind asymmetrisch (es wird Curve25519 verwendet wie bei der Datensicherheitsklasse „Geschützt außer wenn offen“). Auch ein asymmetrischer Keybag wird für das Backup in der Schlüsselbundwiederherstellung des iCloud-Schlüsselbunds verwendet.

Schützen von Schlüsseln in alternativen Startmodi

Datensicherheit ist darauf ausgelegt, ausschließlich autorisierten Benutzern Zugriff auf Benutzerdaten zu geben – und zwar ausschließlich nach der erfolgreichen Authentifizierung. Datensicherheitsklassen sind konzipiert, eine Vielfalt von Anwendungsfällen zu unterstützen, z. B. die Möglichkeit, manche Daten auch dann zu lesen und zu schreiben, wenn ein Gerät gesperrt ist (jedoch nach der ersten Entsperrung). Darüber hinaus werden weitere Schritte unternommen, um Benutzerdaten bei alternativen Startmodi zu schützen – z. B. Startmodi, die für den DFU-Modus (Device Firmware Update), den Wiederherstellungsmodus, Apple-Diagnose oder sogar die Softwareaktualisierung verwendet werden. Diese Funktionen basieren auf einer Kombination aus Hardware- und Softwarefunktionen und wurden im Rahmen der Weiterentwicklung von Apple Chips fortlaufend erweitert.

Funktion	A10	A11, S3	A12, S4	A13, S5	A14, A15, S6, S7, M1-Familie
Wiederherstellung: Datenschutz bei allen Datensicherheitsklassen	✓	✓	✓	✓	✓
Alternative Startmodi des DFU-Modus, Wiederherstellung und Softwareaktualisierungen: Datenschutz bei Klassen A, B und C		✓	✓	✓	✓

Die AES-Engine der Secure Enclave ist mit sperrbaren [Software-Seed-Bits](#) ausgestattet. Werden Schlüssel von der UID erstellt, werden diese Seed-Bits in die KDF (Key Derivation Function) integriert, um zusätzliche Schlüsselhierarchien zu erzeugen. Wie genau das Seed-Bit verwendet wird, hängt vom jeweiligen System on Chip ab:

- Beginnend mit Apple A10- und S3-SoCs wird ein Seed-Bit dafür reserviert, vom Code des Benutzers geschützte Schlüssel zu unterscheiden. Das Seed-Bit wird für Schlüssel gesetzt, die den Code des Benutzers erfordern (einschließlich Datensicherheitsschlüssel der Klassen A, B und C), und für Schlüssel gelöscht, die keinen Benutzercode erfordern (einschließlich der Metaschlüssel des Dateisystems und der Schlüssel der Klasse D).

- Darüber hinaus werden auf Geräten mit iOS 13 (oder neuer) und iPadOS 13.1 (oder neuer) und mit A10 (oder neuer) alle Benutzerdaten mit kryptografischen Mitteln unzugänglich gemacht, wenn die Geräte im Diagnosemodus gebootet oder gestartet werden. Erreicht wird dies durch die Einführung eines zusätzlichen Seed-Bits, dessen Einstellung über die Fähigkeit bestimmt, auf den Medienschlüssel zuzugreifen, der wiederum benötigt wird, um auf die Metadaten (und somit auf den Inhalt aller Dateien) auf dem per Datensicherheit geschützten Datenvolume zuzugreifen. Dieser Schutz erstreckt sich auf Dateien in allen Klassen (A, B, C und D), nicht nur auf Dateien, für die der Code des Benutzers erforderlich ist.
- Bei A12-SoCs sperrt der Boot-ROM der Secure Enclave das Code-Seed-Bit, wenn der Anwendungsprozessor in den Device Firmware Upgrade (DFU)-Modus oder den Wiederherstellungsmodus wechselt. Wenn das Code-Seed-Bit gesperrt ist, wird kein Vorgang zugelassen, durch den es geändert wurde. Auf diese Weise wird der Zugriff auf Daten verhindert, die durch den Code des Benutzers geschützt werden.

Die Wiederherstellung eines Geräts, das in den DFU-Modus wechselt, führt zurück zu einem bekannten sicheren Zustand und gibt die Gewissheit, dass ausschließlich unveränderter und von Apple signierter Code vorhanden ist. Der DFU-Modus kann manuell aufgerufen werden.

Weitere Informationen dazu, wie ein Gerät in den DFU-Modus versetzt werden kann, enthalten die folgenden Apple-Support-Artikel:

Gerät	Artikel
iPhone, iPad, iPod touch	Wenn du deinen iPhone-Code vergessen hast
Apple TV	Falls auf Apple TV ein Warnsymbol angezeigt wird
Mac mit Apple Chips	Reparieren oder Wiederherstellen eines Mac mit Apple Chips in Apple Configurator 2

Schützen von Benutzerdaten bei Angriffen

Angreifer, die Benutzerdaten extrahieren wollen, versuchen dies oft unter Einsatz verschiedener Techniken: das Extrahieren der verschlüsselten Daten auf ein anderes Medium für Brute-Force-Angriffe, das Manipulieren der Betriebssystemversion oder eine anderweitige Veränderung oder Schwächung der Sicherheitsrichtlinie des Geräts, um einen Angriff zu ermöglichen. Der Angriff auf die Daten eines Geräts erfordert in vielen Fällen die Kommunikation mit dem Gerät über physische Schnittstellen wie Lightning oder USB. Apple-Geräte verfügen über Funktionen, die derartige Angriffe verhindern sollen.

Apple-Geräte unterstützen eine *Sealed Key Protection (SKP)* genannte Technologie, die sicherstellt, dass kryptografisches Material außerhalb des Geräts nicht verfügbar ist, und dafür sorgt, dass keine Manipulationen an den Betriebssystemversionen oder den Sicherheitseinstellungen ohne angemessene Benutzerautorisierung stattfinden. Diese Funktion wird *nicht* von der Secure Enclave bereitgestellt. Stattdessen wird sie von Hardwareregistern einer tieferen Ebene unterstützt, um eine zusätzliche Schutzebene für die Schlüssel zu bieten, die zum Entschlüsseln der Benutzerdaten notwendig und nicht von der Secure Enclave abhängig sind.

Hinweis: SKP ist nur auf Geräten mit einem Apple-SoC verfügbar.

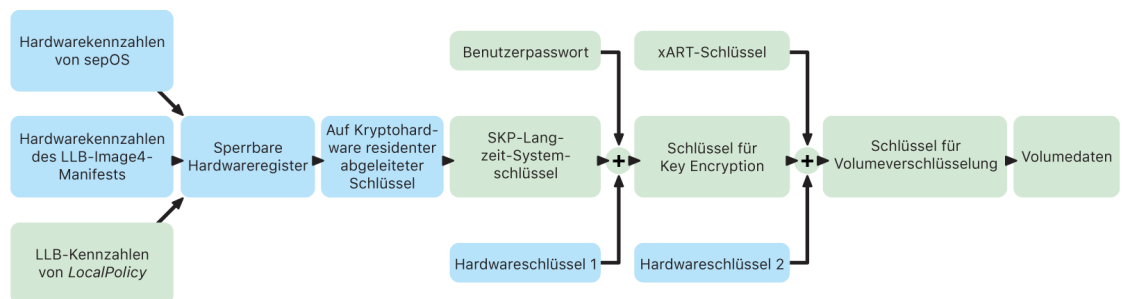
Funktion	A10	A11, S3	A12, S4	A13, S5	A14, A15, S6, S7, M1-Familie
Sealed Key Protection	✓	✓	✓	✓	✓

iPhone- und iPad-Geräte können auch so konfiguriert werden, dass sie Datenverbindungen nur unter Bedingungen aktivieren, die eher darauf schließen lassen, dass sich das Gerät in der physischen Kontrolle des autorisierten Eigentümers befindet.

Sealed Key Protection (SKP)

Auf Apple-Geräten, die Datensicherheit unterstützen, wird der Key Encryption Key (KEK) mit Kennzahlen der auf dem System installierten Software sowie der Verknüpfung mit der UID geschützt (oder „versiegelt“), die nur von der Secure Enclave bereitgestellt werden kann. Auf einem Mac mit Apple Chips wird der Schutz des KEK weiter verstärkt, indem Informationen über die Sicherheitsrichtlinie des Systems einbezogen werden, da macOS kritische Änderungen von Sicherheitsrichtlinien (z. B. das Deaktivieren des sicheren Startvorgangs oder von SIP) unterstützt, die von anderen Plattformen nicht unterstützt werden. Auf einem Mac mit Apple Chips umfasst dieser Schutz auch [FileVault](#)-Schlüssel, da FileVault unter Verwendung von Datensicherheit (Klasse C) implementiert wird.

Der Schlüssel, der auf der Verknüpfung von Benutzerpasswort, SKP-Langzeitschlüssel und Hardware Schlüssel 1 (die UID der Secure Enclave) basiert, wird als *passwortbasierter Schlüssel* bezeichnet. Dieser Schlüssel wird verwendet, um den Keybag „User“ (auf allen unterstützten Plattformen) und den KEK (nur in macOS) zu schützen und das biometrische Entsperren oder automatische Entsperren mit anderen Geräten wie z. B. der Apple Watch zu aktivieren.



Der Boot-Monitor der Secure Enclave erfasst die Kennzahlen des Betriebssystems der Secure Enclave, das geladen wird. Wenn der Boot-ROM des Anwendungsprozessors das Image4-Manifest erfasst, das an LLB gekoppelt ist, enthält dieses Manifest auch die Kennzahlen aller anderen mit dem System gekoppelten Firmwarekomponenten, die geladen werden. Die LocalPolicy-Datei – die lokale Richtlinie – umfasst die zentralen Sicherheitskonfigurationen des macOS-Betriebssystems, die geladen werden. Diese lokale Richtlinie enthält außerdem das „nsih“-Feld, eine Hash-Darstellung des Image4-Manifests von macOS. Das Image4-Manifest von macOS umfasst die Kennzahlen aller mit macOS gekoppelten Firmwarekomponenten und der zentralen Startobjekte von macOS, zum Beispiel die Boot Kernel-Sammlung und den Root-Hash-Wert des Signed System Volume (SSV).

Wenn es einem Angreifer wider aller Erwartungen gelingt, eine der o. g. gemessenen Firmware-, Software- oder Sicherheitskonfigurationskomponenten zu ändern, führt dies zu einer Änderung der Kennzahlen, die in den Hardwareregistern gespeichert sind. Die Modifikation der Kennzahlen bewirkt, dass der per Kryptohardware hergeleitete *SMRK-Schlüssel (System Measurement Root Key)* auf einen anderen Wert abgeleitet wird und so das Seal (Siegel) in der Schlüsselhierarchie gebrochen wird. Dadurch wird der *SMDK-Schlüssel (System Measurement Device Key)* unzugänglich, was wiederum dazu führt, dass der KEK und somit die Daten für den Angreifer unzugänglich werden.

Allerdings muss das System, wenn es nicht angegriffen wird, die Möglichkeit zu legitimen Softwareaktualisierungen bieten, die die Kennzahlen der Firmware und das `nsih`-Feld in der lokalen Richtlinie so ändern, dass sie auf neue macOS-Kennzahlen verweisen. Bei anderen Systemen, die versuchen, Kennzahlen ihrer Firmware einzubinden, die aber keine als verlässlich oder gut klassifizierte Quelle der Wahrheit haben, muss der Benutzer die Sicherheitsfunktionen deaktivieren, die Firmware aktualisieren und eine erneute Aktivierung vornehmen, damit die neue Ausgangsbasis der Kennzahlen erfasst werden kann. Dadurch steigt das Risiko, dass ein Angreifer die Firmware während einer Softwareaktualisierung kompromittieren kann, in signifikanter Weise. Das System profitiert von der Tatsache, dass die Image4-Manifeste alle benötigten Kennzahlen umfassen. Die Hardware, die bei einem normalen Startvorgang bei übereinstimmenden Kennzahlen den SMDK mithilfe des SMRK entschlüsselt, kann auch verwendet werden, um den SMDK im Hinblick auf einen vorgeschlagenen künftigen SMRK zu verschlüsseln. Indem die Kennzahlen bereitgestellt werden, die nach einer Softwareaktualisierung erwartet werden, kann die Hardware einen SMDK, der in einem aktuellen Betriebssystem zugänglich ist, so verschlüsseln, dass er auch in einem künftigen Betriebssystem zugänglich bleibt. Analog gilt dies auch für den Fall, dass ein Kunde legitim Änderungen an den eigenen Sicherheitseinstellungen in der lokalen Richtlinie vornimmt. In diesem Fall muss der SMDK im Hinblick auf den zukünftigen SMRK verschlüsselt werden; Ausgangspunkt hierfür sind die Kennzahlen in der LocalPolicy-Datei, die LLB beim nächsten Neustart errechnen wird.

Sicheres Aktivieren von Datenverbindungen in iOS und iPadOS

Wenn auf iOS- oder iPadOS-Geräten in letzter Zeit keine Datenverbindungen hergestellt wurden, müssen Benutzer Datenverbindungen über Lightning-, USB- oder Smart Connector-Schnittstellen mit Face ID, Touch ID oder einem Code aktivieren. Hierdurch wird die Angriffsfläche für physisch angeschlossene Geräte wie schädliche Ladegeräte reduziert und zugleich die Nutzung von Zubehör innerhalb angemessener Zeitbeschränkungen ermöglicht. Wenn seit dem Sperren des iOS- oder iPadOS-Geräts oder seit dem Beenden der Datenverbindung eines Zubehörs mehr als eine Stunde verstrichen ist, muss das Gerät erst wieder entsperrt werden, um neue Datenverbindungen zuzulassen. Während dieser Stunde werden nur Datenverbindungen von Zubehör zugelassen, das zuvor mit dem entsperrten Gerät verbunden war. Diese Zubehörgeräte werden 30 Tage lang erkannt, nachdem sie zum ersten Mal verbunden wurden. Versuche von einem unbekanntem Zubehör, eine Datenverbindung in diesem Zeitraum herzustellen, führen so lange zum Deaktivieren aller Datenverbindungen über Lightning, USB und Smart Connector, bis das Gerät erneut entsperrt wird. Dieser Zeitraum von einer Stunde:

- stellt sicher, dass Personen, die häufig Verbindungen zu einem Mac, PC oder Zubehör oder Kabelverbindungen zu CarPlay nutzen, nicht jedes Mal ihren Code eingeben müssen, wenn sie ihr Gerät anschließen,
- ist erforderlich, da das Ökosystem der Zubehörgeräte keine kryptografisch zuverlässige Möglichkeit bietet, Zubehör zu identifizieren, bevor eine Datenverbindung hergestellt wird.

Sind mehr als drei Tage vergangen, seit eine Datenverbindung mit einem Zubehör hergestellt wurde, verweigert das Gerät neue Datenverbindungen, unmittelbar nachdem es gesperrt wurde. Für Benutzer, die solches Zubehör eher selten verwenden, soll auf diese Weise der Schutz erhöht werden. Datenverbindungen über Lightning, USB und Smart Connector werden auch deaktiviert, wenn sich das Gerät in einem Status befindet, in dem es einen Code zum erneuten Aktivieren der biometrischen Authentifizierung erfordert.

Der Benutzer kann in seinen Einstellungen die Option „Immer EIN“ für Datenverbindungen reaktivieren. (Bei bestimmten Hilfsgeräten geschieht dies automatisch.)

Rolle von APFS (Apple File System)

Apple File System (APFS) ist ein proprietäres Dateisystem, das gezielt im Hinblick auf Verschlüsselung konzipiert und gestaltet wurde. APFS kommt auf allen Apple-Plattformen zum Einsatz – auf iPhone-, iPad-, iPod touch-, Mac-, Apple TV- und Apple Watch-Geräten. Das für Flash/SSD-Speicher optimierte Dateisystem ermöglicht eine starke Verschlüsselung, Copy-on-Write-Metadaten, die gemeinsame Nutzung von Plattenspeicherbereichen, das Klonen von Dateien und Verzeichnissen, Snapshots (Schnappschüsse) für die Datensicherung, die schnelle Größenbestimmung für Verzeichnisse, Primitive für das atomare Speichern und allgemeine Verbesserungen des Dateisystems sowie ein spezifisches Copy-on-Write-Design, das per E/A-Koaleszenz für ein hohes Maß an Leistung sorgt und zugleich die Zuverlässigkeit der Daten sicherstellt.

Gemeinsame Nutzung von Plattenspeicherbereichen

APFS stellt Speicherplatz nach Bedarf bereit. Wenn ein APFS-Einzelcontainer mehrere Volumes umfasst, wird der freie Speicherbereich des Containers gemeinsam genutzt. Das bedeutet, dass er je nach Bedarf jedem der einzelnen Volumes zugewiesen werden kann. Jedes der Volumes nutzt nur einen Teil des übergeordneten Containers. Der verfügbare Speicherbereich entspricht daher der Gesamtgröße des Containers abzüglich der Gesamtgröße der Bereiche, die auf allen im Container enthaltenen Volumes belegt ist.

Mehrere Volumes

Bei macOS 10.15 (oder neuer) muss ein APFS-Container mindestens fünf Volumes umfassen, wobei die drei ersten Volumes für den Benutzer nicht zu sehen sind:

- *Preboot-Volume*: Auf diesem unverschlüsselten Volume befinden sich die Daten, die zum Booten jedes Systemvolumes im Container benötigt werden.
- *VM-Volume*: Dieses unverschlüsselte Volume wird von macOS zum Speichern von verschlüsselten Swap-Dateien verwendet.
- *Wiederherstellungsvolume*: Dieses unverschlüsselte Volume muss ohne Entsperren eines Systemvolumes verfügbar sein, um recoveryOS starten zu können.

- **Systemvolume:** Hierauf befinden sich die folgenden Daten und Informationen:
 - Alle Dateien, die zum Starten des Mac benötigt werden.
 - Alle von macOS nativ installierten Apps (sie befanden sich im Ordner /Applications und sind jetzt in /System/Applications).

Hinweis: Standardmäßig kann kein Prozess auf das Systemvolume schreiben, nicht einmal Apple-Systemprozesse.
- **Datenvolume:** Hierauf befinden sich Daten, die Änderungen unterworfen sind, zum Beispiel:
 - Alle Daten im Benutzerordner, inkl. Fotos, Musik, Videos und Dokumente.
 - Vom Benutzer installierte Apps, zu denen auch AppleScript und Automator-Programme zählen.
 - Eigene Frameworks und Daemons, die vom Benutzer, der Organisation oder durch Apps anderer Anbieter installiert werden.
 - Sonstige Speicherorte, die als „Eigentum“ des Benutzers angesehen werden und von ihm beschrieben werden können (z. B. /Applications, /Library, /Users, /Volumes, /usr/local, /etc und /tmp).

Für jedes weitere Systemvolume wird ein Datenvolume erstellt. Die Volumes „Preboot“, „VM“ und „Recovery“ werden durchweg geteilt und nicht dupliziert.

Bei macOS 11 (oder neuer) wird das Systemvolume in einem Schnappschuss erfasst. Das Betriebssystem wird aus einem Schnappschuss des Systemvolumes und nicht aus einer nur im Lesezugriff verfügbaren Aktivierung des veränderbaren Systemvolumes gebootet.

Unter iOS und iPadOS ist der Speicher in mindestens zwei APFS-Volumes unterteilt:

- Systemvolume
- Datenvolume

Sicherheit von Schlüsselbunddaten

Viele Apps müssen Passwörter und andere kurze, aber vertrauliche Datenfragmente (z. B. Schlüssel und Anmelde-Token) verarbeiten. Der **Schlüsselbund** stellt eine sichere Methode zum Speichern dieser Elemente zur Verfügung. Die verschiedenen Apple-Betriebssysteme verwenden unterschiedliche Mechanismen, um die Sicherheitsgarantien zu erzwingen, die den unterschiedlichen Sicherheitsklassen des Schlüsselbunds zugeordnet sind. In macOS (einschließlich Mac-Computer mit Apple Chips) wird Datensicherheit nicht direkt verwendet, um diese Sicherheitsgarantien zu erzwingen.

Übersicht

Schlüsselbundobjekte werden mithilfe zweier AES-256-GCM-Schlüssel verschlüsselt: eines Tabellenschlüssels (Metadaten) und eines pro Zeile erzeugten Schlüssels (SecretKey). Die Metadaten des Schlüsselbunds (alle Attribute außer kSecValue) werden zum schnellen Suchen mit dem Metadatenschlüssel verschlüsselt, um Suchen zu beschleunigen, während der geheime Wert (kSecValueData) mit dem geheimen Schlüssel (secretkey) verschlüsselt wird. Der Metadatenschlüssel wird von der Secure Enclave geschützt, aber im Anwendungsprozessor zwischengespeichert, um schnelle Anfragen an den Schlüsselbund zuzulassen. Der Geheimschlüssel erfordert immer einen Durchlauf durch die Secure Enclave.

Der Schlüsselbund ist als SQLite-Datenbank implementiert, die im Dateisystem gespeichert wird. Es gibt nur eine Datenbank, und der `securityd`-Daemon legt fest, auf welche Schlüsselbundelemente ein Prozess oder eine App zugreifen kann. Zugriffe auf die APIs des Schlüsselbunds resultieren in Anfragen an den Daemon, der wiederum die Berechtigungen „`keychain-access-groups`“, „`application-identifier`“ und „`application-group`“ abfragt. Anstatt den Zugriff auf einen einzelnen Prozess einzuschränken, ermöglichen es Zugriffsgruppen, Schlüsselbundeinträge zwischen Apps zu teilen.

Schlüsselbundeinträge können nur zwischen Apps desselben Entwicklers gemeinsam genutzt werden. Für die gemeinsame Nutzung von Schlüsselbundeinträgen werden Apps anderer Anbieter angewiesen, Zugriffsgruppen mit einem Präfix verwenden, das ihnen durch das Apple-Entwicklerprogramm über Anwendungsgruppen zugewiesen wurde. Das Präfix als Erfordernis und die Einzigartigkeit der Anwendungsgruppe werden über die Codesignierung, die [Bereitstellungsprofile](#) und das [Apple-Entwicklerprogramm](#) sichergestellt.

Die Schlüsselbunddaten werden mit einer Klassenstruktur geschützt, die der Klassenstruktur ähnelt, die für die [Datensicherheit](#) bei Dateien zum Einsatz kommt. Diese Klassen weisen ähnliche Verhaltensweisen wie die Datensicherheitsklassen für Dateien auf, verwenden aber separate Schlüssel und Funktionen.

Verfügbarkeit	Sicherheit von Dateidaten	Sicherheit von Schlüsselbunddaten
Wenn entsperrt	<code>NSFileProtectionComplete</code>	<code>kSecAttrAccessibleWhenUnlocked</code>
Wenn gesperrt	<code>NSFileProtectionCompleteUnlessOpen</code>	Nicht verfügbar
Nach erstem Entsperren	<code>NSFileProtectionCompleteUntilFirstUserAuthentication</code>	<code>kSecAttrAccessibleAfterFirstUnlock</code>
Immer	<code>NSFileProtectionNone</code>	<code>kSecAttrAccessibleAlways</code>
Code aktiviert	Nicht verfügbar	<code>kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly</code>

Apps, die Dienste für die Hintergrundaktualisierung verwenden, können `kSecAttrAccessibleAfterFirstUnlock` für Schlüsselbundobjekte nutzen, auf die bei Hintergrundaktualisierungen zugegriffen werden muss.

Die Klasse `kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly` verhält sich ähnlich wie die Klasse `kSecAttrAccessibleWhenUnlocked`, ist aber nur verfügbar, wenn das Gerät mit einem Gerätecode konfiguriert wurde. Diese Klasse existiert nur im systemeigenen Keybag. Sie:

- wird nicht mit dem iCloud-Schlüsselbund synchronisiert,
- wird nicht gesichert,
- wird nicht in Escrow-Keybags integriert.

Wenn der Gerätecode entfernt oder zurückgesetzt wird, werden diese Objekte unbrauchbar, da die Klassenschlüssel verworfen werden.

Andere Schlüsselbundklassen besitzen ein „Nur dieses Gerät“-Gegenstück, das immer mit der UID geschützt wird, wenn eine Kopie bei der Sicherung des Geräts erstellt wird. Dadurch wird es bei der Wiederherstellung auf einem anderen Gerät nutzlos. Apple hat Sicherheit und Benutzerfreundlichkeit sorgfältig abgewogen und Schlüsselbundklassen gewählt, die von der Art der zu sichernden Informationen und davon abhängen, zu welchem Zeitpunkt iOS und iPadOS darauf zugreifen müssen. Ein VPN-Zertifikat muss zum Beispiel immer verfügbar sein, damit das Gerät eine permanente Verbindung besitzt, wird aber als „nicht-migrierend“ klassifiziert, kann also nicht auf ein anderes Gerät übertragen werden.

Sicherheit von Schlüsselbunddatenklassen

Die unten aufgeführten Sicherheitsklassen werden für Schlüsselbundobjekte erzwungen.

Objekt	Zugänglich
WLAN-Passwörter	Nach erstem Entsperren
Mail-Accounts	Nach erstem Entsperren
Microsoft Exchange ActiveSync-Accounts	Nach erstem Entsperren
VPN-Passwörter	Nach erstem Entsperren
LDAP, CalDAV, CardDAV	Nach erstem Entsperren
Account-Token für soziale Netzwerke	Nach erstem Entsperren
Schlüssel für Handoff-Ankündigungen	Nach erstem Entsperren
iCloud-Token	Nach erstem Entsperren
iMessage-Schlüssel	Nach erstem Entsperren
Homeshare-Passwörter	Wenn entsperrt
Safari-Passwörter	Wenn entsperrt
Safari-Lesezeichen	Wenn entsperrt
Finder-/iTunes-Backup	Wenn entsperrt, nicht-migrierend
Von einem Konfigurationsprofil installierte private Schlüssel	Wenn entsperrt, nicht-migrierend
VPN-Zertifikate	Immer, nicht-migrierend
Bluetooth®-Schlüssel	Immer, nicht-migrierend
Apple Push-Benachrichtigungsdienst-Token (APNS)	Immer, nicht-migrierend
iCloud-Zertifikate und privater Schlüssel	Immer, nicht-migrierend
SIM-PIN	Immer, nicht-migrierend
Von einem Konfigurationsprofil installierte Zertifikate	Immer
Token für „Wo ist?“	Immer
Voicemail	Immer

Schlüsselbundzugriffssteuerung

Schlüsselbunde können Zugriffssteuerungslisten (ACLs) verwenden, um Richtlinien für Zugriffs- und Authentifizierungsanforderungen festzulegen. Objekte können mittels Bedingungen die Anwesenheit des Benutzers für den Zugriff auf sie zur Voraussetzung machen, indem sie festlegen, dass der Zugriff nur möglich ist, wenn sich der Benutzer per Face ID oder Touch ID oder durch die Eingabe des Gerätecodes oder -passworts authentifiziert. Der Zugriff auf Objekte kann auch eingeschränkt werden, indem festgelegt wird, dass die Face ID- oder Touch ID-Registrierung seit dem Hinzufügen des Objekts nicht geändert wurde. Mit dieser Einschränkung lässt sich verhindern, dass ein Angreifer seinen eigenen Fingerabdruck hinzufügt, um so auf ein Objekt im Schlüsselbund zuzugreifen. ACLs werden in der Secure Enclave evaluiert und nur dann an den Kernel weitergegeben, wenn die angegebenen Einschränkungen erfüllt sind.

Schlüsselbundarchitektur in macOS

macOS bietet außerdem Zugriff auf den Schlüsselbund, in dem Benutzernamen und die zugehörigen Passwörter sowie digitale Identitäten, Verschlüsselungsschlüssel und geschützte Notizen sicher und bequem gespeichert werden können. Der Zugriff auf den Schlüsselbund erfolgt durch das Öffnen der App „Schlüsselbundzugriff“ im Pfad / Applications/Utilities/. Mit dem Schlüsselbund entfällt für Benutzer die Notwendigkeit, die Anmeldeinformationen für jede Ressource eingeben – oder sie sich überhaupt merken – zu müssen. Für jeden Mac-Benutzer wird ein anfänglicher Standardschlüsselbund erstellt. Benutzer haben aber die Möglichkeit, für spezifische Zwecke andere Schlüsselbunde zu erstellen.

Ergänzend zu den benutzerspezifischen Schlüsselbunden nutzt macOS weitere, auf Systemebene angelegte Schlüsselbunde, um Authentifizierungswerte zu speichern, die nicht benutzerspezifisch sind. Dazu gehören zum Beispiel Anmeldeinformationen für Netzwerke und PKI-Identitäten (Public Key Infrastructure). Einer dieser Schlüsselbunde – System Roots – ist unveränderlich. Darin werden für die Internet-PKI genutzte Root-Zertifikate von Zertifizierungsstellen (CA) gespeichert, die die Grundlage für gängige Tätigkeiten wie Online-Banking und E-Commerce bilden. Auf verwalteten Mac-Computern können Benutzer in ganz ähnlicher Weise von einer internen Zertifizierungsstelle bereitgestellte Zertifikate implementieren, um interne Sites und Dienste zu validieren.

FileVault

Volumeverschlüsselung mit FileVault bei macOS

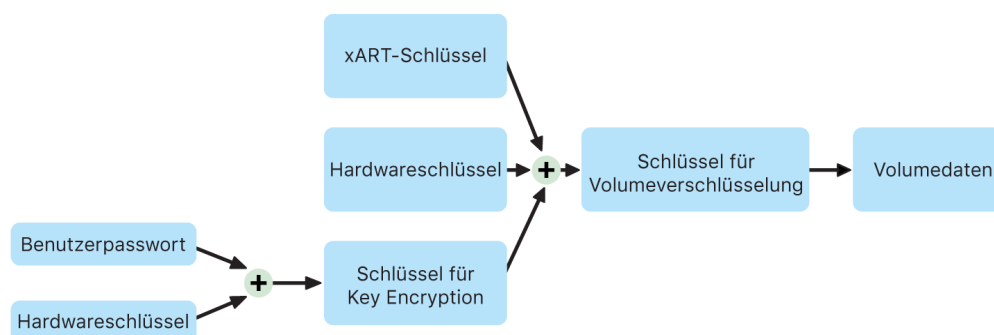
Mac-Computer verfügen mit FileVault über eine integrierte Verschlüsselungstechnologie, die alle nicht im aktiven Zugriff befindlichen Daten schützt. FileVault nutzt den AES-XTS-Algorithmus für die Datenverschlüsselung, um Volumes auf internen Speichergeräten und Wechseldatenträgern zu schützen.

Auf einem Mac mit Apple Chips wird FileVault unter Verwendung der Datensicherheitsklasse C mit einem Volumeschlüssel implementiert. Mac-Computer mit einem Apple T2 Security Chip sowie Mac-Computer mit Apple Chips nutzen verschlüsselte interne Speichergeräte, die direkt mit der Secure Enclave verbunden sind, deren hardwarebasierte Sicherheitsfunktion sowie die der AES-Engine. Wenn ein Benutzer die FileVault-Funktion auf seinem Mac aktiviert, muss er zum Starten (oder Booten) seine Anmeldeinformationen eingeben.

Interne Speichergeräte bei aktivierter FileVault-Funktion

Ohne gültige Anmeldeinformationen oder einen kryptografischen Wiederherstellungsschlüssel bleibt das interne APFS-Volume verschlüsselt und vor unbefugtem Zugriff geschützt, selbst wenn das physische Speichergerät entfernt und an einen anderen Computer angeschlossen wird. Bei macOS 10.15 gilt dies für System- und für Datenvolumes. Ab macOS 11 wird das Systemvolume durch das Signed System Volume (SSV) geschützt, wohingegen das Datenvolume weiterhin durch Verschlüsselung geschützt wird. Bei Mac-Computern mit Apple Chips sowie solchen mit einem T2-Chip erfolgt die interne Volumeverschlüsselung durch den Aufbau und die Verwaltung einer Hierarchie von Schlüsseln, ausgehend von den hardwarebasierten Verschlüsselungstechnologien, die in den Chip integriert sind. Die Hierarchie dieser Schlüssel ist so konzipiert, dass vier Zielsetzungen gleichzeitig realisiert werden:

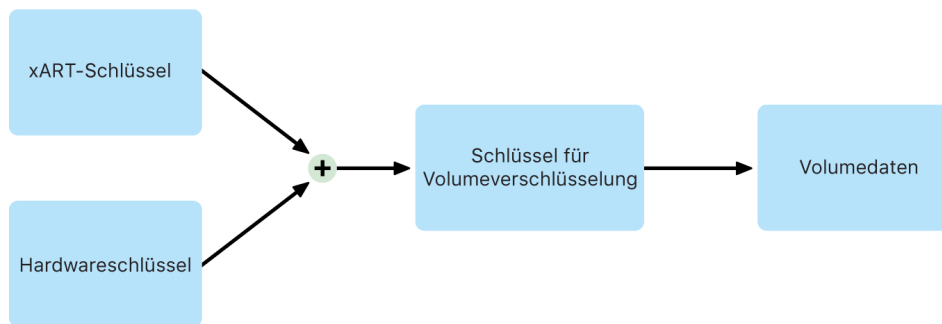
- Für die Entschlüsselung ist das Passwort des Benutzers erforderlich.
- Das System ist vor Brute-Force-Angriffen geschützt, die direkt gegen das vom Mac entfernte Speichermedium gerichtet sind.
- Inhalt kann schnell und sicher gelöscht werden, einfach indem das notwendige kryptografische Material gelöscht wird.
- Benutzer werden in die Lage versetzt, ihr Passwort zu ändern (und damit indirekt auch die kryptografischen Schlüssel zum Schutz ihrer Dateien), ohne dass das gesamte Volume neu verschlüsselt werden muss.



Bei Mac-Computern mit Apple Chips sowie solchen mit einem T2-Chip erfolgt die gesamte Verwaltung der FileVault-Schlüssel in der Secure Enclave. Die für die Verschlüsselung eingesetzten Schlüssel werden der Intel-CPU nie direkt offengelegt. Alle APFS-Volumes werden standardmäßig unter Verwendung eines Volumeverschlüsselungsschlüssels erstellt. Die Volume- und Metadateninhalte werden mithilfe dieses Volumeverschlüsselungsschlüssels verschlüsselt, der seinerseits mit dem Klassenschlüssel verpackt wird. Der Klassenschlüssel wird, sofern FileVault aktiviert ist, durch die Kombination aus dem Passwort des Benutzers und der Hardware-UID geschützt.

Interne Speichergeräte bei deaktivierter FileVault-Funktion

Wenn bei Mac-Computern mit Apple Chips sowie solchen mit einem T2-Chip bei der Erstkonfiguration mithilfe des Systemassistenten FileVault nicht aktiviert wird, wird das Volume zwar verschlüsselt, aber der Volumeverschlüsselungsschlüssel nur durch die Hardware-UID in der Secure Enclave geschützt.



Wenn FileVault zu einem späteren Zeitpunkt aktiviert wird – ein Vorgang, der unmittelbar umgesetzt wird, da die Daten bereits verschlüsselt sind – wird durch einen Anti-Replay-Mechanismus verhindert, dass der alte (nur auf der Hardware-UID basierende) Schlüssel zum Entschlüsseln des Volumes verwendet werden kann. Das Volume wird daraufhin (wie oben beschrieben) durch die Kombination aus dem Passwort des Benutzers und der Hardware-UID geschützt.

Löschen von FileVault-Volumes

Beim Löschen eines Volumes wird dessen Volumeverschlüsselungsschlüssel durch die Secure Enclave sicher gelöscht. Dies verhindert künftige Zugriffe mithilfe dieses Schlüssels selbst durch die Secure Enclave. Darüber hinaus werden alle Volumeverschlüsselungsschlüssel mithilfe eines Medienschlüssels verpackt. Der Medienschlüssel bedeutet keine zusätzliche Vertraulichkeit für die Daten, sondern dient vielmehr dem Zweck, Daten schnell und sicher zu löschen, da ohne den Schlüssel ein Entschlüsseln der Daten nicht möglich ist.

Auf Mac-Computern mit Apple Chips sowie solchen mit T2-Chip wird die Löschung des Medienschlüssels durch von der [Secure Enclave](#) unterstützte Technologie, z. B. MDM-Fernlöschbefehle, garantiert. Wird der Medienschlüssel auf diese Weise gelöscht, wird das gesamte Volume mit kryptografischen Mitteln unzugänglich gemacht.

Wechseldatenträger

Für die Verschlüsselung von Wechseldatenträgern werden die Sicherheitsfunktionen der Secure Enclave nicht eingesetzt. Ihre Verschlüsselung erfolgt daher auf dieselbe Weise wie bei Intel-basierten Mac-Computern ohne T2-Chip.

Verwalten von FileVault in macOS

Bei macOS können Unternehmen FileVault mit Secure Token oder Bootstrap Token verwalten.

Verwendung des Secure Token

Beim Apple File System (APFS) unter macOS 10.13 (oder neuer) ergibt sich eine Änderung hinsichtlich der Art und Weise, wie FileVault-Verschlüsselungsschlüssel generiert werden. Bei früheren Versionen von macOS auf CoreStorage-Volumes wurden die Schlüssel, die für den FileVault-Verschlüsselungsprozess verwendet wurden, zu dem Zeitpunkt erstellt, an dem der Benutzer oder die Organisation FileVault auf einem Mac aktivierte. Bei macOS auf APFS-Volumes werden die Schlüssel entweder beim Erstellen des Benutzers, beim Festlegen des Benutzerpassworts oder bei der erstmaligen Anmeldung eines Benutzers am Mac erstellt. Diese Implementierung der Verschlüsselungsschlüssel, der Zeitpunkt ihrer Generierung und die Art ihrer Speicherung sind Bestandteil einer neuen Funktion, die „*Secure Token*“ oder „sicherer Token“ genannt wird. Genau genommen ist ein sicherer Token die verpackte Version eines KEK (Key Encryption Key), die durch das Passwort des Benutzers geschützt ist.

Bei der Bereitstellung von FileVault auf einem APFS-Volume kann der Benutzer weiterhin:

- vorhandene Werkzeuge und Prozesse verwenden, zum Beispiel das Hinterlegen eines persönlichen Wiederherstellungsschlüssels (PRK) in einer Lösung für die [Mobilgeräteverwaltung \(Mobile Device Management, MDM\)](#)
- einen institutionellen Wiederherstellungsschlüssel (IRK, Institutional Recovery Key) erstellen und verwenden
- die FileVault-Aktivierung bis zum An- oder Abmelden eines Benutzers am Mac verzögern

Bei macOS 11 bewirkt das Festlegen des Initialpassworts für den allerersten Benutzer auf einem Mac, dass diesem Benutzer ein Secure Token zuerkannt wird. Für bestimmte Workflows und Prozesse ist dieses Verhalten möglicherweise nicht wünschenswert, da früher das Zuerkennen des ersten Secure Token voraussetzte, dass sich der Benutzer mit seinem Account anmeldet. Damit dies nicht geschieht, muss `DisabledTags;SecureToken` dem programmseits erstellten Attribut `AuthenticationAuthority` des Benutzers hinzugefügt werden, bevor dessen Passwort festgelegt wird. Dies kann wie folgt geschehen:

```
sudo dscl . append /Users/<user name> AuthenticationAuthority  
";DisabledTags;SecureToken"
```

Verwendung des „Bootstrap Token“

macOS 10.15 umfasst mit *Bootstrap Token* eine neue Funktion, die es ermöglicht, mobilen Accounts und zugleich dem bei der Geräteregistrierung optional erstellten Admin-Account („verwalteter Administrator“) einen Secure Token zuzuerkennen. Bei macOS 11 kann der Bootstrap Token jedem Benutzer, der sich an einem Mac-Computer anmeldet – auch einem lokalen Benutzeraccount –, einen Secure Token zuerkennen. Die Verwendung der Bootstrap-Token-Funktion von macOS 10.15 (oder neuer) ist an die folgenden Bedingungen geknüpft:

- Registrierung des Mac-Computers in Apple School Manager oder Apple Business Manager, was den Mac zu einem betreuten Gerät macht
- Unterstützung für den MDM-Anbieter

Wenn die MDM-Lösung diese Funktion unterstützt, wird in macOS 10.15.4 (oder neuer) ein Bootstrap-Token bei der ersten Anmeldung jedes für Secure Token aktivierten Benutzers generiert und in der MDM-Lösung hinterlegt. Sofern erforderlich, kann ein Bootstrap Token auch mit dem Befehlszeilenprogramm `profiles` generiert und in der MDM-Lösung hinterlegt werden.

Bei macOS 11 kann ein Bootstrap Token außer für das Zuerkennen eines Secure Token zu Benutzeraccounts auch für andere Zwecke genutzt werden. Bei einem Mac mit Apple Chips kann ein Bootstrap Token (sofern verfügbar) dazu genutzt werden, Kernel-Erweiterungen und Softwareaktualisierungen zu autorisieren, wenn die Verwaltung mittels einer MDM-Lösung erfolgt.

Schutz persönlicher und personenbezogener Daten durch Apple

App-Zugriff auf Benutzerdaten schützen

Zusätzlich zum Verschlüsseln gespeicherter Daten tragen Apple-Geräte dazu bei, dass Apps daran gehindert werden, ohne die Zustimmung des Benutzers auf dessen persönliche und personenbezogene Informationen zuzugreifen. Dazu werden verschiedene Technologien wie zum Beispiel [Data Vault](#) eingesetzt. In den Einstellungen von iOS und iPadOS und in den Systemeinstellungen von macOS können Benutzer sehen, welchen Apps sie den Zugriff auf welche Informationen erlaubt haben. Sie können einer App das Zugriffsrecht jederzeit gewähren oder entziehen. In folgenden Apps und Funktionen wird der Zugriff erzwungen:

- *iOS, iPadOS und macOS*: Kalender, Kamera, Kontakte, Mikrofon, Fotos, Erinnerungen, Spracherkennung
- *iOS und iPadOS*: Bluetooth, Home, Medien, Medien-Apps und Apple Music, Motion und Fitness
- *iOS und watchOS*: Health
- *macOS*: Eingabeüberwachung (z. B. Tastatureingaben), Eingabeaufforderung, Bildschirmaufnahme (z. B. statische Bildschirmfotos und Videoaufzeichnungen), Systemeinstellungen

In iOS 13.4 (oder neuer) und iPadOS 13.4 (oder neuer) werden die Daten aller Apps von anderen Anbietern automatisch in einem [Data Vault](#) geschützt. Dieses Data Vault dient dem Schutz vor unbefugten Datenzugriffen, selbst durch Prozesse, die nicht in einer Sandbox ausgeführt werden. Zu den zusätzlichen Klassen in iOS 15 (oder neuer) gehören „Lokales Netzwerk“, „Interaktionen in der Nähe“, „Sensor- und Nutzungsdaten“ und „Fokus“.

Wenn sich der Benutzer bei iCloud anmeldet, erhalten Apps in iOS und iPadOS standardmäßig Zugriff auf iCloud Drive. Benutzer können aber den iCloud-Zugriff für einzelne Apps im Bereich „iCloud“ der Einstellungen ändern. Darüber hinaus unterstützen iOS und iPadOS Einschränkungen, die den Datenaustausch zwischen Apps und Accounts, die durch eine Lösung für die [Mobilgeräteverwaltung \(Mobile Device Management, MDM\)](#) installiert wurden, und den vom Benutzer installierten Apps und Accounts grundsätzlich verhindern.

Zugriff auf Gesundheitsdaten von Benutzern schützen

HealthKit stellt auf einem iPhone und einer Apple Watch einen zentralen Speicherort für Gesundheits- und Fitnessdaten bereit. HealthKit arbeitet direkt mit Gesundheits- und Fitnessgeräten zusammen, z. B. mit kompatiblen BLE-Herzfrequenzmessgeräten (Bluetooth Low Energy) und dem Motion-Coprozessor, der in viele iOS-Geräte integriert ist. Die gesamte Interaktion zwischen HealthKit, den für Gesundheit und Fitness verwendeten Apps und Geräten sowie den Gesundheitseinrichtungen bedarf der Zustimmung durch den Benutzer. Diese Daten werden mit der Datensicherheitsklasse „Geschützt, außer wenn offen“ gespeichert. Zehn Minuten, nachdem das Gerät gesperrt wird, wird der Zugriff auf die Daten aufgehoben. Die Daten werden erst wieder zugänglich, wenn der Benutzer das nächste Mal seinen Code eingibt oder Face ID bzw. Touch ID zum Entsperren des Geräts verwendet.

Erfassen und Speichern von Gesundheits- und Fitnessdaten

HealthKit sammelt und speichert auch Verwaltungsdaten (zum Beispiel Zugriffsrechte für Apps, Namen der mit HealthKit verbundenen Geräte) sowie Planungsinformationen, die verwendet werden, um Apps zu starten, wenn neue Daten verfügbar werden. Diese Daten werden mit der Datensicherheitsklasse „Geschützt bis zur ersten Benutzerauthentifizierung“ gespeichert. In temporären Journal-Dateien werden die Gesundheitsdatensätze gespeichert, die erzeugt werden, wenn das Gerät gesperrt ist, z. B. wenn der Benutzer trainiert. Diese Daten werden mit der Datensicherheitsklasse „Geschützt, außer wenn offen“ gespeichert. Wenn das Gerät entsperrt wird, werden die temporären Journal-Dateien in die Hauptdatenbank für Gesundheitsdaten importiert und anschließend aus dem Journal gelöscht.

Gesundheitsdaten können in iCloud gespeichert werden. Für die Ende-zu-Ende-Verschlüsselung von Gesundheitsdaten sind iOS 12 (oder eine neuere Version) und die Zwei-Faktor-Authentifizierung erforderlich. Andernfalls werden die Daten eines Benutzers zwar beim Speichern und Übertragen verschlüsselt, es erfolgt jedoch keine Ende-zu-Ende-Verschlüsselung. Nach dem Aktivieren der Zwei-Faktor-Authentifizierung und dem Aktualisieren auf iOS 12 (oder eine neuere Version) werden die Gesundheitsdaten eines Benutzers für die Ende-zu-Ende-Verschlüsselung migriert.

Wenn ein Benutzer ein Backup seines Geräts mit dem Finder (macOS 10.15 oder neuer) oder mit iTunes (macOS 10.14 oder älter) erstellt, werden Gesundheitsdaten nur gespeichert, wenn das Backup verschlüsselt wird.

Klinische Gesundheitseinträge

Benutzer können sich innerhalb der App „Health“ bei unterstützten Gesundheitssystemen anmelden, um eine Kopie ihrer klinischen Gesundheitseinträge abzurufen. Wenn ein Benutzer mit einem Gesundheitssystem verbunden wird, muss er sich mit OAuth 2 Client-Anmeldedaten authentifizieren. Steht die Verbindung, werden die Gesundheitseinträge über eine mittels TLS 1.3 geschützte Verbindung direkt von der Gesundheitseinrichtung heruntergeladen. Nach dem Download werden die Gesundheitseinträge zusammen mit anderen Gesundheitsdaten sicher gespeichert.

Integrität von Gesundheitsdaten

Die in der Datenbank gespeicherten Daten enthalten Metadaten, um die Herkunft der Datensätze zurückverfolgen zu können. Zu diesen Metadaten gehört eine App-Kennung, die anzeigt, welche App den Datensatz gespeichert hat. Außerdem kann ein optionales Metadatenobjekt eine digital signierte Kopie des Datensatzes enthalten. Diese Kopie dient der Datenintegrität für Datensätze, die von einem vertrauenswürdigen Gerät erzeugt wurden. Für die digitale Signatur wird gemäß RFC 5652 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc5652.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt) das Format Cryptographic Message Syntax (CMS) verwendet.

Zugriff auf Gesundheitsdaten durch Apps anderer Anbieter

Der Zugriff auf die HealthKit-API wird über Berechtigungen gesteuert und Apps müssen sich an Einschränkungen für die Verwendung der Daten halten. Zum Beispiel dürfen Apps Gesundheitsdaten nicht zu Werbezwecken verwenden. Apps müssen dem Benutzer auch eine Datenschutzrichtlinie bereitstellen, die beschreibt, wie die Gesundheitsdaten verwendet werden.

Der Zugriff von Apps auf die Gesundheitsdaten wird in den Datenschutzeinstellungen des Benutzers festgelegt. Ähnlich wie bei Kontakten, Fotos und anderen iOS-Datenquellen werden Benutzer gefragt, ob der Zugriff gewährt werden soll, wenn Apps auf Gesundheitsdaten zuzugreifen versuchen. Aber bei Gesundheitsdaten werden Lese- und Schreibzugriff und Zugriff auf die einzelnen Arten von Gesundheitsdaten getrennt voneinander gewährt. Benutzer können die für ihre Gesundheitsdaten erteilten Zugriffsrechte über „Einstellungen“ > „Health“ > „Datenzugriff & Geräte“ einsehen und widerrufen.

Apps mit Schreibzugriff können auch die von ihnen geschriebenen Daten lesen. Apps mit Lesezugriff können von allen Quellen geschriebene Daten lesen. Apps können aber nicht den Zugriff anderer Apps auf die Daten bestimmen. Außerdem können Apps nicht überprüfen, ob sie Lesezugriff auf Gesundheitsdaten haben. Apps ohne Lesezugriff erhalten keine Antworten auf ihre Anfragen – das Antwortverhalten entspricht damit dem einer leeren Datenbank. So können die Apps keine Informationen über die Gesundheit des Benutzers davon ableiten, welche Datenarten geschrieben werden.

Notfallpass für Benutzer

Die App „Health“ bietet dem Benutzer die Möglichkeit, ein Formular für den Notfallpass mit Informationen auszufüllen, die bei einem Notfall wichtig sein können. Diese Informationen werden manuell eingegeben bzw. aktualisiert und nicht mit den Daten in den Gesundheitsdatenbanken synchronisiert.

Die Informationen zum Notfallpass werden durch Tippen auf die [Taste „Notfall“](#) im Sperrbildschirm angezeigt. Die Informationen werden mit der Datensicherheitsklasse „Kein Schutz“ auf dem Gerät gespeichert, sie sind also auch ohne Eingabe des Gerätecodes zugänglich. Das Einrichten eines Notfallpasses ist optional und ermöglicht es dem Benutzer, selbst zwischen Sicherheit und Datenschutz abzuwägen. Diese Daten werden bei iOS 13 (oder früheren Versionen) in einem iCloud-Backup gesichert. Bei iOS 14 wird der Notfallpass zwischen den Geräten mittels CloudKit und mit denselben Verschlüsselungsmerkmalen wie die restlichen Health-Daten synchronisiert.

Teilen mit Health

In iOS 15 haben Benutzer mit der Health-App die Möglichkeit, ihre Gesundheitsdaten mit anderen Personen zu teilen. Die Gesundheitsdaten werden zwischen den beiden Personen mithilfe der Ende-zu-Ende-iCloud-Verschlüsselung ausgetauscht. Apple hat keinen Zugriff auf die mit Health geteilten Daten. Um diese Funktion verwenden zu können, müssen Sender und Empfänger iOS 15 (oder neuer) installiert und die Zwei-Faktor-Authentifizierung aktiviert haben.

Benutzer können in der Health-App mit der Funktion „Teilen mit“ ihre Gesundheitsdaten auch mit ihren Gesundheitsdienstleistern teilen. Die mit dieser Funktion geteilten Daten werden nur den vom Benutzer ausgewählten Gesundheitsdienstleistern zur Verfügung gestellt und mit Ende-zu-Ende-Verschlüsselung übertragen. Apple hat keinen Zugriff auf die Verschlüsselungsschlüssel, um die mit der Funktion „Teilen mit“ geteilten Daten zu entschlüsseln, anzuzeigen oder anderweitig zu verarbeiten. Weitere Informationen darüber, wie die Gesundheitsdaten der Benutzer durch diesen Dienst geschützt werden, sind im Abschnitt [Security and Privacy](#) in der Apple-Registrierungsanleitung für Gesundheitsdienstleister verfügbar.

Digitales Signieren und Verschlüsseln

Zugriffssteuerungslisten

Daten des Schlüsselbunds werden partitioniert und mithilfe von Zugriffssteuerungslisten (Access Control Lists, ACLs) geschützt. Auf diese Weise werden Anmeldedaten, die durch Apps anderer Anbieter gespeichert werden, für Apps anderer Identitäten nur zur Verfügung gestellt, wenn der Benutzer diesem Zugriff explizit zustimmt. Dieser Schutzmechanismus sichert zugleich auch die Anmeldedaten ab, die auf Apple-Geräten für eine breite Palette an Apps und Diensten einer Organisation verwendet werden.

Mail

Mit der App „Mail“ können digital signierte und verschlüsselte E-Mail-Nachrichten gesendet werden. Die App „Mail“ erkennt – bei Verwendung von Zertifikaten für das digitale Signieren und Verschlüsseln auf verbundenen PIV-Token (Personal Identification Verification) unterstützter Smart Cards – die jeweiligen RFC 5322-Werte (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc5322.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt) von E-Mail-Adresse, Betreff und alternativem Namen des Inhabers automatisch. Bei diesen Werten werden Groß- und Kleinschreibung unterschieden. Wenn ein konfigurierter E-Mail-Account mit einer E-Mail-Adresse in einem für das digitale Signieren und Verschlüsseln verwendeten Zertifikat auf einem verbundenen PIV-Token übereinstimmt, wird in der App „Mail“ in der Symbolleiste einer neuen E-Mail-Nachricht automatisch die Taste „Signieren“ angezeigt. Wenn die App „Mail“ über das Verschlüsselungszertifikat des E-Mail-Empfängers verfügt oder es in der globalen Adressliste von Microsoft Exchange (GAL) findet, wird in der Symbolleiste einer neuen Nachricht das Symbol eines geöffneten Schlosses angezeigt. Ein Symbol in Form eines geschlossenen Schlosses besagt, dass die Nachricht mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und in dieser Form gesendet wird.

S/MIME für einzelne Nachrichten

iOS, iPadOS und macOS unterstützen S/MIME auf der Ebene einzelner Nachrichten. S/MIME-Benutzer können daher bestimmen, ob standardmäßig alle Nachrichten per S/MIME signiert und verschlüsselt werden sollen oder ob sie individuell für jede Nachricht angeben wollen, ob sie signiert und verschlüsselt werden soll.

Für S/MIME verwendete Identitäten können auf Apple-Geräten über ein Konfigurationsprofil, über eine Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM), über SCEP (Simple Certificate Enrollment Protocol) oder über eine Zertifizierungsstelle für Microsoft Active Directory bereitgestellt werden.

Smart Cards

macOS 10.12 (oder neuer) bietet native Unterstützung für PIV-Karten (d. h. für Karten zur Verifizierung der persönlichen Identität). Derartige Karten sind in Handels- und Regierungsorganisationen weit verbreitet, wo sie für die Zwei-Faktor-Authentifizierung und zum digitalen Signieren und Verschlüsseln verwendet werden.

Smart Cards enthalten eine oder mehrere digitale Identitäten, die aus jeweils einem Paar öffentlicher und privater Schlüssel und dem jeweils zugehörigen Zertifikat bestehen. Das Entsperren einer Smart Card mit ihrer persönlichen Identifikationsnummer (PIN) ermöglicht den Zugriff auf die privaten Schlüssel, die für die Authentifizierung und für Signier- und Verschlüsselungsvorgänge verwendet werden. Hierbei bestimmt das Zertifikat, für welche Zwecke ein Schlüssel verwendet werden kann, welche Attribute ihm zugewiesen sind und ob es durch eine Zertifizierungsstelle validiert (signiert) wurde oder nicht.

Smart Cards können für die Zwei-Faktor-Authentifizierung verwendet werden. Die beiden Faktoren, die zum Entsperren einer Karte benötigt werden, sind „etwas, das sich im Besitz des Benutzers befindet“ (d. h. die Karte) und „etwas, was der Benutzer weiß“ (d. h. die PIN). macOS 10.12 (und neuere Versionen) bieten darüber hinaus native Unterstützung für die Authentifizierung per Smart Card im Anmeldefenster und für die Authentifizierung per Client-Zertifikat gegenüber Websites in Safari. Ebenfalls unterstützt wird die Kerberos-Authentifizierung auf der Basis von Schlüsselpaaren (PKINIT) für die Gesamtauthentifizierung (Single Sign-On) gegenüber Diensten, die von Kerberos unterstützt werden. Näheres über Smart Cards und macOS ist in der [Einführung in die Integration von Smart Cards](#) im Handbuch *Implementierung von Apple-Plattformen* zu finden.

Verschlüsselte Image-Dateien

Bei macOS dienen verschlüsselte Image-Dateien als sichere Container, in die bzw. denen Benutzer vertrauliche Dokumente und andere Dateien transferieren bzw. speichern können. Verschlüsselte Image-Dateien werden mithilfe des Festplattendienstprogramms (im Pfad / Applications/Utilities/) erstellt. Image-Dateien können mit 128-Bit- oder mit 256-Bit-AES-Verschlüsselung verschlüsselt werden. Da eine aktivierte Image-Datei wie ein lokales, mit dem Mac verbundenes Volume behandelt wird, können Benutzer die darin gespeicherten Dateien und Ordner kopieren, verschieben und öffnen. Wie bei FileVault wird der Inhalt der Image-Datei in Echtzeit ent- und verschlüsselt. Auf der Basis verschlüsselter Image-Dateien können Benutzer Dokumente, Dateien und Ordner sicher austauschen. Für den Austausch können sie eine Image-Datei auf einen Wechseldatenträger transferieren, sie als Anhang einer E-Mail versenden oder sie auf einem Server im Netzwerk speichern. Weitere Informationen über verschlüsselte Image-Dateien enthält das [Festplattendienstprogramm – Benutzerhandbuch](#).

Sicherheit bei Apps

Sicherheit bei Apps – Übersicht

Apps gehören heute zu den risikoreichsten Elementen einer Sicherheitsarchitektur. Apps bieten dem Benutzer fantastische Produktivitätsgewinne, haben aber auch das Potenzial, die Systemsicherheit, die Stabilität und die Benutzerdaten zu gefährden, wenn mit ihnen nicht richtig umgegangen wird.

Aus diesem Grund unterstützt Apple einen auf mehreren Ebenen angelegten Schutzmechanismus, mit dem sichergestellt wird, dass Apps frei von bekannten Schadprogrammen sind und nicht von unbefugter Seite manipuliert wurden. Zusätzliche Schutzmaßnahmen sorgen dafür, dass der Zugriff auf Benutzerdaten streng überwacht wird. Diese sicherheitsrelevante Steuerung und Kontrolle ergibt eine stabile, sichere Plattform für Apps und ermöglicht es Tausenden von Entwicklern, Hunderttausende Apps für iOS, iPadOS und macOS zu entwickeln, ohne dass die Systemintegrität beeinträchtigt wird. Benutzer können mit ihrem Apple-Gerät auf diese Apps zugreifen, ohne sich unnötig Sorgen wegen Viren, Malware oder nicht autorisierter Attacken machen zu müssen.

Alle Apps auf dem iPhone, iPad und iPod touch stammen aus dem App Store und sie werden in einer Sandbox ausgeführt, sodass die Kontrolle sehr engmaschig ist.

Viele Apps für den Mac werden aus dem App Store geladen. Mac-Benutzer können Apps aber auch aus dem Internet herunterladen und sie auf dem Mac ausführen. Um das Laden von Apps aus dem Internet sicher zu unterstützen, umfasst macOS sicherheitsrelevante Steuer- und Kontrollmechanismen auf weiteren Ebenen. In einem ersten Schritt müssen in macOS 10.15 (und neueren Versionen) standardmäßig alle Mac-Apps von Apple beglaubigt werden, um gestartet werden zu können. Diese Anforderung soll sicherstellen, dass die Apps frei von bekannten Schadprogrammen sind, ohne dass sie über den App Store bereitgestellt werden müssen. Darüber hinaus umfasst macOS einen hochmodernen Virenschutz, der Schadsoftware (Malware) abblockt und, sofern erforderlich, auch entfernt.

Die Ausführung in einer Sandbox dient als zusätzlicher plattformübergreifender Kontrollmechanismus dem Schutz der Benutzerdaten vor unbefugtem Zugriff durch die Apps. Unter macOS werden Daten in kritischen Bereichen selbst in einer Sandbox gehalten. Dies soll sicherstellen, dass der Benutzer die Kontrolle über den Zugriff auf Dateien behält, die sich auf dem Schreibtisch, im Ordner „Dokumente“, im Ordner „Downloads“ und in einigen anderen Bereichen befinden – unabhängig davon, durch welche App der Zugriff erfolgt, und unabhängig davon, ob die betreffende App ihrerseits in einer Sandbox ausgeführt wird oder nicht.

Native Fähigkeit	Äquivalent von Dritten
Unbestätigte Plug-In-Liste, unbestätigte Safari-Erweiterungsliste	Virus-/Malware-Definitionen
Dateiquarantäne	Virus-/Malware-Definitionen
XProtect-/ YARA-Signaturen	Virus-/Schadprogramm-Definitionen; Endpoint Protection
Gatekeeper	Endgeräteschutz: Codesignierung von Apps obligatorisch, um sicherzustellen, dass nur vertrauenswürdige Software ausgeführt wird
efiheck (notwendig bei einem Mac ohne Apple T2 Security Chip)	Endgeräteschutz; Rootkit-Erkennung
App-Firewall	Endgeräteschutz; Firewall
Paketfilter (pf)	Firewall-Lösungen
Systemintegritätsschutz	Integriert in macOS
Zwingende Zugriffssteuerungen	Integriert in macOS
Kext-Ausschlussliste	Integriert in macOS
Zwingende Codesignierung	Integriert in macOS
App-Beglaubigung	Integriert in macOS

Sicherheit bei Apps für iOS und iPadOS

Sicherheit bei Apps für iOS und iPadOS – Einführung

Anders als andere mobile Plattformen erlauben es iOS und iPadOS ihren Benutzern nicht, potenziell schädliche, unsigned Apps von Websites zu installieren oder nicht vertrauenswürdige Apps auszuführen. Während der Laufzeit wird beim Laden die Codesignatur aller ausführbaren Speicherseiten überprüft, um sicherzustellen, dass die App seit der Installation oder letzten Aktualisierung nicht modifiziert wurde.

Nachdem überprüft wurde, dass die App aus einer vertrauenswürdigen Quelle stammt, setzen iOS und iPadOS Sicherheitsmaßnahmen durch, die so konzipiert wurden, dass andere Apps oder das übrige System durch die App nicht gefährdet werden.

Prozess der App-Codesignierung in iOS und iPadOS

Bei iOS und iPadOS stellt Apple die Sicherheit von Apps mithilfe von Maßnahmen wie der zwingenden Codesignierung und strengen Anmeldevorgaben für Entwickler sicher.

Zwingende Codesignierung

Nachdem der iOS- bzw. iPadOS-Kernel gestartet wurde, bestimmt er, welche Benutzerprozesse und Apps ausgeführt werden dürfen. Um sicherzustellen, dass alle Apps von bekannten und genehmigten Quellen stammen und nicht manipuliert wurden, muss der gesamte ausführbare Code für iOS bzw. iPadOS mit einem von Apple ausgegebenem Zertifikat signiert worden sein. Die ab Werk auf dem Gerät vorhandenen Apps (z. B. Mail und Safari) wurden von Apple signiert. Apps von anderen Anbietern müssen ebenfalls vom Entwickler mit einem von Apple ausgegebenen Zertifikat validiert und signiert werden. Die zwingende Codesignierung weitet das „Chain of Trust“-Konzept des Betriebssystems auf Apps aus und verhindert so, dass Apps anderer Anbieter nicht signierten Code ausführen oder Code verwenden, der sich selbst ändert.

Signieren von Apps durch ihre Entwickler

Entwickler können ihre Apps per Zertifikatvalidierung (über das Apple Developer Program) signieren. Sie haben auch die Möglichkeit, Frameworks in ihre Apps einzubetten und diesen Code mithilfe eines von Apple ausgegebenen Zertifikats validieren zu lassen (über eine als String vorliegende Teamkennung).

- *Zertifikatvalidierung:* Um Apps auf iOS- und iPadOS-Geräten entwickeln und installieren zu können, müssen sich Entwickler bei Apple registrieren und dem Apple-Entwicklerprogramm beitreten. Vor der Ausgabe eines Zertifikats überprüft Apple die reale Identität jedes Entwicklers (Einzelpersonen oder Unternehmen). Mit dem Zertifikat können Entwickler Apps signieren und sie zum Vertrieb an den App Store senden. Das bedeutet, dass alle Apps im App Store von identifizierbaren Personen oder Organisationen eingereicht wurden, was für Entwickler schädlicher Apps als Abschreckung dient. Außerdem wird von Apple die korrekte Funktionsweise der Apps überprüft, um sicherzustellen, dass sie keine offensichtlichen Bugs oder anderen Probleme enthalten. Zusätzlich zu den bereits beschriebenen Technologien können Benutzer dank dieser Kuratierung auf die Qualität der gekauften Apps vertrauen.

- *Validierung der Codesignatur:* iOS und iPadOS erlauben es Entwicklern, Frameworks in ihre Apps zu integrieren, die von der jeweiligen App selbst oder von in der App integrierten Erweiterungen genutzt werden können. Um das System und andere Apps davor zu schützen, dass Code aus anderen Apps in ihrem jeweiligen Adressbereich ausgeführt wird, nimmt das System eine Validierung der Codesignatur aller dynamischen Bibliotheken vor, auf die ein Prozess beim Start zugreift. Diese Verifizierung erfolgt über die Team-ID, die aus einem von Apple ausgegebenem Zertifikat extrahiert wird. Bei einer Team-ID handelt es sich um eine zehnstellige alphanumerische Zeichenfolge – z. B. 1A2B3C4D5F. Ein Programm kann auf jede Plattformbibliothek, die auf dem System vorinstalliert ist, sowie auf jede Bibliothek zugreifen, in deren Team-ID dieselbe Codesignatur wie in der Hauptprogrammdatei enthalten ist. Da die auf dem System vorinstallierten ausführbaren Dateien keine Team-ID haben, können sie nur auf Bibliotheken zugreifen, die ebenfalls auf dem System vorinstalliert sind.

Verifizierung von proprietären firmeninternen Apps

Qualifizierte Unternehmen haben die Möglichkeit, firmeninterne Apps zur Verwendung im Unternehmen zu entwickeln und sie an ihre Mitarbeiter zu verteilen. Unternehmen und Organisationen können sich für das Apple Developer Enterprise Program (ADEP) bewerben. Weitere Informationen und die notwendigen Teilnahmevoraussetzungen sind auf der [Apple Developer Enterprise Program-Website](#) zu finden. Eine Organisation, die als ADEP-Mitglied aufgenommen wurde, kann sich für die Ausstellung eines Bereitstellungsprofils registrieren, das es ermöglicht, proprietäre firmeninterne Apps auf autorisierten Geräten auszuführen.

Benutzer müssen das betreffende Bereitstellungsprofil installieren, damit sie diese Apps ausführen können. Dadurch wird sichergestellt, dass nur die von einer Organisation ins Auge gefassten Benutzer die Apps auf ihre iOS- und iPadOS-Geräte laden können. Apps, die über Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM) installiert werden, sind implizit vertrauenswürdig, da die Beziehung zwischen der Organisation und dem Gerät bereits etabliert ist. In allen anderen Fällen müssen Benutzer dem Bereitstellungsprofil der App in den Einstellungen zustimmen. Organisationen können es Benutzern auch verwehren, Apps zuzulassen, die von unbekanntem Entwicklern stammen. Beim erstmaligen Starten einer proprietären firmeninternen App muss das Gerät daher die positive Bestätigung durch Apple einholen, dass die App ausgeführt werden darf.

Sicherheit des Laufzeitprozesses bei iOS und iPadOS

iOS und iPadOS helfen dabei, die Sicherheit der Laufzeit zu gewährleisten, indem sie eine „Sandbox“, deklarierte Berechtigungen und Address Space Layout Randomization (ASLR) verwenden.

Sandboxing

Apps anderer Anbieter werden in einer Sandbox ausgeführt, damit sie keine Änderungen am Gerät vornehmen oder auf Dateien zugreifen können, die von anderen Apps gespeichert wurden. Dadurch können Apps keine von anderen Apps gespeicherten Informationen abrufen oder verändern. Jede App verfügt über ein eigenes Home-Verzeichnis für die zugehörigen Dateien, das bei der Installation der App nach dem Zufallsprinzip ausgewählt wird. Wenn eine App eines anderen Anbieters auf Informationen zugreifen muss, die ihr nicht zugeordnet sind, kann sie das nur über Dienste tun, die explizit von iOS bzw. iPadOS bereitgestellt werden.

Systemdateien und Ressourcen werden ebenfalls von den Apps des Benutzers abgeschirmt. Die meisten Systemdateien und Ressourcen von iOS und iPadOS werden als nicht privilegierter Benutzer „mobile“ ausgeführt. Dies gilt im Übrigen auch für alle Apps anderer Anbieter. Die gesamte Betriebssystempartition ist nur für den Lesezugriff aktiviert. Nicht notwendige Tools (z. B. Dienste für die entfernte Anmeldung) gehören nicht zur Systemsoftware. Die APIs lassen es nicht zu, dass Apps ihre eigenen Privilegien erhöhen, um andere Apps oder iOS bzw. iPadOS zu verändern.

Berechtigungen

Der Zugriff auf Benutzerinformationen und Funktionen (z. B. iCloud und Erweiterbarkeit) durch Apps anderer Anbieter wird über festgelegte Berechtigungen gesteuert. Berechtigungen sind Schlüssel/Wert-Paare, die zusammen mit einer App signiert werden und die eine Authentifizierung über Laufzeitfaktoren wie die UNIX-Benutzerkennung hinaus ermöglichen. Da die Berechtigungen digital signiert sind, können sie nicht verändert werden. Berechtigungen werden in großem Umfang von System-Apps und Daemons zur Durchführung bestimmter privilegierter Vorgänge verwendet, die andernfalls als „root“-Benutzer ausgeführt werden müssten. Dadurch wird die Gefahr einer Privilegienerhöhung durch eine beschädigte System-App oder einen beschädigten Daemon reduziert.

Außerdem können Apps die Hintergrundverarbeitung nur über vom System bereitgestellte APIs verwenden. Dadurch können Apps ohne Leistungseinbußen oder dramatische Beeinträchtigung der Batteriebensdauer weiterarbeiten.

Speicherverwürfelung

Die Speicherverwürfelung (Address Space Layout Randomization, ASLR) bietet Schutz davor, dass Fehler, die den Speicher modifizieren, ausgenutzt werden können. Integrierte Apps stellen mithilfe von ASLR sicher, dass beim Start alle Speicherbereiche nach dem Zufallsprinzip vergeben werden. ASLR ist schon vor dem Start aktiv und ordnet die Speicheradressen von ausführbarem Code, den Systembibliotheken (System Libraries) und den zugehörigen Programmelementen zufällig an und verringert dadurch die Wahrscheinlichkeit vieler Exploits. Ein „return-to-libc“-Angriff versucht beispielsweise, durch die Manipulation der Speicheradressen von Stack- und System-Libraries ein Gerät zum Ausführen von Schadcode zu zwingen. Werden diese zufällig platziert, erschwert dies den Angriff, insbesondere wenn sich dieser gegen mehrere Geräte richtet. Xcode und die Entwicklungsumgebungen für iOS und iPadOS kompilieren Programme anderer Anbieter automatisch, wenn die ASLR-Unterstützung aktiviert ist.

Funktion „Execute Never“ (XN)

Zusätzlichen Schutz bieten iOS und iPadOS durch die ARM-Funktion „Execute Never“ (XN), die Speicherseiten als „nicht ausführbar“ kennzeichnet. Speicherseiten, die als beschreibbar und ausführbar gekennzeichnet sind, können von Apps nur unter streng kontrollierten Bedingungen verwendet werden: Der Kernel überprüft, ob die Apple vorbehaltene Berechtigung „dynamic code signing“ vorhanden ist. Doch selbst dann kann nur ein einziger mmap-Aufruf genutzt werden, um eine ausführbare und beschreibbare Seite anzufordern, die eine zufällige Adresse erhält. Safari verwendet diese Funktion für seinen JavaScript Just-in-Time (JIT) Compiler.

Unterstützung für Erweiterungen bei iOS, iPadOS und macOS

iOS, iPadOS und macOS ermöglichen es Apps, auf der Basis von Erweiterungen Funktionen für andere Apps bereitzustellen. Erweiterungen sind für einen speziellen Zweck signierte, ausführbare Binärdateien, die in eine App verpackt werden. Bei der Installation erkennt das System diese Erweiterungen automatisch und stellt sie unter Verwendung eines Abgleichsystems für andere Apps zur Verfügung.

Erweiterungspunkte

Ein Systembereich, der Erweiterungen unterstützt, wird *Erweiterungspunkt* (Extension Point) genannt. Jeder Erweiterungspunkt stellt APIs bereit und setzt die Richtlinien für den Bereich durch. Das System legt anhand der jeweiligen Abgleichregeln des Erweiterungspunkts fest, welche Erweiterungen zur Verfügung stehen. Das System startet Erweiterungsprozesse bei Bedarf und verwaltet ihren Lebenszyklus automatisch. Berechtigungen können dafür verwendet werden, die Verfügbarkeit von Erweiterungen für bestimmte System-Apps einzuschränken. Das Widget für die Tagesansicht erscheint beispielsweise nur in der Mitteilungszentrale und eine Freigabeerweiterung ist nur im Bereich „Freigabe“ verfügbar. Beispiele für Erweiterungspunkte sind das Widget „Tagesansicht“, Freigabe/Teilen, Eigene Aktionen, Bildbearbeitung, File Provider und Eigene Tastatur.

Kommunikation unter Erweiterungen

Erweiterungen werden in ihrem eigenen Adressbereich ausgeführt. Die Kommunikation zwischen einer Erweiterung und der App, die sie aktiviert, erfolgt über Verfahren der Interprozesskommunikation, die vom System-Framework vermittelt werden. Sie können nicht auf die Dateien oder Speicherbereiche der jeweils anderen Seite zugreifen. Erweiterungen sind voneinander, von der App, die sie enthält, und von anderen Apps, die sie verwenden, abgeschirmt. Sie werden genau wie alle anderen Apps anderer Anbieter in einer Sandbox ausgeführt und ihr Container ist nicht derselbe Container wie der der App. Sie teilen sich jedoch den Zugriff auf die Datenschutzeinstellungen mit der App, die sie enthält. Wenn ein Benutzer einer App den Zugriff auf seine Kontakte erlaubt, so wird dieser Zugriff auch an die Erweiterungen weitergereicht, die in die App integriert sind, nicht aber an Erweiterungen, die durch die App aktiviert werden.

Verwendung eigener Tastaturen

Eigene Tastaturen stellen eine Sonderform einer Erweiterung dar, da sie vom Benutzer für das gesamte System aktiviert werden. Nach dem Aktivieren wird eine Tastaturerweiterung für alle Textfelder verwendet; ausgenommen sind nur die Eingabe des Codes und verschlüsselte Textfelder. Um die Übertragung von Benutzerdaten einzuschränken, werden eigene Tastaturen standardmäßig in einer besonders restriktiven Sandbox ausgeführt. Diese Sandbox blockiert den Zugriff auf das Netzwerk, auf Dienste, die Netzwerkoperationen stellvertretend für den Prozess ausführen, und auf APIs, die es der Erweiterung erlauben würden, Eingabedaten aufzuzeichnen. Entwickler eigener Tastaturen können anfordern, dass ihre Erweiterung Open Access erhält, sodass das System nach erfolgter Bestätigung durch den Benutzer die Erweiterung in der standardmäßigen Sandbox ausführen kann.

MDM und Erweiterungen

Bei Geräten, die in einer Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM) registriert sind, unterliegen Dokument- und Tastaturerweiterungen der Regel „Verwaltetes Öffnen“ (Managed Open In). Die MDM-Lösung kann beispielsweise verhindern, dass Benutzer ein Dokument aus einer verwalteten App in einen nicht verwalteten Document Provider exportieren oder eine nicht verwaltete Tastatur in einer verwalteten App verwenden. Außerdem können Entwickler festlegen, dass in ihrer App keine Tastaturerweiterungen anderer Anbieter verwendet werden dürfen.

App-Schutz und App-Gruppen in iOS und iPadOS

Bei iOS und iPadOS kann der Schutz für Apps mithilfe des Software Development Kits (SDK) für das jeweilige Betriebssystem und den Beitritt zu einer App-Gruppe im Apple-Entwicklerportal bereitgestellt werden.

Übernahme der Datensicherheit in Apps

Das Software Development Kit (SDK) für iOS und iPadOS bietet ein API-Komplettpaket, mit dem externe und firmeninterne Entwickler Mechanismen für die Datensicherheit ganz einfach übernehmen können, um für ein hohes Maß an Sicherheit in ihren Apps zu sorgen. Diese Datensicherheit ist für Datei- und Datenbank-APIs verfügbar (z. B. für `NSFileManager`, `CoreData`, `NSData` oder `SQLite`).

Die Datenbank der App „Mail“ (inklusive Anhänge), verwaltete Bücher, Safari-Lesezeichen, App-Start-Images und Ortsdaten werden ebenfalls in verschlüsselter Form gesichert. Hierfür werden Schlüssel verwendet, die mit dem Code des Benutzers für das jeweilige Gerät geschützt werden. Kalender (ohne Anhänge), Kontakte, Erinnerungen, Notizen, Nachrichten und Fotos verwenden die Datensicherheitsberechtigung „Geschützt bis zur ersten Benutzerauthentifizierung“.

Vom Benutzer installierte Apps, die keine bestimmte Datensicherheitsklasse besitzen, werden standardmäßig der Klasse „Geschützt bis zur ersten Benutzerauthentifizierung“ zugeordnet.

Beitritt zu einer App-Gruppe

Apps und Erweiterungen eines bestimmten Entwickleraccounts können Inhalte gemeinsam verwenden, wenn sie als Teil einer App-Gruppe konfiguriert sind. Der Entwickler muss die entsprechenden Gruppen im Apple-Entwicklerportal erstellen und die gewünschte Gruppe mit Apps und Erweiterungen hinzufügen. Apps, die einer App-Gruppe zugewiesen wurden, haben Zugriff auf Folgendes:

- Einen geteilten Datenträgercontainer als Speicher, der auf dem Gerät bleibt, solange mindestens eine App aus der Gruppe installiert ist
- Geteilte Einstellungen
- Geteilte Schlüsselbundobjekte

Das Apple-Entwicklerportal gewährleistet, dass Gruppen-IDs (GIDs) für Apps im gesamten Ökosystem eindeutig sind.

Verifizieren von Zubehörgeräten in iOS und iPadOS

Das Lizenzprogramm „Made for iPhone, iPad and iPod touch“ (MFi) gewährt autorisierten Zubehörherstellern den Zugriff auf das „iPod Accessories Protocol“ (iAP) und die notwendigen unterstützenden Hardwarekomponenten.

Wenn ein MFi-Zubehör über einen Lightning- oder USB-C-Anschluss oder über Bluetooth eine Verbindung zu einem iOS- oder iPadOS-Gerät herstellen will, muss das Zubehör nachweisen, dass es von Apple autorisiert wurde, indem es mit einem Zertifikat von Apple antwortet, das vom Gerät überprüft wird. Das Gerät sendet anschließend eine Anfrage, auf die das Zubehör eine signierte Antwort senden muss. Dieses Verfahren wird vollständig von einem spezifischen integrierten Schaltkreis (Integrated Circuit, IC) durchgeführt, den Apple autorisierten Zubehörherstellern zur Verfügung stellt, und ist für das Zubehör selbst transparent.

Zubehör kann Zugriff auf unterschiedliche Übertragungsarten und Funktionen anfordern, zum Beispiel Zugriff auf digitale Audiostreams über das Lightning- oder USB-C-Kabel oder Standortdaten über Bluetooth. Ein IC (integrierter Schaltkreis) für die Authentifizierung sorgt dafür, dass nur genehmigtes Zubehör vollständigen Zugriff auf das Gerät erhält. Wenn ein Zubehörgerät keine Authentifizierung unterstützt, erhält es nur Zugriff auf analoge Audiosignale und in begrenztem Umfang auf die serielle Audiowiedergabesteuerung (UART).

AirPlay verwendet ebenfalls den IC für die Authentifizierung, um zu überprüfen, ob Empfänger von Apple zugelassen wurden. AirPlay-Audiostreams und CarPlay-Videostreams nutzen das Sicherheitsverbindungsprotokoll MFi-SAP (Secure Association Protocol), bei dem die Kommunikation zwischen dem Zubehör und dem Gerät mit AES-128 im CTR-Modus (Zählermodus) verschlüsselt wird. Temporäre Schlüssel werden per ECDH-Schlüsselaustausch (Curve25519) ausgetauscht und mit dem 1024-Bit RSA-Schlüssel des ICs für die Authentifizierung als Teil des Station-to-Station-Protokolls signiert.

Sicherheit bei Apps für macOS

Sicherheit bei Apps für macOS – Einführung

Die Sicherheitsarchitektur bei Apps für macOS umfasst eine Reihe sich überlagernder Ebenen. Die erste Ebene stellt die Option dar, derzufolge nur signierte und vertrauenswürdige Apps aus dem App Store ausgeführt werden dürfen. Darüber hinaus nutzt macOS auf mehreren Ebenen angelegte Schutzmechanismen, um sicherzustellen, dass aus dem Internet geladene Apps frei von bekannten Schadprogrammen sind. Außerdem umfasst das Betriebssystem Technologien, mit denen derartige Malware erkannt und entfernt wird, und weitere Schutzmechanismen, die verhindern, dass nicht vertrauenswürdige Apps auf Benutzerdaten zugreifen. Apple-Dienste wie die Beglaubigung und XProtect-Aktualisierungen sind darauf ausgelegt, die Installation von Schadprogrammen zu verhindern. Wenn es erforderlich ist, lokalisieren diese Dienste die zunächst unerkannten Schadprogramme und entfernen sie schnell und effizient. Schließlich und endlich können macOS-Benutzer frei im Rahmen des Sicherheitsmodells agieren, das für sie am sinnvollsten ist – was ihnen im Extremfall auch die Möglichkeit lässt, Code auszuführen, der nicht signiert und nicht vertrauenswürdig ist.

Prozess der App-Codesignierung in macOS

Alle Apps aus dem App Store sind von Apple signiert. Diese Signatur stellt sicher, dass die Apps nicht manipuliert oder geändert wurden. Apple signiert jede App, die mit einem Apple-Gerät bereitgestellt wird.

Unter macOS 10.15 müssen alle Apps, die nicht über den App Store vertrieben werden, vom jeweiligen Entwickler mithilfe eines von Apple (mit einem privaten Schlüssel) ausgestellten und mit der Entwickler-ID versehenen Zertifikats signiert und von Apple beglaubigt werden, damit sie im Rahmen und im Kontext der standardmäßigen Gatekeeper-Einstellungen ausgeführt werden können. Firmenintern entwickelte Apps müssen ebenfalls mithilfe einer von Apple ausgegebenen Entwickler-ID signiert werden, damit Benutzer deren Integrität verifizieren können.

Bei macOS sind die Codesignierung und die Beglaubigung zwei Vorgänge, die unabhängig voneinander für verschiedene Zwecke – und durch verschiedene Akteure – erfolgen können. Die Codesignierung erfolgt durch den Entwickler unter Verwendung des (von Apple ausgestellten) Zertifikats mit seiner Entwickler-ID. Die Verifizierung dieser Signatur liefert dem Benutzer den Beweis, dass die Software eines Entwicklers weder manipuliert noch geändert wurde, nachdem sie vom Entwickler final generiert und signiert wurde. Die Beglaubigung kann von jeder Person durchgeführt werden, die ein Glied in der Kette der Softwareverteilung ist. Sie dient als Nachweis dafür, dass Apple eine Kopie des Codes erhalten hat, dass Apple diese Kopie auf Schadsoftware geprüft hat und dass bei der Prüfung keine Schadsoftware gefunden wurde. Das Endergebnis der Beglaubigung ist ein Ticket, das auf Apple-Servern gespeichert wird und optional (von einer beliebigen Person) an die App angeheftet werden kann, ohne dass die Signierung durch den Entwickler beeinträchtigt oder ungültig wird.

MAC (Mandatory Access Controls) setzt die Codesignierung zwingend voraus, damit vom System geschützte Berechtigungen aktiviert werden. Beispielsweise muss der Code von Apps, die Zugang durch die Firewall benötigen, mit der entsprechenden MAC-Berechtigung signiert sein.

Gatekeeper und Laufzeitsicherheit bei macOS

Mithilfe der Gatekeeper-Technologie und der Laufzeitsicherheit stellt macOS sicher, dass auf dem Mac eines Benutzers nur vertrauenswürdige Apps ausgeführt werden.

Gatekeeper

macOS enthält eine *Gatekeeper* genannte Sicherheitstechnologie, die dabei hilft, sicherzustellen, dass auf dem Mac eines Benutzers nur vertrauenswürdige Software ausgeführt werden kann. Wenn ein Benutzer eine App, ein Plug-in oder ein Installationspaket, die/das nicht aus dem App Store stammt, lädt und öffnet, stellt Gatekeeper sicher, dass die Software von einem identifizierten Entwickler stammt, dass die Beglaubigung durch Apple vorliegt, dass die Software frei von bekannten Schadprogrammen ist und dass die Software nicht verändert wurde. Außerdem holt Gatekeeper die Einwilligung durch den Benutzer ein, bevor eine heruntergeladene Software erstmals geöffnet wird, um sicherzustellen, dass der Benutzer den ausführbaren Code nicht in der irrigen Annahme ausführt, dass es sich dabei nur um eine einfache Datendatei handelt.

Standardmäßig prüft Gatekeeper, dass jegliche geladene Software aus dem App Store oder von einem registrierten Entwickler signiert und von Apple beglaubigt wurde. Durch den Prüfprozess des App Store wird in Kombination mit der Pipeline für die Beglaubigung gewährleistet, dass Apps keine bekannte Schadsoftware enthalten. Aus diesem Grund wird standardmäßig *unter macOS jegliche Software beim erstmaligen Öffnen auf böartigen oder schädlichen Inhalt überprüft – unabhängig davon, auf welchem Weg sie auf den Mac gelangte*.

Benutzer und Organisationen haben die Möglichkeit, nur solche Software zuzulassen, die aus dem App Store bezogen und installiert wird. Alternativ können Benutzer die Richtlinien von Gatekeeper außer Kraft setzen, jede Software zu öffnen – außer im Fall einer Be- oder Einschränkung durch eine Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM). Organisationen können mithilfe einer MDM-Lösung die Einstellungen für Gatekeeper konfigurieren und u. a. Software zulassen, die mit Alternatividentitäten signiert werden. Bei Bedarf kann Gatekeeper auch gänzlich deaktiviert werden.

Gatekeeper bietet außerdem Schutz davor, dass über eine gutartige App Plug-ins mit böartigem oder schädlichem Code vertrieben werden. Das bedeutet, dass eine App ohne Wissen des Benutzers das Laden eines böartigen oder schädlichen Plug-ins veranlasst. Gatekeeper öffnet (sofern erforderlich) Apps in zufällig ausgesuchten Speicherorten mit reinem Lesezugriff. Damit soll das automatische Laden von Plug-ins unterbunden werden, die zusammen mit der App vertrieben wurden.

Laufzeitsicherheit

Systemdateien, Ressourcen und der Kernel sind vom App-Bereich des Nutzers getrennt und abgeschirmt. Alle Apps aus dem App Store werden in einer Sandbox ausgeführt, um den Zugriff auf Daten zu verhindern, die durch andere Apps gespeichert werden. Wenn eine aus dem App Store stammende App Zugriff auf die Daten einer anderen App benötigt, ist ihr dieser Zugriff nur mithilfe von APIs und Diensten möglich, die durch macOS bereitgestellt werden.

Schutz vor Malware bei macOS

Apple betreibt einen Prozess für die Bedrohungsanalyse, der Malware schnell erkennt und blockiert.

Die drei Abwehrstufen

Die Abwehr von Malware ist in drei Stufen strukturiert:

1. *Verhinderung des Startens oder der Ausführung von Malware:* App Store oder Gatekeeper in Kombination mit Beglaubigung
2. *Blockieren der Ausführung von Malware auf Kundensystemen:* Gatekeeper, Beglaubigung und XProtect
3. *Entfernung und Bereinigung von ausgeführter Malware:* XProtect

Die erste Abwehrstufe ist darauf ausgelegt, die Verteilung von Schadprogrammen zu unterbinden und selbst ihre einmalige Ausführung zu verhindern – dieses Ziel haben der App Store sowie Gatekeeper in Kombination mit der Beglaubigung.

Die nächste Abwehrstufe soll sicherstellen, dass Malware beim Auftreten auf einem Mac schnell erkannt und blockiert wird – sowohl um ihre Ausbreitung zu stoppen als auch um alle Mac-Systeme zu bereinigen, auf denen sie bereits Fuß gefasst hat. XProtect ergänzt diese Abwehrstufe zusätzlich zu Gatekeeper und zur Beglaubigung.

Schließlich entfernt und bereinigt XProtect Schadprogramme, die ausgeführt werden konnten.

Zusammen tragen diese Abwehrmechanismen, die unten detaillierter beschrieben werden, zum optimalen Schutz vor Viren und Schadprogrammen bei. Darüber hinaus stehen (insbesondere auf einem Mac mit Apple Chips) weitere Abwehrmechanismen zur Verfügung, die den potenziellen Schaden von ausgeführter Malware begrenzen. Der Abschnitt „[App-Zugriff auf Benutzerdaten schützen](#)“ enthält Informationen dazu, wie macOS dazu beitragen kann, Benutzerdaten vor Malware zu schützen. Der Abschnitt [Integrität des Betriebssystems](#) enthält Informationen dazu, wie macOS die Aktionen einschränken kann, die Malware auf einem System ausführen kann.

Beglaubigung

Die *Beglaubigung* ist ein von Apple bereitgestellter Malware-Überprüfungsdienst. Entwickler, die Apps für macOS außerhalb des App Store bereitstellen möchten, müssen ihre Apps im Rahmen des Bereitstellungsprozesses zur Überprüfung einreichen. Apple überprüft diese Software auf bekannte Malware und stellt ein Beglaubigungsticket aus, sofern keine Malware entdeckt wird. Üblicherweise heften Entwickler dieses Ticket an ihre App an, sodass Gatekeeper sie verifizieren und starten kann – auch offline.

Außerdem kann Apple ein Sperrungsticket für Apps ausstellen, die als schädlich bekannt sind – auch dann, wenn sie zu einem früheren Zeitpunkt bereits einmal beglaubigt wurden. macOS überprüft regelmäßig, ob neue Sperrungstickets vorliegen, sodass Gatekeeper über die neuesten Informationen verfügt und den Start solcher Dateien blockieren kann. Durch diesen Prozess können schädliche Apps schnell blockiert werden, da die Hintergrundupdates sehr viel öfter erfolgen als jene zur Bereitstellung neuer XProtect-Signaturen. Darüber hinaus kann dieser Schutz sowohl auf Apps, die bereits beglaubigt wurden, als auch auf solche Apps angewendet werden, bei denen noch keine Beglaubigung erfolgte.

XProtect

macOS enthält eine integrierte Antivirustechnologie namens *XProtect* für die signaturbasierte Erkennung und Entfernung von Schadprogrammen. Das System verwendet YARA-Signaturen – ein Werkzeug für die signaturbasierte Erkennung von Malware, das von Apple regelmäßig aktualisiert wird. Apple überwacht Malware-Neuinfizierungen und -Verläufe und aktualisiert die Signaturen automatisch und unabhängig von System-Updates, um einen Mac vor Malware-Infizierungen zu schützen. XProtect erkennt und blockiert die Installation unbekannter Malware automatisch. Unter macOS 10.15 (oder neueren Versionen) führt XProtect immer dann eine Prüfung auf bekannte böartige und schädliche Inhalte durch, wenn:

- eine App zum ersten Mal gestartet wird,
- eine App (im Dateisystem) verändert wurde,
- XProtect-Signaturen aktualisiert werden.

Wenn XProtect bekannte Malware erkennt, wird die Software blockiert. Gleichzeitig wird der Benutzer benachrichtigt und ihm die Möglichkeit geboten, die Software in den Papierkorb zu legen.

Hinweis: Die Beglaubigung schützt effektiv vor bekannten Dateien (oder Datei-Hashes) und kann auf Apps angewendet werden, die bereits zu einem früheren Zeitpunkt geöffnet wurden. Die signaturbasierten Regeln von XProtect sind allgemeiner als bestimmte Datei-Hashes gefasst und können so auch Varianten entdecken, die Apple noch nicht bekannt sind. XProtect scannt nur Apps, die entweder geändert wurden oder das erste Mal gestartet werden.

Sollten sich tatsächlich einmal Schadprogramme auf einem Mac einschleichen, bietet XProtect auch die Technologie, um derartige Infizierungen zu beheben und zu bereinigen. Das Tool enthält eine Engine, die Infektionen basierend auf Updates, die automatisch von Apple bereitgestellt werden, bereinigt. (Diese Updates sind Teil der automatischen Updates von Systemdatendateien und Sicherheitsupdates.) Es entfernt Schadprogramme nach dem Empfang aktualisierter Informationen und prüft regelmäßig auf Infektionen. XProtect startet den Mac nicht automatisch neu.

Automatische XProtect-Sicherheitsupdates

Apple gibt die Updates für XProtect automatisch heraus – basierend auf den neuesten Informationen über aktuelle Bedrohungsszenarien. Standardmäßig prüft macOS täglich, ob derartige Updates verfügbar sind. Aktualisierungen für die Beglaubigung werden über die CloudKit-Synchronisierung bereitgestellt und erfolgen sehr viel öfter.

Reaktion von Apple auf die Entdeckung neuer Schadprogramme

Wenn neue Malware entdeckt wird, kann eine Reihe von Schritten ausgeführt werden:

- Die zugehörigen Zertifikate der Entwickler-ID werden widerrufen.
- Die Beglaubigung stellt die Sperrungstickets für alle Dateien (Apps und verknüpfte Dateien) aus.
- XProtect-Signaturen werden entwickelt und veröffentlicht.

Außerdem werden diese Signaturen rückwirkend auf bereits beglaubigte Software angewendet. Alle Neuentdeckungen von Malware können zu einer oder mehreren bereits genannten Maßnahmen führen.

Schließlich leitet die Malwareerkennung im Laufe der folgenden Sekunden, Stunden und Tage eine Reihe von Schritten ein, um die bestmöglichen Schutzmaßnahmen für Mac-Benutzer zu ergreifen.

Steuerung des Dateizugriffs durch Apps bei macOS

Apple vertritt die Ansicht, dass ein Benutzer volle Transparenz darüber haben muss, was Apps mit seinen Daten anstellen, den Vorgängen zustimmen muss und sie kontrollieren können muss. Unter macOS 10.15 wird dieses Modell systemseitig erzwungen, weshalb alle Apps die Zustimmung des Benutzers einholen müssen, wenn sie auf Dateien im Ordner „Dokumente“, im Ordner „Downloads“, auf dem Schreibtisch, in iCloud Drive oder auf einem Volume im Netzwerk zugreifen möchten. Apps, die den Zugriff auf das gesamte Speichergerät benötigen, müssen unter macOS 10.13 (und neueren Versionen) explizit in den Systemeinstellungen hinzugefügt werden. Darüber hinaus benötigen Funktionen für die Bedienungshilfen und die Automatisierung die Einwilligung des Benutzers, damit sichergestellt ist, dass keine anderen Schutzvorkehrungen umgangen werden. Abhängig von der für den Zugriff geltenden Richtlinie werden Benutzer zur Änderung der folgenden Einstellung entweder aufgefordert oder gezwungen: „Systemeinstellungen“ > „Privatsphäre & Sicherheit“ > „Sicherheit“:

Objekt	Benutzer wird durch App aufgefordert	Benutzer muss die Einstellungen zu Datenschutz und Datensicherheit ändern
Bedienungshilfen		✓
Vollzugriff auf internen Speicher		✓
Dateien und Ordner <i>Hinweis:</i> Umfasst Schreibtisch, Ordner „Dokumente“, Ordner „Downloads“, Volumes im Netzwerk und Wechseldatenträger	✓	
Automatisierung (Apple-Ereignisse)	✓	

Objekte, die sich im Papierkorb des Benutzers befinden, sind vor dem Zugriff durch jede App geschützt, auch wenn diese Vollzugriff auf den internen Speicher hat. Der Benutzer erhält keine Rückfrage bezüglich des Zugriffs durch die App. Wenn der Benutzer möchte, dass Apps auf die Dateien zugreifen können, muss er sie aus dem Papierkorb an einen anderen Speicherort bewegen.

Ein Benutzer, der FileVault auf einem Mac aktiviert, wird zur Eingabe seiner gültigen Anmeldeinformationen aufgefordert, bevor der Boot-Prozess fortgesetzt wird und er Zugriff auf spezialisierte Modi beim Startvorgang erhält. Ohne gültige Anmeldeinformationen oder einen Wiederherstellungsschlüssel bleibt das gesamte Volume verschlüsselt und vor unbefugtem Zugriff geschützt, selbst wenn das physische Speichergerät entfernt und an einen anderen Computer angeschlossen wird.

Um den Schutz in einer Unternehmensumgebung zu gewährleisten, sollten die IT-Verantwortlichen mithilfe einer Lösung für die Mobilgeräteverwaltung ([Mobile Device Management, MDM](#)) FileVault-Konfigurationsrichtlinien definieren und deren Einhaltung erzwingen. Organisationen haben mehrere Möglichkeiten, um verschlüsselte Volumes zu verwalten – zum Beispiel institutionelle Wiederherstellungsschlüssel, persönliche Wiederherstellungsschlüssel (die optional treuhänderisch in der MDM-Lösung gespeichert werden können) oder eine Kombination der beiden Varianten. Auch die Schlüsselrotation kann in der MDM-Lösung als Richtlinie konfiguriert werden.

Sicherheitsfunktionen in der App „Notizen“

Auf iPhone-, iPad- und Mac-Geräten sowie auf der iCloud-Website unterstützt die App „Notizen“ eine Funktion für geschützte Notizen, mit deren Hilfe Benutzer den Inhalt bestimmter Notizen schützen können. Benutzer können ebenfalls Notizen mit anderen Personen sicher teilen.

Geschützte Notizen

Geschützte Notizen werden mit einem vom Benutzer bereitgestellten Passwort Ende-zu-Ende-verschlüsselt; dieses muss zum Anzeigen der Notizen auf iOS-, iPadOS- und macOS-Geräten und auf der iCloud-Website eingegeben werden. Jeder iCloud-Account (auch Accounts vom Typ „Auf meinem Gerät“) kann ein separates Passwort (Passphrase) haben.

Beim Schützen einer Notiz wird aus dem Passwort des Benutzers mittels PBKDF2 und SHA-256 ein 16-Byte-Schlüssel abgeleitet. Die Notiz wird mitsamt ihren Anhängen unter Verwendung von AES im Galois/Counter Mode (AES-GCM) verschlüsselt. In Core Data und CloudKit werden neue Datensätze erstellt, in denen die verschlüsselte Notiz, ihre Anhänge, ihr Tag und ihr Initialisierungsvektor gespeichert werden. Nachdem die neuen Datensätze erstellt wurden, werden die unverschlüsselten Originaldaten gelöscht. Zu den Anhängen, die verschlüsselt werden können, gehören Bilder, Skizzen, Tabellen, Karten und Websites. Notizen mit anderen Arten von Anhängen können nicht verschlüsselt werden. Es ist außerdem nicht möglich, zu geschützten Notizen Anhangsformate hinzuzufügen, die nicht unterstützt werden.

Damit ein Benutzer eine geschützte Notiz anzeigen kann, muss er deren Passwort (Passphrase) eingeben oder sich per Face ID oder Touch ID authentifizieren. Nachdem sich der Benutzer erfolgreich authentifiziert hat, öffnet die App „Notizen“ eine sichere Sitzung, um eine geschützte Notiz anzuzeigen oder zu erstellen. Solange die sichere Sitzung geöffnet bleibt, kann der Benutzer weitere Notizen öffnen und sichern, ohne dass er sich erneut authentifizieren muss. Die sichere Sitzung gilt allerdings nur für Notizen, die mit dem eingegebenen Passwort geschützt sind. Für Notizen, die durch ein anderes Passwort geschützt sind, muss sich der Benutzer erneut authentifizieren. Die sichere Sitzung wird in folgenden Situationen geschlossen:

- Der Benutzer tippt in der App „Notizen“ auf die Taste „Jetzt sperren“.
- Die App „Notizen“ verbleibt länger als 3 Minuten im Hintergrund (8 Minuten bei macOS).
- Das iOS- oder iPadOS-Gerät wird gesperrt.

Zum Ändern des Passworts einer geschützten Notiz muss der Benutzer deren aktuelles Passwort eingeben. Beim Ändern des Passworts stehen Face ID und Touch ID nicht zur Verfügung. Nachdem das neue Passwort festgelegt wurde, leitet die App „Notizen“ die Neuverpackung der Schlüssel aller vorhandenen Notizen im gleichen Account ein, die mit dem bisherigen Passwort verschlüsselt waren.

Wenn ein Benutzer das Passwort drei Mal nacheinander falsch eingibt, blendet die App „Notizen“ eine vom Benutzer vordefinierte Merkhilfe ein, sofern der Benutzer bei der Konfiguration eine solche Merkhilfe erfasst hat. Wenn sich der Benutzer trotz Merkhilfe nicht an das Passwort erinnern kann, hat er die Möglichkeit, die Einstellungen der App „Notizen“ zurückzusetzen. Der Benutzer wird dadurch in die Lage versetzt, neue geschützte Notizen unter Verwendung eines neuen Passworts zu erstellen, er kann aber keine der zuvor geschützten Notizen anzeigen. Zuvor geschützte Notizen können weiterhin angezeigt werden, wenn sich der Benutzer wieder an das alte Passwort erinnert. Zum Zurücksetzen des Passworts ist das Passwort für den iCloud-Account des Benutzers erforderlich.

Geteilte Notizen

Notizen, die nicht mit einem Passwort Ende-zu-Ende-verschlüsselt sind, können mit anderen Personen geteilt werden. Geteilte Notizen verwenden weiterhin den mit CloudKit verschlüsselten Datentyp für Texte oder Anhänge, die der Benutzer einer Notiz hinzufügt. Medien werden immer mit einem Schlüssel aus dem Datensatz [CKRecord](#) verschlüsselt. Metadaten wie Erstellungs- und Änderungsdatum werden nicht verschlüsselt. CloudKit verwaltet den Prozess, durch den Teilnehmer die Daten der jeweils anderen Teilnehmer ver- und entschlüsseln können.

Sicherheitsfunktionen in der App „Kurzbefehle“

Kurzbefehle in der App „Kurzbefehle“ können optional via iCloud auf Apple-Geräten synchronisiert werden. Kurzbefehle können auch via iCloud mit anderen Benutzern geteilt werden. Kurzbefehle werden lokal in verschlüsselter Form gespeichert.

Eigene Kurzbefehle sind vielseitig und haben Ähnlichkeit mit Skripts oder Programmen. Beim Herunterladen eines Kurzbefehls aus dem Internet wird der Benutzer gewarnt, dass der Kurzbefehl nicht von Apple überprüft wurde. Er erhält dann die Möglichkeit, den Kurzbefehl zu inspizieren. Zum Schutz vor Kurzbefehlen mit böswilligem oder schädlichem Inhalt werden aktualisierte Malware-Definitionen heruntergeladen, mit denen schädliche Kurzbefehle zum Zeitpunkt der Ausführung identifiziert werden.

Beim Aufruf über die Freigabeseite („Share Sheet“) können eigene Kurzbefehle verwendet werden, um in Safari auf Websites vom Benutzer angegebenen JavaScript-Code auszuführen. Zum Schutz vor bösartigem JavaScript-Code (der zum Beispiel den Benutzer verleitet, ein Skript auf einer Social Media-Website auszuführen, um seine Daten abzugreifen), wird der JavaScript-Code ebenfalls mit den oben erwähnten Malware-Definitionen abgeglichen. Wenn der Benutzer erstmals JavaScript-Code für eine Domain ausführt, wird er aufgefordert, Kurzbefehlen, die JavaScript-Code enthalten, die Ausführung auf der aktuellen Webseite für diese Domain zu erlauben.

Sicherheit bei Diensten

Sicherheit bei Diensten – Übersicht

Apple hat eine zuverlässige Reihe von Diensten eingerichtet, die es den Benutzern ermöglichen, ihre Geräte noch effizienter und produktiver einzusetzen. Diese Dienste bieten leistungsfähige Funktionen u. a. für Cloud-Speicher, Synchronisierung, Passwortsicherung, Authentifizierung, Zahlung, Messaging und Kommunikation, wobei Schutz und Sicherheit der Benutzerdaten immer im Vordergrund stehen.

Dieses Kapitel enthält Informationen über die Sicherheitstechnologien, die bei iCloud, „Mit Apple anmelden“, Apple Pay, iMessage, Apple Messages for Business, FaceTime, „Wo ist?“ und „Integration“ verwendet werden.

Hinweis: Nicht alle Dienste und Inhalte von Apple sind in allen Ländern und Regionen verfügbar.

Apple-ID und verwaltete Apple-ID

Sicherheit bei Apple ID – Übersicht

Eine Apple-ID ist der Account, der verwendet wird, um sich bei Apple-Diensten anzumelden. Es ist wichtig, dass Benutzer ihre Apple-IDs sicher aufbewahren, um einen unbefugten Zugriff auf den Account zu verhindern. Außerdem ist es wichtig zu wissen, dass die Apple-IDs starke Passwörter voraussetzen, die:

- mindestens achtstellig sein müssen,
- sowohl Buchstaben als auch Zahlen enthalten müssen,
- nicht mehr als drei (oder mehr) identische Zeichen hintereinander aufweisen dürfen,
- nicht mit einem häufig genutzten Passwort übereinstimmen dürfen.

Es wird Benutzern nachdrücklich empfohlen, über diese Richtlinien hinaus weitere Zeichen und Interpunktionszeichen hinzuzufügen, um ein noch sichereres Passwort zu erhalten.

Apple sendet auch E-Mails, Push-Benachrichtigungen oder beides an Benutzer, wenn wichtige Änderungen am Account durchgeführt werden, z. B. wenn Passwort oder Rechnungsdaten geändert wurden oder die Apple-ID auf einem neuen Gerät für die Anmeldung verwendet wurde. Wenn Benutzern etwas verdächtig erscheint, sollten sie sofort das Passwort für ihre Apple-ID ändern.

Apple nutzt darüber hinaus eine Vielzahl von Richtlinien und Verfahren, die darauf abzielen, Benutzeraccounts zu schützen. Dazu gehören die Beschränkung der Eingabeversuche beim Anmelden oder Zurücksetzen des Passworts, eine aktive Betrugsüberwachung, um Angriffe sofort beim Auftreten zu entdecken, sowie regelmäßige Überprüfungen der Richtlinien, die es Apple ermöglichen, diese an neue Informationen anzupassen, die Auswirkungen auf die Sicherheit der Benutzer haben könnten.

Hinweis: Die Passwortrichtlinie für die verwaltete Apple-ID wird von einem Administrator in [Apple School Manager](#) oder [Apple Business Manager](#) definiert.

Zwei-Faktor-Authentifizierung

Um Benutzern dabei zu helfen, ihre Accounts noch besser zu sichern, verwendet Apple standardmäßig eine *Zwei-Faktor-Authentifizierung* als weitere Sicherheitsstufe für Apple-IDs. Auf diese Weise soll sichergestellt werden, dass nur der Accountinhaber auf den Account zugreifen kann, auch wenn eine andere Person das Passwort kennt. Bei der Zwei-Faktor-Authentifizierung kann nur mit vertrauenswürdigen Geräten auf den Account eines Benutzers zugegriffen werden, etwa mit dessen iPhone, iPad, iPod touch oder Mac oder mit anderen Geräten, nachdem von einem dieser vertrauenswürdigen Geräte oder einer vertrauenswürdigen Telefonnummer eine Überprüfung durchgeführt wurde. Für die erste Anmeldung auf einem neuen Gerät sind zwei Informationen erforderlich: zum einen das Passwort der Apple-ID und zum anderen ein sechsstelliger Bestätigungscode, der auf den vertrauenswürdigen Geräten des Benutzers angezeigt oder an eine vertrauenswürdige Telefonnummer gesendet wird. Durch die Eingabe des Codes bestätigt der Benutzer, dass er diesem neuen Gerät vertraut und dass die Anmeldung sicher ist. Da ein Passwort allein nicht mehr ausreicht, um auf einen Benutzeraccount zuzugreifen, trägt die Zwei-Faktor-Authentifizierung dazu bei, die Sicherheit der Apple-ID des Benutzers und aller seiner bei Apple gespeicherten persönlichen Daten zu verbessern. Die Funktion ist direkt in iOS, iPadOS, macOS, tvOS, watchOS und in die auf Apple-Websites verwendeten Authentifizierungssysteme integriert.

Meldet sich ein Benutzer mit einem Webbrowser bei einer Apple-Website an, wird als zweite Stufe für die Zwei-Faktor-Authentifizierung eine Anfrage an alle vertrauenswürdigen Geräte gesendet, die dem iCloud-Account des Benutzers zugeordnet sind, um die Einwilligung zur Websitzung einzuholen. Wenn sich der Benutzer auf einem vertrauenswürdigen Gerät mit einem Browser bei einer Apple-Website anmeldet, wird der Verifizierungscode lokal auf dem von ihm verwendeten Gerät angezeigt. Wenn der Benutzer den Code auf diesem Gerät eingibt, wird die Websitzung genehmigt.

Passwort zurücksetzen und Accountwiederherstellung

Wenn Benutzer das Passwort für einen Apple-ID-Account vergessen, können sie es auf einem vertrauenswürdigen Gerät zurücksetzen. Wenn kein vertrauenswürdigenes Gerät verfügbar, das Passwort jedoch bekannt ist, können Benutzer über die SMS-Verifizierung auch eine vertrauenswürdige Telefonnummer für die Authentifizierung verwenden. Darüber hinaus kann für die sofortige Wiederherstellung einer Apple-ID ein zuvor verwendeter Code verwendet werden, um sie in Verbindung mit einer SMS zurückzusetzen. Können diese Optionen nicht verwendet werden, muss der Prozess für die Accountwiederherstellung ausgeführt werden. Weitere Informationen findest du im Apple Support-Artikel [So verwendest du die Accountwiederherstellung, wenn du dein Apple-ID-Passwort nicht zurücksetzen kannst](#).

Sicherheit bei verwalteter Apple ID

Verwaltete Apple-IDs funktionieren ähnlich wie eine Apple-ID, befinden sich aber im Besitz eines Unternehmens oder einer Bildungseinrichtung und unterliegen dessen bzw. deren Kontrolle. Die Organisationen können Passwörter zurücksetzen, die Kaufmöglichkeiten und Kommunikationsfunktionen wie FaceTime und Nachrichten einschränken und rollenbasierte Zugriffsrechte für Angestellte, Mitarbeiter, Lehrkräfte und Schüler/Studierende festlegen.

Für verwaltete Apple-IDs werden einige Dienste deaktiviert (beispielsweise Apple Pay, iCloud-Schlüsselbund, HomeKit und „Wo ist?“).

Verwaltete Apple-IDs überprüfen

Verwaltete Apple-IDs unterstützen darüber hinaus *Überprüfungen*, sodass Organisationen die Einhaltung gesetzlicher Vorschriften und Datenschutzregelungen durchsetzen können. Ein Administrator von Apple School Manager, ein Manager oder eine Lehrkraft kann bestimmte verwaltete Apple-ID-Accounts überprüfen.

Die Prüfer können nur Accounts überwachen, die in der Hierarchie der Organisation unter ihnen stehen. Das heißt, Lehrkräfte können Schüler/Studierende überwachen, Manager können Lehrkräfte und Schüler/Studierende überprüfen und Administratoren können wiederum Manager, Lehrkräfte und Schüler/Studierende überprüfen.

Wenn mithilfe von Apple School Manager Überprüfungsanmeldedaten angefordert werden, wird ein spezieller Account angelegt, der nur auf die verwaltete Apple-ID zugreifen kann, für die eine Überprüfung angefragt wurde. Der Prüfer kann dann die in iCloud- oder in CloudKit-fähigen Apps gespeicherten Inhalte des Benutzers lesen und Änderungen daran vornehmen. Alle Anfragen für einen solchen Überprüfungszugriff werden in Apple School Manager protokolliert. Die Protokolle zeigen den Namen des Prüfers, die verwaltete Apple-ID, auf die der Prüfer Zugriff angefordert hat, die Uhrzeit der Anfrage und die Angabe, ob die Überprüfung stattgefunden hat oder nicht.

Verwaltete Apple-IDs und persönliche Geräte

Verwaltete Apple-IDs können auch mit persönlichen iOS-Geräten und iPadOS-Geräten sowie Mac-Computern verwendet werden. Die Schüler/Studierenden melden sich dazu bei iCloud mit der von der Einrichtung ausgegebenen verwalteten Apple-ID und einem zusätzlichen privaten Passwort an, das als zweite Stufe für die Zwei-Faktor-Authentifizierung der Apple-ID dient. Wenn Schüler oder Studierende eine verwaltete Apple-ID auf einem persönlichen Gerät verwenden, ist der iCloud-Schlüsselbund nicht verfügbar; außerdem kann die Nutzung bestimmter Funktionen wie „FaceTime“ oder „Nachrichten“ eingeschränkt sein. Alle iCloud-Dokumente, die Schüler/Studierende erstellen, wenn sie angemeldet sind, können wie zuvor in diesem Abschnitt beschrieben überprüft werden.

iCloud

iCloud-Sicherheit – Übersicht

iCloud speichert Kontakte, Kalender, Fotos, Dokumente und andere Informationen für den Benutzer und synchronisiert die Daten geräteübergreifend. iCloud kann auch von Apps anderer Anbieter verwendet werden, um Dokumente und Schlüsselwerte für App-Daten in der vom Entwickler festgelegten Weise zu speichern und zu synchronisieren. Benutzer konfigurieren iCloud, indem sie sich mit einer Apple-ID anmelden und dann die Dienste auswählen, die sie verwenden wollen. Bestimmte iCloud-Funktionen, etwa iCloud Drive und iCloud-Backup können von IT-Admins mittels [MDM \(Mobile Device Management\)](#)-Konfigurationsprofilen deaktiviert werden.

iCloud nutzt strenge Sicherheitsmethoden und wendet strikte Richtlinien an, um Benutzerdaten zu schützen. Die meisten iCloud-Daten werden vor dem Upload auf die iCloud-Server zunächst auf dem Gerät des Benutzers mittels vom Gerät generierten iCloud-Schlüsseln verschlüsselt. Bei Daten ohne Ende-zu-Ende-Verschlüsselung lädt das Gerät des Benutzers diese iCloud-Schlüssel sicher in die iCloud-Hardwaresicherheitsmodule in den Rechenzentren von Apple hoch. So kann Apple den Benutzer bei der Datenwiederherstellung unterstützen und die Daten bei Bedarf für den Benutzer entschlüsseln (beispielsweise wenn sich der Benutzer bei einem neuen Gerät anmeldet, mit einem Backup eine Wiederherstellung ausführt oder auf seine iCloud-Daten im Web zugreift). Daten, die zwischen den Geräten des Benutzers und den iCloud-Servern übertragen werden, werden während der Übertragung mit TLS separat verschlüsselt, und iCloud-Server speichern Benutzerdaten mit einer zusätzlichen Verschlüsselungsebene im Ruhezustand.

Verschlüsselungsschlüssel werden, wenn sie Apple zur Verfügung stehen, in den Rechenzentren von Apple zuverlässig geschützt. Wenn Daten in Rechenzentren von Drittanbietern verarbeitet werden, greift nur die Apple-Software, die auf sicheren Servern ausgeführt wird, auf diese Verschlüsselungsschlüssel zu, und zwar nur zum Zweck der notwendigen Verarbeitung. Für noch mehr Datenschutz und zusätzliche Sicherheit verwenden viele Apple-Dienste die Ende-zu-Ende-Verschlüsselung. Dies bedeutet, dass die iCloud-Daten des Benutzers nur für den Benutzer selbst zugänglich sind, und zwar nur mit vertrauenswürdigen Geräten, auf denen sich der Benutzer mit seiner Apple-ID angemeldet hat.

Apple bietet zwei Optionen, um die in iCloud gespeicherten Daten zu verschlüsseln und zu schützen:

- **Standardverschlüsselung (die Standardeinstellung):** Die iCloud-Daten des Benutzers werden verschlüsselt, die Verschlüsselungsschlüssel werden in den Rechenzentren von Apple geschützt, und Apple kann mit Daten- und Accountwiederherstellung Hilfestellung leisten. Nur bestimmte iCloud-Daten – 14 Datenkategorien, einschließlich Gesundheitsdaten und Passwörter im iCloud-Schlüsselbund – werden mittels Ende-zu-Ende-Verschlüsselung geschützt.

- **Erweiterter Datenschutz für iCloud:** Diese optionale Einstellung bietet die höchste Stufe der iCloud-Datensicherheit von Apple. Wenn ein Benutzer den erweiterten Datenschutz aktiviert, erhalten seine vertrauenswürdigen Geräte den alleinigen Zugriff auf die Verschlüsselungsschlüssel für die Mehrzahl der iCloud-Daten, wobei diese mit Ende-zu-Ende-Verschlüsselung geschützt werden. Wenn du den erweiterten Datenschutz aktivierst, erhöht sich die Anzahl der Datenkategorien, die durch eine Ende-zu-Ende-Verschlüsselung geschützt sind, auf 23 und umfasst dein iCloud-Backup, Fotos, Notizen und mehr.

Die mit Ende-zu-Ende-Verschlüsselung geschützten Kategorien von iCloud-Daten werden im Apple Support-Artikel [Sicherheitsüberblick – iCloud-Daten](#) aufgelistet.

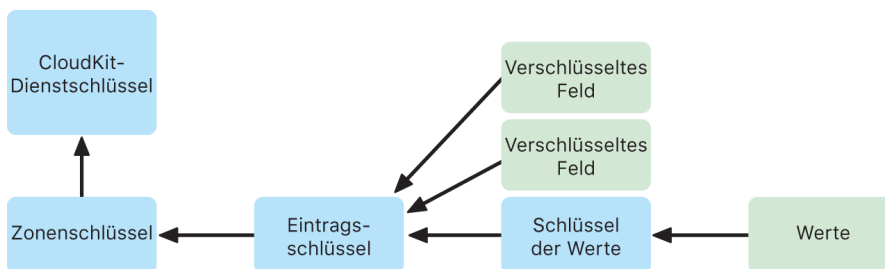
iCloud-Verschlüsselung

Die Datenverschlüsselung in iCloud ist eng mit dem Datenspeichermodell verbunden, beginnend mit den CloudKit-Frameworks und APIs, die es Apps und Systemsoftware ermöglichen, Daten im Namen des Benutzers in iCloud zu speichern und diese geräteübergreifend und im Web auf dem neuesten Stand zu halten.

CloudKit-Verschlüsselung

[CloudKit](#) ist ein Framework, das es App-Entwicklern ermöglicht, Schlüssel-Wert-Daten, strukturierte Daten und Medien (größere Datenmengen, die separat von der Datenbank gespeichert werden, etwa Bilder oder Videos) in iCloud zu speichern. CloudKit unterstützt sowohl öffentliche als auch private Datenbanken, die in Containern gruppiert werden. Öffentliche Datenbanken werden global geteilt und in der Regel für generische Medien genutzt, sie werden nicht verschlüsselt. Private Datenbanken speichern die iCloud-Daten aller Benutzer.

CloudKit nutzt eine Hierarchie von Schlüsseln, die zur Struktur der Daten passen. Die private Datenbank jedes Containers wird durch eine Schlüsselhierarchie geschützt, die in einem asymmetrischen Schlüssel hinterlegt ist, der als *CloudKit-Dienstschlüssel* bezeichnet wird. Diese Schlüssel sind für jeden iCloud-Benutzer einmalig und werden auf seinem vertrauenswürdigen Gerät generiert. Wenn Daten in das CloudKit geschrieben werden, werden alle Datensatzschlüssel auf dem vertrauenswürdigen Gerät des Benutzers generiert und vor dem Upload der Daten in die entsprechende Schlüsselhierarchie eingebunden.



Viele Apple-Dienste, die im Apple Support-Artikel [Sicherheitsüberblick – iCloud-Daten](#) genannt werden, nutzen die Ende-zu-Ende-Verschlüsselung unter Verwendung eines CloudKit-Dienstschlüssels, der per Synchronisierung mit dem iCloud-Schlüsselbund geschützt ist. Die Dienstschlüssel werden für diese CloudKit-Container im iCloud-Schlüsselbund des Benutzers gespeichert und teilen die Sicherheitsmerkmale des iCloud-Schlüsselbunds. Die Dienstschlüssel sind nur auf den als vertrauenswürdig eingestuften Geräten des Benutzers verfügbar und für Apple oder einen anderen Anbieter nicht zugänglich. Sollte ein Gerät verloren gehen, lassen sich die Daten des iCloud-Schlüsselbunds mittels [sicherer Wiederherstellung des iCloud-Schlüsselbunds](#), [Accountwiederherstellungskontakten](#) oder einem Schlüssel für die Accountwiederherstellung wiederherstellen.

Verwaltung von Verschlüsselungsschlüsseln

Die Sicherheit von verschlüsselten Daten in CloudKit hängt von der Sicherheit der entsprechenden Verschlüsselungsschlüssel ab. Es gibt zwei Kategorien von CloudKit-Dienstschlüsseln: Ende-zu-Ende-verschlüsselt und nach-Authentifizierung-verfügbar.

- **Dienstschlüssel mit Ende-zu-Ende-Verschlüsselung:** Bei Ende-zu-Ende-verschlüsselten iCloud-Diensten werden die relevanten privaten CloudKit-Dienstschlüssel niemals für die Apple-Server zur Verfügung gestellt. Dienstschlüsselpaare, einschließlich privater Schlüssel, werden lokal auf dem vertrauenswürdigen Gerät eines Benutzers erstellt und unter Verwendung der [Sicherheit des iCloud-Schlüsselbunds](#) auf die anderen Geräte des Benutzers übertragen. Obwohl die Wiederherstellungs- und Synchronisierungsabläufe des iCloud-Schlüsselbunds von Apple-Servern vermittelt werden, werden diese Server kryptografisch daran gehindert, auf die Schlüsselbunddaten des Benutzers zuzugreifen. Sollte im schlimmsten Fall kein Zugriff mehr auf den iCloud-Schlüsselbund und all seine Wiederherstellungsmechanismen bestehen, gehen die Ende-zu-Ende-verschlüsselten Daten im CloudKit verloren. Apple kann bei der Wiederherstellung dieser Daten keine Unterstützung leisten.
- **Nach-Authentifizierung-verfügbare Dienstschlüssel:** Bei anderen Diensten wie Fotos und iCloud Drive werden die Dienstschlüssel in iCloud-Hardwaresicherheitsmodulen in den Rechenzentren von Apple gespeichert und einige Apple-Dienste können darauf zugreifen. Wenn sich ein Benutzer auf einem neuen Gerät bei iCloud anmeldet und seine Apple-ID authentifiziert, sind diese Schlüssel ohne weitere Interaktion oder Eingabe des Benutzers für die Apple-Server zugänglich. Beispielsweise kann der Benutzer nach der Anmeldung bei iCloud.com sofort seine Fotos online ansehen. Diese Dienstschlüssel sind *nach-Authentifizierung-verfügbare* Schlüssel.

Erweiterter Datenschutz für iCloud

Der erweiterte Datenschutz für iCloud ist eine optionale Einstellung, die die höchste Sicherheitsstufe für Cloud-Daten von Apple bietet. Wenn ein Benutzer den erweiterten Datenschutz aktiviert, erhalten seine vertrauenswürdigen Geräte den alleinigen Zugriff auf die Verschlüsselungsschlüssel für die Mehrzahl der iCloud-Daten, wobei diese mit *Ende-zu-Ende-Verschlüsselung* geschützt werden. Wenn Benutzer den erweiterten Datenschutz aktivieren, steigt die Anzahl der Datenkategorien, die mittels Ende-zu-Ende-Verschlüsselung geschützt werden, von 14 auf 23 an und umfasst das iCloud-Backup, Fotos, Notizen und mehr.

Der erweiterte Datenschutz für iCloud wird bis Ende 2022 für Benutzer in den USA verfügbar sein und Anfang 2023 für den Rest der Welt eingeführt werden.

Das Konzept des erweiterten Datenschutzes ist einfach: Alle CloudKit-Dienstschlüssel, die auf dem Gerät generiert und später in die *nach-Authentifizierung-verfügbaren* iCloud-Hardwaresicherheitsmodule (Hardware Security Modules, HSMs) in den Apple-Rechenzentren hochgeladen wurden, werden von diesen HSMs gelöscht und verbleiben stattdessen vollständig innerhalb der iCloud-Schlüsselbund-Sicherheitsdomain des Accounts. Sie werden wie die vorhandenen *Ende-zu-Ende-verschlüsselten* Dienstschlüssel behandelt. Dies bedeutet, dass Apple diese Schlüssel nicht länger lesen und auch nicht mehr darauf zugreifen kann.

Mit dem erweiterten Datenschutz werden auch automatisch CloudKit-Felder geschützt, die Entwickler anderer Anbieter als verschlüsselt markieren, und alle anderen CloudKit-Daten.

Erweiterten Datenschutz aktivieren

Wenn ein Benutzer den erweiterten Datenschutz aktiviert, führt sein vertrauenswürdiges Gerät zwei Aktionen aus: Zuerst teilt es die Absicht des Benutzers, den erweiterten Datenschutz zu aktivieren, seinen anderen Geräten mit, die an der Ende-zu-Ende-Verschlüsselung beteiligt sind. Dazu wird ein neuer Wert, der von lokalen Schlüsseln auf dem Gerät signiert wird, in die Metadaten des iCloud-Schlüsselbund-Geräts geschrieben. Die Apple-Server können diesen Nachweis nicht entfernen oder ändern, während er mit den anderen Geräten des Benutzers synchronisiert wird.

Im zweiten Schritt initiiert das Gerät die Beseitigung der *nach-Authentifizierung-verfügbaren* Dienstschlüssel aus den Rechenzentren von Apple. Da diese Schlüssel durch iCloud HSMs geschützt sind, ist dieser Löschvorgang unmittelbar, dauerhaft und unwiderruflich. Nach dem Löschen der Schlüssel hat Apple keinen Zugriff mehr auf *jegliche* der Daten, die von den Dienstschlüssel des Benutzers geschützt werden. Zu diesem Zeitpunkt beginnt das Gerät mit einem asynchronen Schlüsselrotationsvorgang, bei dem für jeden Dienst, dessen Schlüssel zuvor für Apple-Server verfügbar war, ein neuer Dienstschlüssel erstellt wird. Wenn die Schlüsselrotation aufgrund einer Netzwerkstörung oder eines anderen Fehlers fehlschlägt, wiederholt das Gerät die Schlüsselrotation, bis sie erfolgreich ist.

Nach erfolgreicher Dienstschlüsselrotation können neue Daten, die in den Dienst geschrieben wurden, nicht mit dem alten Dienstschlüssel entschlüsselt werden. Sie werden mit dem neuen Schlüssel geschützt, der ausschließlich von den vertrauenswürdigen Geräten des Benutzers gesteuert wird und nie für Apple verfügbar war.

Erweiterter Datenschutz und Webzugriff auf iCloud.com

Wenn ein Benutzer den erweiterten Datenschutz erstmals aktiviert, wird der Webzugriff auf seine Daten auf iCloud.com automatisch deaktiviert. Der Grund dafür ist, dass die iCloud-Webserver keinen Zugriff mehr auf die Schlüssel haben, die zum Entschlüsseln und Anzeigen der Daten des Benutzers erforderlich sind. Der Benutzer kann dann den Webzugriff wieder aktivieren und die Teilnahme seines vertrauenswürdigen Geräts nutzen, um auf seine verschlüsselten iCloud-Daten im Web zuzugreifen.

Nach Aktivieren des Webzugriffs muss der Benutzer jedes Mal, wenn er iCloud.com besucht, die Webanmeldung auf einem seiner vertrauenswürdigen Geräte autorisieren. Die Autorisierung bereitet das Gerät für den Webzugriff vor. Für die nächste Stunde akzeptiert dieses Gerät Anfragen von bestimmten Apple-Servern, einzelne Dienstschlüssel hochzuladen, aber nur diejenigen, die einer Positivliste von Diensten entsprechen, die normalerweise auf iCloud.com zugänglich sind. Anders gesagt: Auch nachdem der Benutzer eine Webanmeldung autorisiert hat, kann eine Serveranfrage das Gerät des Benutzers nicht veranlassen, Dienstschlüssel für Daten hochzuladen, die nicht zum Anzeigen auf iCloud.com vorgesehen sind (etwa Gesundheitsdaten oder Passwörter im iCloud-Schlüsselbund). Apple-Server fordern nur die Dienstschlüssel an, die zum Entschlüsseln der spezifischen Daten erforderlich sind, auf die der Benutzer im Web zugreifen möchte. Bei jedem Upload eines Dienstschlüssels wird dieser mit einem temporären Schlüssel verschlüsselt, der an die Websitzung gebunden ist, die der Benutzer autorisiert hat. Dabei wird auf dem Gerät des Benutzers eine Mitteilung angezeigt, die den iCloud-Dienst zeigt, dessen Daten den Apple-Servern vorübergehend bereitgestellt werden.

Benutzerauswahlen erhalten

Die Einstellungen für den erweiterten Datenschutz und den Webzugriff auf iCloud.com können nur vom Benutzer geändert werden. Diese Werte werden in den Metadaten des iCloud-Schlüsselbund-Geräts gespeichert und können nur von einem der vertrauenswürdigen Geräte des Benutzers geändert werden. Die Apple-Server können diese Einstellungen weder im Namen des Benutzers ändern, noch können sie sie auf eine vorherige Konfiguration zurückstufen.

Sicherheitsüberlegungen zu Freigabe und Zusammenarbeit

Wenn Benutzer Inhalte für die Zusammenarbeit miteinander teilen – beispielsweise mit geteilten Notizen, geteilten Erinnerungen und geteilten Ordnern auf iCloud Drive oder in der geteilten iCloud-Fotomediathek – und wenn alle Benutzer den erweiterten Datenschutz aktiviert haben, werden die Apple-Server nur verwendet, um die Freigabe einzurichten, haben aber keinen Zugriff auf die Verschlüsselungsschlüssel für die geteilten Daten. Die Inhalte bleiben Ende-zu-Ende-verschlüsselt und sind nur auf den vertrauenswürdigen Geräten der Teilnehmenden zugänglich. Für jede Freigabeaktion kann ein Titel und eine repräsentative Miniatur von Apple mit dem Standarddatenschutz gespeichert werden, um den Empfängern eine Vorschau anzuzeigen.

Wird beim Aktivieren der Zusammenarbeit die Option „Jeder mit einem Link“ ausgewählt, werden die Inhalte den Apple-Servern unter Standarddatenschutz zur Verfügung gestellt, da die Server in der Lage sein müssen, jedem, der die URL öffnet, Zugriff zu gewähren.

Der erweiterte Datenschutz wird von der iWork-Zusammenarbeit und der Funktion für geteilte Alben in der App „Fotos“ nicht unterstützt. Wenn Benutzer gemeinsam an einem iWork-Dokument arbeiten oder ein iWork-Dokument in einem geteilten Ordner auf iCloud Drive öffnen, werden die Verschlüsselungsschlüssel für das Dokument sicher auf die iWork-Server in den Apple-Rechenzentren hochgeladen. Das liegt daran, dass die Echtzeit-Zusammenarbeit in iWork erfordert, dass Dokumentänderungen durch serverseitige Vermittlung zwischen den Teilnehmenden koordiniert werden. Zu geteilten Alben hinzugefügte Fotos werden mit Standarddatenschutz gespeichert, da die Funktion es erlaubt, dass Alben öffentlich im Web geteilt werden.

Erweiterten Datenschutz deaktivieren

Der Benutzer kann den erweiterten Datenschutz jederzeit deaktivieren. Dabei geschieht Folgendes:

1. Das Gerät des Benutzers zeichnet zuerst die neue Auswahl in den beteiligten Metadaten des iCloud-Schlüsselbunds auf. Diese Einstellung wird dann sicher auf allen Geräten synchronisiert.
2. Das Gerät des Benutzers lädt die Dienstschlüssel für alle *nach-Authentifizierung-verfügbaren* Dienste auf die iCloud HSMS in den Rechenzentren von Apple hoch. Dies umfasst niemals Schlüssel für solche Dienste, die unter Standarddatenschutz Ende-zu-Ende-verschlüsselt sind, etwa iCloud-Schlüsselbund und Health.

Das Gerät lädt sowohl die ursprünglichen Dienstschlüssel hoch, die vor der Aktivierung des erweiterten Datenschutzes generiert wurden, als auch die neuen Dienstschlüssel, die generiert wurden, nachdem der Benutzer die Funktion aktiviert hatte. Dadurch werden alle Daten in diesen Diensten nach der Authentifizierung zugänglich gemacht und für den Account gilt wieder der Standarddatenschutz, bei dem Apple dem Benutzer erneut helfen kann, die meisten seiner Daten wiederherzustellen, falls er den Zugriff auf seinen Account verlieren sollte.

iCloud-Daten vom erweiterten Datenschutz nicht abgedeckt

Aufgrund der Notwendigkeit, mit den globalen E-Mail-, Kontakt- und Kalendersystemen zusammenzuarbeiten, werden iCloud Mail, Kontakte und Kalender nicht Ende-zu-Ende-verschlüsselt.

iCloud speichert einige Daten ohne den Schutz der benutzerspezifischen CloudKit-Dienstschlüssel, selbst wenn der erweiterte Datenschutz aktiviert ist. Felder für CloudKit-Datensätze müssen im zu schützenden Containerschema explizit als „verschlüsselt“ deklariert werden, und das Lesen und Schreiben verschlüsselter Felder erfordert die Verwendung dedizierter [APIs](#). Datum und Uhrzeit, zu der eine Datei oder ein Objekt geändert wurde, werden verwendet, um die Informationen eines Benutzers zu sortieren, und Prüfsummen von Datei- und Fotodaten werden verwendet, um Apple dabei zu unterstützen, den iCloud- und Gerätespeicher des Benutzers zu deduplizieren und zu optimieren – und das alles ohne Zugriff auf die Dateien und Fotos selbst. Details dazu, wie die Verschlüsselung für bestimmte Datenkategorien verwendet wird, enthält der Apple Support-Artikel [Sicherheitsüberblick – iCloud-Daten](#).

Entscheidungen wie die Verwendung von Prüfsummen für die Datenduplizierung – eine bekannte Technik, die als *konvergente Verschlüsselung* bezeichnet wird – waren Teil des ursprünglichen Designs von iCloud-Diensten, als sie eingeführt wurden. Diese Metadaten sind immer verschlüsselt, aber die Verschlüsselungsschlüssel werden von Apple mit Standarddatenschutz gespeichert. Um den Sicherheitsschutz für alle Benutzer weiter zu stärken, setzt sich Apple dafür ein, dass mehr Daten, einschließlich dieser Art von Metadaten, Ende-zu-Ende-verschlüsselt werden, wenn der erweiterte Datenschutz aktiviert ist.

Anforderungen an den erweiterten Datenschutz

Für die Aktivierung des erweiterten Datenschutzes für iCloud gelten folgende Anforderungen:

- Der Account des Benutzers muss die Ende-zu-Ende-Verschlüsselung unterstützen. Die Ende-zu-Ende-Verschlüsselung erfordert die Zwei-Faktor-Authentifizierung für die Apple-ID und einen Code oder ein Passwort, der bzw. das auf den vertrauenswürdigen Geräten festgelegt ist. Weitere Informationen sind im Apple Support-Artikel [Zwei-Faktor-Authentifizierung für die Apple-ID](#) zu finden.
- Geräte, bei denen der Benutzer mit seiner Apple-ID angemeldet ist, müssen auf iOS 16.2, iPadOS 16.2, macOS 13.1, tvOS 16.2, watchOS 9.2 und die neueste Version von iCloud für Windows aktualisiert werden. Diese Vorgabe verhindert, dass die neu erstellten Dienstschlüssel von einer vorherigen Version von iOS, iPadOS, macOS, tvOS oder watchOS falsch behandelt werden. Beispielsweise könnten sie sonst in einem fehlgeleiteten Versuch, den Account-Status zu reparieren, diese neuen Dienstschlüssel erneut in die *nach-Authentifizierung-verfügbaren* Hardwaresicherheitsmodule (HSMs) hochladen.
- Der Benutzer muss mindestens eine alternative Wiederherstellungsmethode einrichten, etwa einen oder mehrere Wiederherstellungskontakte oder einen Wiederherstellungsschlüssel, die er dann zum Wiederherstellen seiner iCloud-Daten verwenden kann, falls er den Zugriff auf seinen Account verlieren sollte.

Falls die Wiederherstellungsmethoden fehlschlagen, beispielsweise wenn die Informationen des Wiederherstellungskontakts veraltet sind oder wenn der Benutzer sie vergisst, kann Apple bei der Wiederherstellung der Ende-zu-Ende-verschlüsselten iCloud-Daten keine Unterstützung leisten.

Der erweiterte Datenschutz für iCloud kann nur für Apple-IDs aktiviert werden. Verwaltete Apple-IDs und Accounts von Kindern (variiert je nach Land oder Region) werden nicht unterstützt.

Sicherheit bei iCloud Backup

iCloud sichert Informationen – einschließlich Geräteeinstellungen, App-Daten, Fotos und Videos in „Aufnahmen“ sowie Konversationen in der App „Nachrichten“ – täglich per WLAN. iCloud-Backup wird nur ausgeführt, wenn das Gerät gesperrt und mit einer Stromquelle verbunden ist und WLAN-Zugriff auf das Internet besteht. iCloud-Backup ist für den Schutz der Daten mit dem Augenmerk auf die in iOS und iPadOS verwendete Speicherverschlüsselung konzipiert und erlaubt inkrementelle, unbeaufsichtigte Sicherungen und Wiederherstellungen. Standardmäßig wird der iCloud-Backup-Dienstschlüssel in iCloud-Hardwaresicherheitsmodulen in den Rechenzentren von Apple zuverlässig gesichert und ist Bestandteil der nach-Authentifizierung-verfügbaren Datenkategorie. Für Benutzer, die den erweiterten Datenschutz für iCloud aktivieren, wird der iCloud-Backup-Dienstschlüssel mit Ende-zu-Ende-Verschlüsselung geschützt und steht nur Benutzern auf vertrauenswürdigen Geräten zur Verfügung.

Wenn Dateien in Datensicherheitsklassen erstellt werden, auf die bei einem gesperrten Gerät nicht zugegriffen werden kann, werden ihre pro Datei erzeugten Schlüssel mit den Klassenschlüsseln des iCloud-Backup-Keybag verschlüsselt und die Dateien in iCloud in ihrem ursprünglichen, verschlüsselten Zustand gesichert. Alle Dateien werden bei der Übertragung verschlüsselt, und beim Speichern werden accountbasierte Schlüssel verwendet (siehe [CloudKit-Verschlüsselung](#)).

Der iCloud-Backup-Keybag enthält asymmetrische (Curve25519) Schlüssel für Datenschutzzklassen, auf die nicht zugegriffen werden kann, wenn das Gerät gesperrt ist. Die Sicherungen werden im iCloud-Account des Benutzers gespeichert und bestehen aus einer Kopie der Dateien des Benutzers und dem iCloud-Backup-Keybag. Der iCloud-Backup-Keybag wird mit einem zufälligen Schlüssel geschützt, der mit den Sicherungen gespeichert wird. Das iCloud-Passwort des Benutzers wird nicht für die Verschlüsselung eingesetzt, sodass bestehende Sicherungen bei einer Passwortänderung nicht ungültig werden.

Bei der Wiederherstellung werden die gesicherten Dateien, der iCloud-Backup-Keybag und der Schlüssel für den Keybag vom iCloud-Account des Benutzers abgerufen. Der iCloud-Backup-Keybag wird mit seinem Schlüssel entschlüsselt. Anschließend werden die pro Datei erzeugten Schlüssel im Keybag verwendet, um die Dateien in den Sicherungen zu entschlüsseln, die als neue Dateien auf das Dateisystem geschrieben und so gemäß ihrer Datensicherheitsklasse neu verschlüsselt werden.

Die folgenden Inhalte werden mit iCloud-Backup gesichert:

- Einträge für gekaufte Musik, Filme, TV-Sendungen, Apps und Bücher. Das iCloud-Backup eines Benutzers umfasst Informationen über gekaufte Inhalte, die auf dem Gerät des Benutzers vorhanden sind, nicht jedoch die gekauften Inhalte selbst. Wenn der Benutzer ein iCloud-Backup wiederherstellt, werden seine gekauften Inhalte automatisch aus dem iTunes Store, dem App Store, der Apple TV App oder von Apple Books geladen. Einige Arten von Inhalten werden nicht in allen Ländern und Regionen automatisch heruntergeladen und bereits getätigte Einkäufe sind möglicherweise nicht verfügbar, wenn sie zurückerstattet wurden oder im jeweiligen Store nicht mehr angeboten werden. Der vollständige Einkaufsverlauf ist der Apple-ID des Benutzers zugewiesen.

- Fotos und Videos auf Geräten des Benutzers. Zur Beachtung: Aktiviert ein Benutzer die iCloud-Fotomediathek unter iOS 8.1, iPadOS 13.1 oder OS X 10.10.3 (oder neuer), sind seine Fotos und Videos bereits in iCloud gespeichert. Sie werden also nicht in das iCloud-Backup des Benutzers einbezogen.
- Kontakte, Kalender-Ereignisse, Erinnerungen und Notizen
- Geräteeinstellungen
- App-Daten
- Home-Bildschirm und Anordnung der Apps
- HomeKit-Konfiguration
- Notfallpass-Daten
- Passwort für Sprachmemos (bei Bedarf, erfordert die physische SIM-Karte, die während des Backups verwendet wurde)
- Nachrichten, Apple Messages for Business, Text- (SMS-) und MMS-Nachrichten (bei Bedarf, erfordert die physische SIM-Karte, die während des Backups verwendet wurde)

iCloud-Backup wird auch verwendet, um den lokalen Schlüsselbund des Geräts zu sichern, verschlüsselt mit einem Schlüssel, der vom kryptografischen Secure Enclave UID Root-Schlüssel des Geräts abgeleitet ist. Dieser Schlüssel ist für das Gerät einzigartig und Apple nicht bekannt. Dadurch kann die Datenbank auf dem Gerät, von dem er ursprünglich stammt, wiederhergestellt werden, und niemand außer dem Benutzer – noch nicht einmal Apple – kann ihn lesen. Weitere Informationen sind unter [Secure Enclave](#) zu finden.

Nachrichten in iCloud

„Nachrichten in iCloud“ hält den gesamten Nachrichtenverlauf eines Benutzers auf dem aktuellen Stand und auf allen Geräten verfügbar.

Mit dem Standarddatenschutz wird „Nachrichten in iCloud“ Ende-zu-Ende-verschlüsselt, wenn iCloud-Backup deaktiviert wird. Wenn iCloud-Backup aktiviert wird, umfasst das Backup eine Kopie des Verschlüsselungsschlüssels von „Nachrichten in iCloud“, damit Apple dem Benutzer beim Wiederherstellen seiner Nachrichten behilflich sein kann, auch wenn er den Zugriff auf den iCloud-Schlüsselbund und seine vertrauenswürdigen Geräte verloren hat. Wenn der Benutzer iCloud-Backup deaktiviert, wird auf dem Gerät ein neuer Schlüssel generiert, um künftige „Nachrichten in iCloud“ zu schützen. Der neue Schlüssel wird nur im iCloud-Schlüsselbund gespeichert, ist nur für den Benutzer auf seinen vertrauenswürdigen Geräten zugänglich, und neue Daten, die in den Container geschrieben werden, können nicht mit dem alten Container-Schlüssel entschlüsselt werden.

Beim erweiterten Datenschutz ist „Nachrichten in iCloud“ immer Ende-zu-Ende-verschlüsselt. Wenn iCloud-Backup aktiviert wird, ist alles, was sich darin befindet, Ende-zu-Ende-verschlüsselt, einschließlich des Verschlüsselungsschlüssels von „Nachrichten in iCloud“. Der iCloud-Backup-Dienstschlüssel sowie der Containerschlüssel von „Nachrichten in iCloud“ werden beide aktiviert, wenn der Benutzer den erweiterten Datenschutz aktiviert. Weitere Informationen sind im Apple Support-Artikel [Sicherheitsüberblick – iCloud-Daten](#) zu finden.

Sicherheit beim Kontakt für die Accountwiederherstellung

Benutzer können bis zu fünf Personen, denen sie vertrauen, als Kontakte für die Accountwiederherstellung hinzufügen, um ihren iCloud-Account und ihre iCloud-Daten wiederherzustellen. Hierzu gehören unter anderem alle mit der Ende-zu-Ende-Verschlüsselung geschützte Daten, egal, ob der erweiterte Datenschutz aktiviert ist oder nicht. Weder Apple noch der Wiederherstellungskontakt haben die notwendigen Informationen, um die mit der Ende-zu-Ende-Verschlüsselung geschützten iCloud-Daten des Benutzers selbst wiederherzustellen.

Der Wiederherstellungskontakt wurde mit Augenmerk auf den Datenschutz entwickelt. Die vom Benutzer gewählten Wiederherstellungskontakte sind Apple nicht bekannt. Erst spät im Verlauf eines Wiederherstellungsversuchs, nachdem der Benutzer den Kontakt um Hilfe gebeten und dieser tatsächlich begonnen hat, bei der Wiederherstellung zu helfen, erfahren die Apple-Server Informationen über den Wiederherstellungskontakt. Diese Informationen werden nicht gesichert, nachdem die Wiederherstellung abgeschlossen ist.

Sicherheitsprozess für den Wiederherstellungskontakt

Wenn ein Benutzer einen Account-Wiederherstellungskontakt einrichtet, wird der Zugangsschlüssel zu den iCloud-Daten des Benutzers – einschließlich mit Ende-zu-Ende-Verschlüsselung geschützter CloudKit-Daten – mit einem starken zufälligen Schlüssel verschlüsselt. Dieser zufällige Schlüssel wird dann zwischen dem Wiederherstellungskontakt und Apple aufgeteilt. Nur wenn beide Teile zum Zeitpunkt der Wiederherstellung erneut kombiniert werden, kann der Originalschlüssel wiederhergestellt und auf die iCloud-Daten des Benutzers zugegriffen werden.

Für die Konfiguration eines Account-Wiederherstellungskontakts kommuniziert das Benutzergerät mit Apple-Servern, um den Teil der Schlüsselinformationen hochzuladen, der von Apple gespeichert wird. Danach wird ein Ende-zu-Ende-verschlüsselter CloudKit-Container mit dem Wiederherstellungskontakt eingerichtet, um den Teil der Schlüsselinformationen für den Wiederherstellungskontakt zu teilen. Sowohl Apple als auch der Wiederherstellungskontakt erhalten vom Benutzer dieselben Secret-Daten für die Autorisierung, die später für die Wiederherstellung erforderlich sind. Die Kommunikation zum Einladen und Akzeptieren des Wiederherstellungskontakts erfolgt durch einen gegenseitig authentifizierten IDS-Kanal. Der Wiederherstellungskontakt speichert die empfangenen Informationen automatisch im iCloud-Schlüsselbund. Apple kann weder auf den Inhalt des CloudKit-Containers, noch auf den iCloud-Schlüsselbund zugreifen, in dem diese Informationen gespeichert sind. Wenn dieser Vorgang ausgeführt wird, können Apple-Server nur eine anonyme ID für den Wiederherstellungskontakt sehen.

Wenn Benutzer zu einem späteren Zeitpunkt ihren Account und ihre iCloud-Daten wiederherstellen möchten, können sie ihren Wiederherstellungskontakt um Hilfe bitten. In diesem Fall wird ein Wiederherstellungscode auf dem Gerät des Wiederherstellungskontakts erzeugt, den der Wiederherstellungskontakt dann dem Benutzer übermitteln kann (z. B. persönlich oder per Telefon). Der Benutzer gibt dann den Wiederherstellungscode auf seinem Gerät ein, um über das Protokoll SPAKE2+ eine sichere Verbindung zwischen den Geräten herzustellen. Diese Inhalte sind für Apple nicht zugänglich. Diese Interaktion wird von den Apple-Servern abgestimmt. Apple kann den Wiederherstellungsprozess jedoch nicht einleiten.

Nachdem die sichere Verbindung hergestellt ist und alle notwendigen Sicherheitsüberprüfungen durchgeführt wurden, gibt das Gerät des Wiederherstellungskontakts den jeweiligen Teil der Schlüsselinformationen zurück und das zuvor ausgewiesene Autorisierungs-Secret wird an den Benutzer zurückgegeben, der die Wiederherstellung angefordert hat. Der Benutzer stellt einem Apple-Server diese Secret-Daten für die Autorisierung bereit. Im Gegenzug erhält er Zugriff auf die Verschlüsselungsinformationen, die Apple aufbewahrt. Die Bereitstellung der Secret-Daten autorisiert außerdem das Zurücksetzen des Passworts für den Account, um den Zugriff auf den Account wiederherzustellen.

Im letzten Schritt entschlüsselt das Benutzergerät die Schlüsseldaten, die es von Apple und dem Account-Wiederherstellungskontakt erhalten hat, und verwendet diese Daten dann, um die iCloud-Daten zu entschlüsseln und wiederherzustellen.

Integrierte Sicherheitsmechanismen verhindern, dass ein Wiederherstellungskontakt eine Wiederherstellung ohne Benutzereinwilligung initiiert. Zu diesen gehört die Überprüfung, dass ein Benutzeraccount gültig ist. Wenn der Account aktiv ist, setzt die Wiederherstellung über einen Wiederherstellungskontakt außerdem die Kenntnis eines zuletzt vergebenen Gerätecodes oder des iCloud-Sicherheitscodes voraus.

Sicherheit beim Nachlasskontakt

Wenn Benutzer begünstigten Personen den Zugriff auf ihre iCloud-Daten ermöglichen wollen, nachdem sie verstorben sind, können sie Nachlasskontakte in ihrem Account festlegen. Die begünstigte Person eines Nachlasskontakts erhält Zugriff auf alle iCloud-Daten der verstorbenen Person, einschließlich aller Ende-zu-Ende-verschlüsselter Daten, aber ausschließlich der Daten des iCloud-Schlüsselbunds, wie etwa Account-Passwörter. Die Technologie, die einem Nachlasskontakt zugrunde liegt, ähnelt der Funktionsweise eines Wiederherstellungskontakts – ein zufälliger Schlüssel wird zwischen Apple und dem Nachlasskontakt aufgeteilt, sodass keine der beiden Parteien unabhängig von der jeweils anderen Daten entschlüsseln kann. Eine begünstigte Person erhält die gleichen Datenklassen, wobei es keine Rolle spielt, ob der Benutzer den erweiterten Datenschutz aktiviert hat oder nicht.

Die Schlüsselinformation, die eine begünstigte Person erhält, wird in der Benutzerdokumentation als „Zugriffsschlüssel“ bezeichnet und automatisch auf unterstützten Geräten gesichert. Er kann aber auch ausgedruckt und für die Offlinenutzung aufbewahrt werden. Weitere Informationen sind im Apple Support-Artikel [Einen Nachlasskontakt für deine Apple-ID hinzufügen](#) zu finden.

Nach dem Tod des Benutzers können Nachlasskontakte auf der von Apple bereitgestellten Website den Zugriff beantragen. Hierzu ist ein Totenschein erforderlich, der zum Teil mit den Secret-Daten autorisiert wird, wie im vorherigen Abschnitt beschrieben. Nachdem alle Sicherheitsüberprüfungen abgeschlossen sind, gibt Apple einen Benutzernamen und ein Passwort für den neuen Account aus und stellt dem Nachlasskontakt die Informationen bereit.

Damit der Zugriffsschlüssel bei Bedarf einfacher eingegeben werden kann, wird dieser als alphanumerischer Code mit einem zugehörigen QR-Code bereitgestellt. Nach der Eingabe wird der Zugriffsschlüssel verwendet, um den Zugriff auf die iCloud-Daten der verstorbenen Person wiederherzustellen. Dieser Vorgang kann auf einem Gerät oder auch online durchgeführt werden. Weitere Informationen sind im Apple Support-Artikel [Anfordern des Zugriffs auf einen Apple-Account als Nachlasskontakt](#) zu finden.

Sicherheit von iCloud Privat-Relay

Das iCloud Privat-Relay hilft in erster Linie dabei, Nutzer bei Browsing-Aktivitäten mit Safari zu schützen. Darüber hinaus laufen über den Dienst alle Anfragen für DNS-Namensauflösungen. Dadurch ist sichergestellt, dass niemand, auch nicht Apple, die IP-Adresse des Benutzers mit seinen Browsing-Aktivitäten verknüpfen kann. Um dies zu ermöglichen, nutzt der Dienst verschiedene Proxys: einen von Apple verwalteten Eingangsproxy und einen von einem Inhaltsanbieter verwalteten Ausgangsproxy. Zum Verwenden des iCloud Privat-Relay muss iOS 15, iPadOS 15 oder macOS 12.0.1 (oder neuer) installiert sein, und der Benutzer muss mit seiner Apple-ID beim Account von iCloud+ angemeldet sein. Sind diese Voraussetzungen erfüllt, kann das iCloud Privat-Relay unter „Einstellungen“ > „iCloud“ oder „Systemeinstellungen“ > „iCloud“ aktiviert werden.

Weitere Informationen sind unter [iCloud Private Relay Overview](#) zu finden.

Code- und Passwortverwaltung

Passwortsicherheit – Übersicht

iOS, iPadOS und macOS ermöglichen es Benutzern, sich einfach und bequem bei Apps und Websites von Drittanbietern zu authentifizieren, die Passwörter verwenden. Die beste Methode der Passwortverwaltung besteht darin, kein Passwort verwenden zu müssen. „Mit Apple anmelden“ ermöglicht es Benutzern, sich bei Apps und Websites anderer Anbieter anzumelden, ohne zusätzliche Accounts oder Passwörter erstellen und verwalten zu müssen. Gleichzeitig wird die Anmeldung durch die Zwei-Faktor-Authentifizierung für die Apple-ID geschützt. Für Websites, die „Mit Apple anmelden“ nicht unterstützen, ermöglicht es die Funktion „Automatisch starke Passwörter“ dem Gerät eines Benutzers, automatisch eindeutige, starke Passwörter für Websites und Apps zu erstellen, zu synchronisieren und einzugeben. In iOS und iPadOS werden Passwörter in einem speziellen Schlüsselbund zum automatischen Ausfüllen von Passwörtern gesichert. Dieser Schlüssel wird vom Benutzer gesteuert und kann unter „Einstellungen“ > „Passwörter“ verwaltet werden.

In macOS können gesicherte Passwörter in Safari unter „Einstellungen“ > „Passwörter“ verwaltet werden. Auch zum Synchronisieren von Passwörtern, die manuell vom Benutzer erstellt wurden, kann dieses Synchronisierungssystem verwendet werden.

Sicherheit bei Option „Mit Apple anmelden“

„Mit Apple anmelden“ ist eine mit Datenschutzrichtlinien konforme Alternative zu anderen SSO-Systemen (Single Sign-On). Diese Option erlaubt das bequeme und effiziente Anmelden durch einmaliges Tippen und bietet dem Benutzer gleichzeitig mehr Transparenz und Kontrolle über seine persönlichen Daten.

Mit der Option „Mit Apple anmelden“ können Benutzer einen Account einrichten und sich mit der Apple-ID, die sie bereits haben, bei Apps und Websites anmelden. Außerdem erhalten sie mehr Kontrolle über ihre persönlichen Daten. Beim Einrichten eines Accounts können Apps nur nach dem Namen und der E-Mail-Adresse des Benutzers fragen. Der Benutzer hat dann die Wahl: Er kann seine persönliche E-Mail-Adresse mit einer App teilen oder sich dafür entscheiden, seine persönliche E-Mail-Adresse privat zu halten und stattdessen den neuen Apple-Relay-Dienst für private E-Mail-Adressen zu verwenden. Dieser Dienst stellt eine eindeutige, anonymisierte E-Mail-Adresse bereit, die E-Mails an die persönliche Adresse des Benutzers weiterleitet. Auf diese Weise kann der Benutzer weiterhin nützliche Mitteilungen vom Entwickler empfangen und zugleich ein gewisses Maß an Schutz und Kontrolle über seine persönlichen Daten erhalten.

Die Option „Mit Apple anmelden“ ist auf Sicherheit ausgelegt. Jeder Benutzer, der „Mit Apple anmelden“ nutzen möchte, muss für seine Apple-ID die Zwei-Faktor-Authentifizierung aktivieren. Zwei-Faktor-Authentifizierung trägt dazu bei, nicht nur die Apple-ID des Benutzers zu schützen, sondern auch die Accounts, die er mit seinen Apps erstellt. Darüber hinaus hat Apple ein Antibetrugssignal entwickelt und in die Funktion „Mit Apple anmelden“ integriert, das den Datenschutzvorgaben entspricht. Das Signal gibt Entwicklern die Gewissheit, dass es sich bei neuen Benutzern, die sich anmelden, um reale Personen und nicht um Bots oder per Skript erstellte Accounts handelt.

Option „Automatisch starke Passwörter“

Wenn der iCloud-Schlüsselbund aktiviert ist, erstellen iOS, iPadOS und macOS nach dem Zufallsprinzip starke eindeutige Passwörter, wenn sich die Benutzer bei einer App oder einer Website in Safari anmelden oder dort ihre Passwörter ändern. Bei iOS und iPadOS steht „Automatisch starke Passwörter“ auch in Apps zur Verfügung. Die Benutzer müssen sich gegen die Nutzung starker Passwörter entscheiden. Generierte Passwörter werden im Schlüsselbund gesichert und mithilfe des iCloud-Schlüsselbunds (wenn dieser aktiviert ist) geräteübergreifend auf dem neuesten Stand gehalten.

Von iOS und iPadOS generierte Passwörter sind standardmäßig 20 Zeichen lang. Sie enthalten eine Ziffer, einen Großbuchstaben, zwei Bindestriche und 16 Kleinbuchstaben. Diese generierten Passwörter sind stark und umfassen eine Entropie mit 71 Bits.

Die Passwörter werden basierend auf einer Heuristik generiert, die bestimmt, dass zum Erstellen eines Passworts ein Passwortfeld erforderlich ist. Für den Fall, dass die Heuristik nicht erkennt, dass ein kontextspezifisches Passwort bei der Passwörterstellung verwendet wird, können App-Entwickler `UITextFieldContentType.newPassword` für das Textfeld und Web-Entwickler `autocomplete= "new-password"` für ihre `<input>`-Elemente festlegen.

Apps und Websites können Regeln bereitstellen, um sicherzustellen, dass generierte Passwörter mit den relevanten Diensten kompatibel sind. Entwickler stellen diese Regeln mithilfe von `UITextFieldInputPasswordRules` oder mithilfe des Attributs `passwordrules` in ihren `input`-Elementen bereit. Geräte generieren dann die stärksten Passwörter, die ihnen möglich sind, die diese Regeln erfüllen können.

Sicherheit beim automatischen Ausfüllen von Passwörtern

Die Funktion für das automatische Ausfüllen von Passwörtern füllt Anmeldedaten aus, die im Schlüsselbund gespeichert sind. Der Passwort-Manager des iCloud-Schlüsselbunds und der Schlüsselbund zum automatischen Ausfüllen von Passwörtern bieten folgende Funktionen:

- Ausfüllen von Anmeldedaten in Apps und auf Websites
- Generieren starker Passwörter
- Sichern von Passwörtern in Apps und auf Websites in Safari
- Sicheres Teilen von Passwörtern mit den Kontakten des Benutzers
- Bereitstellen von Passwörtern für ein Apple TV in der Nähe, das Anmeldedaten anfordert

Das Generieren und Sichern von Passwörtern innerhalb von Apps sowie das Bereitstellen von Passwörtern für das Apple TV ist nur in iOS und iPadOS verfügbar.

Automatisches Ausfüllen von Passwörtern in Apps

Unter iOS und iPadOS können Benutzer gespeicherte Benutzernamen und Passwörter in Felder mit Anmeldedaten in Apps eingeben – ähnlich wie beim automatischen Ausfüllen von Passwörtern in Safari. In iOS und iPadOS tippen die Benutzer dazu auf eine Einverständnistaste in der QuickType-Leiste der Softwaretastatur. In macOS wird für Apps, die mit Mac Catalyst erstellt wurden, unter den Feldern mit Anmeldedaten ein Einblendmenü „Passwörter“ angezeigt.

Wenn eine App eng mit einer Website verbunden ist, die denselben App-Website-Zuordnungsmechanismus verwendet, den die Datei „apple-app-site-association“ einsetzt (wie oben beschrieben), schlagen die QuickType-Leiste von iOS und iPadOS sowie das macOS-Einblendmenü direkt Anmeldedaten zum Ausfüllen für die App vor, sofern entsprechende Informationen im Schlüsselbund zum automatischen Ausfüllen von Passwörtern gespeichert sind. So können Benutzer in Safari gespeicherte Anmeldedaten für Apps mit denselben Sicherheitsmerkmalen offenlegen, ohne dass Apps eine API übernehmen müssen.

Die Option zum automatischen Ausfüllen von Passwörtern legt einer App keine Anmeldedaten offen, es sei denn, der Benutzer stimmt zu, sie für die App freizugeben. Die Listen mit Anmeldedaten werden außerhalb des Prozesses der App erstellt oder präsentiert.

Wenn eine App und eine Website eine vertrauenswürdige Verbindung haben und ein Benutzer Anmeldedaten innerhalb einer App sendet, können iOS und iPadOS den Benutzer auffordern, diese Anmeldedaten für spätere Zwecke im Schlüsselbund zum automatischen Ausfüllen von Passwörtern zu sichern.

Zugriff durch Apps auf gesicherte Passwörter

iOS-, iPadOS- und macOS-Apps können mittels `ASAuthorizationPasswordProvider` und `SecAddSharedWebCredential` die Hilfe des Schlüsselbundes zum automatischen Ausfüllen von Passwörtern anfordern. Der Passwortanbieter und seine Anfrage lassen sich in Verbindung mit der Option „Mit Apple anmelden“ verwenden, sodass zur Unterstützung der Benutzer beim Anmelden an Apps dieselbe API aufgerufen wird – unabhängig davon, ob der Account des Benutzers passwortbasiert ist oder mit „Mit Apple anmelden“ erstellt wurde.

Der Zugriff von Apps auf gesicherte Passwörter wird nur gewährt, wenn sowohl der Entwickler der App als auch der Administrator der Website dies erlauben und der Benutzer zugestimmt hat. App-Entwickler können angeben, dass sie auf die von Safari gesicherten Passwörter zugreifen möchten, indem sie in ihre App eine Berechtigung einfügen. Diese Berechtigung umfasst die vollständig qualifizierten Domain-Namen verknüpfter Websites. Auf dem Server der Website muss sich eine Datei mit der eindeutigen App-Kennung der von Apple zugelassenen Apps befinden.

Wird eine App mit der Berechtigung „com.apple.developer.associated-domains“ installiert, stellen iOS und iPadOS eine TLS-Anfrage an alle aufgeführten Websites und fordern eine der folgenden Dateien an:

- apple-app-site-association
- .well-known/apple-app-site-association

Wenn in der Datei die Kennung der zu installierenden App aufgelistet ist, markieren iOS und iPadOS die Beziehung zwischen Website und App als vertrauenswürdig. Nur bei vertrauenswürdigen Beziehungen führen Aufrufe dieser beiden APIs zu einer Eingabeaufforderung für den Benutzer, der zustimmen muss, bevor Passwörter für die App freigegeben, aktualisiert oder gelöscht werden.

Sicherheitsempfehlungen für Passwörter

Aus der Liste der Passwörter für das automatische Ausfüllen von Passwörtern von iOS, iPadOS und macOS geht hervor, welches der gesicherten Passwörter eines Benutzers für andere Websites *wiederholt verwendet* wird, welche Passwörter als *schwach* eingestuft werden und welche Passwörter infolge eines *Datenlecks* kompromittiert wurden.

Übersicht

Wird dasselbe Passwort für mehrere Dienste verwendet, macht dies die betreffenden Accounts anfällig für Credential-Stuffing-Angriffe. Diese funktionieren folgendermaßen: Wenn ein Angreifer in einen Dienst eingedrungen ist und Passwörter abgegriffen hat, kann er versuchen, dieselben Anmeldedaten bei anderen Diensten auszuprobieren, um weitere Accounts zu schädigen.

- Passwörter werden als *wiederholt verwendet* gekennzeichnet, wenn erkannt wird, dass ein bestimmtes Passwort in verschiedenen Domains wiederholt als gesichertes Passwort verwendet wird.
- Passwörter werden als „*schwach*“ gekennzeichnet, wenn Angreifer sie leicht erraten können. iOS, iPadOS und macOS erkennen gängige Muster, die zum Erstellen leicht zu merkender Passwörter verwendet werden. Hierzu gehören beispielsweise Wörter, die in einem Lexikon zu finden sind, gängige Zeichenersetzungen (etwa „p4ssw0rt“ anstelle von „passwort“), Muster auf einer Tastatur (etwa „q12we34r“ einer QWERTZ-Tastatur) oder sich wiederholende Zeichenfolgen (etwa „123123“). Diese Muster werden häufig zum Erstellen von Passwörtern verwendet, die die Mindestanforderungen für Passwörter zum Anmelden bei Diensten erfüllen. Doch auch Angreifer bedienen sich dieser Muster, wenn sie mittels Brute-Force-Attacken versuchen, in den Besitz von Passwörtern zu kommen.

Da viele Dienste gezielt einen vier- oder sechststelligen PIN-Code abfragen, gelten für diese kurzen Codes andere Regeln. PIN-Codes gelten als schwach, wenn sie zu einem der am häufigsten verwendeten PIN-Codes gehören, wenn sie eine auf- oder absteigende Ziffernfolge darstellen, z. B. „1234“ oder „8765“, oder wenn sie einem Wiederholungsmuster folgen, etwa „123123“ oder „123321“.

- Passwörter werden als *kompromittiert* gekennzeichnet, wenn die Funktion für die Passwortüberwachung belegen kann, dass sie Gegenstand eines Datenlecks waren. Weitere Informationen sind unter [Passwortüberwachung](#) zu finden.

Schwache, wiederholt verwendete und kompromittierte Passwörter werden entweder in der Liste gekennzeichnet (macOS) oder in der dedizierten Oberfläche für Sicherheitsempfehlungen präsentiert (iOS und iPadOS). Wenn sich der Benutzer in Safari bei einer Website anmeldet und dabei ein zuvor gesichertes Passwort verwendet, das sehr schwach ist oder durch ein Datenleck kompromittiert wurde, erscheint ein Warnhinweis mit der Aufforderung, ein Upgrade auf ein automatisch starkes Passwort zu vorzunehmen.

Aktualisieren der Authentifizierungssicherheit für Accounts in iOS und iPadOS

Durch die Implementierung von Account Authentication Modification Extensions (im Authentication Services Framework) können Apps Benutzer in die Lage versetzen, durch bloßes Tippen auf eine Taste Upgrades für passwortbasierte Accounts und den Wechsel zur Option „Mit Apple anmelden“ vorzunehmen oder automatisch starke Passwörter zu verwenden. Dieser Erweiterungspunkt ist für iOS und iPadOS verfügbar.

Wenn der Erweiterungspunkt in einer App implementiert ist und die App auf einem Gerät installiert ist, sehen Benutzer die Upgrade-Optionen der Erweiterung, wenn sie sich die Sicherheitsempfehlungen für Anmeldedaten zu ihrer App ansehen (im Passwortmanager des iCloud-Schlüsselbunds unter „Einstellungen“). Die Upgrades werden auch angeboten, wenn Benutzer für die Anmeldung bei einer App risikobehaftete Anmeldedaten verwenden. Apps können das System jedoch anweisen, Benutzern nach der Anmeldung die Upgrade-Optionen nicht zu präsentieren. Über die neue API AuthenticationServices können Apps außerdem eigene Erweiterungen aufrufen und selbst Upgrades ausführen – im Idealfall ausgehend von einem Bildschirm der App, der für Accounteinstellungen oder die Accountverwaltung verwendet wird.

Apps ist es freigestellt, ob sie die Upgrade-Option für „Automatisch starke Passwörter“ oder die Upgrade-Option für „Mit Apple anmelden“ oder beide Optionen nutzen wollen. Beim Upgrade auf die Option „Automatisch starke Passwörter“ generiert das System automatisch ein starkes Passwort für den Benutzer. Im Bedarfsfall kann die App eigene Passwortrichtlinien bereitstellen, die beim Generieren des neuen Passworts eingehalten werden müssen. Wenn ein Benutzer den Wechsel von einem passwortbasierten Account zur Option „Mit Apple anmelden“ vornimmt, stellt das System im Hinblick auf die Option „Mit Apple anmelden“ neue Anmeldedaten für die Erweiterung bereit, die diese dann dem Account zuordnet. Die E-Mail-Adresse zur Apple-ID des Benutzers wird nicht als Teil der Anmeldedaten bereitgestellt. Nach dem erfolgreichen Upgrade auf die Option „Mit Apple anmelden“ löscht das System die bisherigen Passwortdaten aus dem Schlüsselbund des Benutzers, sofern sie dort gesichert waren.

Account Authentication Modification Extensions bieten die Möglichkeit, eine zusätzliche Benutzerauthentifizierung vorzunehmen, bevor das Upgrade durchgeführt wird. Bei Upgrades, die über den Passwortmanager oder nach der Anmeldung an einer App initiiert werden, stellt die Erweiterung den Benutzernamen und das Passwort des Accounts bereit, für den das Upgrade vorgenommen werden soll. Bei Upgrades aus der App heraus wird nur der Benutzername bereitgestellt. Wenn die Erweiterung eine weitere Benutzerauthentifizierung erfordert, kann sie hierfür eine eigene spezifische Oberfläche einblenden, bevor das Upgrade initiiert wird. Sinn und Zweck dieses Vorgehens ist es, den Benutzer zur Eingabe eines zweiten Authentifizierungsfaktors zu zwingen, um das Upgrade zu autorisieren.

Passwortüberwachung

Die Passwortüberwachung (Password Monitoring) ist eine Funktion, die Passwörter, die im Schlüsselbund zum automatischen Ausfüllen von Passwörtern hinterlegt sind, mit einer kuratierten und fortlaufend aktualisierten Liste solcher Passwörter abgleicht, die infolge von Datenlecks bei anderen Online-Organisationen kompromittiert wurden. Wenn die Funktion aktiviert ist, gleicht das Überwachungsprotokoll fortlaufend die Passwörter im Schlüsselbund für das automatische Ausfüllen von Passwörtern mit der kuratierten Liste ab.

So funktioniert die Überwachung

Das Gerät des Benutzers führt fortlaufend Round-Robin-Prüfungen für die Passwörter eines Benutzers durch, wobei das Abfrageintervall unabhängig von den Passwörtern des Benutzers oder der Verwendungsmuster eines Passwortmanagers ist. Auf diese Weise wird sichergestellt, dass jeder einzelne Prüfstatus stets mit dem neuesten Stand der aktuellen kuratierten Liste kompromittierter Passwörter korrespondiert. Um Informations- und Datenlecks im Zusammenhang damit vorzubeugen, wie viele eindeutige Passwörter ein Benutzer hat, werden die Anfragen zu Stapeln zusammengefasst und parallel ausgeführt. Bei jeder Prüfung wird eine fest vorgegebene Anzahl von Passwörtern parallel überprüft. Liegt die Anzahl der Passwörter eines Benutzers unter dieser vorgegebenen Anzahl, werden zum Ausgleich der Differenz zufällige Passwörter generiert und zu den Anfragen hinzugefügt.

Wie Passwörter abgeglichen werden

Das Abgleichen der Passwörter erfolgt in einem zweiteiligen Prozess. Die am häufigsten kompromittierten Passwörter befinden sich in einer lokalen Liste auf dem Gerät des Benutzers. Wenn das Passwort des Benutzers in dieser Liste auftaucht, wird der Benutzer sofort und ohne weitere Interaktion von außen benachrichtigt. Dadurch wird sichergestellt, dass keine Informationen über Passwörter des Benutzers nach außen dringen, die infolge einer Passwortlücke ein besonders hohes Risiko darstellen.

Wenn das Passwort nicht in der Liste der sehr häufig kompromittierten Passwörter enthalten ist, wird es mit einer Liste seltener kompromittierter Passwörter abgeglichen.

Vergleich von Benutzerpasswörtern mit einer kuratierten Liste

Für die Prüfung, ob ein nicht in der lokalen Liste enthaltenes Passwort zu einer Übereinstimmung mit der externen Quelle führt, ist die Interaktion mit Apple-Servern erforderlich. Damit sichergestellt ist, dass legitime Passwörter des Benutzers nicht an Apple gesendet werden, wird ein kryptografisches *privates Set als Schnittmenge* bereitgestellt, sodass die Passwörter des Benutzers mit einer großen Anzahl kompromittierter Passwörter abgeglichen werden. Auf diese Weise wird sichergestellt, dass über Passwörter, bei denen das Risiko, kompromittiert zu werden, gering ist, nur wenige Informationen mit Apple geteilt werden. Genau genommen sind bei Benutzerpasswörtern diese Informationen auf das 15-Bit-Präfix eines kryptografischen Hash-Werts beschränkt. Durch das Heraushalten der am häufigsten kompromittierten Passwörter aus diesem interaktiven Prozess und durch die Verwendung der lokalen Liste häufig kompromittierter Passwörter verringert sich das Delta bei der relativen Häufigkeit der Passwörter in den Buckets der Webdienste. Dies erschwert es, die Passwörter von Benutzern aus diesen Listen abzuleiten.

Das zugrunde liegende Protokoll partitioniert die Liste der kuratierten Passwörter, die zum Zeitpunkt der Redaktion dieses Dokuments ca. 1,5 Milliarden Passwörter umfasst, in 2^{15} einzelne Buckets. Das Bucket, zu dem ein Passwort zugewiesen wird, wird anhand der ersten 15 Bit des SHA-256-Hash-Werts des Passworts ermittelt. Außerdem wird jedes kompromittierte Passwort („pw“) einem Punkt auf der elliptischen NIST P256-Kurve zugeordnet: $P_{pw} = \alpha \cdot H_{SWU}(pw)$ – Dabei ist „ α “ ein geheimer zufälliger Schlüssel, der nur Apple bekannt ist, und H_{SWU} eine Orakelfunktion (Zufallsgenerator), die Passwörter nach der Shallue-van de Woestijne-Ulas-Methode Kurvenpunkten zuordnet. Diese Transformation ist darauf ausgelegt, mit rechnerischen Mitteln die Werte von Passwörtern zu verbergen und die Offenlegung neu kompromittierter Passwörter im Zuge der Passwortüberwachung zu verhindern.

Zur Berechnung der Schnittmenge für das private Set ermittelt das Gerät des Benutzers den Bucket, zu dem das Benutzerpasswort gehört; hierfür wird „ λ “, das 15-Bit-Präfix von SHA-256(upw) verwendet, wobei „upw“ für eines der Passwörter des Benutzers steht. Das Gerät generiert eine eigene Zufallskonstante „ β “ und sendet den Punkt „ $P_c = \beta \cdot H_{SWU}(upw)$ “ zusammen mit einer Anforderung für den Bucket, der „ λ “ entspricht, an den Server. „ β “ bewirkt, dass Apple die Informationen über das Benutzerpasswort vorenthalten werden und dass die Informationen, die Apple gegenüber offengelegt werden, auf „ λ “ beschränkt bleiben. Der Server übernimmt den vom Gerät des Benutzers gesendeten Punkt, berechnet „ $\alpha P_c = \alpha \beta \cdot H_{SWU}(upw)$ “ und meldet das Ergebnis zusammen mit dem zugehörigen Bucket von Punkten – „ $B_\lambda = \{ P_{pw} \mid \text{SHA-256}(pw) \text{ beginnt mit Prefix } \lambda \}$ “ – zurück an das Gerät.

Anhand dieser zurückgemeldeten Informationen kann das Gerät „ $B'_\lambda = \{ \beta \cdot P_{pw} \mid P_{pw} \in B_\lambda \}$ “ berechnen und im Falle, dass „ $\alpha P_c \in B'_\lambda$ “ zutrifft, bestätigen, dass das Passwort des Benutzers Gegenstand eines Datenlecks war und kompromittiert wurde.

Senden von Passwörtern an andere Benutzer oder Apple Geräte

Apple sendet Passwörter per AirDrop sicher an andere Benutzer oder Apple-Geräte oder an Apple TV.

Sichern von Anmeldedaten auf einem anderen Gerät per AirDrop

Wenn iCloud aktiviert ist, können Benutzer gesicherte Anmeldedaten via AirDrop an ein anderes Gerät übertragen. Zu den Anmeldedaten gehören Benutzername und Passwort sowie die Websites, für die sie gesichert wurden. Die Übertragung von Anmeldedaten mit AirDrop erfolgt immer im Modus „Nur Kontakte“, unabhängig von den Einstellungen des Benutzers. Die Anmeldedaten werden nach der Einwilligung des Benutzers auf dem Empfängergerät im Schlüsselbund zum automatischen Ausfüllen von Passwörtern dieses Benutzers gespeichert.

Ausfüllen von Anmeldedaten in Apple TV

Mit der Option zum automatischen Ausfüllen von Passwörtern werden Anmeldedaten in Apps auf Apple TV ausgefüllt. Sobald der Benutzer in tvOS ein Textfeld für den Benutzernamen oder das Passwort auswählt, stellt Apple TV eine „Automatisch Ausfüllen“-Anfrage via Bluetooth Low Energy (BLE).

Ein in der Nähe befindliches Gerät (iPhone, iPad oder iPod touch) zeigt daraufhin eine Aufforderung an und lädt den Benutzer ein, Anmeldedaten mit Apple TV zu teilen. Die Verschlüsselungsmethode wird wie folgt eingerichtet:

- Wenn das Gerät und Apple TV denselben iCloud-Account verwenden, erfolgt die Verschlüsselung zwischen den Geräten automatisch.
- Wenn das Gerät bei einem iCloud-Account angemeldet ist, der nicht auch von Apple TV verwendet wird, erhält der Benutzer die Aufforderung, eine verschlüsselte Verbindung mithilfe eines PIN-Codes herzustellen. Das iPhone muss entsperrt sein und sich in unmittelbarer Nähe der mit diesem Apple TV gekoppelten Siri Remote-Fernbedienung befinden, um diese Aufforderung zu empfangen.

Nachdem die verschlüsselte Verbindung mit der BLE-Link-Verschlüsselung hergestellt wurde, werden die Anmeldedaten an Apple TV gesendet und automatisch in die relevanten Textfelder der App eingesetzt.

Erweiterungen für Credential-Provider

In iOS, iPadOS und macOS können Benutzer eine App eines teilnehmenden Fremdanbieters zum Credential-Provider für die Funktion „Passwort automatisch ausfüllen“ bestimmen; (in iOS und iPadOS kann dies im Bereich „Einstellungen“ > „Passwörter“ und in macOS im Bereich „Systemeinstellungen“ > „Erweiterungen“ bewerkstelligt werden). Dieser Mechanismus ist in App-Erweiterungen integriert. Die Erweiterung für Credential-Provider muss eine Darstellung für die Auswahl von Anmeldedaten bereitstellen. Optional kann die Erweiterung auch Metadaten über gesicherte Anmeldedaten zur Verfügung stellen, sodass diese direkt in der QuickType-Leiste (iOS und iPadOS) oder in einer Vorschlagsliste für die automatische Vervollständigung (macOS) angeboten werden können. Diese Metadaten umfassen die Website der Anmeldedaten und den zugehörigen Benutzernamen, nicht aber das zugehörige Passwort. iOS, iPadOS und macOS kommunizieren mit der Erweiterung, um das Passwort abzurufen, wenn der Benutzer Anmeldeinformationen in einer App oder auf einer Website in Safari ausfüllen möchte. Die Metadaten für die Anmeldedaten werden im Container der App des Credential-Providers gespeichert; sie werden automatisch entfernt, wenn die zugehörige App deinstalliert wird.

iCloud-Schlüsselbund

Sicherheit des iCloud-Schlüsselbunds – Übersicht

iCloud bietet Benutzern die Möglichkeit, ihre Passwörter sicher zwischen iOS-, iPadOS- und Mac-Geräten zu synchronisieren, ohne dass die Informationen für Apple offengelegt werden. Zusätzlich zu Datenschutz und Sicherheit waren Benutzerfreundlichkeit und Wiederherstellbarkeit eines Schlüsselbunds weitere Ziele, die sich nachhaltig auf die Konzeption und Architektur des iCloud-Schlüsselbunds ausgewirkt haben. Der iCloud-Schlüsselbund umfasst zwei Dienste: Schlüsselbund synchronisierung und Schlüsselbundwiederherstellung.

Apple hat den iCloud-Schlüsselbund und die Funktion zur Wiederherstellung des Schlüsselbunds so konzipiert, dass die Passwörter selbst unter den folgenden Umständen sicher sind:

- Der iCloud-Account eines Benutzers wurde kompromittiert.
- iCloud wird von einem Angreifer von außen oder einem Mitarbeiter kompromittiert.
- Ein anderer Anbieter greift auf Benutzeraccounts zu.

Integration des Passwortmanagers mit dem iCloud-Schlüsselbund

iOS, iPadOS und macOS können automatisch kryptografisch starke, randomisierte Zeichenfolgen (Strings) generieren, um sie als Account-Passwörter in Safari zu verwenden. iOS und iPadOS können auch starke Passwörter für Apps erstellen. Die generierten Passwörter werden im [Schlüsselbund](#) gespeichert und geräteübergreifend synchronisiert. Schlüsselbundobjekte werden von einem Gerät über die Apple-Server auf ein anderes Gerät übertragen, werden dabei aber so verschlüsselt, dass weder Apple noch andere Geräte den Inhalt lesen können.

Sichere Schlüsselbundsynchronisierung

Wenn ein Benutzer den iCloud-Schlüsselbund erstmals aktiviert, richtet das Gerät einen „Circle of Trust“ (einen Kreis von Vertrauensbeziehungen) ein und erstellt für sich selbst eine Identität für die Synchronisierung. Diese Synchronisierungsidentität besteht aus einem privaten und einem öffentlichen Schlüssel und wird im Schlüsselbund des Geräts gespeichert. Der öffentliche Schlüssel dieser Synchronisierungsidentität wird Teil des „Circle of Trust“. Der Kreis wird zwei Mal signiert: zuerst mit dem privaten Schlüssel der Synchronisierungsidentität und anschließend mittels eines Schlüssels, der (unter Verwendung von P-256) per asymmetrischer elliptischer Kurven-Kryptografie vom Passwort des iCloud-Accounts des Benutzers abgeleitet wird. Außerdem werden im Circle of Trust die Parameter gespeichert, mit denen der Schlüssel aus dem Passwort für den iCloud-Account erstellt wurde (zufälliger Salt-Wert und Iterationen).

Bei Accounts mit Zwei-Faktor-Authentifizierung wird ein ähnlicher zusätzlicher Synchronisationskreis erstellt und in CloudKit gespeichert. Die Geräteidentitäten in diesem System bestehen aus zwei Paaren von asymmetrischen elliptischen Schlüsseln (die P-384 verwenden), die ebenfalls im Schlüsselbund gespeichert werden. Jedes Gerät verwaltet eine eigene Liste mit Identitäten, denen es vertraut, und signiert diese Liste mit einem seiner Identitätsschlüssel.

iCloud-Speicher des Synchronisationskreises

Der signierte Circle of Trust wird im Schlüssel-Wert-Speicherbereich des Benutzers gespeichert. Er kann ohne Kenntnis des iCloud-Passworts des Benutzers nicht ausgelesen und ohne den privaten Schlüssel der Synchronisierungsidentität der Mitglieder nicht modifiziert werden.

Bei Accounts mit Zwei-Faktor-Authentifizierung wird die Synchronisationsliste jedes Geräts in CloudKit gespeichert. Die Listen können ohne Kenntnis des iCloud-Passworts des Benutzers nicht ausgelesen und ohne die privaten Schlüssel des Geräts, dem sie gehören, nicht gültig verändert werden.

Verfahrensweise zum Hinzufügen anderer Geräte zum Circle of Trust für die Synchronisierung

Neue Geräte treten dem Synchronisationskreis des iCloud-Schlüsselbunds auf einer von zwei möglichen Arten bei, wenn sie sich bei iCloud anmelden: **entweder** durch Kopplung mit bestehendem iCloud-Schlüsselbund-Gerät und dessen Unterstützung oder durch Verwenden der iCloud-Schlüsselbund-Wiederherstellung.

Während des Kopplungsvorgangs erstellt das Anwärtergerät neue Synchronisationsidentitäten für den Synchronisationskreis und die Synchronisationslisten (für Accounts mit Zwei-Faktor-Authentifizierung) und legt sie dem Unterstützengerät vor. Das Unterstützengerät fügt den öffentlichen Schlüssel des Anwärtergeräts dem Circle of Trust hinzu und signiert ihn erneut mit der Synchronisierungsidentität und dem Schlüssel, der aus dem iCloud-Passwort des Benutzers hergeleitet wird. Der neue Synchronisationskreis wird in iCloud gespeichert und dort analog vom neuen Mitglied signiert. Bei Accounts mit Zwei-Faktor-Authentifizierung stellt das Unterstützengerät dem Anwärtergerät einen *Voucher* bereit, der mit seinen Identitätsschlüsseln signiert ist. Dadurch wird angezeigt, dass dem Anwärtergerät vertraut werden kann. Im Anschluss aktualisiert es dann seine eigene Liste mit vertrauenswürdigen Synchronisationsidentitäten, um das Anwärtergerät aufzunehmen.

Der Signierungskreis hat nun zwei Mitglieder und jedes Mitglied besitzt den öffentlichen Schlüssel des jeweils anderen Mitglieds. Die Mitglieder tauschen nun untereinander einzelne Schlüsselbundobjekte über den Schlüssel-Wert-Speicher von iCloud aus oder speichern sie bei Bedarf in CloudKit. Wenn beide Mitglieder des Kreislaufs Updates für dasselbe Objekt haben, wird entweder das eine oder das andere ausgewählt, um alle auf dem gleichen Stand zu halten. Jedes synchronisierte Objekt wird verschlüsselt. Es kann also nur von einem Gerät im Circle of Trust des Benutzers entschlüsselt werden. Es kann nicht von anderen Geräten oder von Apple entschlüsselt werden.

Dieser „Beitrittsprozess“ wird für jedes Gerät wiederholt, das dem Synchronisationskreis beiträgt. Wenn zum Beispiel ein drittes Gerät beiträgt, kann es entweder mit dem einen oder dem anderen Gerät gekoppelt werden. Werden neue Mitglieder hinzugefügt, synchronisieren sich alle Mitglieder mit dem neuen Mitglied. So soll sichergestellt werden, dass alle Mitglieder über dieselben Schlüsselbundobjekte verfügen.

Einschränken der Synchronisierung auf bestimmte Objekte

Einige Schlüsselbundobjekte wie zum Beispiel iMessage-Schlüssel sind gerätespezifisch und müssen daher auf dem Gerät verbleiben. Aus diesem Grund muss jedes zu synchronisierende Objekt explizit mit dem Attribut `kSecAttrSynchronizable` markiert werden.

Apple setzt dieses Attribut für Safari-Benutzerdaten (dazu gehören Benutzernamen, Passwörter und Kreditkartennummern) sowie für WLAN-Passwörter, HomeKit-Verschlüsselungsschlüssel und andere Schlüsselbundobjekte, die die Ende-zu-Ende-iCloud-Verschlüsselung unterstützen.

Außerdem werden Schlüsselbundobjekte, die von Apps anderer Anbieter hinzugefügt wurden, standardmäßig nicht synchronisiert. Entwickler müssen das Attribut „`kSecAttrSynchronizable`“ verwenden, wenn Objekte dem Schlüsselbund hinzugefügt werden sollen.

Sichere Wiederherstellung des iCloud-Schlüsselbunds

Der iCloud-Schlüsselbund ermöglicht es einem Benutzer, seine Schlüsselbunddaten bei Apple treuhänderisch zu hinterlegen (Escrow), *ohne* dass Apple die Passwörter und andere darin enthaltene Daten lesen kann. Auch wenn ein Benutzer nur ein einzelnes Gerät hat, bietet die Funktion zur Wiederherstellung des Schlüsselbunds Schutz vor möglichem Datenverlust. Dies ist insbesondere wichtig, wenn Safari verwendet wird, um zufällige starke Passwörter für Webaccounts zu generieren, weil diese Passwörter ausschließlich im Schlüsselbund hinterlegt werden.

Ein Eckpfeiler der Schlüsselbundwiederherstellung ist die sekundäre Authentifizierung und ein sicherer Treuhanddienst (Escrow), der von Apple speziell für diese Funktion erstellt wurde. Der Schlüsselbund des Benutzers wird mit einem sicheren Code verschlüsselt, und der Treuhanddienst stellt nur dann eine Kopie des Schlüsselbunds bereit, wenn eine Reihe strikter Bedingungen erfüllt ist.

Verwenden der sekundären Authentifizierung

Es gibt verschiedene Möglichkeiten, ein starkes Passwort einzurichten:

- Wenn der iCloud-Schlüsselbund und die Zwei-Faktor-Authentifizierung für den Benutzeraccount aktiviert sind, wird der Code des Geräts verwendet, um einen hinterlegten (Escrow) Schlüsselbund wiederherzustellen.
- Wenn die Zwei-Faktor-Authentifizierung nicht eingerichtet ist, wird der Benutzer gebeten, einen iCloud-Sicherheitscode zu erstellen, indem er einen sechsstelligen Code eingibt. Benutzer ohne Zwei-Faktor-Authentifizierung können aber auch eigene, längere Codes erstellen oder von ihren Geräten einen zufälligen kryptografischen Code erzeugen lassen, den sie selbst speichern.

Treuhänderische Hinterlegung des Schlüsselbunds

Nachdem der Code erstellt wurde, wird der Schlüsselbund treuhänderisch bei Apple gespeichert. Das iOS-, iPadOS- oder macOS-Gerät exportiert zunächst eine Kopie des Schlüsselbunds des Benutzers und verschlüsselt diesen dann, wobei die Schlüssel in einem asymmetrischen [Keybag](#) verpackt und in iCloud im Schlüssel-Wert-Speicher des Benutzers abgelegt werden. Der Keybag wird mit dem iCloud-Sicherheitscode des Benutzers und dem öffentlichen Schlüssel [des HSM-Clusters \(Hardware Sicherheitsmodul\)](#) verpackt, der den Escrow-Eintrag speichert. Daraus setzt sich der *iCloud Escrow-Eintrag* des Benutzers zusammen. Bei Accounts mit Zwei-Faktor-Authentifizierung wird der Schlüsselbund außerdem im CloudKit gespeichert und zu Zwischenschlüsseln verpackt, die nur mit dem Inhalt des in iCloud vorliegenden Escrow-Eintrags wiederhergestellt werden können. Dies entspricht einem gleich hohen Schutzniveau.

Die Inhalte des Escrow-Eintrags ermöglichen dem Wiederherstellungsgerät auch den erneuten Beitritt zum iCloud-Schlüsselbund und bestätigen gegenüber jedem vorhandenen Gerät, dass das Wiederherstellungsgerät den Escrow-Prozess erfolgreich durchgeführt hat und deshalb vom Eigentümer des Accounts autorisiert ist.

Hinweis: Wenn der Benutzer sich dafür entscheidet, einen zufälligen kryptografischen Sicherheitscode anstelle eines eigenen oder eines vierstelligen Werts zu akzeptieren, wird kein Escrow-Eintrag benötigt. Stattdessen wird der zufällige Schlüssel direkt mit dem iCloud-Sicherheitscode verpackt.

Über die Erstellung eines Sicherheitscodes hinaus müssen Benutzer eine Telefonnummer registrieren. Dies etabliert eine zweite Authentifizierungsstufe für die Wiederherstellung des Schlüsselbunds. Benutzer erhalten eine SMS-Nachricht, auf die sie antworten müssen, damit die Wiederherstellung durchgeführt werden kann.

Escrow-Sicherheit für iCloud-Schlüsselbund

Mit der sicheren iCloud-Infrastruktur wird beim treuhänderischen Hinterlegen des Schlüsselbunds (Escrow) sichergestellt, dass nur autorisierte Benutzer und Geräte eine Wiederherstellung durchführen können. Hinter iCloud stehen HSM-Cluster (Hardware Security Modules), mit denen die Escrow-Einträge geschützt werden. Jedes besitzt einen Schlüssel, mit dem die geschützten Escrow-Einträge verschlüsselt werden, wie bereits in diesem Dokument beschrieben.

Um einen Schlüsselbund wiederherzustellen, muss sich der Benutzer mit seinem iCloud-Account, seinem Passwort und dem Code aus der an die registrierte Telefonnummer gesendeten SMS authentifizieren. Im Anschluss daran muss der Benutzer seinen iCloud-Sicherheitscode eingeben. Der HSM-Cluster überprüft mit dem SRP-Protokoll (Secure Remote Password), ob der Benutzer den iCloud-Sicherheitscode kennt. Der Code selbst wird nicht an Apple gesendet. Alle Bestandteile des Clusters verifizieren unabhängig voneinander, dass der Benutzer die maximale Anzahl zulässiger Versuche, die zum Abrufen des Eintrags zulässig sind, nicht überschritten hat (siehe nachfolgende Beschreibung). Wenn dies mehrheitlich bestätigt wird, entpackt der Cluster den Escrow-Eintrag und sendet ihn an das Gerät des Benutzers.

Anschließend verwendet das Gerät den iCloud-Sicherheitscode, um die zufälligen Schlüssel zu entpacken, mit denen der Schlüsselbund des Benutzers verschlüsselt wurde. Mit diesem Schlüssel wird der Schlüsselbund – der aus iCloud aus dem Schlüssel-Wert-Speicher oder aus CloudKit abgerufen wird – entschlüsselt und auf dem Gerät wiederhergestellt. iOS, iPadOS und macOS erlauben maximal 10 Versuche, einen Escrow-Eintrag zu authentifizieren und abzurufen. Nach mehreren fehlgeschlagenen Versuchen wird der Eintrag gesperrt und der Benutzer muss sich für zusätzliche Versuche an den Apple Support wenden. Nach dem 10. fehlgeschlagenen Versuch löscht der HSM-Cluster den Escrow-Eintrag, sodass der Schlüsselbund unwiderruflich verloren ist. Diese Vorgehensweise bietet Schutz vor Brute-Force-Angriffen auf den Eintrag – wenngleich zum Preis, dass die Schlüsselbunddaten geopfert werden müssen.

Diese Richtlinien sind in der HSM-Firmware codiert. Die Zugangskarten, mit denen die Firmware geändert werden kann, wurden zerstört. Alle Versuche, die Firmware zu ändern oder auf den privaten Schlüssel zuzugreifen, führen dazu, dass der HSM-Cluster den privaten Schlüssel löscht. In diesem Fall erhält der Eigentümer jedes mit dem Cluster geschützten Schlüsselbunds eine Mitteilung, dass der Escrow-Eintrag gelöscht wurde. Er kann sich dann erneut registrieren.

Apple Pay

Apple Pay-Sicherheit – Übersicht

Mit Apple Pay können Benutzer mit einem unterstützten iPhone-, iPad-, Mac- oder Apple Watch-Gerät schnell, sicher und vertraulich in Ladengeschäften, in Apps und im Web in Safari bezahlen. Benutzer können auch für Apple Pay aktivierte ÖPNV-Karten und Studierendenausweise sowie Zugangskarten zu Apple Wallet hinzufügen. Dies ist nicht nur praktisch, sondern bietet auch in Hard- und Software integrierte Sicherheit.

Apple Pay ist auch dafür ausgelegt, die persönlichen Daten des Benutzers zu schützen. Apple Pay sammelt keine Informationen über Transaktionen, die mit dem Benutzer verknüpft werden können. Die Transaktion findet zwischen dem Benutzer, dem Händler und dem Kartenaussteller statt.

Sicherheit bei Apple Pay-Komponenten

Apple Pay nutzt verschiedene Hardware- und Softwarefunktionen, um sichere, zuverlässige Käufe zu ermöglichen.

Secure Element

Secure Element ist ein zertifizierter Chip, der die Branchenstandards erfüllt und auf dem die Java Card-Plattform ausgeführt wird, die mit den Anforderungen für elektronische Zahlungen der Finanzbranche kompatibel ist. Der Secure Element-IC und die Java Card-Plattform sind gemäß dem Prozess EMVCo Security Evaluation zertifiziert. Nachdem die Sicherheitsevaluierung erfolgreich abgeschlossen wurde, gibt EMVCo eindeutige Zertifikate für den IC und die Plattform aus.

Der Secure Element-IC wurde basierend auf dem CC-Standard (Common Criteria) zertifiziert. Weitere Informationen sind unter [Sicherheitszertifizierungen für den Secure Enclave-Prozessor](#) auf der Website für Zertifizierungen der Apple-Plattformen zu finden.

NFC-Controller

Der NFC-Controller verarbeitet die NFC-Protokolle (Near Field Communication) und steuert die Kommunikation zwischen Anwendungsprozessor und Secure Element bzw. zwischen Secure Element und Kassenterminal.

Apple Wallet

Die Apple Wallet-App wird verwendet, um Kredit-, Debit- und Kundenkarten hinzuzufügen und zu verwalten und Zahlungen mit Apple Pay abzuwickeln. Der Benutzer kann in Apple Wallet seine Karten sehen und möglicherweise auch vom Kartenaussteller bereitgestellte Zusatzinformationen (z. B. dessen Datenschutzrichtlinien, die letzten Zahlungen und weitere Informationen). Karten für Apple Pay können außerdem hier hinzugefügt werden:

- Im Systemassistenten und in den Einstellungen für iOS und iPadOS
- In der App „Watch“ für die Apple Watch
- Im Bereich „Wallet & Apple Pay“ in den Systemeinstellungen für Mac-Computer mit Touch ID.

Darüber hinaus können Benutzer Apple Wallet verwenden, um u. a. ÖPNV-Karten, Bonuskarten, Bordkarten, Eintrittskarten, Geschenkkarten, Studierendenausweise und Zugangskarten hinzuzufügen und zu verwalten.

Secure Enclave

Auf einem iPhone, einem iPad, einer Apple Watch, auf Mac-Computern mit Touch ID und Mac-Computern mit Apple Chips, die das Magic Keyboard mit Touch ID verwenden, führt die Secure Enclave das Authentifizierungsverfahren durch und ermöglicht die Abwicklung der Zahlung.

Die Apple Watch muss entsperrt sein und der Benutzer muss die Seitentaste doppelklicken. Der Doppelklick wird erkannt und ohne Umweg über den Anwendungsprozessor direkt an das Secure Element oder die Secure Enclave (sofern verfügbar) weitergeleitet.

Apple Pay-Server

Die Apple Pay-Server verwalten die Konfiguration und Bereitstellung der Kredit-, Debit- und ÖPNV-Karten, der Studierendenausweise und der Zugangskarten in Apple Wallet. Die Server verwalten auch die im Secure Element gespeicherten DANs (Device Account Number). Sie kommunizieren sowohl mit dem Gerät als auch mit den Zahlungsservern der Kartenaussteller im Netzwerk. Die Apple Pay-Server sind auch für die Neuverschlüsselung der Zahlungsdaten in den Apps bzw. im Web zuständig.

Wie Apple Pay die Käufe von Benutzern schützt

Secure Element

Das Secure Element nutzt ein extra entwickeltes Applet zur Verwaltung von Apple Pay. Außerdem besitzt es Applets, die von den Zahlungsnetzwerken oder Kartenausstellern zertifiziert werden. Die Daten der Kredit-, Debit- oder Prepaid-Karte werden vom Zahlungsnetzwerk oder dem Kartenaussteller in verschlüsselter Form an diese Applets gesendet, wobei Schlüssel verwendet werden, die nur das Zahlungsnetzwerk oder der Kartenaussteller und die Sicherheits-Domain des Applets kennen. Diese Daten werden in den Applets gespeichert und mit den Sicherheitsfunktionen des Secure Element geschützt. Bei einer Transaktion kommuniziert das Terminal über den NFC-Controller und einen eigenen Hardware-Bus direkt mit dem Secure Element.

NFC-Controller

Als Tor zum Secure Element sorgt der NFC-Controller dafür, dass alle kontaktlosen Zahlungen über ein Kassenterminal in unmittelbarer Nähe des Geräts durchgeführt werden. Nur Zahlungsanfragen von Terminals in unmittelbarer Nähe werden vom NFC-Controller als kontaktlose Zahlungen markiert.

Nachdem eine Zahlung mit einer Kredit-, Debit- oder Prepaid-Karte (einschließlich Kundenkarten) vom Besitzer der Karte per Face ID, Touch ID oder Code oder auf einer entsperrten Apple Watch durch Doppelklicken auf die Seitentaste in der Secure Enclave autorisiert wurde, werden die kontaktlosen Antworten der Zahlungs-Applets im Secure Element über den Controller exklusiv an das NFC-Feld geleitet. Daher verbleiben die Daten der Zahlungsanweisung für kontaktlose Zahlungstransaktionen im lokalen NFC-Feld und werden niemals dem Anwendungsprozessor gegenüber offengelegt. Die Daten von Zahlungsanweisungen in Apps und im Web werden dagegen über den Anwendungsprozessor an den Apple Pay-Server gesendet, allerdings erst nach der Verschlüsselung durch das Secure Element.

Kredit-, Debit- und Prepaid-Karten

Sicherheit bei der Kartenbereitstellung – Übersicht

Wenn der Benutzer in Apple Wallet eine Kredit-, Debit- oder Prepaid-Karte (Kundenkarten inbegriffen) hinzufügt, sendet Apple die Kartendaten zusammen mit anderen Informationen über den Benutzeraccount und das Gerät über eine sichere Verbindung an den jeweiligen Kartenaussteller oder den autorisierten Dienstleister des Kartenausstellers. Anhand dieser Informationen bestimmt der Kartenaussteller, ob die Karte für Apple Wallet hinzugefügt werden kann. Apple Pay verwendet zum Zweck der Kartenbereitstellung drei serverseitige Aufrufe, um mit dem Kartenaussteller oder dem Netzwerk zu kommunizieren:

- Erforderliche Felder
- Kartenüberprüfung
- Verknüpfen und Bereitstellen

Der Kartenaussteller bzw. das Netzwerk verwendet diese Aufrufe, um Karten für Apple Wallet zu verifizieren, zu bestätigen und hinzuzufügen. Für den Datentransfer verwenden die Client-Server-Sitzungen TLS 1.2.

Die vollständige Kartenummer wird weder auf dem Gerät noch auf den Apple Pay-Servern gespeichert. Stattdessen wird eine eindeutige DAN (Device Account Number) erstellt, verschlüsselt und im Secure Element gespeichert. Diese eindeutige DAN wird so verschlüsselt, dass Apple nicht darauf zugreifen kann. Diese DAN ist eindeutig und unterscheidet sich von anderen Kredit- oder Debitkartenummern darin, dass der Kartenaussteller oder das Zahlungsnetzwerk verhindern kann, dass sie auf einer Magnetkarte, am Telefon oder auf Websites verwendet wird. Die DAN (Device Account Number) im Secure Element wird niemals auf Apple Pay-Servern gespeichert oder in iCloud gesichert und ist isoliert von iOS-, iPadOS-, watchOS- und Mac-Geräten mit Touch ID.

Karten zur Verwendung mit der Apple Watch werden für Apple Pay mit der App „Watch“ auf dem iPhone oder innerhalb der iPhone-App eines Kartenausstellers aktiviert. Um eine Karte zur Apple Watch hinzuzufügen, muss sich die Uhr im Bluetooth-Kommunikationsradius befinden. Karten, die speziell für die Verwendung mit der Apple Watch aktiviert wurden, haben eine eigene DAN, die im Secure Element auf der Apple Watch gespeichert ist.

Wenn Kredit-, Debit- oder Prepaid-Karten (einschließlich Kundenkarten) hinzugefügt werden, erscheinen sie in einer Kartenliste im Systemassistenten auf Geräten, die bei demselben iCloud-Account angemeldet sind. Diese Karten verbleiben in dieser Liste, solange sie auf mindestens einem Gerät aktiv sind. Sieben Tage, nachdem die Karten von allen Geräten entfernt wurden, werden sie aus dieser Liste entfernt. Diese Funktion setzt voraus, dass die Zwei-Faktor-Authentifizierung in dem jeweiligen iCloud-Account aktiviert ist.

Hinzufügen von Kredit- oder Debitkarten zu Apple Pay

Kreditkarten können auf Apple-Geräten manuell zu Apple Pay hinzugefügt werden.

Manuelles Hinzufügen von Kredit- oder Debitkarten

Beim manuellen Hinzufügen einer Karte werden für den Bereitstellungsprozess der Name des Kartenhalters, die Kartenummer, das Ablaufdatum und die CVV-Nummer verwendet. Der Benutzer kann die Informationen in den Einstellungen, in Apple Wallet oder in der Apple Watch-App eintippen oder mithilfe der Kamera des Geräts erfassen. Wenn die Kamera die Karteninformationen erfasst, versucht Apple, Name, Kartenummer und Ablaufdatum einzugeben. Das Foto wird nicht auf dem Gerät oder in der Foto-Mediathek gespeichert. Nachdem alle Felder ausgefüllt wurden, überprüft der Kartenüberprüfungsprozess (Check Card) alle Felder mit Ausnahme der CVV-Nummer. Sie werden dann verschlüsselt und an den Apple Pay-Server gesendet.

Wenn der Kartenüberprüfungsprozess eine ID für die Geschäftsbedingungen umfasst, lädt Apple die Geschäftsbedingungen des jeweiligen Kartenausstellers und zeigt sie dem Benutzer an. Wenn der Benutzer die Geschäftsbedingungen annimmt, übergibt Apple die ID der Geschäftsbedingungen sowie die CVV an den Prozess zum Verknüpfen und Bereitstellen. Apple stellt als Teil des Prozesses zum Verknüpfen und Bereitstellen auch Informationen über das Gerät für den Kartenaussteller oder das Netzwerk bereit. Hierzu gehören Informationen über (a) Aktivitäten des iTunes- und App Store-Accounts des Benutzers (z. B. die Anzahl seiner Transaktionen in iTunes), (b) Informationen über das Gerät des Benutzers (z. B. Telefonnummer, Name und Modell des Geräts sowie weitere für das Einrichten von Apple Pay erforderliche Apple-Geräte) und (c) den ungefähren Standort des Benutzers zu der Zeit, zu der seine Karte hinzugefügt wurde, sofern die Ortungsdienste aktiviert sind. Anhand dieser Informationen bestimmt der Kartenaussteller, ob die Karte für Apple Pay hinzugefügt werden kann.

Der Prozess zum Verknüpfen und Bereitstellen löst zwei Ereignisse aus:

- Das Gerät beginnt mit dem Laden der Apple Wallet-Kartendatei für die jeweilige Kredit- bzw. Debitkarte.
- Das Gerät bindet die Karte an das Secure Element.

Die Kartendatei enthält URLs zum Laden von Kartengrafiken, Metadaten zur Karte wie Kontaktinformationen, zugehörige App des Kartenausstellers und unterstützte Funktionen. Außerdem enthält sie den Kartenstatus, der Informationen darüber enthält, ob das Secure Element vollständig personalisiert wurde, ob die Karte aktuell durch den Kartenaussteller gesperrt ist oder ob eine zusätzliche Verifizierung nötig ist, bevor in Apple Pay mit dieser Karte gezahlt werden kann.

Hinzufügen von Kredit- oder Debitkarten, die für einen iTunes Store-Account verwendet werden

Bei Kredit- oder Debitkarten, die bereits für iTunes verwendet werden, muss der Benutzer das Passwort für seine Apple-ID erneut eingeben. Die Kreditkartennummer wird von iTunes abgerufen, und der Kartenüberprüfungsprozess „Check Card“ wird gestartet. Wenn die Karte für Apple Pay verwendet werden kann, lädt das Gerät die Geschäftsbedingungen, zeigt sie an und übergibt die dazugehörige ID und den Kartensicherheitscode an den Prozess zum Verknüpfen und Bereitstellen. Für im iTunes-Account vorhandene Karten werden möglicherweise zusätzliche Überprüfungen durchgeführt.

Kredit- oder Debitkarten aus der App des Kartenausstellers zu Apple Pay hinzufügen

Wenn eine App für die Verwendung mit Apple Pay registriert wird, werden Schlüssel für die App und den Server des Kartenausstellers erstellt. Diese Schlüssel werden verwendet, um die Karteninformationen zu verschlüsseln, die an den Kartenaussteller gesendet werden. Auf diese Weise wird verhindert, dass diese Informationen vom Apple-Gerät gelesen werden. Der Datenfluss für die Bereitstellung ähnelt dem Datenfluss beim manuellen Hinzufügen von Karten (siehe oben). Die einzige Abweichung besteht darin, dass anstelle der CVV-Nummer ein nur einmal nutzbares Passwort verwendet wird.

Hinzufügen von Kredit- oder Debitkarten von der Website des Kartenausstellers zu Apple Pay

Einige Kartenaussteller bieten die Möglichkeit, den Bereitstellungsprozess der Karte für Apple Wallet direkt auf ihren Websites zu starten. In diesem Fall startet der Benutzer den Vorgang, indem er auf der Website des Kartenausstellers eine Karte zum Bereitstellen auswählt. Der Benutzer wird anschließend zur eigenständigen Anmeldeplattform von Apple (auf der Domain von Apple) weitergeleitet und dort aufgefordert, sich mit seiner Apple-ID anzumelden. Nach der erfolgreichen Anmeldung wählt der Benutzer ein oder mehrere Geräte aus, auf dem bzw. denen die Karte bereitgestellt werden soll. Anschließend muss die erfolgte Bereitstellung auf jedem Zielgerät bestätigt werden.

Zusätzliche Prüfung hinzufügen

Der Kartenaussteller kann festlegen, ob für eine Kredit- oder Debitkarte eine zusätzliche Überprüfung notwendig ist. Der Benutzer kann für die Überprüfung aus verschiedenen Methoden wählen, abhängig davon, welche vom Kartenaussteller angeboten werden, z. B. SMS, E-Mail, Anruf beim Kundendienst oder über eine App eines anderen Anbieters. Bei SMS- oder E-Mail-Nachrichten kann der Benutzer aus den beim Kartenaussteller hinterlegten Kontaktdaten auswählen. Es wird ein Code gesendet, der in Apple Wallet, in den Einstellungen oder in der Apple Watch-App eingegeben werden muss. Beim Kundendienst oder bei der Überprüfung über eine App führt der Aussteller einen eigenen Kommunikationsprozess durch.

Autorisierung von Zahlungen mit Apple Pay

Auf Geräten mit einer Secure Enclave wird eine Zahlung erst dann zugelassen, wenn es die Autorisierung von der Secure Enclave erhalten hat. Auf dem iPhone oder iPad erfordert dies die Bestätigung, dass sich der Benutzer per Face ID, Touch ID oder Gerätecode authentifiziert hat. Die Standardmethode ist Face ID oder Touch ID, es kann aber auch jederzeit der Code verwendet werden. Der Code wird automatisch angeboten, wenn dreimal in Folge kein registrierter Fingerabdruck erkannt wurde oder zwei Versuche zum Gesichtsabgleich nicht erfolgreich waren. Nach fünf Fehlversuchen muss der Code eingegeben werden. Der Code wird auch benötigt, wenn Face ID oder Touch ID nicht konfiguriert oder nicht für Apple Pay aktiviert wurde. Damit auf der Apple Watch eine Zahlung erfolgen kann, muss der Benutzer das Gerät mittels Code entsperren und einen Doppelklick auf die Seitentaste ausführen.

Verwenden eines gemeinsamen Kopplungsschlüssels

Die Kommunikation zwischen der Secure Enclave und dem Secure Element findet über eine serielle Schnittstelle statt, wobei das Secure Element mit dem NFC-Controller verbunden ist, der wiederum mit dem Anwendungsprozessor verbunden ist. Obwohl sie nicht direkt verbunden sind, können Secure Enclave und Secure Element über ein gemeinsames Schlüsselpaar, das bei der Fertigung festgelegt wird, sicher kommunizieren. Die Verschlüsselung und Authentifizierung der Kommunikation basiert auf AES, wobei beide Seiten zum Schutz vor Replay-Angriffen kryptografische **Nonces** verwenden. Der Kopplungsschlüssel wird in der Secure Enclave aus dem UID-Schlüssel und der eindeutigen Kennung des Secure Element erzeugt. Der Kopplungsschlüssel wird dann bei der Fertigung von der Secure Enclave sicher an ein **Hardware-Sicherheitsmodul (HSM)** übertragen, das die benötigten Schlüsselmaterialien besitzt, um den Kopplungsschlüssel mit dem Secure Element zu verknüpfen.

Autorisieren einer sicheren Transaktion

Wenn der Benutzer eine Transaktion autorisiert, die eine physische Geste enthält, die direkt an die Secure Enclave übertragen wird, sendet die Secure Enclave signierte Daten über die Art der Authentifizierung und Details zur Transaktion (kontaktlos oder in Apps) zusammen mit einem Zufallswert für die Authentifizierung (AR) an das Secure Element. Der AR-Wert wird in der Secure Enclave erzeugt, wenn der Benutzer zum ersten Mal eine Kreditkarte hinzufügt, und bleibt erhalten, solange Apple Pay aktiviert ist. Er wird von den Mechanismen der Secure Enclave für Verschlüsselung und Anti-Rollback geschützt. Er wird mithilfe des Kopplungsschlüssels sicher an das Secure Element übertragen. Bei Erhalt eines neuen AR-Werts löscht das Secure Element alle zuvor hinzugefügten Karten.

Verwenden eines Zahlungskryptogramms für dynamische Sicherheit

Zahlungsvorgänge aus Zahlungs-Applets umfassen ein Zahlungskryptogramm zusammen mit einer Device Account Number. Dieses Kryptogramm, ein einmal nutzbarer Code, wird mit einem Transaktionszähler und einem Schlüssel errechnet. Der Transaktionszähler wird bei jeder neuen Transaktion um einen Schritt erhöht. Der Schlüssel wird während der Personalisierung im Zahlungs-Applet bereitgestellt und ist dem Zahlungsnetzwerk, dem Kartenaussteller oder beiden bekannt. Abhängig vom Zahlungssystem werden auch andere Daten für die Berechnung verwendet, darunter:

- Eine TUN (Terminal Unpredictable Number) für NFC-Transaktionen
- Eine Nonce des Apple Pay-Servers für Transaktionen innerhalb von Apps

Diese Sicherheitscodes werden an das Zahlungsnetzwerk und den Kartenaussteller weitergegeben, sodass der Aussteller die Transaktion überprüfen kann. Die Länge dieser Sicherheitscodes hängt von der jeweiligen Transaktion ab.

Kartenzahlung mit Apple Pay

Apple Pay kann zum Bezahlen von Käufen verwendet werden, die in Ladengeschäften, in Apps und auf Websites getätigt werden.

Kartenzahlung in Geschäften

Wenn das iPhone oder die Apple Watch eingeschaltet ist und ein NFC-Feld erkennt, bietet es dem Benutzer entweder die angeforderte Karte (sofern für diese Karte die automatische Auswahl aktiviert ist) oder alternativ die Standardkarte an, die in den Einstellungen festgelegt wurde. Der Benutzer kann auch Apple Wallet öffnen und eine Karte auswählen oder, falls das Gerät gesperrt ist:

- Auf Geräten mit Face ID zweimal auf die Seitentaste drücken
- Auf Geräten mit Touch ID zweimal auf die Home-Taste drücken
- Die Funktionen der Bedienungshilfen verwenden, die Apple Pay auf dem Sperrbildschirm erlauben

Anschließend muss sich der Benutzer mit Face ID, Touch ID oder dem Code authentifizieren, bevor die Zahlungsinformationen übertragen werden. Wenn die Apple Watch nicht gesperrt ist, kann durch Doppelklicken auf die Seitentaste die Standardkarte zur Bezahlung aktiviert werden. Ohne Benutzerauthentifizierung werden keine Zahlungsinformationen gesendet.

Nach der Authentifizierung werden die Device Account Number und der transaktionsspezifische dynamische Sicherheitscode für die Zahlungsabwicklung verwendet. Weder Apple noch das Gerät des Benutzers sendet die vollständige Kartenummer an den Händler. Apple kann anonymisierte Informationen zur Zahlung erhalten, z. B. ungefähre Zeit und Ort der Transaktion, mit denen Apple Pay und andere Produkte und Dienste von Apple verbessert werden können.

Kartenzahlung in Apps

Apple Pay kann auch für Zahlungen in Apps auf iPhone, iPad-, Mac- und Apple Watch-Geräten verwendet werden. Wenn Benutzer innerhalb von Apps mit Apple Pay bezahlen, erhält Apple die verschlüsselten Transaktionsdaten. Bevor diese Daten an den Entwickler oder Händler gesendet werden, entschlüsselt Apple die Transaktion mit einem entwicklerspezifischen Schlüssel. Apple Pay speichert anonymisierte Informationen zu Transaktionen, wie den ungefähren Betrag. Diese Informationen können nicht mit dem Benutzer verknüpft werden und enthalten keine Angaben dazu, wofür der Benutzer gezahlt hat.

Wenn in einer App eine Apple Pay-Transaktion gestartet wird, erhalten die Apple Pay-Server die verschlüsselte Transaktion vom Gerät, bevor sie an den Händler gesendet wird. Die Apple Pay-Server verschlüsseln sie dann erneut mit einem händlerspezifischen Schlüssel, bevor sie die Transaktion an den Händler weiterleiten.

Wenn eine App eine Zahlung anfordert, ruft sie eine API auf, um festzustellen, ob das Gerät Apple Pay unterstützt und der Benutzer Kredit- oder Debitkarten besitzt, mit denen auf einem vom Händler akzeptierten Zahlungsnetzwerk Zahlungen abgewickelt werden können. Die App fragt an, welche Informationen für die Transaktion benötigt werden, z. B. Rechnungs- und Lieferadresse oder Kontaktdaten. Danach ergeht an iOS, iPadOS oder watchOS die Aufforderung, die Apple Pay-Seite anzuzeigen, um Informationen für die App und andere benötigte Informationen abzufragen, z. B. welche Karte verwendet werden soll.

Zu diesem Zeitpunkt erhält die App die Adressdaten (Ort, Staat/Land und PLZ), um die Versandkosten berechnen zu können. Die App erhält erst dann alle Informationen, wenn der Benutzer die Zahlung mit Face ID, Touch ID oder dem Code bestätigt. Ist die Zahlung autorisiert, werden die Informationen der Apple Pay-Seite an den Händler übertragen.

Autorisierung von Zahlungen in Apps

Wenn der Benutzer eine Zahlung bestätigt, erfolgt ein Aufruf an die Apple Pay-Server, um eine kryptografische Nonce anzufordern, die eine ähnliche Funktion erfüllt wie der vom NFC-Terminal für In-Store-Transaktionen gesendete Wert. Die Nonce wird zusammen mit anderen Daten zur Transaktion an das Secure Element weitergegeben, um die Zahlungsdaten zu erzeugen, die mit einem Apple-Schlüssel verschlüsselt werden. Die verschlüsselten Zahlungsdaten werden an die Apple Pay-Server zurückgegeben, die die Daten entschlüsseln, die enthaltene Nonce mit der ursprünglich von den Apple Pay-Servern gesendeten Nonce abgleichen und die Zahlungsdaten mit dem zur Händler-ID gehörenden Händlerschlüssel erneut verschlüsseln. Die Zahlungsdaten werden anschließend an das Gerät zurückgegeben, das sie über die API an die App zurückgibt. Die App überträgt sie dann an das Händlersystem zur Bearbeitung. Der Händler kann nun die Zahlungsdaten mit seinem privaten Schlüssel zur Bearbeitung entschlüsseln. Dadurch und mit der Signatur der Server von Apple kann der Händler überprüfen, ob diese Transaktion für ihn bestimmt ist.

Die APIs benötigen eine Berechtigung, die die unterstützten Händler-IDs angibt. Eine App kann auch weitere Daten bereitstellen, etwa eine Bestellnummer oder Kunden-ID, die zum Signieren an das Secure Element gesendet werden, sodass sichergestellt ist, dass die Transaktion nicht mit einem anderen Kunden vertauscht werden kann. Der App-Entwickler kann „applicationData“ bei „PKPaymentRequest“ angeben. Ein Hash-Wert dieser Daten wird dann mit den anderen Zahlungsdaten verschlüsselt. Der Händler muss nun überprüfen, ob sein „applicationData“ Hash-Wert mit dem in den Zahlungsdaten übereinstimmt.

Kartenzahlung auf Websites

Apple Pay kann verwendet werden, um auf iPhone-, iPad-, Apple Watch- und Mac-Geräten mit Touch ID Käufe zu bezahlen, die auf Websites getätigt werden. Apple Pay-Transaktionen können auch auf einem Mac begonnen und dann unter Verwendung desselben iCloud-Accounts auf einem iPhone oder einer Apple Watch, auf dem bzw. der Apple Pay aktiviert ist, abgeschlossen werden.

Apple Pay im Internet setzt voraus, dass alle teilnehmenden Websites bei Apple registriert sind. Nachdem die Domain registriert wurde, wird ihr Name verifiziert; dies geschieht allerdings erst, nachdem Apple ein TLS-Client-Zertifikat ausgegeben hat. Websites mit Apple Pay-Unterstützung müssen ihre Inhalte über HTTPS bereitstellen. Für jede Zahlungstransaktion müssen Websites mit dem von Apple ausgestellten TLS-Client-Zertifikat eine sichere und eindeutige Händlersitzung mit einem Apple-Server starten. Die Daten der Händlersitzung werden von Apple signiert. Nachdem die Signatur der Händlersitzung verifiziert wurde, kann eine Website abfragen, ob der Benutzer über ein Apple Pay-fähiges Gerät verfügt und ob eine Kredit-, Debit- oder Prepaid-Karte auf dem Gerät aktiviert ist. Eine Freigabe weiterer Details erfolgt nicht. Wenn der Benutzer seine Informationen nicht teilen möchte, kann er Apple Pay-Anfragen auf seinem iPhone-, iPad- oder Mac-Gerät in den Datenschutzeinstellungen von Safari deaktivieren.

Nach der Validierung einer Händlersitzung sind alle Datenschutz- und Sicherheitsmaßnahmen identisch mit denen, die bei Zahlungsvorgängen des Benutzers innerhalb einer App gelten.

Beim Handoff vom Mac zum iPhone oder zur Apple Watch nutzt Apple Pay das Ende-zu-Ende-verschlüsselte Apple IDS-Protokoll (Identity Service), um zahlungsrelevante Informationen zwischen dem Mac des Benutzers und dem autorisierenden Gerät zu übertragen. Der IDS-Client auf dem Mac nutzt die Benutzergeräteschlüssel für die Verschlüsselung, sodass kein anderes Gerät diese Informationen entschlüsseln kann. Diese Schlüssel stehen Apple nicht zur Verfügung. Die Geräteerkennung für Apple Pay-Handoff umfasst den Typ und die eindeutige Kennung der Kreditkarten des Benutzers sowie einige Metadaten. Die gerätespezifische Accountnummer der Karte des Benutzers wird nicht preisgegeben und verbleibt sicher auf dem iPhone oder der Apple Watch des Benutzers. Apple sorgt mit dem iCloud-Schlüsselbund für die sichere Übertragung der kürzlich verwendeten Kontakt-, Liefer- und Rechnungsadressen des Benutzers.

Nachdem der Benutzer die Zahlung per Face ID, Touch ID, Code oder durch Doppelklicken auf die Seitentaste der Apple Watch autorisiert hat, wird ein für das Händlerzertifikat jeder Website verschlüsselter Zahlungs-Token auf sichere Weise vom iPhone bzw. von der Apple Watch des Benutzers an den Mac übertragen und dann der Website des Händlers bereitgestellt.

Nur Geräte, die sich in unmittelbarer Nähe zueinander befinden, können eine Zahlung anfordern und abschließen. Die Nähe wird durch Bluetooth Low Energy-Ankündigungen (BLE) bestimmt.

Kontaktlose Ausweise in Apple Pay

Um Daten von unterstützten Ausweisen an kompatible NFC-Terminals zu übertragen, verwendet Apple das Apple-Protokoll VAS (Value Added Services). Das VAS-Protokoll kann auf den kontaktlosen Terminals oder in iPhone-Apps implementiert werden. Für die Kommunikation mit den unterstützten Apple-Geräten wird NFC verwendet. Das VAS-Protokoll funktioniert nur über kurze Distanzen und kann verwendet werden, um kontaktlose Ausweise entweder unabhängig oder im Rahmen einer Apple Pay-Zahlung vorzulegen.

Wenn das Gerät in die Nähe des NFC-Terminals gehalten wird, beginnt das Terminal mit dem Empfang der Ausweisinformationen, indem es eine Anfrage für den Ausweis sendet. Ist der Benutzer im Besitz eines Ausweises mit der Kennung des Ausweisausstellers, wird er aufgefordert, die Verwendung dieses Ausweises mittels Face ID, Touch ID oder Code zu autorisieren. Es wird ein Verschlüsselungscode für die Kartendaten generiert, mit dem dann die an das Terminal gesendeten Daten verschlüsselt werden. Basis für diesen Verschlüsselungscode sind die Ausweisinformationen, ein Zeitstempel und ein nur einmal nutzbarer ECDH P-256-Zufallscode.

In iOS 12.0.1 bis einschließlich iOS 13 können Benutzer manuell einen Ausweis auswählen, bevor sie ihn dem NFC-Terminal des Händlers vorlegen. In iOS 13.1 (oder neuer) können Ausweisaussteller ausgewählte Ausweise manuell so konfigurieren, dass entweder eine Benutzerauthentifizierung erforderlich ist oder dass die Ausweise ohne Authentifizierung verwendet werden können.

Unbrauchbarmachung von Karten mit Apple Pay

Dem Secure Element hinzugefügte Kredit-, Debit- und Prepaid-Karten können nur verwendet werden, wenn das Secure Element eine Autorisierung mit dem Kopplungsschlüssel und dem AR-Wert (Authorization Random) erhält, die beim Hinzufügen der Karte verwendet wurden. Bei Erhalt eines neuen AR-Werts löscht das Secure Element alle zuvor hinzugefügten Karten. Dadurch kann das Betriebssystem die Secure Enclave anweisen, eine Karte unbrauchbar zu machen, indem sie ihre Kopie des AR unter den folgenden Umständen als ungültig kennzeichnet:

Methode	Gerät
Der Code ist deaktiviert.	iPhone, iPad, Apple Watch
Das Passwort ist deaktiviert.	Mac
Der Benutzer meldet sich bei iCloud ab.	iPhone, iPad, Mac, Apple Watch
Der Benutzer löscht alle Inhalte und Einstellungen.	iPhone, iPad, Mac, Apple Watch
Das Gerät wird aus dem Wiederherstellungsmodus wiederhergestellt.	iPhone, iPad, Mac, Apple Watch
Das Gerät wird entkoppelt.	Apple Watch

Karten sperren, entfernen und löschen

Benutzer können Apple Pay auf iPhone, iPad und Apple Watch sperren, indem sie ihr Gerät mit der App „Wo ist?“ in den Modus „Verloren“ versetzen. Benutzer haben auch die Möglichkeit, ihre Karten mit der App „Wo ist?“, über iCloud.com oder direkt auf den Geräten in Apple Wallet aus Apple Pay zu entfernen und zu löschen. Bei einer Apple Watch lassen sich Karten mit den iCloud-Einstellungen, der App „Apple Watch“ auf dem iPhone oder direkt auf der Watch entfernen. Die Möglichkeit, auf dem Gerät per Karte zu bezahlen, kann für Apple Pay vom Kartenaussteller oder dem Zahlungsnetzwerk gesperrt oder entfernt werden, auch wenn das Gerät offline und nicht mit einem Mobilfunknetz oder WLAN verbunden ist. Benutzer können sich auch an ihren Kartenaussteller wenden, um Karten für Apple Pay sperren oder davon entfernen zu lassen.

Wenn ein Benutzer das Gerät im Modus „Wo ist?“ mit der Option „Inhalte & Einstellungen löschen“ komplett löscht oder das Gerät im Wiederherstellungsmodus wiederherstellt, weisen iPhone-, iPad-, iPod touch-, Mac- und Apple Watch-Geräte das Secure Element an, alle Karten als gelöscht zu markieren. Dadurch werden die Karten sofort in Apple Pay unbrauchbar, bis die Apple Pay-Server angewiesen werden können, die Karten völlig aus dem Secure Element zu löschen. Unabhängig davon markiert die Secure Enclave den AR als ungültig, sodass keine weiteren Zahlungen für die registrierten Karten mehr autorisiert werden können. Wenn das Gerät online ist, kontaktiert es die Apple Pay-Server, um sicherzustellen, dass alle Karten im Secure Element gelöscht werden.

Sicherheit bei Apple Card

Auf unterstützten iPhone und Mac-Modellen können Benutzer eine Apple Card auf sichere Weise beantragen.

Sicherheit bei der Beantragung einer Apple Card

In iOS 12.4 (oder neuer), macOS 10.14.6 (oder neuer) und watchOS 5.3 (oder neuer) kann Apple Card mit Apple Pay verwendet werden, um in Geschäften, in Apps und im Web Zahlungen zu leisten.

Zum Beantragen von Apple Card muss ein Benutzer mit einem iCloud-Account auf einem Apple Pay-fähigen iOS- oder iPadOS-Gerät angemeldet sein und die Zwei-Faktor-Authentifizierung im iCloud-Account eingerichtet haben. Wird der Antrag bewilligt, wird Apple Card in Apple Wallet oder unter „Einstellungen“ > „Wallet & Apple Pay“ auf allen berechtigten Geräten bereitgestellt, auf denen sich der Benutzer mit seiner Apple-ID angemeldet hat.

Wenn der Benutzer Apple Card beantragt, werden seine Identitätsinformationen auf sichere Weise von den als Identitätsanbieter tätigen Apple-Partnern verifiziert und danach zur Identitäts- und Kreditprüfung an die Goldman Sachs Bank USA weitergegeben (gemäß der Apple-Datenschutzrichtlinie, die unter www.apple.com/de/legal/privacy verfügbar ist).

Im Rahmen des Antrags bereitgestellte Informationen (etwa die Sozialversicherungsnummer oder eine Kopie eines amtlichen Lichtbildausweises) werden sicher an die Identitätsanbieter von Apple und/oder an die Goldman Sachs Bank USA übertragen und mit deren jeweiligen Schlüsseln verschlüsselt. Apple kann diese Daten nicht entschlüsseln.

Die beim Antrag bereitgestellten Gehaltsdaten und die für die Zahlung von Rechnungen verwendeten Bankverbindungsdaten werden sicher an die Goldman Sachs Bank USA übertragen und mit ihrem Schlüssel verschlüsselt. Die Bankverbindungsdaten werden im Schlüsselbund gesichert. Apple kann diese Daten nicht entschlüsseln.

Beim Hinzufügen von Apple Card zu Apple Wallet können dieselben Informationen, die beim Hinzufügen einer Kredit- oder Debitkarte eingegeben wurden, mit der Apple-Partnerbank Goldman Sachs Bank USA und mit Apple Payments Inc. geteilt werden (gemäß der Apple-Datenschutzrichtlinie, die unter www.apple.com/de/legal/privacy verfügbar ist). Die Informationen werden nur bei der Fehlerbehebung, zur Verhinderung von Betrug und zu Regulierungszwecken verwendet.

In iOS 14.6 (oder neuer), iPadOS 14.6 (oder neuer) und watchOS 7.5 (oder neuer) kann der Organisator einer iCloud-Familie mit einer Apple Card seine Karte mit allen Mitgliedern einer iCloud-Familie teilen, die mindestens 13 Jahre alt sind. Eine Benutzerauthentifizierung ist für die Bestätigung der Einladung erforderlich. Apple Wallet verwendet einen Schlüssel in der Secure Enclave, um eine Signatur zu erzeugen, die den Besitzer und die eingeladene Person verknüpft. Die Signatur wird auf Apple-Servern überprüft und bestätigt.

Der Organisator kann ein Transaktionslimit für die Teilnehmenden festlegen. Die Karten von Teilnehmenden können in Apple Wallet ebenfalls gesperrt werden, um ihre Möglichkeit für Ausgaben jederzeit auszusetzen. Wenn ein Miteigentümer oder eine teilnehmende Person über 18 Jahren die Einladung akzeptiert und anwendet, durchlaufen sie denselben Antragsprozess, der im Abschnitt für den Apple Card-Antrag in Apple Wallet definiert ist.

Verwendung der Apple Card

Es ist möglich, eine physische Karte über Apple Card in Apple Wallet zu bestellen. Nachdem der Benutzer die physische Karte in Empfang genommen hat, wird die Karte mit dem NFC-Tag aktiviert, das sich im Begleitumschlag der physischen Karte befindet. Das Tag ist für jede Karte einzigartig und kann nicht verwendet werden, um die Karte eines anderen Benutzers zu aktivieren. Alternativ lässt sich die Karte manuell in den Apple Wallet-Einstellungen aktivieren. Darüber hinaus hat der Benutzer die Möglichkeit, die physische Karte jederzeit über Apple Wallet zu sperren oder zu entsperren.

Apple Card-Zahlungen und Details zu Apple Wallet-Karten

Beim Apple Card-Account fällige Zahlungen können über Apple Wallet in iOS mit Apple Cash und einem Bankkonto veranlasst werden. Die Begleichung von Rechnungen kann als wiederkehrende Zahlung oder als Einmalzahlung zu einem bestimmten Datum mit Apple Cash und einem Bankkonto eingerichtet werden. Beim Tätigen einer Zahlung erfolgt ein Aufruf an die Apple Pay-Server, um eine kryptografische **Nonce** ähnlich wie bei Apple Cash anzufordern. Die Nonce wird zusammen mit den Details zur Zahlung an das Secure Element weitergegeben, um eine Signatur zu erzeugen. Die Signatur wird dann an die Apple Pay-Server zurückgegeben. Die Apple Pay-Server überprüfen die Authentifizierung, Integrität und Korrektheit der Zahlung über die Signatur und die Nonce und die Zahlungsanweisung wird an die Goldman Sachs Bank USA zur Bearbeitung übergeben.

Die Nummer der Apple Card wird dann durch Bereitstellen eines Zertifikats von Apple Wallet abgerufen. Der Apple Pay-Server überprüft das Zertifikat, um zu bestätigen, dass der Schlüssel in der Secure Enclave erzeugt wurde. Er verwendet dann diesen Schlüssel zum Entschlüsseln der Apple Card-Nummer, bevor er diese an Apple Wallet zurückgibt. So kann nur das iPhone, das die Apple Card-Nummer angefordert hat, sie entschlüsseln. Nach der Entschlüsselung wird die Apple Card-Nummer im iCloud-Schlüsselbund gesichert.

Das Anzeigen der Apple Card-Nummer mit Apple Wallet erfordert die Benutzerauthentifizierung mit Face ID, Touch ID oder einem Code. Der Benutzer kann diesen Code im Abschnitt mit den Karteninformationen ändern. Der vorherige Code wird ersetzt.

Erweiterter Betrugsschutz

In iOS 15 (oder neuer) und iPadOS 15 (oder neuer) kann der Apple Card-Benutzer den erweiterten Betrugsschutz in Apple Wallet aktivieren. Ist diese Funktion aktiviert, wird der Sicherheitscode für die Karte alle paar Tage aktualisiert.

Sicherheit bei Apple Cash

Bei iOS 11.2 (oder neuer), iPadOS 13.1 (oder neuer) und watchOS 4.2 (oder neuer) kann Apple Pay auf einem iPhone-, iPad- oder Apple Watch-Gerät verwendet werden, um Geld an andere Benutzer zu senden bzw. von diesen zu empfangen oder anzufordern. Wenn ein Benutzer Geld empfängt, wird es zu einem Apple Cash-Account hinzugefügt, auf den in Apple Wallet oder unter „Einstellungen“ > „Wallet & Apple Pay“ auf allen berechtigten Geräten zugegriffen werden kann, auf denen sich der Benutzer mit seiner Apple-ID angemeldet hat.

In iOS 14, iPadOS 14 und watchOS 7 kann der Organisator einer iCloud-Familie, der die Identität der Familienmitglieder bei Apple Cash verifiziert hat, Apple Cash für minderjährige Familienmitglieder (Mitglieder unter 18 Jahre) aktivieren. Der Organisator kann wahlweise die Möglichkeit, Geld zu versenden, für diese Benutzer so einschränken, dass sie nur Familienmitgliedern oder nur Kontakten Geld zukommen lassen können. Wenn ein minderjähriges Familienmitglied eine Wiederherstellung eines Apple-ID-Accounts durchläuft, muss der Organisator der Familie die Apple Cash-Karte für den betreffenden Benutzer manuell reaktivieren. Das minderjährige Familienmitglied ist nicht mehr Teil der iCloud-Familie; sein Apple Cash-Guthaben wird automatisch auf den Account des Organisators transferiert.

Wird Apple Cash eingerichtet, können dieselben Informationen, die beim Hinzufügen einer Kredit- oder Debitkarte eingegeben wurden, mit unserer Partnerbank Green Dot Bank und mit Apple Payments Inc. geteilt werden. Dies ist ein eigenes Tochterunternehmen, das zum Schutz der Privatsphäre des Benutzers gegründet wurde, in dem Informationen getrennt von den restlichen Apple-Funktionen gespeichert und verarbeitet werden und zwar so, dass sie in den restlichen Apple-Funktionen nicht bekannt sind. Die Informationen werden nur bei der Fehlerbehebung, zur Verhinderung von Betrug und zu Regulierungszwecken verwendet.

Verwendung von Apple Cash in iMessage

Um Apple Cash und Zahlungen zwischen Personen zu verwenden, muss ein Benutzer mit einem iCloud-Account auf einem Apple Cash-fähigen Gerät angemeldet sein und die Zwei-Faktor-Authentifizierung im iCloud-Account eingerichtet haben. Geldanforderungen und -überweisungen zwischen Benutzern werden in der App „Nachrichten“ oder über Siri gestartet. Wenn ein Benutzer versucht, Geld zu senden, wird in iMessage die Apple Pay-Seite angezeigt. Das Apple Cash-Guthaben wird immer zuerst verwendet. Falls erforderlich, wird ein zusätzlicher Betrag von einer zweiten Kredit- oder Debitkarte abgebucht, die der Benutzer in Apple Wallet hinzugefügt hat.

Verwenden von Apple Cash in Ladengeschäften, in Apps und im Internet

Die Apple Cash-Karte in Apple Wallet kann mit Apple Pay verwendet werden, um in Geschäften, in Apps und im Web zu bezahlen. Das Geld auf dem Apple Cash-Konto kann auch auf ein Bankkonto überwiesen werden. Zusätzlich zum Geld, das von anderen Benutzern überwiesen wird, kann dem Apple Cash-Konto auch Geld über eine Debit- oder Prepaid-Karte in Apple Wallet hinzugefügt werden.

Nach dem Abschluss einer Transaktion speichert Apple Payments Inc. die Transaktionsdaten des Benutzers möglicherweise für die Fehlerbehebung, zur Verhinderung von Betrug und zu Regulierungszwecken. Die restlichen Apple-Funktionen wissen nicht, an wen der Benutzer Geld gesendet hat, von wem Geld gesendet wurde oder wo der Benutzer mit seiner Apple Cash-Karte eingekauft hat.

Wird Geld mit Apple Pay gesendet, einem Apple Cash-Konto Geld hinzugefügt oder Geld auf ein Bankkonto überwiesen, wird ein Aufruf an die Apple Pay-Server gesendet, um eine kryptografische Nonce anzufordern, die dem Wert ähnelt, der in Apps für Apple Pay zurückgegeben wird. Die Nonce wird zusammen mit anderen Daten zur Transaktion an das Secure Element weitergegeben, um eine Zahlungssignatur zu erzeugen. Die Signatur wird an die Apple Pay-Server zurückgegeben. Die Apple Pay-Server überprüfen die Authentifizierung, Integrität und Korrektheit der Transaktion anhand der Zahlungssignatur und der Nonce. Anschließend wird die Geldüberweisung gestartet und der Benutzer wird über den Abschluss der Transaktion informiert.

Wenn bei einer Transaktion:

- eine Debitkarte verwendet wird, um Geld zu Apple Cash hinzuzufügen,
- oder eine ergänzende Zahlung getätigt wird, wenn das Apple Cash-Guthaben nicht ausreicht,

dann werden auch verschlüsselte Zahlungsdaten generiert und an die Apple Pay-Server gesendet – ähnlich wie bei Daten, die innerhalb von Apps und auf Websites für Apple Pay verwendet werden.

Nachdem das Guthaben auf dem Apple Cash-Konto einen bestimmten Betrag überschritten hat oder wenn ungewöhnliche Aktivitäten festgestellt werden, wird der Benutzer aufgefordert, seine Identität zu verifizieren. Bereitgestellte Informationen zum Verifizieren der Benutzeridentität wie die Sozialversicherungsnummer oder Antworten auf Fragen (z. B. die Bestätigung des Namens einer Straße) werden sicher an den Apple-Partner übertragen und mit seinem Schlüssel verschlüsselt. Apple kann diese Daten nicht entschlüsseln. Der Benutzer wird aufgefordert, seine Identität erneut zu verifizieren, wenn er die Wiederherstellung eines zu einer Apple-ID gehörenden Accounts durchführen möchte, bevor er wieder Zugang zu seinem Apple Cash-Guthaben erhält.

Sicherheit der Funktion „Tap to Pay“ auf dem iPhone

Die in iOS 15.4 verfügbare Funktion „Tap to Pay“ auf dem iPhone ermöglicht es Händlern in den USA, Apple Pay und andere kontaktlose Zahlungsmethoden mit dem iPhone und einer iOS-App von Partnern zu akzeptieren. Mit diesem Dienst können Benutzer mit unterstützten iPhone-Geräten kontaktlose Zahlungen und durch *Apple Pay* bereitgestellte NFC-fähige Karten auf sichere Weise akzeptieren. Dank der Funktion „Tap to Pay“ auf dem iPhone benötigen Händler keine zusätzliche Hardware für kontaktlose Zahlungen.

Die Funktion „Tap to Pay“ auf dem iPhone ist dafür ausgelegt, die persönlichen Daten der zahlenden Person zu schützen. Der Dienst sammelt keine Informationen über Transaktionen, die Rückschlüsse auf die zahlende Person zulassen. Die Informationen der Zahlungskarte wie die Kredit-/Debitkartennummer (PAN) werden von dem Secure Element geschützt und sind für den Händler nicht verfügbar. Nur der Zahlungsdienstleister des Händlers, die zahlende Person und der Kartenaussteller haben Zugriff auf die Kartendaten. Darüber hinaus sammelt der „Tap to Pay“-Dienst keine Namen, Adressen oder Telefonnummern der zahlenden Personen.

Die Funktion „Tap to Pay“ auf dem iPhone wurde extern von einem anerkannten Sicherheitslabor bewertet und von American Express, Discover, Mastercard und Visa zugelassen.

Sicherheit der Komponenten bei kontaktlosen Zahlungen

- *Secure Element*: Das Secure Element [Link zum Abschnitt „Apple Pay Secure Element“] hostet die Zahlungskernel, mit denen die Kartendaten für die kontaktlose Zahlung ausgelesen und gesichert werden.
- *NFC-Controller*: Der NFC-Controller verarbeitet die NFC-Protokolle (Near Field Communication) und steuert die Kommunikation zwischen Anwendungsprozessor und Secure Element bzw. zwischen Secure Element und der kontaktlosen Zahlungskarte.
- *„Tap to Pay“-Server*: Die „Tap to Pay“-Server verwalten die Konfiguration und Bereitstellung der Zahlungskernel im Gerät. Außerdem überwachen die Server die Sicherheit der „Tap to Pay“-Geräte so, dass sie mit dem CPoC-Standard (Contactless Payments on COTS) des Payment Card Industry Security Standards Council (PCI SSC) kompatibel sind und PCI DSS einhalten.

Auslesen von Kredit-, Debit- und Prepaid-Karten durch die Funktion „Tap to Pay“

Sicherheit bei der Bereitstellung – Übersicht

Vor der ersten Verwendung der Funktion „Tap to Pay“ mit einer berechtigten App, überprüfen die „Tap to Pay“-Server, ob das Gerät die notwendigen Voraussetzungen erfüllt, z. B. das Gerätemodell, die iOS-Version und ob ein Code festgelegt wurde. Nachdem diese Überprüfung abgeschlossen ist, wird das Zahlungsbestätigungs-Applet von den „Tap to Pay“-Servern mit der zugehörigen Konfiguration für die Zahlungskernel geladen und im Secure Element installiert. Der Vorgang wird sicher zwischen den „Tap to Pay“-Servern und dem Secure Element durchgeführt. Vor der Installation überprüft das Secure Element die Integrität und Authentizität der Daten.

Sicherheit beim Auslesen der Karte – Übersicht

Wenn eine „Tap to Pay“-App das Auslesen einer Karte aus dem ProximityReader-Framework anfordert, wird ein von iOS gesteuertes Sheet angezeigt und der Benutzer darin aufgefordert, auf eine Zahlungskarte zu tippen. iOS initialisiert den Payment Card Reader und weist die Zahlungskernel im Secure Element an, das Auslesen der Karte zu starten.

Zu diesem Zeitpunkt übernimmt das Secure Element die Steuerung des NFC-Controllers im Lesemodus (Reader Mode). In diesem Modus ist es nur erlaubt, die Kartendaten zwischen der Zahlungskarte und dem Secure Element über den NFC-Controller auszutauschen. Zahlungskarten können nur in diesem Modus ausgelesen werden.

Nachdem das Zahlungsbestätigungs-Applet auf dem Secure Element das Auslesen der Karte abgeschlossen hat, verschlüsselt und signiert es die Kartendaten. Die Kartendaten bleiben verschlüsselt und authentifiziert, bis sie den Zahlungsdienstleister erreichen. Nur der Zahlungsdienstleister, der von der App verwendet wurde, um das Auslesen der Karte anzufordern, kann die Kartendaten entschlüsseln. Der Zahlungsdienstleister muss den Schlüssel für die Entschlüsselung der Kartendaten von den „Tap to Pay“-Servern anfordern. Die „Tap to Pay“-Server geben die Entschlüsselungsschlüssel an den Zahlungsdienstleister aus, wenn die Integrität und Authentizität der Daten überprüft und sichergestellt wurde, dass das Auslesen der Karte innerhalb von 60 Sekunden auf dem Gerät durchgeführt wurde.

Auf diese Weise ist sichergestellt, dass die Kartendaten nur vom Zahlungsdienstleister entschlüsselt werden können, der die Transaktion für den Händler verarbeitet.

Verwenden der Apple Wallet

Zugriff mit Apple Wallet

Auf unterstützten iPhone- und Apple Watch-Geräten können Benutzer in Apple Wallet Schlüssel für ihr Zuhause, ihre Autos und ihre Hotelzimmer speichern. Sie können sogar Unternehmens-Badges und Studierendenausweise sichern. An der Tür wird automatisch der richtige Schlüssel vorgelegt, damit Benutzer nur einmal Tippen müssen, um per NFC Zugang zu erhalten.

Benutzerkomfort

Wenn Schlüssel, Karten oder Studierendenausweise zu Apple Wallet hinzugefügt werden, wird standardmäßig der Expressmodus aktiviert. Karten im Expressmodus interagieren mit akzeptierenden Terminals, ohne dass eine Authentifizierung per Face ID, Touch ID oder Code oder aber ein Doppelklick auf die Seitentaste der Apple Watch erforderlich ist. Der Benutzer kann diese Funktion deaktivieren, indem er auf die Taste „Mehr“ vorne auf dem Ausweis in Apple Wallet tippt. Um den Expressmodus wieder zu aktivieren, muss der Benutzer Face ID, Touch ID oder einen Code verwenden.

Datenschutz und Sicherheit

Die Schlüssel in Apple Wallet nutzen die Datenschutz- und Sicherheitsfunktionen, die im iPhone und der Apple Watch integriert sind. Wann oder wo eine Person ihre Schlüssel in Apple Wallet verwendet, wird nicht mit Apple geteilt oder auf Apple-Servern gespeichert. Die Anmeldedaten werden sicher im Secure Element (SE) von unterstützten Geräten gespeichert. Das SE hostet speziell entwickelte Applets, um Zugangsschlüssel sicher zu verwalten und zu speichern. So ist sichergestellt, dass Schlüssel nicht extrahiert werden können.

Bevor Zugangsschlüssel bereitgestellt werden, muss ein Benutzer bei seinem iCloud-Account auf einem kompatiblen iPhone angemeldet sein und die Zwei-Faktor-Authentifizierung für den iCloud-Account aktiviert haben. Bei Studierendenausweisen muss die Zwei-Faktor-Authentifizierung nicht aktiviert sein.

Wenn ein Benutzer den Bereitstellungsprozess einleitet, werden ähnliche Schritte ausgeführt wie bei der Bereitstellung von Kredit- und Debitkarten, z. B wie unter [Verknüpfen und Bereitstellen](#) beschrieben. Bei einer Transaktion kommuniziert das Lesegerät über den NFC-Controller (Near Field Communication) mit dem Secure Element. Diese Kommunikation erfolgt über einen bestehenden sicheren Kanal.

Die Anzahl der Geräte, einschließlich iPhone und Apple Watch, denen ein Zugangsschlüssel bereitgestellt werden kann, wird vom jeweiligen Partner definiert und verwaltet und kann je nach Partner variieren. Dank dieses Ansatzes hat jeder Partner die Kontrolle über die maximale Anzahl der bereitgestellten Sicherheitsschlüssel pro Gerätetyp, um die eigenen Anforderungen erfüllen zu können. Zu diesem Zweck stellt Apple den Partnern die Gerätetypen und anonymisierten Geräte-IDs zur Verfügung. Aus Datenschutz- und Sicherheitsgründen variieren die IDs für jeden Partner.

Schlüssel können wie folgt deaktiviert oder entfernt werden:

- Das Gerät mit „Wo ist?“ fernlöschen
- Modus „Verloren“ mit „Wo ist?“ aktivieren
- MDM-Fernlöschbefehl (Mobile Device Management) empfangen
- Alle Karten auf der Accountseite der Apple-ID entfernen
- Alle Karten auf iCloud.com entfernen
- Alle Karten aus Apple Wallet entfernen
- Die Karte in der App des Ausstellers entfernen

Wenn Benutzer in iOS 15.4 (oder neuer) die Seitentaste eines iPhone mit Face ID oder die Home-Taste auf einem iPhone mit Touch ID zweimal drücken, werden die Karten- und Zugangsschlüsseldetails erst angezeigt, wenn sie sich auf dem Gerät authentifiziert haben. Eine Authentifizierung mit Face ID, Touch ID oder einem Code ist erforderlich, bevor kartenspezifische Informationen wie Hotelbuchungsdetails in Apple Wallet angezeigt werden.

Zugangsdatentypen

Es gibt verschiedene Zugangstypen von Apple Wallet, z. B. Hotelzimmerschlüssel, Unternehmens-Badges, Studierendenausweise, Hausschlüssel und Autoschlüssel.

Hotels

Hotelzimmerschlüssel in Apple Wallet ermöglichen einen komfortablen und kontaktlosen Aufenthalt – vom Check-in bis zum Check-out. Darüber hinaus bieten sie im Gegensatz zu normalen Hotelschlüsselkarten zusätzlichen Datenschutz und zusätzliche Sicherheit für die Gäste. Hotelgäste können an unterstützten Orten auf ihr kompatibles [iPhone](#) oder ihre kompatible Apple Watch Series 4 (oder neuer) tippen, um mit dem in Apple Wallet gesicherten Zimmerschlüssel die Tür aufzuschließen.

Die Funktionen in Apple Wallet sind speziell entwickelt worden, um einen reibungslosen Ablauf für die Kunden zu ermöglichen:

- Bereitstellung des Zimmerschlüssels durch die Hotel-App vor der Anreise, um eine Karte vor dem Aufenthalt zu Apple Wallet hinzuzufügen
- Kartenkacheln für den Check-in, um über Apple Wallet Check-ins und Zimmerzuweisungen durchzuführen
- Schlüsselaktualisierungen nach der Bereitstellung, um Aufenthaltsverlängerungen oder -änderungen zu ermöglichen
- Unterstützung von mehreren Zimmerschlüsseln für eine Karte in Apple Wallet
- Automatische Archivierung von abgelaufenen Schlüsseln in Apple Wallet

Unternehmens-Badges

Mitarbeiter-Badges von unterstützten Partnern können zu Apple Wallet auf dem iPhone und der Apple Watch hinzugefügt werden. Dies ermöglicht Mitarbeitern weltweit einen kontaktlosen Zugang zu ihren Arbeitsplätzen. Zum Hinzufügen eines Badges muss die betreffende Person die Multi-Faktor-Authentifizierung für ihren Account aktiviert haben, mit dem sie sich in der vom Arbeitgeber bereitgestellten App anmeldet.

Mitarbeiter-Badges nutzen die Zugangsfunktionen von Apple, mit denen Personen:

- Mitarbeiter-Badges zu einer gekoppelten Apple Watch über eine Push-Bereitstellung ohne Installation einer Partner-App hinzufügen können
- Reibungslosen Zugang zu Unternehmensgebäuden dank Expressmodus erhalten
- Zugang zu ihrem Arbeitsplatz erhalten, selbst wenn die Batterie des iPhone leer ist

Studierendenausweise

Bei iOS 12 (oder neuer) können Studierende, Lehrkräfte und Mitarbeiter teilnehmender Bildungseinrichtungen ihren Ausweis zu Apple Wallet auf unterstützten iPhone- und Apple Watch-Modellen hinzufügen, um Zutritt zu Orten zu erhalten und überall dort zu zahlen, wo ihr Ausweis akzeptiert wird.

Mithilfe einer App, die vom Ausweisaussteller oder von der teilnehmenden Bildungseinrichtung bereitgestellt wird, kann ein Benutzer seinen Ausweis zu Apple Wallet hinzufügen. Hierbei ist der technische Vorgang mit dem identisch, der im Abschnitt „[Kredit- oder Debitkarten aus der App des Kartenausstellers zu Apple Pay hinzufügen](#)“ beschrieben wird. Darüber hinaus müssen ausstellende Apps die Zwei-Faktor-Authentifizierung in den Accounts unterstützen, die den Zugriff auf ihre Studierendenausweise schützen. Ein Ausweis kann auf maximal zwei unterstützten Apple-Geräten, die mit derselben Apple-ID angemeldet sind, gleichzeitig eingerichtet werden.

Haushalte mit mehreren Personen

Mieter und Mitarbeiter von unterstützten Partnereinrichtungen können ihren Hausschlüssel in Apple Wallet verwenden, um Zugang zu Gebäuden, Abteilungen oder allgemeinen Bereichen zu erhalten. Der Hausschlüssel kann über die App des Partners bereitgestellt werden. Bei Partnern, die die reibungslose Bereitstellung unterstützen, können Gebäudeverwalter den Mietern einen Link senden, mit dem die Bereitstellung mithilfe ihres bevorzugten Kanals (z. B. E-Mail oder SMS) gestartet werden kann. So muss der Mieter nur auf den Link klicken, um den Schlüssel abzurufen. App Clips sorgen darüber hinaus für eine sichere und reibungslose Erfahrung und ermöglichen die Bereitstellung eines Schlüssels, ohne die App eines Partners installieren zu müssen. Weitere Informationen sind im Apple Support-Artikel [Verwenden von App Clips auf dem iPhone](#) zu finden.

Hausschlüssel

Ein Hausschlüssel in Apple Wallet kann mit NFC-fähigen Türschlössern verwendet werden. Dafür reicht das Tippen auf ein iPhone oder eine Apple Watch. Weitere Informationen zur Konfiguration und Verwendung eines Hausschlüssels sind im Apple Support-Artikel [Aufschließen deiner Tür mit einem digitalen Hausschlüssel auf dem iPhone](#) zu finden.

Wenn eine Person einen Hausschlüssel konfiguriert, erhalten alle Bewohner in ihrem Haushalt automatisch diesen Hausschlüssel. Der Eigentümer eines Zuhauses kann die Home-App verwenden, um einen Hausschlüssel mit weiteren Personen zu teilen, ein Mitglied in einem geteilten Zuhause zu entfernen oder Einladungen und Mitglieder zu verwalten. Wenn eine Person eine Einladung für ein Zuhause mit einem Hausschlüssel annimmt, wird der Hausschlüssel auf ihrem Gerät in Apple Wallet bereitgestellt. Wenn eine Person ein Zuhause verlässt oder der Eigentümer des Zuhauses den Zugang für diese Person widerruft, wird der Hausschlüssel auch aus Apple Wallet entfernt.

Autoschlüssel

Das digitale Speichern von Autoschlüsseln in Apple Wallet wird von unterstützten iPhone-Geräten und damit gekoppelten Apple Watch-Geräten in nativer Form unterstützt. Die Autoschlüssel werden in Apple Wallet durch Karten repräsentiert (die von Apple im Namen und im Auftrag des jeweiligen Autoherstellers erstellt werden) und unterstützen den kompletten Lebenszyklus von für Apple Pay genutzten Karten (iCloud-Modus „Verloren“, Fernlöschung, lokales Löschen einer Karte und Option „Alle Inhalte und Einstellungen löschen“). Ergänzend zur Standardverwaltung von für Apple Pay genutzten Karten können geteilte Autoschlüssel über das iPhone und die Apple Watch des Benutzers und über die HMI-Schnittstelle (Human Machine Interface) des Fahrzeugs gelöscht werden.

Mit der Funktion „Autoschlüssel“ kann das Fahrzeug ent- und verriegelt, der Motor gestartet und das Fahrzeug in den Modus „Drive“ versetzt werden. Die „Standardtransaktion“ unterstützt die wechselseitige Authentifizierung und ist obligatorisch zum Starten des Motors. Zum Ent- und Verriegeln des Fahrzeugs kann die „schnelle Transaktion“ verwendet werden, sofern dies aus Performance-Gründen notwendig ist.

Die Schlüssel werden erstellt, indem ein iPhone mit einem unterstützten Fahrzeug im Besitz des Benutzers gekoppelt wird. Alle Schlüssel werden auf dem eingebetteten Secure Element basierend auf der ECC-OBKG-Technologie (Elliptic Curve (NIST P-256) On-Board Key Generation) erstellt. Die privaten Schlüssel verlassen niemals das Secure Element. Die Kommunikation zwischen den Geräten und dem Fahrzeug erfolgt entweder über NFC oder einer Kombination aus Bluetooth LE und UWB. Für die Schlüsselverwaltung wird eine API für die Verbindung von Apple zum Server des Autoherstellers basierend auf dem TLS-Protokoll mit wechselseitiger Authentifizierung verwendet. Nachdem ein Schlüssel mit einem iPhone gekoppelt wurde, kann auch eine mit dem iPhone gekoppelte Apple Watch einen Schlüssel empfangen. Ein Schlüssel, der im Fahrzeug oder auf dem Gerät gelöscht wird, kann nicht wiederhergestellt werden. Schlüssel auf verloren gegangenen oder gestohlenen Geräten können vorübergehend deaktiviert und später ggf. wieder aktiviert werden. Ihre Bereitstellung auf einem neuen Gerät erfordert hingegen das erneute Koppeln und Teilen.

Sicherheit der Funktion „Autoschlüssel“ in iOS

Entwickler haben die Möglichkeit, eine Option für den sicheren und schlüsselfreien Zugang zu einem Fahrzeug auf einem unterstützten iPhone und einer damit gekoppelten Apple Watch zur Verfügung zu stellen.

Koppeln für Eigentümer

Der Eigentümer muss den Nachweis erbringen, dass sich das Fahrzeug in seinem Besitz befindet (wobei die Art, wie dieser Nachweis erbracht wird, abhängig vom Autohersteller ist). Er kann den Kopplungsvorgang in der App des Autoherstellers starten. Ihm steht dafür ein Link zur Verfügung, den er direkt vom Autohersteller oder über das Menü im Fahrzeug erhält. In allen Fällen muss der Eigentümer für den Kopplungsprozess ein vertrauliches Einmalpasswort bereitstellen. Es wird verwendet, um mithilfe des Protokolls SPAKE2+ und der NIST-P-256-Kurve einen sicheren Kopplungskanal zu generieren. Bei Verwendung der App oder des Links in der E-Mail wird das Passwort automatisch auf das iPhone transferiert. Beim Starten des Kopplungsvorgangs über das Fahrzeug muss das Passwort hingegen manuell eingegeben werden.

Teilen von Schlüsseln

Das gekoppelte iPhone des Eigentümers kann Schlüssel mit iPhone-Geräten qualifizierter Familienmitglieder und Freunde (und mit ihren gekoppelten Apple Watch-Geräten) teilen, indem es mithilfe von iMessage und [IDS \(Apple Identity Service\)](#) eine gerätespezifische Einladung versendet. Alle das Teilen betreffenden Befehle werden unter Verwendung von IDS mit Ende-zu-Ende-Verschlüsselung ausgetauscht. Das gekoppelte iPhone-Gerät des Eigentümers erzwingt, dass der IDS-Kanal während des Kopplungsvorgangs nicht geändert wird, um das Weiterleiten von Einladungen zu verhindern.

Wird die Einladung akzeptiert, erstellt das iPhone-Gerät des Familienmitglieds oder der befreundeten Person einen digitalen Schlüssel und sendet danach die Zertifikatskette für die Schlüsselgenerierung zurück an das gekoppelte iPhone-Gerät des Eigentümers. Auf diese Weise wird sichergestellt, dass der Schlüssel auf einem authentischen Apple-Gerät erstellt wird. Das gekoppelte iPhone des Eigentümers signiert den öffentlichen ECC-Schlüssel des iPhone-Geräts des Familienmitglieds oder der befreundeten Person und sendet diese Signatur zurück an das iPhone-Gerät des Familienmitglieds oder der befreundeten Person. Der Signiervorgang auf dem Gerät des Eigentümers setzt die Benutzerauthentifizierung (per Face ID, Touch ID oder durch die Eingabe des Codes) und eine sichere Benutzerabsicht voraus (siehe [Verwendung von Face ID und Touch ID](#)). Die Autorisierung wird angefordert, wenn die Einladung versendet und im Secure Element gespeichert wird, damit sie zur Verfügung steht, wenn das befreundete Gerät die Signieranforderung zurücksendet. Die Schlüsselberechtigungen werden dem Fahrzeug entweder online durch den Fahrzeug-OEM-Server oder während der ersten Verwendung des geteilten Schlüssels bereitgestellt.

Löschen eines Schlüssels

Schlüssel können vom Gerät des Fahrzeugeigentümers aus oder im Fahrzeug selbst vom Gerät des Schlüsselnutzers gelöscht werden. Das Löschen wird auf dem iPhone des Schlüsselnutzers sofort umgesetzt, selbst wenn der Schlüsselnutzer den Schlüssel aktiv in Gebrauch hat. Daher wird eine deutliche Warnung angezeigt, bevor der Löschvorgang ausgeführt wird. Das Löschen von Schlüsseln im Fahrzeug kann jederzeit oder nur dann möglich sein, wenn das Fahrzeug online ist.

In beiden Fällen wird das Löschen auf dem Gerät des Schlüsselnutzers und im Fahrzeug an einen Schlüsselbestandsserver (KIS, Key Inventory Server) des Autoherstellers gemeldet, auf dem die für ein Fahrzeug ausgegebenen Schlüssel für Versicherungszwecke registriert werden.

Der Eigentümer kann das Löschen von der Rückseite seines Eigentümergebietes anfordern. Die Anforderung wird zunächst an den Autohersteller gesendet, damit der Schlüssel im Fahrzeug gelöscht wird. Die Bedingungen, unter denen der Schlüssel vom Fahrzeug entfernt wird, werden vom jeweiligen Autohersteller vorgegeben. Erst nachdem der Schlüssel vom Fahrzeug entfernt wurde, sendet der Server des Autoherstellers eine Aufkündigungsanforderung an das Gerät des Schlüsselnutzers.

Nachdem der Schlüssel auf dem Gerät aufgekündigt wurde, erstellt das Applet, mit dem die digitalen Autoschlüssel verwaltet werden, eine kryptografisch verschlüsselte Bescheinigung für die erfolgte Aufkündigung. Sie dient als Nachweis für das Löschen durch den Autohersteller und wird verwendet, um den Schlüssel vom KIS-Server zu entfernen.

NFC-Standardtransaktionen

Bei Fahrzeugen mit einem NFC-Schlüssel wird zwischen dem Lesegerät und einem iPhone ein sicherer Kanal initiiert, indem temporäre Schlüsselpaare auf dem Lesegerät und dem iPhone generiert werden. Mit einer Methode der Schlüsselvereinbarung kann auf beiden Seiten ein Shared Secret hergeleitet und für die Generierung eines geteilten symmetrischen Schlüssels verwendet werden. Hierbei kommen die Diffie-Hellman-Methode, eine Funktion für die Schlüsselherleitung, und Signaturen aus dem beim Kopplungsvorgang etablierten Langzeitschlüssel zum Einsatz.

Der temporäre öffentliche Schlüssel, der auf Fahrzeugseite generiert wird, wird mit dem privaten Langzeitschlüssel des Lesegeräts signiert, was seinerseits die Authentifizierung des Lesegeräts durch das iPhone zur Folge hat. Aus der Sicht des iPhone-Geräts ist dieses Protokoll darauf ausgelegt, dass sensible persönliche und personenbezogene Daten nicht gegenüber einem Angreifer offengelegt werden, der die Kommunikation abfängt.

Das iPhone-Gerät verwendet schließlich den etablierten sicheren Kanal, um die ID seines öffentlichen Schlüssels zusammen mit der Signatur, die aus einer datenbasierten Anfrage des Lesegeräts hergeleitet wird, und weiteren App-spezifischen Daten zu verschlüsseln. Durch diese Verifizierung der iPhone-Signatur durch das Lesegerät kann das Lesegerät das Gerät authentifizieren.

Schnelle Transaktionen

Das iPhone generiert ein Kryptogramm auf der Basis eines Shared Secret aus einer vorhergehenden Standardtransaktion. Dies ermöglicht dem Fahrzeug das schnelle Authentifizieren des Geräts in Performance-sensitiven Szenarien. Optional kann ein sicherer Kanal zwischen dem Fahrzeug und dem Gerät etabliert werden, indem Sitzungsschlüssel aus einem Shared Secret einer vorhergehenden Standardtransaktion und einem neuen temporären Schlüsselpaar hergeleitet werden. Erbringt das Fahrzeug den Nachweis, dass es den sicheren Kanal etablieren kann, wird es gegenüber dem iPhone authentifiziert.

BLE-/UWB-Standardtransaktionen

Bei Fahrzeugen mit einem UWB-Schlüssel wird eine Bluetooth LE-Sitzung zwischen dem Fahrzeug und dem iPhone hergestellt. Ähnlich wie bei NFC-Transaktionen wird ein gemeinsames geteiltes Secret auf beiden Seiten hergeleitet und für die Herstellung der sicheren Sitzung verwendet. Mit dieser Sitzung wird nachfolgend ein UWB Ranging Secret Key (URSK) hergeleitet und akzeptiert. Der URSK wird dem UWB-Radio auf dem Gerät des Benutzers und im Fahrzeug bereitgestellt, um die genaue Position des Benutzergeräts in der Nähe oder innerhalb des Fahrzeugs zu ermitteln. Das Fahrzeug verwendet dann die Position des Geräts, um zu entscheiden, ob das Aufschließen und Starten des Fahrzeugs erlaubt ist. URSKs besitzen eine vordefinierte TTL. Um eine Unterbrechung bei der Bereichsermittlung zu verhindern, wenn eine TTL abläuft, können URSKs vorab im Geräte-SE und im Fahrzeug-HSM/SE hergeleitet werden, während die sichere Bereichsermittlung nicht aktiv, aber BLE verbunden ist. Dadurch ist keine Standardtransaktion notwendig, um einen neuen URSK in zeitkritischen Situationen zu erzeugen. Der vorab hergeleitete URSK kann sehr schnell an die UWB-Radios des Fahrzeugs und des Geräts übertragen werden, um eine Unterbrechung der UWB-Bereichsermittlung zu verhindern.

Datenschutz und Privatsphäre

Der KIS (Key Inventory Server) des Autoherstellers speichert weder die Geräte-ID noch die SEID oder die Apple-ID. Auf dem Server wird nur eine veränderliche Kennung – die CA-Kennung – der jeweiligen Instanz gespeichert. Diese Kennung ist an keinerlei persönliche oder personenbezogene Daten auf dem Gerät oder an den Server gebunden. Sie wird gelöscht, wenn vom Benutzer (mit der Option „Alle Inhalte und Einstellungen löschen“) alle Daten und der gesamte Inhalt vom Gerät gelöscht werden.

Hinzufügen von ÖPNV- und eMoney-Karten zu Apple Wallet

In vielen Ländern weltweit können Benutzer auf unterstützten iPhone- oder Apple Watch-Geräten unterstützte ÖPNV- und eMoney-Karten zu Apple Wallet hinzufügen. Abhängig vom jeweiligen Betreiber kann dies durchgeführt werden, indem der Wert oder die ÖPNV-Karte einer physischen Karte in digitaler Form in Apple Wallet übernommen oder eine neue ÖPNV- oder eMoney-Karte über Apple Wallet oder die App des Kartenausstellers bereitgestellt wird. Nachdem ÖPNV-Karten zu Apple Wallet hinzugefügt wurden, können die Benutzer öffentliche Verkehrsmittel nutzen, indem sie ihr iPhone oder ihre Apple Watch in die Nähe eines entsprechenden Lesegeräts des Verkehrsbetriebs halten. Einige ÖPNV-Karten können auch für Zahlungen verwendet werden.

Funktionsweise von ÖPNV- und eMoney-Karten

Hinzugefügte ÖPNV- und eMoney-Karten werden mit dem iCloud-Account eines Benutzers verknüpft. Wenn der Benutzer mehr als eine Karte zu Apple Wallet hinzufügt, können Apple oder der Kartenaussteller die persönlichen Daten des Benutzers und die verknüpften Accountdaten möglicherweise zwischen Karten verbinden. ÖPNV- und eMoney-Karten und Transaktionen werden durch eine Reihe von hierarchischen kryptografischen Schlüsseln geschützt.

Während der Übertragung des Guthabens von einer physischen Karte an Apple Wallet müssen Benutzer kartenspezifische Informationen eingeben. Benutzer müssen möglicherweise auch persönliche Informationen bereitstellen, um nachzuweisen, dass sie im Besitz der Karte sind. Wird eine Karte von einem iPhone auf die Apple Watch übertragen, müssen beide Geräte online sein.

Das Guthaben kann über Beträge von einer Kredit-, Debit- oder Prepaid-Karte in Apple Wallet oder über die App des ÖPNV- oder eMoney-Anbieters aufgeladen werden. Weitere Informationen zur Sicherheit beim Aufladen des Guthabens mit Apple Pay sind unter [Kartenzahlung in Apps](#) zu finden. Weitere Informationen zum Bereitstellen der Karte in der App des Ausstellers sind unter [Kredit- oder Debitkarten aus der App des Kartenausstellers zu Apple Pay hinzufügen](#) zu finden.

Wenn das Bereitstellen von einer physischen Karte unterstützt wird, besitzt der ÖPNV- oder eMoney-Kartenaussteller die kryptografischen Schlüssel, die erforderlich sind, um die physische Karte zu authentifizieren und die eingegebenen Benutzerdaten zu überprüfen. Nach der Überprüfung kann das System eine Gerätekontonummer für das Secure Element erstellen und die neu hinzugefügte Karte mit dem übertragenen Guthaben in Apple Wallet aktivieren. Bei einigen Karten wird, nachdem die Bereitstellung der physischen Karte in digitaler Form abgeschlossen ist, die physische Karte deaktiviert.

Zum Abschluss beider Arten der Bereitstellung wird das Guthaben der Karte auf dem Gerät gespeichert, verschlüsselt und in einem spezifischen Applet im Secure Element gespeichert. Der Betreiber besitzt die Schlüssel, um kryptografische Operationen für Guthabentransaktionen mit den Kartendaten auszuführen.

Den Benutzern von ÖPNV-Karten stehen standardmäßig die Vorteile von „Express-ÖPNV“ zur Verfügung, damit sie ohne Face ID, Touch ID oder einen Code zahlen oder den Nahverkehr nutzen können. Wenn der Expressmodus aktiviert ist, kann an kontaktlosen Kartenlesegeräten auf Informationen wie kürzlich besuchte Stationen, letzte Transaktionen und zusätzliche Fahrscheine zugegriffen werden. Benutzer können die Anforderung einer Autorisierung per Face ID, Touch ID oder Code in der Einstellung „Wallet & Apple Pay“ aktivieren, indem sie „Express-ÖPNV“ deaktivieren. Der Expressmodus wird bei eMoney-Karten nicht unterstützt.

Wie bei anderen Apple Pay-Karten können Benutzer eMoney-Karten wie folgt sperren oder entfernen:

- Das Gerät mit „Wo ist?“ fernlöschen
- Modus „Verloren“ mit „Wo ist?“ aktivieren
- MDM-Fernlöschbefehl (Mobile Device Management) eingeben
- Alle Karten auf der Accountseite der Apple-ID entfernen
- Alle Karten auf iCloud.com entfernen
- Alle Karten aus Apple Wallet entfernen
- Die Karte in der App des Ausstellers entfernen

Die Apple Pay-Server fordern den Kartenaussteller auf, diese Karten zu sperren oder zu deaktivieren. Wenn ein Benutzer eine ÖPNV- oder eMoney-Karte von einem Online-Gerät entfernt, kann das Guthaben erstattet werden, indem sie wieder auf einem Gerät hinzugefügt wird, das mit derselben Apple-ID angemeldet ist. Die Wiederherstellung ist unter Umständen nicht möglich, wenn das Gerät offline, ausgeschaltet oder unbrauchbar ist.

Hinzufügen von ÖPNV- und eMoney-Karten zur Apple Watch eines Familienmitglieds

In iOS 15 und watchOS 8 kann der Organisator einer iCloud-Familie ÖPNV- und eMoney-Karten zu den Apple Watch-Geräten eines Familienmitglieds über die Watch-App vom iPhone hinzufügen. Beim Bereitstellen einer dieser Karten auf einer Apple Watch eines Familienmitglieds muss sich die Apple Watch in der Nähe des iPhone des Organisators befinden und mit diesem über WLAN oder Bluetooth verbunden sein. Die Familienmitglieder müssen die Zwei-Faktor-Authentifizierung für ihre Apple-ID aktiviert haben, damit dies möglich ist.

Familienmitglieder können eine Anfrage senden, um mit iMessage auf ihrer Apple Watch Geld zu einer ÖPNV- und eMoney-Karte hinzuzufügen. Der Nachrichteninhalt wird durch die Ende-zu-Ende-Verschlüsselung wie unter [Sicherheit bei iMessage – Übersicht](#) beschrieben geschützt. Das Hinzufügen von Geld zu einer Karte auf der Apple Watch eines Familienmitglieds kann über eine WLAN- oder Mobilfunkverbindung aus der Ferne durchgeführt werden. Das Gerät muss sich nicht in der Nähe befinden.

Hinweis: Diese Funktion ist möglicherweise nicht in allen Ländern oder Regionen verfügbar.

Kredit- und Debitkarten

In einigen Städten werden EMV-Karten (Europay International, MasterCard und VISA) beim Bezahlen von Fahrten mit öffentlichen Verkehrsmitteln von ÖPNV-Lesegeräten akzeptiert. Wenn der Benutzer eine Kredit- oder Debitkarte an ein solches Lesegerät hält, ist wie beim „Bezahlen mit Kredit- und Debitkarten in Geschäften“ eine Benutzerauthentifizierung erforderlich.

In iOS 12.3 (oder neuer) können einige vorhandene EMV-Kredit-/Debitkarten in Apple Wallet für „Express-ÖPNV“ aktiviert werden. Mit „Express-ÖPNV“ können Benutzer ohne Face ID, Touch ID oder Code bei unterstützten Nahverkehrsbetrieben für eine Fahrt bezahlen. Wenn ein Benutzer eine EMV-Kredit- oder -Debitkarte hinzufügt, wird die zuerst in Apple Wallet hinzugefügte Karte für „Express-ÖPNV“ aktiviert. Der Benutzer kann auf die Taste „Mehr“ vorne auf der Karte in Apple Wallet tippen und den Expressmodus ausschalten, indem er „Express-ÖPNV-Einstellungen“ > „Keine“ auswählt. Der Benutzer hat auch die Möglichkeit, in Apple Wallet eine andere Kredit- oder Debitkarte als seine „Express-ÖPNV-Karte“ auszuwählen. Für eine erneute Aktivierung des Expressmodus oder die Auswahl einer anderen Karte sind Face ID, Touch ID oder der Code erforderlich.

Apple Card und Apple Cash können für Express-ÖPNV verwendet werden.

Ausweisdokumente in Apple Wallet

Auf dem iPhone 8 (oder neueren Modellen) mit iOS 15.4 (oder neuer) und auf der Apple Watch Series 4 (oder neueren Modellen) mit watchOS 8.4 (oder neuer) können Benutzer ihren staatlichen Ausweis oder Führerschein zu Apple Wallet hinzufügen und auf ihr iPhone oder ihre Apple Watch tippen, um diese Dokumente reibungslos und sicher bei teilnehmenden Orten vorzulegen.

Hinweis: Diese Funktion ist nur bei teilnehmenden US-Bundesstaaten verfügbar.

Die Ausweisdokumente in Apple Wallet verwenden Sicherheitsfunktionen, die in die Hardware und Software des Geräts integriert sind. Diese Funktionen schützen die Identität des Benutzers und seine persönlichen Daten.

Hinzufügen eines Führerscheins oder staatlichen Ausweises zu Apple Wallet

Auf dem iPhone können Benutzer oben auf dem Bildschirm in Apple Wallet auf die Taste „Hinzufügen“ (+) tippen, um den Führerschein oder den staatlichen Ausweis hinzuzufügen. Wenn die Benutzer zum Zeitpunkt der Konfiguration eine gekoppelte Apple Watch besitzen, werden sie aufgefordert, ihren Führerschein oder staatlichen Ausweis auch zu Apple Wallet auf der Apple Watch hinzuzufügen.

Die Benutzer werden zuerst aufgefordert, mit ihrem iPhone die Vorder- und Rückseite ihres physischen Führerscheins oder staatlichen Ausweises zu scannen. Das iPhone prüft die Bildqualität und den Bildtyp, um sicherzustellen, dass die bereitgestellten Bilder für die ausstellende Behörde akzeptabel sind. Diese Bilder der Ausweisdokumente sind mit dem Schlüssel der ausstellenden Behörde auf dem Gerät verschlüsselt und werden an die ausstellende Behörde gesendet.

Als Nächstes wird der Benutzer aufgefordert, eine Reihe von Gesichts- und Kopfbewegungen durchzuführen. Diese Bewegungen werden vom Gerät des Benutzers und von Apple geprüft, um das Risiko zu minimieren, dass eine Person ein Foto, ein Video oder eine Maske verwendet, um das Ausweisdokument einer anderen Person zu Apple Wallet hinzuzufügen. Die Analyseergebnisse werden dann an die ausstellende Behörde gesendet, aber nicht das Video mit den Bewegungen selbst.

Um sicherzustellen, dass die Person, die das Ausweisdokument zu Apple Wallet hinzufügt, auch die Person ist, der das Ausweisdokument gehört, wird sie aufgefordert, ein Selfie aufzunehmen. Bevor das Foto des Benutzers an die ausgebende Behörde übermittelt wird, vergleichen die Apple-Server und das Gerät des Benutzers das Foto mit der Person, die die Gesichts- und Kopfbewegungen durchgeführt hat. Dadurch ist sichergestellt, dass das übermittelte Foto von einer lebenden Person stammt und mit der Person auf dem Ausweisdokument übereinstimmt. Nach dem Abgleich wird das Foto auf dem Gerät verschlüsselt und dann an die ausstellende Behörde gesendet, um es mit dem in der Behörde hinterlegten Foto für das Ausweisdokument zu vergleichen.

In einem letzten Schritt werden die Benutzer aufgefordert, eine Authentifizierung per Face ID oder Touch ID durchzuführen. Das Gerät des Benutzers verknüpft die biometrischen Daten von Face ID oder Touch ID mit dem staatlichen Ausweis, um sicherzustellen, dass nur die Person, die das Ausweisdokument zu diesem iPhone hinzugefügt hat, dieses Dokument auch vorlegen kann. Andere registrierte biometrische Daten können nicht verwendet werden, um das Vorlegen des Ausweises zu autorisieren. Dieser Vorgang wird ausschließlich auf dem Gerät durchgeführt und nicht an die ausstellende Behörde übermittelt.

Die ausstellende Behörde erhält nur die notwendigen Informationen, um den digitalen Ausweis zu konfigurieren. Dazu gehören die Bilder der Vorder- und Rückseite des Ausweises, die Daten aus dem PDF417-Barcode und das Selfie, das der Benutzer im Rahmen des Ausweisüberprüfungsprozesses erstellt hat. Der ausstellende US-Bundesstaat erhält außerdem einen einstelligen Wert für den Betrugsschutz, der auf den Gerätenutzungsmustern, den Einstellungsdaten und der persönlichen Apple-ID des Benutzers basiert. Letztendlich ist es aber die Entscheidung der ausstellenden Behörde, das Hinzufügen des Ausweises zu Apple Wallet zu genehmigen oder abzulehnen.

Nachdem die ausstellende Behörde das Hinzufügen des staatlichen Ausweises oder des Führerscheins zu Apple Wallet autorisiert hat, wird durch das iPhone ein Schlüsselpaar im Secure Element erzeugt, das den Ausweis des Benutzers an das entsprechende Gerät bindet. Beim Hinzufügen zur Apple Watch wird von der Apple Watch ein Schlüsselpaar im Secure Element erzeugt.

Wenn sich der Ausweis auf dem iPhone befindet, werden die Ausweisinformationen des Benutzers in Apple Wallet in einem verschlüsselten Format gespeichert und von der Secure Enclave geschützt.

Verwenden eines Führerscheins oder staatlichen Ausweises in Apple Wallet

Damit Benutzer ihren Ausweis in Apple Wallet nutzen können, müssen sie sich auf dem Face ID- oder Touch ID-Gerät authentifizieren, das mit dem Ausweis in Apple Wallet verknüpft ist, bevor das iPhone die Daten an das Identitätslesegerät übergibt.

Damit Benutzer ihren Ausweis in Apple Wallet auf der Apple Watch nutzen können, müssen sie ihr iPhone jedes Mal mit dem entsprechenden Face ID-Erscheinungsbild oder Touch ID-Fingerabdruck entsperren, wenn sie ihre Apple Watch anziehen. Im Anschluss können sie ihren Ausweis in Apple Wallet so lange ohne Authentifizierung nutzen, bis sie die Apple Watch wieder ablegen. Um dies zu ermöglichen, werden grundlegende Funktionen für das automatische Entsperren verwendet, die unter [Systemsicherheit bei watchOS](#) detailliert beschrieben sind.

Wenn ein Benutzer das iPhone oder die Apple Watch in die Nähe eines Identitätslesegeräts hält, wird ein Dialogfeld auf dem Gerät mit den Informationen angezeigt, die angefordert werden, von wem sie angefordert werden und ob beabsichtigt wird, diese Daten zu speichern. Nach der Autorisierung mit der zugehörigen Face ID oder Touch ID werden die angeforderten Identitätsdaten auf dem Gerät freigegeben.

Wichtig: Die Benutzer müssen zur Vorlage ihres Ausweises ihr Gerät nicht entsperren, vorzeigen oder übergeben.

Wenn Benutzer eine Bedienungshilfe wie Sprachsteuerung, Schaltersteuerung oder AssistiveTouch anstelle von Face ID oder Touch ID aktiviert haben, können sie ihren Code verwenden, um auf ihre Daten zuzugreifen oder sie vorzulegen.

Die Übermittlung von Identitätsdaten an das Identitätslesegerät erfolgt gemäß dem Standard ISO/IEC 18013-5, der mehrere Sicherheitsmechanismen bietet, mit denen Sicherheitsrisiken erkannt, verhindert und entschärft werden können. Zu diesen Maßnahmen gehören Integrität der Identitätsdaten und Fälschungsschutz, Gerätebindung, Einverständniserklärung und Vertraulichkeit der Benutzerdaten durch Funkverbindung.

Integrität der Identitätsdaten und Fälschungsschutz

Die Ausweisdokumente in Apple Wallet verwenden eine vom Aussteller bereitgestellte Signatur, um es ISO/IEC 18013-5-konformen Lesegeräten zu ermöglichen, den Ausweis eines Benutzers in Apple Wallet zu überprüfen. Darüber hinaus werden alle Datensätze auf dem Ausweisdokument in Wallet individuell gegen Fälschung geschützt. So kann das Identitätslesegerät einen bestimmten Teil des Datensatzes auf dem Ausweisdokument in Apple Wallet abfragen und Apple Wallet hat die Möglichkeit, nur diesen Datensatz bereitzustellen, um nur die angeforderten Daten zu teilen und dadurch den Datenschutz zu erhöhen.

Gerätebindung

Die Authentifizierung von Ausweisdokumenten in Apple Wallet verwendet eine Gerätesignatur, um das Klonen von Ausweisdokumenten und das Wiederholen einer Identitätstransaktion zu verhindern. Indem der private Schlüssel für die Authentifizierung eines Ausweisdokuments im Secure Element des iPhone gespeichert wird, ist das Ausweisdokument an das Gerät gebunden, für das die ausstellende Behörde es erstellt hat.

Einverständniserklärung

Die Authentifizierung von Lesegeräten für Ausweisdokumente in Apple Wallet authentifiziert den Identitätsleser mit dem im Standard ISO/IEC 18013-5 definierten Protokoll. Beim Vorlegen wird den Benutzern ein Symbol angezeigt, das vom Zertifikat des Lesegeräts stammt, um ihnen zu versichern, dass sie mit der beabsichtigten Partei interagieren.

Vertraulichkeit der Benutzerdaten durch Funkverbindung

Die Verschlüsselung der Sitzung stellt sicher, dass alle persönlichen Daten (PII), die zwischen dem Ausweisdokument in Apple Wallet und dem Identitätslesegerät ausgetauscht werden, verschlüsselt sind. Die Verschlüsselung erfolgt durch die Anwendungsschicht (Application Layer). Daher ist die Sicherheit der Sitzungsverschlüsselung nicht von der Sicherheit der Transportschicht (Transmission Layer) abhängig (z. B. NFC, Bluetooth und WLAN).

Ausweisdokumente in Apple Wallet schützen die Daten der Benutzer

Ausweisdokumente in Apple Wallet halten sich an den Prozess für die Geräteabfrage (Device Retrieval), der in ISO/IEC 18013-5 beschrieben ist. Mit der Geräteabfrage ist es nicht notwendig, während der Vorlage des Ausweisdokuments Serverabfragen durchzuführen. Auf diese Weise kann verhindert werden, dass Apple oder der Aussteller die Benutzer tracken können.

iMessage

Sicherheit bei iMessage – Übersicht

Apple iMessage ist ein Messaging-Dienst für iOS- und iPadOS-Geräte, Apple Watch und Mac-Computer. iMessage unterstützt Text und Anhänge (beispielsweise Fotos, Kontakte, Standorte und Links) sowie Anhänge direkt an einer Nachricht, etwa das „Daumen hoch“-Symbol. Nachrichten werden auf allen registrierten Geräten des Benutzers angezeigt, sodass ein Chat auf jedem beliebigen Gerät des Benutzers fortgesetzt werden kann. iMessage nutzt den Apple-Dienst für Push-Benachrichtigungen ([Apple Push Notification Service – APNS](#)) in vollem Umfang. Apple zeichnet keine Nachrichten oder Anhänge auf und der Inhalt wird mit einer Ende-zu-Ende-Verschlüsselung geschützt, sodass nur Sender und Empfänger darauf zugreifen können. Apple kann die Daten nicht entschlüsseln.

Wenn ein Benutzer iMessage auf einem Gerät aktiviert, erzeugt das Gerät Schlüsselpaare für die Verschlüsselung und die Signatur, die mit dem Dienst verwendet werden können. Für die Verschlüsselung gibt es einen RSA 1280-Bit-Verschlüsselungsschlüssel und einen EC 256-Bit-Verschlüsselungsschlüssel auf der NIST P-256-Kurve. Für Signaturen werden [Elliptic Curve Digital Signature Algorithm \(ECDSA\)](#) 256-Bit-Signaturschlüssel verwendet. Die privaten Schlüssel werden im Schlüsselbund des Geräts gespeichert und sind nur nach dem ersten Entsperren verfügbar. Die öffentlichen Schlüssel werden an den [Identitätsdienst von Apple \(IDS\)](#) gesendet, wo sie mit der Telefonnummer oder E-Mail-Adresse des Benutzers und der Adresse des Geräts für den APNS verknüpft werden.

Wenn Benutzer zusätzliche Geräte für iMessage aktivieren, werden ihre Verschlüsselung und öffentlichen Schlüssel für die Signatur, APNS-Adressen und verknüpften Telefonnummern ebenfalls dem Verzeichnisdienst hinzugefügt. Benutzer können auch mehrere E-Mail-Adressen hinzufügen, die über den Link in einer Bestätigungsmail verifiziert werden. Telefonnummern werden über das Mobilfunknetz und die SIM-Karte verifiziert. Bei einigen Netzwerken muss eine SMS verwendet werden. (Dem Benutzer wird ein Bestätigungsdialo angezeigt, wenn die SMS nicht kostenfrei ist.) Eine Verifizierung der Telefonnummer kann neben iMessage für einige Systemdienste wie FaceTime und iCloud erforderlich sein. Außerdem zeigen alle registrierten Geräte eines Benutzers eine Warnung an, wenn ein neues Gerät, eine neue Telefonnummer oder eine neue E-Mail-Adresse hinzugefügt wird.

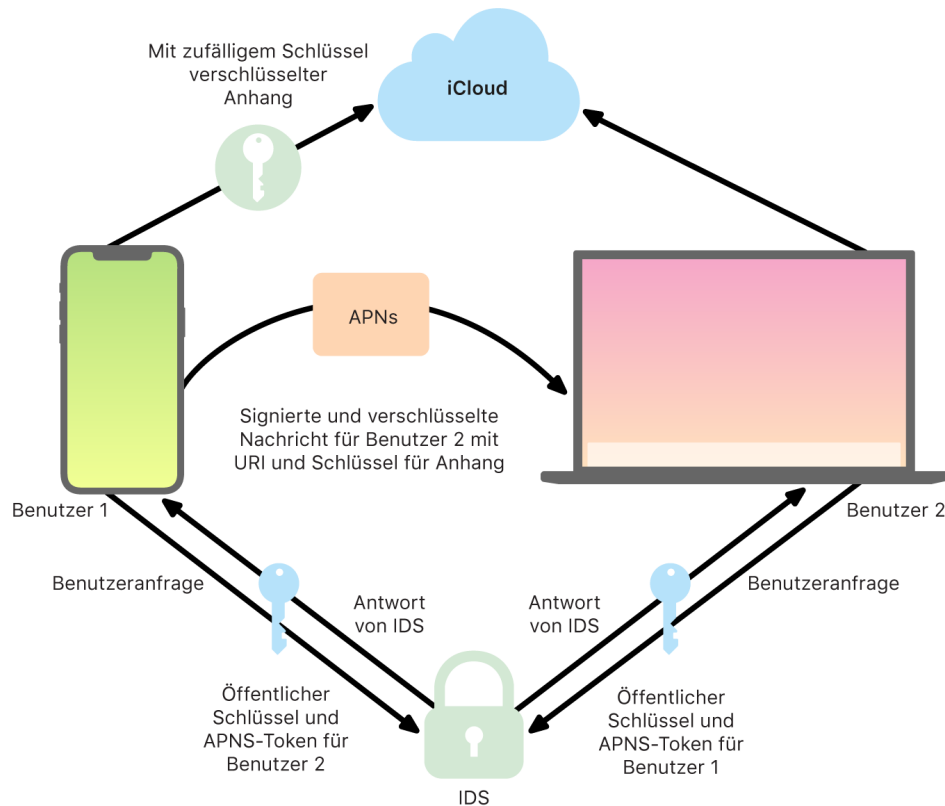
Sicheres Senden und Empfangen von Nachrichten in iMessage

Benutzer können in iMessage eine neue Konversation starten, indem sie eine Adresse oder einen Namen eingeben. Wenn sie eine Telefonnummer oder eine E-Mail-Adresse eingeben, kontaktiert das Gerät den Verzeichnisdienst (Apple Identity Service (IDS)), um die öffentlichen Schlüssel und APNS-Adressen für alle mit dem Benutzer verknüpften Geräte abzurufen. Wenn der Benutzer einen Namen eingibt, sucht das Gerät zuerst in den Kontakten des Benutzers nach Telefonnummern und E-Mail-Adressen, die mit diesem Namen verknüpft sind, und ruft dann die öffentlichen Schlüssel und APNS-Adressen vom IDS ab.

Die vom Benutzer gesendeten Nachrichten werden für alle Geräte des Empfängers einzeln verschlüsselt. Die öffentlichen Verschlüsselungsschlüssel und Signaturschlüssel der Empfangsgeräte werden vom IDS abgerufen. Das Sendegerät erzeugt für jedes Empfangsgerät einen zufälligen 88-Bit-Wert und verwendet diesen als HMAC-SHA-256-Schlüssel, um einen 40-Bit-Wert zu erstellen, der aus dem öffentlichen Schlüssel des Senders und des Empfängers und dem Klartext hergeleitet wird. Durch Verkettung des 88-Bit-Werts mit dem 40-Bit-Wert ergibt sich ein 128-Bit-Wert, der die Nachricht mittels AES im Zählermodus (CTR-Modus) verschlüsselt. Der 40-Bit-Wert wird auf Seiten des Empfängers verwendet, um die Integrität des entschlüsselten Klartexts zu überprüfen. Dieser pro Nachricht erzeugte AES-Schlüssel wird mit RSA-OAEP für den öffentlichen Schlüssel des Empfangsgeräts verschlüsselt. Die Kombination aus dem verschlüsselten Nachrichtentext und dem verschlüsselten Nachrichtenschlüssel wird mit SHA-1 als Hash codiert und der Hash wird mittels dem Elliptic Curve Digital Signature Algorithm (ECDSA) mit dem privaten Signaturschlüssel des Sendegeräts signiert. In iOS 13 (oder neuer) und iPadOS 13.1 (oder neuer), können Geräte anstelle der RSA-Verschlüsselung eine Elliptic Curve Integrated Encryption Scheme (ECIES)-Verschlüsselung verwenden.

Die dadurch entstehenden Nachrichten, je eine pro Empfangsgerät, bestehen aus dem verschlüsselten Nachrichtentext, dem verschlüsselten Nachrichtenschlüssel und der digitalen Signatur des Senders. Sie werden dann an APNS zur Weitersendung übertragen. Metadaten wie der Zeitstempel und die Routing-Informationen des APNS werden nicht verschlüsselt. Die Kommunikation mit APNS wird über einen TLS-Kanal mit Forward Secrecy verschlüsselt.

APNS können nur Nachrichten weiterleiten, die bis zu 4 bzw. 16 KB (abhängig von der iOS- oder iPadOS-Version) groß sind. Wenn der Nachrichtentext zu lang ist oder einen Anhang, z. B. ein Foto, enthält, wird der Anhang mit einem per AES im CTR-Modus zufällig erzeugten 256-Bit-Schlüssel verschlüsselt und in iCloud hochgeladen. Der AES-Schlüssel für den Anhang, sein **URI (Uniform Resource Identifier)** und ein SHA-1-Hash der verschlüsselten Form werden an den Empfänger als Inhalt der iMessage gesendet, wobei Vertraulichkeit und Integrität mit der normalen iMessage-Verschlüsselung geschützt werden, die im folgenden Diagramm gezeigt wird.



Bei Gruppenkonversationen wird dieser Prozess für jeden Empfänger und jedes Gerät wiederholt.

Auf der Empfängerseite erhält jedes Gerät eine Kopie der Nachricht vom APNS und ruft gegebenenfalls den Anhang aus iCloud ab. Die Telefonnummer oder E-Mail-Adresse des Absenders wird mit den Kontakten des Empfängers abgeglichen, sodass nach Möglichkeit ein Name angezeigt werden kann.

Wie bei allen Push-Benachrichtigungen wird die Nachricht nach der Zustellung beim APNS gelöscht. Im Gegensatz zu anderen APNS-Mitteilungen werden iMessage-Nachrichten in eine Warteliste eingefügt, wenn sich das Empfänger-Gerät im Offline-Modus befindet. Nachrichten werden auf Apple-Servern für maximal 30 Tage gespeichert.

Sichere Freigabe von Namen und Fotos in iMessage

Die Funktion von iMessage für die Freigabe von Namen und Fotos ermöglicht es dem Benutzer, mit iMessage einen Namen und ein Foto zu teilen. Der Benutzer kann seine Visitenkarteninformationen auswählen oder den Namen anpassen und ein Bild seiner Wahl hinzufügen. Die Funktion von iMessage für die Namens- und Fotofreigabe verwendet ein zweistufiges System zum Verteilen des Namens und des Fotos.

Die Daten werden in Felder unterteilt, jeweils separat verschlüsselt und authentifiziert und zusätzlich auch zusammen authentifiziert. Der genaue Vorgang wird im Folgenden beschrieben. Es gibt drei Felder:

- Name
- Foto
- Fotodateiname

Bei einem der ersten Schritte bei der Datenerstellung wird nach dem Zufallsprinzip ein 128-Bit-Eintragsschlüssel auf dem Gerät generiert. Dieser Eintragsschlüssel wird dann mit HKDF-HMAC-SHA-256 abgerufen, um drei Teilschlüssel zu erstellen: Key 1:Key 2:Key 3 = HKDF (Eintragsschlüssel, „Spitznamen“). Für jedes Feld wird ein zufälliger 96-Bit-IV-Wert (Initialization Vector) erzeugt und die Daten werden mittels AES-CTR und Key 1 verschlüsselt. Anschließend wird eine Authentifizierung (MAC) mit HMAC-SHA-256 unter Verwendung von Key 2 berechnet, die den Feldnamen, das Feld IV und den Feld-Chiffretext abdeckt. Abschließend wird ein Set einzelner Feld-MAC-Werte verkettet und deren MAC wird mit HMAC-SHA-256 unter Verwendung von Key 3 berechnet. Die 256-Bit-MAC wird mit den verschlüsselten Daten gespeichert. Die ersten 128 Bit dieser MAC-Adresse werden als RecordID verwendet.

Dieser verschlüsselte Eintrag wird dann in der öffentlichen CloudKit-Datenbank unter der RecordID abgelegt. Dieser Eintrag wird niemals verändert. Wann immer der Benutzer seinen Namen und sein Foto ändert, wird ein neuer verschlüsselter Eintrag generiert. Wenn Benutzer 1 entscheidet, seinen Namen und sein Foto mit Benutzer 2 zu teilen, sendet er den Eintragsschlüssel zusammen mit der RecordID in seiner iMessage-Payload, die [verschlüsselt](#) wird.

Wenn Benutzer 2 diese iMessage-Payload empfängt, stellt dessen Gerät fest, dass die Payload eine Kurznamen- und Foto-RecordID und einen Schlüssel enthält. Das Gerät von Benutzer 2 greift dann auf die öffentliche CloudKit-Datenbank zu, um den verschlüsselten Namen und das verschlüsselte Foto für die betreffende RecordID abzurufen und per iMessage zu senden.

Nach dem Abruf der Nachricht entschlüsselt das Gerät von Benutzer 2 die Payload und überprüft die Signatur anhand der RecordID. Ist die Prüfung erfolgreich, werden der Name und das Foto für Benutzer 2 präsentiert, der dann entscheiden kann, ob er diese zu seinen Kontakten hinzufügen oder für Nachrichten verwenden möchte.

Sicherheit bei Apple Messages for Business

Apple Messages for Business ist ein Messaging-Dienst, der es Benutzern ermöglicht, mittels der App „Nachrichten“ mit Unternehmen zu kommunizieren. Bei Apple Messages for Business behält der Benutzer stets die Kontrolle über die geführte Konversation. Er hat auch die Möglichkeit, die Konversation zu löschen und künftige Nachrichten von dem Unternehmen blockieren zu lassen. Aus Gründen des Datenschutzes erhält das Unternehmen weder die Telefonnummer oder die E-Mail-Adresse des Benutzers noch dessen iCloud-Accountinformationen. Stattdessen wird durch [IDS \(Apple Identity Service\)](#) eine eindeutige „undurchsichtige“ Kennung – die sogenannte „*Opaque ID*“ – generiert und mit dem Unternehmen geteilt. Die „*Opaque ID*“ ist eine eindeutige Kennung für die Beziehung zwischen der Apple-ID des Benutzers und der Geschäftskennung (Business ID) des Unternehmens. Für jedes Unternehmen, das per Apple Messages for Business kontaktiert wird, erhält der Benutzer eine andere undurchsichtige Kennung. Der Benutzer entscheidet, ob und wann persönliche Daten mit dem Unternehmen geteilt werden. Apple Messages for Business speichert keinen Konversationsverlauf.

Apple Messages for Business unterstützt verwaltete Apple-IDs aus [Apple Business Manager](#) und stellt fest, ob sie in [Apple School Manager](#) für iMessage und FaceTime aktiviert wurden.

Nachrichten, die an das Unternehmen gesendet werden, werden zwischen dem Gerät des Benutzers und den Messaging-Servern von Apple verschlüsselt, die auch für iMessage verwendet werden. Die Messaging-Server von Apple entschlüsseln diese Nachrichten im RAM und leiten sie über einen verschlüsselten Link unter Verwendung von TLS 1.2 an das Unternehmen. Nachrichten werden zu keinem Zeitpunkt in unverschlüsselter Form gespeichert, während sie mit dem Dienst Apple Messages for Business transferiert werden. Die Antworten des Unternehmens werden ebenfalls unter Verwendung von TLS 1.2 an die Messaging-Server von Apple gesendet, wo sie mithilfe des spezifischen öffentlichen Schlüssels des jeweiligen Empfängergeräts entschlüsselt werden.

Wenn das Gerät des Benutzers online ist, wird die Nachricht sofort zugestellt und nicht im Cache der Messaging-Server von Apple zwischengespeichert. Wenn das Gerät des Benutzers nicht online ist, wird die Nachricht in verschlüsselter Form für einen Zeitraum von maximal 30 Tagen im Cache aufbewahrt, damit sie dem Benutzer zugestellt werden kann, wenn sein Gerät wieder online ist. Sobald das Gerät wieder online ist, wird die Nachricht zugestellt und danach aus dem Cache gelöscht. Nach Ablauf der Aufbewahrungsfrist von 30 Tagen verfällt eine nicht zugestellte Nachricht; sie wird dann unwiderruflich gelöscht.

FaceTime-Sicherheit

FaceTime ist der Dienst für Video- und Audioanrufe von Apple. Für FaceTime-Anrufe wird, genau wie für iMessage, [APNS \(Apple-Dienst für Push-Benachrichtigungen\)](#) zum Herstellen der Verbindung mit den registrierten Geräten des Benutzers verwendet. Die Audio/Video-Inhalte eines FaceTime-Anrufs werden mit einer Ende-zu-Ende-Verschlüsselung geschützt, sodass nur Sender und Empfänger auf sie zugreifen können. Apple kann die Daten nicht entschlüsseln.

Die erste FaceTime-Verbindung wird über die Apple-Serverinfrastruktur hergestellt, die Datenpakete zwischen den registrierten Geräten der Benutzer weiterleitet. Die Geräte verifizieren ihre Identitätszertifikate und erstellen ein Shared Secret für jede Sitzung mittels APNS-Benachrichtigungen und STUN-Nachrichten (Session Traversal Utilities for NAT) über die weitergeleitete Verbindung. Das Shared Secret wird verwendet, um Sitzungsschlüssel für Medienkanäle abzuleiten, die mit SRTP (Secure Real-time Transport Protocol) gestreamt werden. SRTP-Pakete werden mit AES-256 im Counter Mode verschlüsselt und mit HMAC-SHA-1 überprüft. Falls möglich, verwendet FaceTime nach der ersten Verbindung und der Sicherheitskonfiguration STUN und ICE (Internet Connectivity Establishment) zum Erstellen einer Peer-to-Peer-Verbindung zwischen den Geräten.

Die Option „FaceTime-Gruppe“ erweitert FaceTime und unterstützt jetzt bis zu 33 Teilnehmer gleichzeitig. Wie bei den klassischen Eins-zu-Eins-FaceTime-Anrufen werden auch diese Anrufe zwischen den Geräten der eingeladenen Teilnehmer Ende-zu-Ende-verschlüsselt. Während Infrastruktur und Design der Eins-zu-Eins-FaceTime-Anrufe zum Großteil weiterverwendet werden, bieten diese Gruppenanrufe einen Verschlüsselungsmechanismus, der auf der vom Apple-IDS (Apple Identity Service) bereitgestellten Authentizität aufgebaut ist. Dieses Protokoll bietet Forward Secrecy, was bedeutet, dass selbst bei Missbrauch eines Benutzergeräts die Inhalte früherer Anrufe nicht nach außen dringen. Sitzungsschlüssel werden via AES-SIV verpackt und mit einer ECIES-Konstruktion mit temporären P-256-ECDH-Schlüsseln an die Teilnehmer verteilt.

Wird eine neue Telefonnummer oder E-Mail-Adresse zu einem aktiven FaceTime-Gruppenanruf hinzugefügt, richten die aktiven Geräte neue Medienschlüssel ein und teilen zuvor verwendete Schlüssel niemals mit den zuletzt eingeladenen Geräten.

Wo ist?

Sicherheit von „Wo ist?“

Die Funktion „Wo ist?“ für Apple-Geräte basiert auf einer wegweisenden Verschlüsselung öffentlicher Schlüssel.

Übersicht

Die App „Wo ist?“ kombiniert „Mein iPhone suchen“ und „Freunde suchen“ in iOS, iPadOS und macOS zu einer App. Die Option „Wo ist?“ kann Benutzer dabei unterstützen, ein verlorenes Gerät zu finden, selbst dann, wenn ein Mac offline ist. Ein Online-Gerät kann seinen Standort einfach via iCloud melden. Die Option „Wo ist?“ funktioniert offline, indem Bluetooth-Signale vom verlorenen Gerät über kurze Distanzen gesendet werden, die von anderen Apple-Geräten in der Nähe entdeckt werden können. Diese Geräte senden die entdeckten Ortsdaten des verlorenen Geräts dann an iCloud weiter, sodass Benutzer das Gerät in der App „Wo ist?“ orten können. Hierbei bleiben die Daten aller beteiligten Benutzer geschützt. Die Option „Wo ist?“ funktioniert auch mit einem Mac, der offline oder im Ruhemodus ist.

Mithilfe von Bluetooth und unzähligen iOS-, iPadOS- und macOS-Geräten, die weltweit aktiv sind, kann der Benutzer ein verlorenes Gerät aufspüren, auch wenn es sich nicht mit einem WLAN oder Mobilfunknetz verbinden kann. Jedes iOS-, iPadOS- oder macOS-Gerät, auf dem die Offline-Suche in den Einstellungen von „Wo ist?“ aktiviert ist, kann als „Suchgerät“ fungieren. Dies bedeutet, dass dieses Gerät das Vorhandensein eines verlorenen Offline-Geräts mittels Bluetooth erkennen und dann seine Netzwerkverbindung nutzen kann, um dem Eigentümer den ungefähren Standort zu melden. Wenn ein Gerät die Offline-Suche aktiviert hat, so bedeutet dies auch, dass es von anderen Teilnehmern auf die gleiche Weise lokalisiert werden kann. Die gesamte Interaktion erfolgt mit Ende-zu-Ende-Verschlüsselung, ist anonym und auf Batterie- und Dateneffizienz ausgelegt. Daher sind die Auswirkungen auf die Batterielebensdauer und die Nutzung des Datentarifs minimal und der Datenschutz wird verbessert.

Hinweis: Die Option „Wo ist?“ ist möglicherweise nicht in allen Ländern oder Regionen verfügbar.

Ende-zu-Ende-Verschlüsselung

„Wo ist?“ basiert auf einer fortschrittlichen Verschlüsselung öffentlicher Schlüssel. Wenn die Offline-Suche in den Einstellungen von „Wo ist?“ aktiviert ist, wird direkt auf dem Gerät auf Basis einer elliptischen Kurve (EC) ein privates P-224-Verschlüsselungspaar erstellt, das als $\{d, P\}$ bezeichnet wird. Dabei ist d der private Schlüssel und P der öffentliche Schlüssel. Zusätzlich werden ein 256-Bit-Secret SK_0 und ein Zähler i als null initialisiert. Dieses private Schlüsselpaar und das Secret werden nicht an Apple gesendet. Sie werden nur mit den anderen Geräten des Benutzers mittels Ende-zu-Ende-Verschlüsselung im iCloud-Schlüsselbund synchronisiert. Das Secret und der Zähler werden verwendet, um den aktuellen symmetrischen Schlüssel SK_i mit der folgenden rekursiven Konstruktion abzuleiten: $SK_i = \text{KDF}(SK_{i-1}, \text{„update“})$.

Basierend auf dem Schlüssel SK_i werden zwei große Ganzzahlen u_i und v_i mit $(u_i, v_i) = KDF(SK_i, \text{"diversify"})$ berechnet. Sowohl der als d bezeichnete private P-224-Schlüssel als auch der als P bezeichnete zugehörige öffentliche Schlüssel werden dann mit einer affinen Relation abgeleitet, die zwei Ganzzahlen zum Berechnen eines kurzzeitig gültigen Schlüsselpaars nutzt: der abgeleitete private Schlüssel ist d_i wobei $d_i = u_i * d + v_i$ (modulo Ordnung von P-224-Kurve) und der entsprechende öffentliche Teil P_i ist und $P_i = u_i * P + v_i * G$ verifiziert.

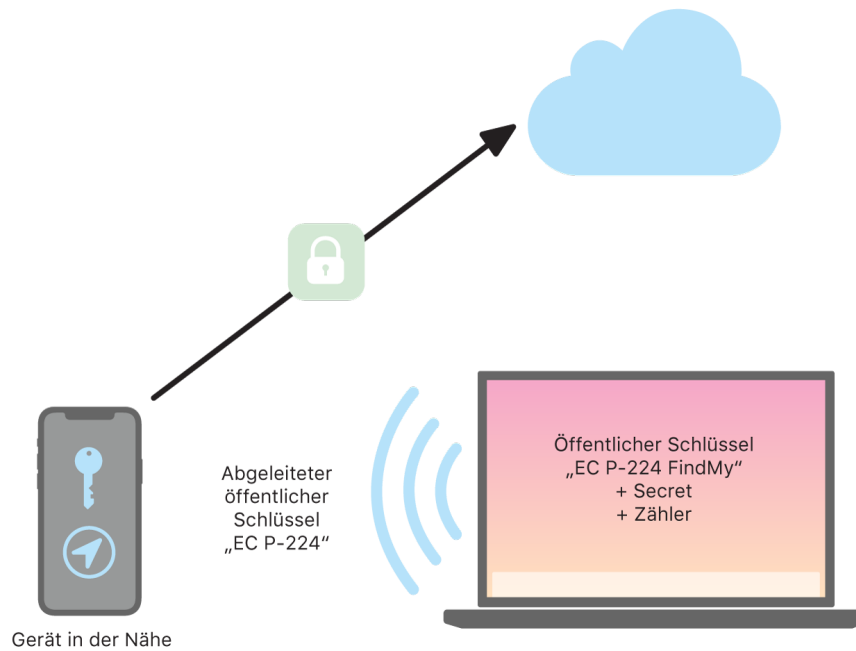
Wenn ein Gerät verloren geht und sich nicht mit dem WLAN oder Mobilfunknetz verbinden kann (etwa ein MacBook Pro, das auf einer Parkbank vergessen wird), beginnt es, in regelmäßigen Abständen den abgeleiteten öffentlichen Schlüssel P_i für einen begrenzten Zeitraum in einer Bluetooth-Payload zu übertragen. Dank P-224 passt die Darstellung des öffentlichen Schlüssels in eine einzige Bluetooth-Payload. Die in der Nähe befindlichen Geräte können beim Auffinden des Offline-Geräts helfen, indem sie ihren Standort mit dem öffentlichen Schlüssel verschlüsseln. Ungefähr alle 15 Minuten wird der öffentliche Schlüssel durch einen neuen ersetzt, wobei ein inkrementierter Wert des Zählers und der oben beschriebene Vorgang verwendet werden, sodass der Benutzer nicht mittels einer dauerhaften Kennung verfolgt werden kann. Der Ableitungsmechanismus verhindert, dass die verschiedenen öffentlichen Schlüssel P_i mit demselben Gerät verknüpft werden.

Wahren der Anonymität von Benutzern und Geräten

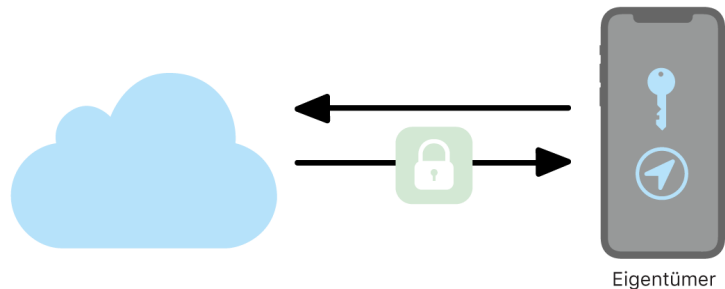
Es wird nicht nur sichergestellt, dass die Ortsdaten und weitere Daten vollständig verschlüsselt sind, sondern die Identitäten der Teilnehmer bleiben auch füreinander und für Apple anonym. Der Datenverkehr, der von den Suchgeräten an Apple gesendet wird, enthält keinerlei Authentifizierungsinformationen – weder in den Inhalten noch in den Headern. Infolgedessen erfährt Apple nicht, wer der Finder ist oder wessen Gerät gefunden wurde. Darüber hinaus protokolliert Apple keine Informationen, die die Identität des Finders verraten, und speichert keine Informationen, mit denen jemand Finder und Eigentümer in Beziehung zueinander setzen könnte. Der Eigentümer des Geräts erhält lediglich die verschlüsselten Ortsdaten, die in der App „Wo ist?“ entschlüsselt und angezeigt werden und keinerlei Hinweis darauf geben, wer das Gerät gefunden hat.

Auffinden verlorener Apple-Geräte mit „Wo ist?“

Apple-Geräte innerhalb der Bluetooth-Reichweite, bei denen die Offline-Suche aktiviert ist, können ein Signal von einem anderen Apple-Gerät erkennen, auf dem „Wo ist?“ aktiviert ist, und den aktuellen Broadcast-Schlüssel P_i lesen. Mit einer ECIES-Konstruktion und dem öffentlichen Schlüssel P_i der Übertragung verschlüsseln Suchgeräte ihre aktuellen Ortsdaten und leiten sie an Apple weiter. Der verschlüsselte Standort wird einem Serverindex zugewiesen, der als SHA-256-Hash-Wert des öffentlichen P-224-Schlüssels P_i berechnet und von der Bluetooth-Payload abgerufen wird. Apple hat keinerlei Möglichkeit, den Verschlüsselungsschlüssel zu erfahren, und kann daher den vom Suchgerät verschlüsselten Standort nicht lesen. Der Eigentümer des verlorenen Geräts kann den Index rekonstruieren und den verschlüsselten Ort entschlüsseln.



Beim Versuch das verlorene Gerät zu finden, wird ein erwarteter Bereich von Zählerwerten für den Zeitraum der Standortsuche geschätzt. Mit Kenntnis des ursprünglichen privaten P-224-Schlüssels d und des geheimen Werts SK_i im Bereich der Zählerwerte für den Suchzeitraum kann der Eigentümer dann die Wertegruppe $\{d_i, \text{SHA-256}(P_i)\}$ für den gesamten Suchzeitraum rekonstruieren. Das zum Lokalisieren des verlorenen Geräts verwendete Eigentümergerät kann mit dem Satz von Indexwerten $\text{SHA-256}(P_i)$ Abfragen an den Server stellen und die verschlüsselten Standorte vom Server laden. Die App „Wo ist?“ entschlüsselt daraufhin die verschlüsselten Standorte mit den passenden privaten Schlüsseln d_i und zeigt den ungefähren Standort des verlorenen Geräts in der App an. Ortsberichte mehrerer Suchgeräte werden von der App des Eigentümers kombiniert, um so den Standort genauer ermitteln zu können.



Auffinden von Geräten, die offline sind

Wenn ein Benutzer „Mein iPhone suchen“ auf seinem Gerät aktiviert hat, wird standardmäßig auch die Option für die Offline-Suche aktiviert, wenn er sein Gerät auf iOS 13 (oder neuer), iPadOS 13.1 (oder neuer) bzw. macOS 10.15 (oder neuer) aktualisiert. Dadurch wird sichergestellt, dass jeder Benutzer die bestmöglichen Chancen hat, sein Gerät bei Verlust wiederzufinden. Will der Benutzer an der Offline-Suche nicht mehr teilnehmen, kann er sie jederzeit in den Einstellungen von „Wo ist?“ auf seinem Gerät deaktivieren. Wenn die Offline-Suche deaktiviert ist, fungiert das Gerät nicht mehr als Suchgerät und kann von anderen Suchgeräten nicht mehr erkannt werden. Der Benutzer kann das Gerät aber weiterhin lokalisieren, sofern es mit einem WLAN oder Mobilfunknetz verbunden ist.

Wenn ein verlorenes Gerät lokalisiert wird, erfährt der Benutzer in einer Mitteilung und einer E-Mail, dass das Gerät gefunden wurde. Der Benutzer öffnet die App „Wo ist?“ und wählt den Tab „Geräte“, um den Standort des fehlenden Geräts anzuzeigen. Anstatt das Gerät auf einer leeren Karte zu sehen, wie es vor dem Auffinden dargestellt worden wäre, zeigt „Wo ist?“ einen Ort auf der Karte mit einer ungefähren Adresse und Informationen dazu, wie lange es her ist, seit das Gerät gefunden wurde. Wenn weitere Ortsberichte eintreffen, werden der aktuelle Standort und die Zeitangabe automatisch aktualisiert. Der Benutzer kann zwar auf einem Offline-Gerät keinen Signalton abspielen und es nicht per Fernzugriff löschen, er kann aber die Standortdaten nutzen, um Schritte nachzuverfolgen, oder andere Maßnahmen ergreifen, um das Gerät wieder in seinen Besitz zu bringen.

Continuity (Integration)

Sicherheit bei Continuity (Integration) – Übersicht

Continuity nutzt Technologien wie iCloud, Bluetooth und WLAN, um es dem Benutzer zu ermöglichen, eine Aktivität von einem Gerät auf ein anderes zu übertragen, Telefonanrufe zu tätigen und zu empfangen, SMS zu senden und zu empfangen und die Internetverbindung über Mobilfunk gemeinsam zu nutzen.

Handoff-Sicherheit

Apple handhabt Handoffs auf sichere Weise – von einem Gerät auf ein anderes Gerät ebenso wie zwischen einer nativen App und einer Website, sogar Handoffs großer Datenmengen.

Sichere Funktionsweise von Handoff

Mit Handoff kann der Benutzer automatisch alles, woran er auf seinem iOS-, iPadOS- und macOS-Gerät arbeitet, von einem Gerät an ein anderes, in der Nähe befindliches Gerät übertragen. Mit Handoff kann der Benutzer das Gerät wechseln und sofort weiterarbeiten.

Meldet sich der Benutzer auf einem zweiten Handoff-fähigen Gerät bei iCloud an, wird zwischen den Geräten mittels Bluetooth Low Energy (BLE) 4.2 eine Out-of-Band-Kopplung über APNS hergestellt. Die einzelnen Nachrichten werden ähnlich wie für iMessage verschlüsselt. Wenn die Geräte gekoppelt wurden, erstellt jedes von ihnen einen symmetrischen 256-Bit AES-Schlüssel, der im [Schlüsselbund](#) des jeweiligen Geräts gespeichert wird. Mit diesem Schlüssel können die BLE-Ankündigungen, die den anderen in iCloud gekoppelten Geräten die aktuelle Aktivität des Geräts mitteilen, verschlüsselt und authentifiziert werden, wobei zum Schutz vor Replay-Attacken AES-256 im GCM-Modus verwendet wird.

Wenn ein Gerät zum ersten Mal eine Ankündigung von einem neuen Schlüssel erhält, stellt es eine BLE-Verbindung zum Absender her und führt einen Schlüsselaustausch für die Verschlüsselung von Ankündigungen durch. Diese Verbindung wird mit der Standardverschlüsselung für BLE 4.2 gesichert und die einzelnen Nachrichten werden zusätzlich verschlüsselt. Die Verschlüsselung ähnelt damit der für iMessage. In einigen Situationen werden diese Nachrichten via APNS statt BLE gesendet. Die Payload der Aktivität wird genau wie eine iMessage geschützt und übertragen.

Handoff zwischen nativen Apps und Websites

Handoff ermöglicht es nativen iOS-, iPadOS- oder macOS-Apps, Webseiten auf Domains, deren Eigentümer der Entwickler der App ist, fortzusetzen. Die Benutzeraktivität in der nativen App kann auch in einem Internetbrowser fortgesetzt werden.

Damit native Apps keine Websites fortsetzen können, deren Eigentümer nicht der Entwickler ist, muss die App belegen, dass sie die Web-Domain, die sie fortsetzen will, tatsächlich kontrolliert. Die Kontrolle über die Domain einer Website wird über die Methode für die gemeinsam genutzten Internetanmeldedaten überprüft. [Zugriff durch Apps auf gesicherte Passwörter](#) enthält weitere Details hierzu. Das System muss die Kontrolle der App über den Domain-Namen validieren, bevor sie Handoff für die Benutzeraktivität verwenden kann.

Ursprung für das Handoff einer Webseite kann jeder Browser sein, der die Handoff-APIs unterstützt. Wenn der Benutzer eine Webseite öffnet, kündigt das System den Domain-Namen der Webseite in den verschlüsselten Ankündigungsbytes für Handoff an. Nur die anderen Geräte des Benutzers können die Ankündigungsbytes entschlüsseln.

Auf dem Empfängergerät erkennt das System, dass eine installierte native App Handoff von dem angekündigten Domain-Namen annimmt, und zeigt das Symbol für die native App als Handoff-Option an. Wird sie gestartet, empfängt die App die vollständige URL und den Titel der Webseite. Es werden keine anderen Informationen vom Browser an die native App übertragen.

Demgegenüber kann eine native App eine Fallback-URL angeben, wenn auf dem Empfängergerät die native App nicht installiert ist. In diesem Fall zeigt das System den Standardbrowser des Benutzers als Handoff-Option an (wenn dieser Browser Handoff-APIs übernommen hat). Wird Handoff angefordert, öffnet der Browser die Fallback-URL, die die App gesendet hat. Die Fallback-URL ist nicht auf Domain-Namen beschränkt, deren Eigentümer der Entwickler der nativen App ist.

Handoff für größere Datenmengen

Zusätzlich zu der Grundfunktion von Handoff können manche Apps auch APIs zum Senden größerer Datenmengen über die von Apple entwickelte (und AirDrop ähnliche) Peer-To-Peer-WLAN-Technologie senden. Beispielsweise verwendet Mail diese APIs, um das Handoff eines E-Mail-Entwurfs, einschließlich großer Anhänge, zu ermöglichen.

Nutzt eine App diese APIs, wird der Austausch zwischen den Geräten wie bei Handoff gestartet. Nach dem Empfang der ersten Nutzerdaten über Bluetooth Low Energy (BLE) startet das Empfängergerät eine neue Verbindung über WLAN. Diese Verbindung ist verschlüsselt (mit TLS) und sorgt mit einer über den iCloud-Schlüsselbund geteilten Identität für Vertrauen. Die Identität in den Zertifikaten wird mit der Identität des Benutzers abgeglichen. Die nachfolgenden Nutzerdaten werden über diese verschlüsselte Verbindung gesendet, bis die Übertragung abgeschlossen ist.

Universelle Zwischenablage

Die universelle Zwischenablage nutzt Handoff für die sichere Übertragung der Inhalte der Zwischenablage zwischen Geräten. Dadurch ist es möglich, etwas auf einem Gerät zu kopieren und auf einem anderen Gerät einzufügen. Der Inhalt wird dabei auf gleiche Weise wie andere Handoff-Daten geschützt und standardmäßig mit der universellen Zwischenablage geteilt, es sei denn, der App-Entwickler lässt die Freigabe nicht zu.

Die Daten in der Zwischenablage sind für Apps zugänglich, unabhängig davon, ob der Benutzer die Zwischenablage in die App eingesetzt hat. Mit der universellen Zwischenablage wird dieser Datenzugriff auf Apps ausgeweitet, die auf anderen Geräten des Benutzers ausgeführt werden (wie bei der iCloud-Anmeldung festgelegt).

Sicherheit bei iPhone-Mobilanrufumleitung

Wenn ein Mac, iPad, iPod touch oder HomePod des Benutzers dasselbe WLAN wie sein iPhone verwendet, können über die Mobilfunkverbindung des iPhone Telefonanrufe gestartet oder empfangen werden. Für die Konfiguration müssen die Geräte mit demselben Apple-ID-Account bei iCloud und FaceTime angemeldet sein.

Bei einem eingehenden Anruf werden alle konfigurierten Geräte über den Apple-Dienst für Push-Benachrichtigungen (APNS) benachrichtigt, wobei für jede Benachrichtigung dieselbe Ende-zu-Ende-Verschlüsselung wie für iMessage verwendet wird. Geräte im selben Netzwerk zeigen die Mitteilung für eingehende Anrufe an. Nimmt der Benutzer den Anruf entgegen, werden die Audiodaten nahtlos über eine sichere Peer-To-Peer-Verbindung zwischen den beiden Geräten übertragen.

Wird ein Anruf auf einem Gerät angenommen, wird das Klingeln auf über iCloud gekoppelten Geräten in der Nähe durch eine kurze Ankündigung mit Bluetooth Low Energy (BLE) beendet. Die Ankündigungsbytes werden mit derselben Methode wie Handoff-Ankündigungen verschlüsselt.

Ausgehende Anrufe werden ebenfalls über APNS umgeleitet und die Audiodaten werden in ähnlicher Weise über eine sichere Peer-To-Peer-Verbindung zwischen den Geräten übertragen. Der Benutzer kann die Mobilanrufumleitung auf einem Gerät deaktivieren, indem er in den FaceTime-Einstellungen „iPhone-Mobilanrufe“ deaktiviert.

Sicherheit bei iPhone-Nachrichtenweiterleitung

Mit der Nachrichtenweiterleitung werden SMS vom iPhone automatisch auf die registrierten iPad-, iPod touch- und Mac-Geräte des Benutzers übertragen. Alle Geräte müssen beim Dienst „iMessage“ mit derselben Apple-ID angemeldet sein. Wenn die Nachrichtenweiterleitung und die Zwei-Faktor-Authentifizierung aktiviert sind, erfolgt die Registrierung auf Geräten im Circle of Trust eines Benutzers automatisch. Andernfalls wird die Registrierung für jedes Gerät bestätigt, indem ein vom iPhone erzeugter zufälliger sechsstelliger Code auf ihm eingegeben wird.

Sind die Geräte verknüpft, verschlüsselt das iPhone eingehende SMS-Textnachrichten und leitet sie an die anderen Geräte weiter, wobei die im Abschnitt [Sicherheit bei iMessage – Übersicht](#) beschriebenen Methoden zum Einsatz kommen. Die Antworten werden mit derselben Methode an das iPhone zurückgesendet, das die Antwort dann über den Mobilfunkanbieter als SMS verschickt. Die Nachrichtenweiterleitung kann in den Nachrichteneinstellungen aktiviert oder deaktiviert werden.

Sicherheit bei Instant Hotspot

Ein Instant Hotspot ermöglicht es anderen Geräten, eine Verbindung zum persönlichen Hotspot eines iOS- oder iPadOS-Geräts herzustellen. iOS- und iPadOS-Geräte, die Instant Hotspot unterstützen, verwenden Bluetooth Low Energy (BLE), um Geräte zu erkennen, die beim selben iCloud-Account angemeldet sind, und mit ihnen zu kommunizieren. Dies trifft auch auf Accounts zu, die mit der Familienfreigabe (in iOS 13 und iPadOS) verwendet werden. Kompatible Mac-Computer mit OS X 10.10 (oder neuer) verwenden dieselbe Technologie, um iOS- und iPadOS-Geräte mit Instant Hotspot zu erkennen und um mit ihnen zu kommunizieren.

Wenn der Benutzer auf einem Gerät die WLAN-Einstellungen eingibt, sendet das Gerät eine BLE-Ankündigung mit einer Kennung, die alle bei demselben iCloud-Account angemeldeten Geräte verwenden. Diese Kennung wird aus einer mit dem iCloud-Account verknüpften DSID (Destination Signaling Identifier) erzeugt und regelmäßig geändert. Wenn sich andere bei demselben iCloud-Account angemeldeten Geräte in unmittelbarer Nähe befinden und den persönlichen Hotspot unterstützen, erkennen sie das Signal und signalisieren ihre Verfügbarkeit, um Instant Hotspot zu nutzen.

Wenn ein Benutzer, der nicht Teil der Familienfreigabe ist, ein iPhone oder iPad als persönlichen Hotspot auswählt, wird eine Anfrage für die Aktivierung des persönlichen Hotspots an das Gerät gesendet. Die Anfrage wird über eine Verbindung gesendet, die die BLE-Standardverschlüsselung verwendet, und die Anfrage selbst wird ähnlich wie eine iMessage-Nachricht verschlüsselt. Das Gerät sendet dann über dieselbe BLE-Verbindung mit derselben nachrichtenspezifischen Verschlüsselung die Verbindungsinformationen für den persönlichen Hotspot.

Für Benutzer, die an der Familienfreigabe teilnehmen, werden Verbindungsinformationen für Personal Hotspot auf sichere Weise mit einem Mechanismus geteilt, die dem ähneln, der von HomeKit-Geräten zum Synchronisieren von Informationen verwendet wird. Vor allem wird die Verbindung, die Hotspot-Informationen zwischen Benutzern teilt, mit einem kurzlebigen ECDH-Schlüssel (Curve25519) gesichert, der mit den gerätespezifischen öffentlichen Ed25519-Schlüsseln des Benutzers authentifiziert wird. Die verwendeten öffentlichen Schlüssel wurden zuvor zwischen den Teilnehmern der Familienfreigabe synchronisiert. Dazu wurde beim Einrichten der Familienfreigabe IDS eingesetzt.

Netzwerksicherheit

Netzwerksicherheit – Übersicht

Zusätzlich zu den integrierten Sicherheitsmaßnahmen, die Apple zum Schutz der auf dem Apple-Gerät gespeicherten Daten verwendet, gibt es viele Maßnahmen, die Organisationen ergreifen können, damit die Sicherheit von Informationen bei der Übertragung an und von einem Gerät gewahrt bleibt. Alle diese Sicherheitsvorkehrungen und Sicherheitsmaßnahmen sind Teil der Netzwerksicherheit.

Benutzer müssen von überall auf der Welt auf Unternehmensnetzwerke zugreifen können. Daher muss sichergestellt werden, dass sie über die entsprechenden Zugriffsrechte verfügen und ihre Daten während der Übertragung zuverlässig geschützt sind. Um diese Sicherheitsziele zu erreichen, sind in iOS, iPadOS und macOS bewährte Technologien und die aktuellen Standards für Datenverbindungen sowohl über WLAN als auch über Mobilfunk integriert. Aus diesem Grund verwenden unsere Betriebssysteme Standardnetzwerkprotokolle für die authentifizierte, autorisierte und verschlüsselte Kommunikation – und sie ermöglichen Entwicklern den Zugriff auf diese Protokolle.

TLS-Sicherheit

iOS, iPadOS und macOS unterstützen Transport Layer Security (TLS 1.0, TLS 1.1, TLS 1.2, TLS 1.3) und Datagram Transport Layer Security (DTLS). Das Protokoll TLS unterstützt sowohl AES-128 als auch AES-256 und bevorzugt Cipher-Suites mit Forward Secrecy. „Safari“, „Kalender“, „Mail“ und andere Internetprogramme verwenden dieses Protokoll automatisch, um einen verschlüsselten Kommunikationskanal zwischen einem Gerät und den Netzwerkdiensten herzustellen. High-Level-APIs (z. B. CFNetwork) erleichtern es Entwicklern, TLS in ihren Apps umzusetzen. Im Gegensatz dazu bieten Low-Level-APIs (z. B. Network.framework) präzise Einstellungsmöglichkeiten. CFNetwork lässt SSL 3 nicht zu. Apps (z. B. Safari), die von WebKit Gebrauch machen, ist es verwehrt, eine SSL 3-Verbindung herzustellen.

Bei iOS 11 (und neueren Versionen) und macOS 10.13 (oder neuer) können SHA-1-Zertifikate nicht mehr für TLS-Verbindungen verwendet werden, solange sie der Benutzer nicht als vertrauenswürdig einstuft. Zertifikate mit RSA-Schlüsseln mit weniger als 2048 Bit sind ebenfalls nicht zulässig. Die RC4-Cipher-Suite für die symmetrische Verschlüsselung findet in iOS 10 und macOS 10.12 keine Anwendung. Standardmäßig sind RC4-Cipher-Suites auf TLS-Clients oder -Servern, die mit SecureTransport APIs implementiert wurden, nicht aktiviert und können keine Verbindung herstellen, wenn es sich bei RC4 um die einzige verfügbare Cipher-Suite handelt. Dienste oder Apps, die RC4 erfordern, sollten für die Verwendung von sicheren Cipher-Suites aktualisiert werden, um ein höheres Maß an Sicherheit zu erzielen. In iOS 12.1 müssen Zertifikate, die nach dem 15. Oktober 2018 von einem vom System als vertrauenswürdig eingestuften Root-Zertifikat ausgegeben wurden, in einem vertrauenswürdigen CT-Protokoll (Certificate Transparency) registriert sein, damit sie für TLS-Verbindungen zugelassen werden. In iOS 12.2 ist TLS 1.3 für Network.framework und für NSURLSession-APIs standardmäßig aktiviert. TLS-Clients, die SecureTransport APIs verwenden, können TLS 1.3 nicht verwenden.

Transportsicherheit für Apps

App Transport Security umfasst Anforderungen für Standardverbindungen, um bei Verwendung von NSURLConnection-, CFURL- oder NSURLSession-APIs sicherzustellen, dass von den Apps die besten Verfahren für sichere Verbindungen eingehalten werden. Standardmäßig beschränkt App Transport Security die Cipher-Auswahl auf solche Cipher-Suites, die Forward Secrecy bereitstellen, insbesondere:

- ECDHE_ECDSA_AES und ECDHE_RSA_AES im Galois/Counter Mode (GCM)
- Cipher Block Chaining (CBC)-Modus

Apps sind in der Lage, das Forward Secrecy-Erfordernis pro Domain zu deaktivieren. In diesem Fall wird RSA_AES zur Gruppe der verfügbaren Ciphers hinzugefügt.

Server müssen TLS 1.2 und Forward Secrecy unterstützen und Zertifikate müssen auf der Basis von SHA-256 (oder stärker) validiert und signiert sein, wobei ein 2048-Bit-RSA-Schlüssel oder ein 256-Bit-ECC-Schlüssel (elliptischer Kurven-Schlüssel) die Mindestanforderung darstellt.

Netzwerkverbindungen, die diese Anforderungen nicht erfüllen, kommen nicht zustande, es sei denn, die App setzt die in App Transport Security definierten Anforderungen für die Transportsicherheit außer Kraft. Ungültige Zertifikate führen in jedem Fall zu einem nicht behebbaren Fehler (Hard Failure) und zum Nichtzustandekommen der Verbindung. App Transport Security wird automatisch auf alle Apps angewendet, die für iOS 9 (oder neuer) und für macOS 10.11 (oder neuer) kompiliert werden.

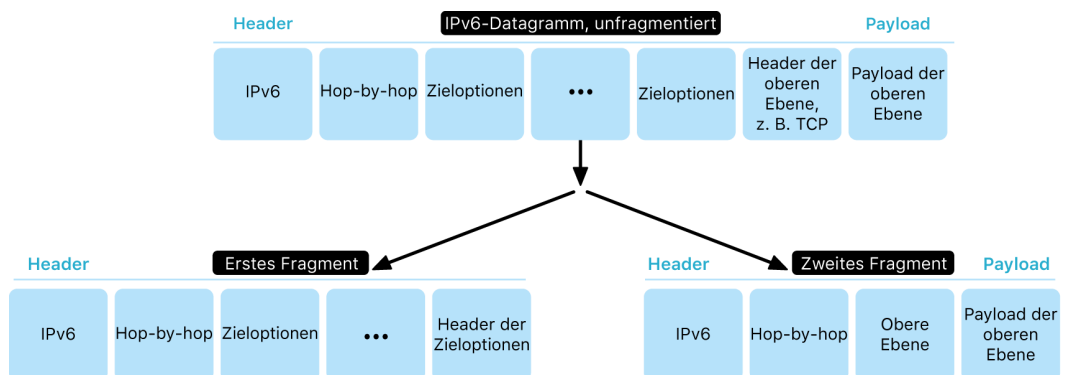
Prüfung der Zertifikatgültigkeit

Die Beurteilung der Vertrauenswürdigkeit eines TLS-Zertifikats erfolgt gemäß Industriestandards, die in RFC 5280 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc5280.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt) definiert sind, und unter Berücksichtigung neu aufkommender Standards wie beispielsweise RFC 6962 (Zertifikattransparenz) (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc6962.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt). Apple-Geräte mit iOS 11 (oder neuer) oder mit macOS 10.13 (oder neuer) erhalten periodisch eine laufend aktualisierte Liste widerrufener oder eingeschränkter Zertifikate. Diese Liste wird aus den Zertifikatperrlisten (Certificate Revocation Lists, CRLs) zusammengestellt, die von den Zertifizierungsstellen, die in die Apple-Produkte integriert sind und von Apple als vertrauenswürdig eingestuft werden, und/oder von ihnen nachgeordneten Stellen veröffentlicht werden. Apple behält sich das Recht vor, zusätzliche Einschränkungen in die Liste aufzunehmen. Diese Informationen werden immer dann zu Rate gezogen, wenn eine API-Netzwerkfunktion verwendet wird, um eine sichere Verbindung herzustellen. Wenn eine einzelne Zertifizierungsstelle mehr Zertifikate zurückgezogen hat, als einzeln aufgelistet werden können, kann eine Beurteilung der Vertrauenswürdigkeit zu dem Ergebnis führen, dass eine OCSP-Anfrage (Online Certificate Status Response) erforderlich ist. Wenn auf die Anfrage keine Antwort erfolgt, schlägt die Beurteilung fehl.

IPv6-Sicherheit

Alle Apple-Betriebssysteme unterstützen IPv6 und implementieren verschiedene Mechanismen zum Schutz der Privatsphäre der Benutzer und der Stabilität des Netzwerkstapels. Bei einer SLAAC-Konfiguration (Stateless Address Autoconfiguration) werden die IPv6-Adressen aller Schnittstellen so generiert, dass das Tracking der Geräte über Netzwerke verhindert wird, gleichzeitig aber ein gutes Benutzererlebnis gewährleistet ist, indem die Stabilität der Adressen sichergestellt wird, sofern kein Wechsel des Netzwerks erfolgt. Der Algorithmus für die Adressgenerierung basiert auf kryptografisch generierten Adressen gemäß RFC 3972 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc3972.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt); er wird durch einen schnittstellenspezifischen Modifikator erweitert, sodass gewährleistet ist, dass auch unterschiedliche Schnittstellen im gleichen Netzwerk unterschiedliche Adressen haben. Darüber hinaus werden temporäre Adressen mit einem bevorzugten Zeithorizont von 24 Stunden erstellt, die standardmäßig für neue Verbindungen verwendet werden. In Übereinstimmung mit der Funktion der privaten WLAN-Adresse, die mit iOS 14, iPadOS 14 und watchOS 7 eingeführt wurde, wird für jedes WLAN, dem sich ein Gerät anschließt, eine eindeutige Link-Local-Adresse generiert. Daran anschließend wird die SSID des Netzwerks als zusätzliches Element für die Adressgenerierung einbezogen – ähnlich dem Parameter „Network_ID“ gemäß RFC 7217 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc7217.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt). Dieser Ansatz wird bei iOS 14, iPadOS 14 und watchOS 7 verfolgt.

Zum Schutz vor Angriffen, die auf Headern von IPv6-Erweiterungen und Fragmentierung basieren, sind in Apple-Geräten die Schutzvorkehrungen implementiert, die in RFC 6980 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc6980.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt), RFC 7112 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc7112.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt) und RFC 8021 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc8021.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt) spezifiziert sind. Neben anderen Maßnahmen werden dadurch Angriffe verhindert, bei denen der Header der oberen Schicht nur im zweiten Fragment (siehe unten) zu finden ist, was zu Mehrdeutigkeiten bei Sicherheitsvorkehrungen wie Paketfilterung ohne Status führen kann.



Um die Zuverlässigkeit des IPv6-Stapels von Apple-Betriebssystemen zu gewährleisten, erzwingen Apple-Geräte darüber hinaus verschiedene Limits für IPv6-relevante Datenstrukturen, zum Beispiel für die Anzahl der Präfixe pro Schnittstelle.

VPN-Sicherheit (Virtual Private Network)

Bei iOS-, iPadOS- und macOS-Geräten erfordert die Einbindung sicherer Netzwerkdienste wie „Virtual Private Network“ nur minimalen Einrichtungs- und Konfigurationsaufwand.

Unterstützte Protokolle

Diese Geräte arbeiten mit VPN-Servern, die die folgenden Protokolle und Authentifizierungsmethoden unterstützen:

- IKEv2/IPsec mit Authentifizierung per Shared Secret, RSA-Zertifikate, Elliptic Curve Digital Signature Algorithm (ECDSA)-Zertifikate, EAP-MSCHAPv2 oder EAP-TLS
- SSL-VPN unter Verwendung der entsprechenden Client-App aus dem App Store
- L2TP/IPsec mit Benutzerauthentifizierung per MS-CHAPV2-Passwort und Systemauthentifizierung mittels Shared Secret (iOS, iPadOS und macOS) und RSA SecurID oder CRYPTOCARD (nur macOS)
- Cisco IPsec mit Benutzerauthentifizierung per Passwort, RSA SecurID oder CRYPTOCARD sowie Systemauthentifizierung mittels Shared Secret und Zertifikaten (nur macOS)

Unterstützte VPN-Implementierungen

iOS, iPadOS und macOS unterstützen Folgendes:

- *VPN On Demand*: für Netzwerke, die eine zertifikatbasierte Authentifizierung verwenden. IT-Richtlinien legen über ein VPN-Konfigurationsprofil fest, welche Domains eine VPN-Verbindung benötigen.
- *Per App VPN*: für VPN-Verbindungen, die noch detaillierter und präziser eingestellt werden können. Lösungen für die Mobilgeräteverwaltung ([Mobile Device Management, MDM](#)) können eine Verbindung für jede verwaltete App und spezielle Domains in Safari vorgeben. So lässt sich sicherstellen, dass nur sichere Daten und keine privaten Daten des Benutzers in das Unternehmensnetzwerk hinein- und daraus hinausgelangen.

iOS und iPadOS unterstützen Folgendes:

- *Always On VPN*: Für Geräte, die über eine MDM-Lösung verwaltet und mithilfe von Apple Configurator für den Mac, [Apple School Manager](#) oder [Apple Business Manager](#) betreut werden. Mit „Always-on VPN“ entfällt für Benutzer die Notwendigkeit, beim Verbindungsaufbau zu einem Mobilfunknetz oder WLAN das VPN aus Gründen der Sicherheit explizit zu aktivieren. Darüber hinaus hat die Organisation die uneingeschränkte Kontrolle über den Datenverkehr von und zu Geräten, da der gesamte IP-Datenverkehr zurück zur Organisation getunnelt wird. IKEv2, das Standardprotokoll für den Austausch von Parametern und Schlüsseln für die anschließende Verschlüsselung, sichert den Datenverkehr durch Datenverschlüsselung ab. Die Organisation kann den Datenverkehr von und zu ihren Geräten überwachen und filtern, Daten im Netzwerk sichern und den Internetzugriff der Geräte einschränken.

WLAN-Sicherheit

Sicherer Zugriff auf Funknetzwerke

Alle Apple-Plattformen unterstützen die Industriestandardprotokolle für die Authentifizierung und Verschlüsselung bei WLAN-Verbindungen, damit bei Verbindungen mit folgenden Funknetzwerken der authentifizierte Zugriff und die Vertraulichkeit gewahrt bleiben:

- WPA2 Personal
- WPA2 Enterprise
- WPA2/WPA3 Transitional
- WPA3 Personal
- WPA3 Enterprise
- WPA3 Enterprise 192-Bit-Sicherheit

WPA2 und WPA3 authentifizieren jede Verbindung und stellen die 128-Bit-AES-Verschlüsselung bereit, damit beim drahtlosen Senden die Vertraulichkeit der Daten gewahrt bleibt. Benutzer haben so die größtmögliche Gewissheit, dass ihre Daten geschützt bleiben, wenn sie Informationen über eine WLAN-Verbindung senden und empfangen.

WPA3-Unterstützung

WPA3 wird von den folgenden Apple-Geräten unterstützt:

- iPhone 7 (oder neuer)
- iPad 5. Generation (oder neuer)
- Apple TV 4K (oder neuer)
- Apple Watch Series 3 (oder neuer)
- Mac-Computer (ab Spätherbst 2013 mit 802.11ac (oder neuer))

Neuere Geräte unterstützen bei Verbindungen zu kompatiblen Zugangspunkten die Authentifizierung per WPA3 Enterprise 192-Bit-Sicherheit inklusive der Unterstützung für die 256-Bit-AES-Verschlüsselung. Dadurch wird der Schutz der Vertraulichkeit von Daten beim Senden über Mobilfunk zusätzlich verstärkt. WPA3 Enterprise 192-Bit-Sicherheit wird von den Geräten iPhone 11, iPhone 11 Pro, iPhone 11 Pro Max und neueren iOS- und iPadOS-Geräten unterstützt.

PMF-Unterstützung

Zusätzlich zum Schutz von Daten beim drahtlosen Senden erstreckt sich bei Apple-Plattformen der WPA2- und WPA3-Schutz basierend auf dem PMF-Dienst (Protected Management Frame) gemäß 802.11w auch auf Unicast- und Multicast-Management-Frames. Die PMF-Unterstützung ist auf den folgenden Apple-Geräten verfügbar:

- iPhone 6 (oder neuer)
- iPad Air 2 (oder neuer)
- Apple TV HD (oder neuer)

- Apple Watch Series 3 (oder neuer)
- Mac-Computer (ab Spätherbst 2013 mit 802.11ac (oder neuer))

Dank der Unterstützung für 802.1X können Apple-Geräte in eine Vielzahl von RADIUS-Authentifizierungsumgebungen integriert werden. Für die Mobilfunkauthentifizierung werden folgende 802.1X-Methoden unterstützt: EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAPv0 und PEAPv1.

Plattformschutz

Apple-Betriebssysteme schützen die Geräte gegenüber Schwachstellen in der Firmware von Netzwerkprozessoren. Das bedeutet, dass Netzwerkcontroller mit WLAN beschränkten Zugriff auf den Speicher der Anwendungsprozessoren haben.

- Wenn USB oder SDIO (Secure Digital Input Output) als Schnittstelle zu einem Netzwerkprozessor verwendet wird, kann der Netzwerkprozessor keine DMA-Transaktionen (Direct Memory Access) mit dem Anwendungsprozessor starten.
- Wenn PCIe verwendet wird, befindet sich jeder Netzwerkprozessor auf einem eigenen isolierten PCIe-Bus. Der DMA-Zugriff eines Netzwerkprozessors wird mittels einer [Input/Output Memory Management Unit \(IOMMU\)](#) auf jedem PCIe-Bus auf die Speicherbereiche und Ressourcen eingeschränkt, die dessen Netzwerkpakete und Kontrollstrukturen enthalten.

Veraltete Protokolle

Apple-Produkte unterstützen die folgenden veralteten WLAN-Protokolle für die Authentifizierung und Verschlüsselung:

- WEP Open mit 40-Bit- und mit 104-Bit-Schlüsseln
- WEP Shared mit 40-Bit- und mit 104-Bit-Schlüsseln
- Dynamic WEP
- Temporal Key Integrity Protocol (TKIP)
- WPA
- WPA/WPA2 Transitional

Diese Protokolle gelten nicht mehr als sicher, weshalb aus Gründen der Kompatibilität, der Zuverlässigkeit, der Leistung und der Sicherheit davon abgeraten wird, sie weiter zu verwenden. Sie werden allein aus Gründen der Abwärtskompatibilität unterstützt und werden möglicherweise in künftigen Softwareversionen ganz entfernt.

Es empfiehlt sich, alle WLAN-Implementierungen auf WPA3 Personal oder WPA3 Enterprise umzustellen, um die größtmögliche Robustheit, Sicherheit und Kompatibilität von WLAN-Verbindungen zu gewährleisten.

Datenschutz bei WLAN-Verbindungen

MAC-Adressrandomisierung

Apple-Plattformen verwenden eine nach dem Zufallsprinzip bestimmte MAC-Adresse (Media Access Control) für die Suche nach WLAN-Netzwerken, wenn sie keinem bestimmten WLAN-Netzwerk zugewiesen sind. Solche Suchen oder Scans können ausgeführt werden, um ein bekanntes WLAN-Netzwerk aufzufinden und eine Verbindung zu ihm herzustellen oder um Ortungsdienste für Apps zu unterstützen, die Geofencing nutzen – zum Beispiel für ortsbasierte Erinnerungen oder zum Fixieren eines Standorts in der Apple-App „Karten“. Zu beachten ist, dass WLAN-Scans, die beim Versuch erfolgen, die Verbindung zu einem bevorzugten WLAN herzustellen, nicht randomisiert sind. Die MAC-Adressrandomisierung für WLAN-Verbindungen steht auf iPhone 5 (oder neuer) zur Verfügung.

Apple verwendet eine randomisierte MAC-Adresse auch für ePNO-Scans (enhanced Preferred Network Offload), wenn ein Gerät nicht einem bestimmten WLAN zugewiesen ist oder sich der Prozessor im Ruhezustand befindet. ePNO-Scans werden durchgeführt, wenn ein Gerät die Ortungsdienste für Apps verwendet, die Geofencing nutzen – zum Beispiel für ortsbasierte Erinnerungen, die feststellen müssen, ob ein Gerät an einem bestimmten Ort ist.

Da sich die MAC-Adresse eines Geräts ändert, wenn die Verbindung zu einem WLAN getrennt wird, kann sie nicht dafür genutzt werden, per passiver Beobachtung des WLAN-Datenverkehrs ein Bewegungsprofil für ein Gerät zu erstellen, selbst wenn dieses mit dem Mobilfunknetz verbunden ist. Apple hat WLAN-Hersteller darüber informiert, dass in iOS oder iPadOS durchgeführte WLAN-Scans eine randomisierte MAC-Adresse verwenden und dass weder Apple noch der jeweilige Hersteller diese randomisierten MAC-Adressen vorhersehen kann.

Wenn in iOS 14 (oder neuer), iPad OS 14 (oder neuer) und watchOS 7 (oder neuer) iPhone-, iPad-, iPod touch- oder Apple Watch-Geräte eine Verbindung zu einem WLAN herstellen, identifizieren sie sich mit individuellen (randomisierten) MAC-Adresse pro Netzwerk. Diese Funktion kann entweder vom Benutzer oder mittels einer neuen Option in der WLAN-Payload deaktiviert werden. Unter bestimmten Umständen greift das Gerät auf seine tatsächliche MAC-Adresse zurück.

Weitere Informationen enthält der Apple Support-Artikel [Private WLAN-Adressen auf dem iPhone, iPad, iPod touch und der Apple Watch verwenden](#).

Randomisierung von WLAN-Frame-Sequenzen

WLAN-Frames haben eine fortlaufende Nummer, die vom Low-Level-Protokoll 802.11 verwendet wird, um für eine effiziente und verlässliche WLAN-Kommunikation zu sorgen. Da diese Folgenummer sich bei jedem weiteren übertragenen Frame erhöht, könnte sie genutzt werden, um Informationen, die im Zuge eines WLAN-Scans übermittelt werden, mit anderen Frame in Beziehung zu setzen, die von demselben Gerät übermittelt werden.

Um derartigen Versuchen vorzubeugen, randomisieren Apple-Geräte die Folgenummern, wann immer eine MAC-Adresse gegen eine neue randomisierte Adresse ausgetauscht wird. Dies betrifft auch die Randomisierung der Folgenummern für jede neue Scananfrage, die initiiert wird, solange das Gerät keinem WLAN zugewiesen ist. Diese Randomisierung wird von den folgenden Geräten unterstützt:

- iPhone 7 (oder neuer)
- iPad 5. Generation (oder neuer)
- Apple TV 4K (oder neuer)
- Apple Watch Series 3 (oder neuer)
- iMac Pro (Retina 5K, 27-Zoll, 2017) oder neuer
- MacBook Pro (13-Zoll, 2018) oder neuer
- MacBook Pro (15-Zoll, 2018) oder neuer
- MacBook Air (Retina, 13-Zoll, 2018) oder neuer
- Mac mini (2018) oder neuer
- iMac (Retina 4K, 21,5-Zoll, 2019) oder neuer
- iMac (Retina 5K, 27-Zoll, 2019) oder neuer
- Mac Pro (2019) oder neuer

WLAN-Verbindungen

Apple generiert randomisierte MAC-Adressen für WLAN-Verbindungen auf Peer-to-Peer-Basis, die für AirDrop und AirPlay genutzt werden. Randomisierte Adressen werden auch für den persönlichen Hotspot unter iOS und iPadOS (bei eingesetzter SIM-Karte) und für die Freigabe im Internet unter macOS verwendet.

Bei jedem Starten dieser Netzwerkschnittstellen werden neue zufällige Adressen generiert und für jede Schnittstelle wird je nach Bedarf und unabhängig voneinander eine eindeutige Adresse generiert.

Versteckte Netzwerke

WLAN-Netzwerke werden anhand ihres Netzwerknamens identifiziert, der unter der Bezeichnung *SSID (Service Set Identifier)* bekannt ist. Einige WLAN-Netzwerke sind so konfiguriert, dass ihre SSID versteckt bleibt. Dies führt dazu, dass der Zugangspunkt den Namen des Netzwerks nicht mitteilt. Solche Netzwerke werden *versteckte Netzwerke* genannt. iPhone 6s und neuere Geräte erkennen automatisch, wenn ein Netzwerk „versteckt“ ist. Wenn ein Netzwerk versteckt ist, sendet das iOS- oder iPadOS-Gerät eine Anfrage, in die die SSID eingebettet ist – ansonsten geschieht dies nicht. Dadurch wird vermieden, dass das Gerät die Namen von zuvor versteckten Netzwerken mitteilt, mit denen ein Benutzer verbunden war, was wiederum den Datenschutz stärkt.

Bluetooth-Sicherheit

Bei Apple-Geräten werden zwei Bluetooth-Arten unterschieden: Bluetooth Classic und Bluetooth Low Energy (BLE). Das Bluetooth-Sicherheitsmodell umfasst bei beiden Versionen die folgenden distinkten Sicherheitsfunktionen:

- *Pairing*: Dies ist der Prozess, mit dem ein oder mehrere Shared-Secret-Schlüssel ausgehandelt werden.
- *Bonding*: Dies ist der Vorgang, bei dem die beim Pairing ausgehandelten Schlüssel so gespeichert werden, dass sie für nachfolgende Verbindungen verwendet werden können, um ein vertrauenswürdiges Gerätepaar zu bilden.
- *Authentifizierung*: Dies ist der Vorgang, mit dem sichergestellt wird, dass die beiden Geräte dieselben Schlüssel haben.
- *Verschlüsselung*: Dadurch wird die Vertraulichkeit der Nachrichten gewahrt.
- *Nachrichtenintegrität*: Dies dient als Schutz gegen die Verfälschung von Nachrichten.
- *Secure Simple Pairing*: Dies ist eine Schutzvorkehrung gegen passives Abhören und gegen MITM-Angriffe (Man-in-the-Middle- oder auch Janusangriffe).

Bei Bluetooth Version 4.1 wurde der physische BR/EDR-Transport für Bluetooth Classic um die Funktion für sichere Verbindungen erweitert.

Die Sicherheitsfunktionen für die verschiedenen Bluetooth-Typen sind im Folgenden zusammengestellt.

Unterstützung	Bluetooth Classic	Bluetooth Low Energy
Pairing	Elliptische P-256-Kurve	FIPS-konforme Algorithmen (AES-CMAC und elliptische P-256-Kurve)
Bonding	Die beim Pairing ausgehandelten Informationen werden in einem sicheren Speicherbereich auf dem iOS-, iPadOS-, macOS-, tvOS- oder watchOS-Gerät gespeichert.	Die beim Pairing ausgehandelten Informationen werden in einem sicheren Speicherbereich auf dem iOS-, iPadOS-, macOS-, tvOS- oder watchOS-Gerät gespeichert.
Authentifizierung	FIPS-konforme Algorithmen (HMAC-SHA-256 und AES-CTR)	FIPS-konforme Algorithmen
Verschlüsselung	AES-CCM-Kryptografie im Controller	AES-CCM-Kryptografie im Controller
Nachrichtenintegrität	AES-CCM für die Integrität von Nachrichten	AES-CCM für die Integrität von Nachrichten
Secure Simple Pairing: Schutz gegen passives Abhören	Elliptic Curve Diffie-Hellman Exchange Ephemeral (ECDHE)	Elliptic Curve Diffie-Hellman Exchange (ECDHE)
Secure Simple Pairing: Schutz gegen MITM-Angriffe	Es gibt zwei numerische Methoden mit Assistenz durch den Benutzer: numerischer Vergleich und Passkey-Eingabe.	Es gibt zwei numerische Methoden mit Assistenz durch den Benutzer: numerischer Vergleich und Passkey-Eingabe. Pairing-Vorgänge erfordern die Antwort durch den Benutzer. Dies schließt alle Nicht-MITM-Pairing-Modi ein.

Unterstützung	Bluetooth Classic	Bluetooth Low Energy
Bluetooth 4.1 (oder neuer)	iMac Ende 2015 (oder neuer) MacBook Pro ab Anfang 2015 (oder neuer)	iOS 9 (oder neuer) iPadOS 13.1 (oder neuer) macOS 10.12 (oder neuer) tvOS 9 (oder neuer) watchOS 2.0 (oder neuer)
Bluetooth 4.2 (oder neuer)	iPhone 6 (oder neuer)	iOS 9 (oder neuer) iPadOS 13.1 (oder neuer) macOS 10.12 (oder neuer) tvOS 9 (oder neuer) watchOS 2.0 (oder neuer)

Datenschutz bei Bluetooth Low Energy

Zum Schutz der Privatsphäre und zur Wahrung des Datenschutzes verfügt BLE über die beiden folgenden Funktionen: Adressrandomisierung und Cross-Transport-Schlüsselableitung.

Die *Adressrandomisierung* schränkt die Möglichkeit ein, ein BLE-Gerät über längere Zeit zu verfolgen, da die Bluetooth-Geräteadressen häufig geändert werden. Damit ein Gerät die Datenschutzfunktion verwenden kann, um die Verbindung zu bekannten Geräten ein weiteres Mal herzustellen, muss die Geräteadresse – sie wird als *private Adresse* bezeichnet – vom jeweils anderen Gerät aufgelöst werden können. Die private Adresse wird unter Verwendung des IRK-Schlüssels (Identity Resolving Key) des Geräts generiert, der beim Koppeln ausgetauscht wird.

iOS 13 (oder neuer) und iPadOS 13.1 (oder neuer) haben die Möglichkeit, Verknüpfungsschlüssel während des Transports abzuleiten. Diese Funktion wird als *Cross-Transport-Schlüsselableitung* bezeichnet. Beispielsweise kann ein per BLE generierter Verknüpfungsschlüssel dazu verwendet werden, einen Verknüpfungsschlüssel von Bluetooth Classic abzuleiten. Darüber hinaus hat Apple den BLE-Support für Geräte, die die mit Bluetooth CoreSpec Version 4.1 eingeführte Funktion für sichere Verbindungen unterstützen, um Bluetooth Classic (siehe [Bluetooth Core Spezifikation 5.1](#)) erweitert.

Sicherheit der Ultrabreitbandtechnologie von iOS

Der neue von Apple entwickelte U1-Chip nutzt die Ultrabreitband-Technologie für die räumliche Orientierung, sodass iPhone 11, iPhone 11 Pro und iPhone 11 Pro Max oder neuere iPhone-Geräte die Standorte anderer Apple-Geräte mit U1-Chip präzise bestimmen können. Die Ultrabreitband-Technologie nutzt zum Randomisieren von Daten dieselbe Technologie, die auch in anderen unterstützten Apple-Geräten zu finden ist:

- MAC-Adressrandomisierung
- Randomisierung von WLAN-Frame-Sequenzen

Sicherheit der Gesamtauthentifizierung (Single Sign-On)

Gesamtauthentifizierung (Single Sign-On)

Sicherheit der Gesamtauthentifizierung (Single Sign-On)

iOS und iPadOS unterstützen die Gesamtauthentifizierung (Single Sign-On, SSO) für Unternehmensnetzwerke. SSO kann bei auf Kerberos basierenden Netzwerken verwendet werden, um Benutzer für Dienste, auf die sie zugreifen dürfen, zu authentifizieren. SSO kann für eine Reihe verschiedener Netzwerkaktivitäten verwendet werden, z. B. in sicheren Safari-Sitzungen oder in Apps anderer Anbieter. Die zertifikatbasierte Authentifizierung wie mit PKINIT wird ebenfalls unterstützt.

macOS verwendet Kerberos für die Authentifizierung gegenüber Unternehmensnetzwerken. Apps können Kerberos verwenden, um Benutzer gegenüber Diensten zu authentifizieren, für die sie zugriffsberechtigt sind. Kerberos kann darüber hinaus für eine breite Palette von Netzwerkaktivitäten verwendet werden – von sicheren Sitzungen in Safari über die Authentifizierung bei Netzwerkdateisystemen bis hin zu Apps anderer Anbieter. Die zertifikatbasierte Authentifizierung wird unterstützt, wobei aber eine App-Anpassung an eine Entwickler-API erforderlich ist.

iOS, iPadOS und macOS nutzen SPNEGO-Token und das Protokoll HTTP Negotiate für die Gesamtauthentifizierung gegenüber Gateways für die Kerberos-basierte Authentifizierung und gegenüber in Windows integrierten Authentifizierungssystemen, die Kerberos-Tickets unterstützen. Die SSO-Unterstützung basiert auf dem Open-Source-Projekt Heimdal.

In iOS, iPadOS und macOS werden die folgenden Verschlüsselungstypen unterstützt:

- AES-128-CTS-HMAC-SHA-1-96
- AES-256-CTS-HMAC-SHA-1-96
- DES3-CBC-SHA-1
- ARCFOUR-HMAC-MD5

Safari unterstützt die Gesamtauthentifizierung. Apps anderer Anbieter, die die standardmäßigen Netzwerk-APIs von iOS und iPadOS verwenden, können ebenfalls für SSO konfiguriert werden. Für die Konfiguration von SSO unterstützen iOS und iPadOS eine neue Payload für Konfigurationsprofile, die es Lösungen für die [Mobilgeräteverwaltung \(Mobile Device Management, MDM\)](#) ermöglicht, die notwendigen Einstellungen per Push auf ein Gerät zu transferieren. Dies schließt die Festlegung des Prinzipalnamens eines Benutzers (d. h. dessen Active-Directory-Benutzeraccount) und der Realm-Einstellungen für Kerberos sowie das Konfigurieren der Apps und URLs in Safari ein, für die SSO zulässig sein soll.

Die Konfiguration von Kerberos in macOS kann durch das Beziehen von Tickets mithilfe des Ticket-Viewers in Kerberos, durch die Anmeldung an einer AD-Domain (Active Directory) von Windows oder mithilfe des Befehlszeilenprogramms `kinit` erfolgen.

Erweiterungen für Gesamtauthentifizierung

App-Entwickler können mithilfe von SSO-Erweiterungen eigene Implementierungen der Gesamtauthentifizierung realisieren. SSO-Erweiterungen werden aufgerufen, wenn eine native App oder eine Web-App für die Authentifizierung eines Benutzers die Verwendung eines Identitätsanbieters (IdP, Identity Provider) benötigt. Entwickler können hierfür zwei Arten von Erweiterungen bereitstellen: Erweiterungen, die die Umleitung an HTTPS bewirken, und Erweiterungen, die einen Challenge/Response-Mechanismus wie Kerberos verwenden. Auf diese Weise können OpenID-, OAuth-, SAML2- und Kerberos-Authentifizierungsschemata als Erweiterungen für die Gesamtauthentifizierung verwendet werden.

Um eine SSO-Erweiterung zu verwenden, kann eine App entweder die API „AuthenticationServices“ nutzen oder auf den URL-Abfangmechanismus des Betriebssystems zurückgreifen. WebKit und CFNetwork umfassen eine Ebene für diese Abfangvorgänge, die die nahtlose Unterstützung der Gesamtauthentifizierung für jede native App und jede WebKit-App ermöglicht. Damit eine SSO-Erweiterung aufgerufen wird, muss mithilfe eines MDM-Profiles eine von einem Administrator definierte Konfiguration installiert werden. Außerdem müssen Erweiterungen vom Typ „Redirect“ die Payload „Associated Domains“ (Zugehörige Domains) verwenden, um den Nachweis zu erbringen, dass dem von ihnen unterstützten Identitätsserver ihre Existenz bekannt ist.

Die einzige Erweiterung, die vom Betriebssystem selbst bereitgestellt wird, ist die SSO-Erweiterung für Kerberos.

AirDrop-Sicherheit

Apple-Geräte, die AirDrop unterstützen, verwenden BLE (Bluetooth Low Energy) sowie von Apple entwickelte, Peer-To-Peer-basierte WLAN-Technologien, um Dateien und Informationen an Geräte in der Nähe zu senden, zum Beispiel an iOS- und iPad-Geräte mit iOS 7 (und neueren Versionen) und an Mac-Computer mit OS X 10.11 (und neueren Versionen). Die Geräte kommunizieren direkt per WLAN-Funk miteinander, ohne dass eine Internetverbindung oder ein Zugangspunkt (Access Point, AP) benötigt wird. Diese Verbindung ist mithilfe von TLS verschlüsselt.

AirDrop verwendet standardmäßig die Freigabeeinstellung „Nur Kontakte“. Benutzer können in AirDrop aber auch die Freigabe für „Alle“ auswählen oder die Funktion komplett deaktivieren. Für Geräte und Apps, die mit einer Lösung für die [Mobilgeräteverwaltung \(Mobile Device Management, MDM\)](#) verwaltet werden, kann die Verwendung von AirDrop eingeschränkt werden.

AirDrop-Funktionsweise

AirDrop nutzt iCloud-Dienste, um Benutzer bei der Authentifizierung zu unterstützen. Wenn sich ein Benutzer bei iCloud anmeldet, wird eine 2048-Bit-RSA-Identität auf seinem Gerät gespeichert. Wenn ein Benutzer AirDrop aktiviert, wird – als Hash-Wert – eine AirDrop-Kurzidentität erstellt, die auf den E-Mail-Adressen und den Telefonnummern basiert, die für die Apple-ID des Benutzers hinterlegt sind.

Wenn ein Benutzer AirDrop als Methode für die Freigabe (das Teilen) eines Objekts wählt, übermittelt das sendende Gerät per BLE ein AirDrop-Signal, das den Hash-Wert der AirDrop-Kurzidentität des Benutzers enthält. Andere Geräte in der unmittelbaren Umgebung, die sich nicht im Ruhezustand befinden und auf denen AirDrop aktiviert ist, erkennen dieses Signal und antworten per Peer-To-Peer-WLAN, sodass das sendende Gerät die Identität eines antwortenden Geräts erkennen kann.

Im Modus „Nur Kontakte“ wird der empfangene Hash-Wert der AirDrop-Kurzidentität mit den Hash-Werten der Personen abgeglichen, die in der App „Kontakte“ auf dem empfangenden Gerät enthalten sind. Bei einer Übereinstimmung antwortet das empfangende Gerät per Peer-To-Peer-WLAN mit seinen Identitätsinformationen. Kommt es zu keiner Übereinstimmung, antwortet das Gerät nicht.

Im Modus „Alle“ kommt derselbe Gesamtprozess zum Einsatz. Allerdings antwortet bei diesem Modus das empfangende Gerät auch, wenn in der App „Kontakte“ des Geräts keine Übereinstimmung gefunden wird.

Das sendende Gerät initiiert daraufhin eine AirDrop-Verbindung per Peer-To-Peer-WLAN und verwendet diese Verbindung, um den Hash-Wert der Identität in der Langversion an das empfangende Gerät zu senden. Wenn diese Langversion zu einer Übereinstimmung mit einer bekannten Person in der App „Kontakte“ auf dem empfangenden Gerät führt, antwortet das empfangende Gerät mit der Langversion des Hash-Werts der eigenen Identität.

Nachdem die Hash-Werte verifiziert wurden, werden der Vorname und (sofern in der App „Kontakte“ enthalten) das Foto des Empfängers auf dem sendenden Gerät auf der Freigabeseite (Share Sheet) für AirDrop angezeigt. In iOS und iPadOS erscheinen sie im Abschnitt „Personen“ oder „Geräte“. Für Geräte, die nicht verifiziert und authentifiziert wurden, werden auf dem sendenden Gerät auf der Freigabeseite (Share Sheet) für AirDrop ein Symbol in Form einer Silhouette und der Gerätename angezeigt, der im Bereich „Einstellungen“ > „Allgemein“ > „Info“ > „Name“ eingestellt ist. In iOS und iPadOS erscheinen sie auf der Freigabeseite (Share Sheet) für AirDrop im Abschnitt „Andere Personen“.

Der Benutzer des sendenden Geräts kann daraufhin angeben, mit welcher Person er seine Inhalte teilen möchte. Bei der Benutzerauswahl initiiert das sendende Gerät zum Austausch der iCloud-Identitätszertifikate eine (per TLS) verschlüsselte Verbindung zum empfangenden Gerät. Die Identität in den Zertifikaten wird mit den Angaben in der App „Kontakte“ der beteiligten Benutzer abgeglichen.

Wenn die Zertifikate verifiziert werden, wird der Benutzer des empfangenden Geräts aufgefordert, die eingehende Dateiübertragung vom identifizierten Benutzer bzw. Gerät zu akzeptieren. Wenn mehrere Empfänger ausgewählt wurden, wird dieses Verfahren für jede Zielperson wiederholt.

Sicherheit bei WLAN-Passwortfreigabe auf iPhone und iPad

iOS- und iPadOS-Geräte, die die WLAN-Passwortfreigabe unterstützen, verwenden ein Verfahren, das AirDrop ähnelt, um ein WLAN-Passwort von einem Gerät an ein anderes Gerät zu senden.

Wenn ein Benutzer ein WLAN-Netzwerk auswählt (der Antragsteller) und nach dem WLAN-Passwort gefragt wird, startet das Apple-Gerät die BLE-Ankündigung (Bluetooth Low Energy), dass es das WLAN-Passwort benötigt. Andere Geräte in der unmittelbaren Umgebung, die sich nicht im Ruhezustand befinden und die über das Passwort für das ausgewählte WLAN-Netzwerk verfügen, stellen eine BLE-Verbindung zum anfordernden Gerät her.

Das Gerät, das über das WLAN-Passwort verfügt (der Aussteller), fordert die Kontaktinformationen des Antragstellers an. Der Antragsteller muss daraufhin seine Identität nachweisen, wofür wiederum ein AirDrop-ähnliches Verfahren verwendet wird. Nachdem die Identität nachgewiesen wurde, sendet der Aussteller dem Antragsteller den Code für den Beitritt zum Netzwerk.

Für Geräte und Apps, die mit einer Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM) verwaltet werden, kann die Verwendung der Funktion für die WLAN-Passwortfreigabe eingeschränkt werden.

Firewall-Sicherheit bei macOS

macOS verfügt über eine integrierte Firewall zum Schutz des Mac-Computers vor Zugriffen aus dem Netzwerk und vor DoS-Angriffen (Denial-of-Service). Die Firewall kann im Bereich „Privatsphäre & Sicherheit“ der Systemeinstellungen konfiguriert werden. Unterstützt werden die folgenden Konfigurationen:

- Alle eingehenden Verbindungen blockieren (unabhängig von der jeweiligen App).
- Integrierter Software automatisch erlauben, eingehende Verbindungen zu empfangen.
- Geladener und signierter Software automatisch erlauben, eingehende Verbindungen zu empfangen.
- Zugriff für einzelne vom Benutzer ausgewählte Apps erlauben oder verweigern.
- Verhindern, dass der Mac-Computer auf ICMP (Internet Control Message Protocol)- Probing- und Portscan-Anfragen reagiert.

Sicherheit bei Developer Kits

Sicherheit bei Developer Kits – Übersicht

Apple bietet eine Reihe von „Kit“-Frameworks, die es Entwicklern anderer Anbieter ermöglichen, Apple-Dienste zu erweitern. Das Hauptaugenmerk liegt bei diesen Frameworks auf Datenschutz und Sicherheit der Benutzer:

- HomeKit
- CloudKit
- SiriKit
- DriverKit
- ReplayKit
- ARKit

Sicherheit beim HomeKit

Sicherheit bei HomeKit-Kommunikation

HomeKit bietet eine Infrastruktur zur Automatisierung häuslicher Umgebungen, die Sicherheitsfunktionen von iCloud und von iOS, iPadOS und macOS nutzt, um private Daten zu schützen und zu synchronisieren, ohne dass Apple darauf zugreifen kann.

Identität und Sicherheit basieren bei HomeKit auf einem Ed25519-Schlüsselpaar mit einem öffentlichen und einem privaten Schlüssel. Auf iOS-, iPadOS- und macOS-Geräten wird für jeden HomeKit-Benutzer ein Ed25519-Schlüsselpaar erzeugt, das die HomeKit-Identität des jeweiligen Benutzers darstellt. Die Identität wird für die Kommunikation zwischen iOS-, iPadOS- und macOS-Geräten untereinander und zwischen iOS-, iPadOS- und macOS-Geräten und Zubehörgeräten verwendet.

Die Schlüssel – sie werden im Schlüsselbund gespeichert und nur in verschlüsselte Backups des Schlüsselbunds einbezogen – werden (sofern zugänglich) unter Verwendung des iCloud-Schlüsselbunds auf dem aktuellen Stand gehalten. HomePod und Apple TV empfangen Schlüssel per „Antippen für die Konfiguration“ oder über den unten beschriebenen Konfigurationsmodus. Schlüssel werden von einem iPhone via Apple IDS (Identity Service) mit einer gekoppelten Apple Watch geteilt.

Kommunikation zwischen HomeKit-Zubehörgeräten

HomeKit-Geräte erstellen ihr eigenes Ed25519-Schlüsselpaar für die Kommunikation mit iOS-, iPadOS- und macOS-Geräten. Beim Zurücksetzen eines Geräts auf die Werkseinstellungen wird ein neues Schlüsselpaar generiert.

Um eine Verbindung zwischen einem iOS-, iPadOS- oder macOS-Gerät und einem HomeKit-Gerät herzustellen, werden die Schlüssel über das 3072-Bit-Protokoll „Secure Remote Password“ ausgetauscht, wofür ein achtstelliger Code des Geräteherstellers verwendet wird, der vom Benutzer auf dem iOS- oder iPadOS-Gerät eingegeben und per ChaCha20-Poly1305 AEAD mit per HKDF-SHA-512 hergeleiteten Schlüsseln verschlüsselt wird. Die MFi-Zertifizierung des Geräts wird bei der Konfiguration ebenfalls überprüft. Geräte ohne einen MFi-Chip können die Unterstützung für die Softwareautorisierung unter iOS 11.3 (und neueren Versionen) integrieren.

Wenn das iOS-, iPadOS- oder macOS-Gerät und das HomeKit-Gerät bei laufendem Betrieb kommunizieren, authentifizieren sie sich gegenseitig über die zuvor ausgetauschten Schlüssel. Jede Sitzung wird über ein Station-to-Station-Protokoll hergestellt und mit von HKDF-SHA-512 abgeleiteten Schlüsseln verschlüsselt, die auf für diese Sitzung erzeugten Curve25519-Schlüsseln basieren. Dies gilt sowohl für IP-basierte Geräte als auch für BLE-Geräte (Bluetooth Low Energy).

BLE-Geräte, die Broadcast-Mitteilungen unterstützen, werden mit einem Broadcast-Verschlüsselungsschlüssel ausgestattet und durch ein gekoppeltes iOS-, iPadOS- oder macOS-Gerät über eine sichere Sitzung bereitgestellt. Dieser Schlüssel wird verwendet, um die Daten über Statusänderungen auf den Geräten zu verschlüsseln, die mittels BLE-Ankündigungen benachrichtigt werden. Der Broadcast-Verschlüsselungsschlüssel ist ein per HKDF-SHA-512 hergeleiteter Schlüssel; die Daten werden mithilfe des ChaCha20-Poly1305 AEAD-Algorithmus verschlüsselt. Der Broadcast-Verschlüsselungsschlüssel wird regelmäßig vom iOS-, iPadOS- oder macOS-Gerät geändert und via iCloud auf anderen Geräten aktualisiert (siehe Beschreibung unter [HomeKit-Datensicherheit](#)).

HomeKit und Siri

Siri kann verwendet werden, um Geräte abzufragen und zu steuern und Szenen zu aktivieren. Minimale Informationen zur Konfiguration des Zuhauses werden anonym an Siri gesendet, um Namen von Zimmern, Geräten oder Szenen bereitzustellen, die für die Erkennung von Befehlen erforderlich sind. Audiodaten, die an Siri gesendet werden, können bestimmte Geräte oder Befehle enthalten. Die Siri-Daten werden aber nicht mit anderen Apple-Funktionen (z. B. HomeKit) assoziiert.

Siri-fähige HomeKit-Geräte

Mit der Home-App können Benutzer auf Siri-fähigen Geräten neue Funktionen wie Siri und andere HomePod-Funktionen wie Timer, Wecker, Intercom und Türklingel aktivieren. Wenn diese Funktionen aktiviert sind, kommuniziert das Gerät mit dem gekoppelten HomePod, der diese Apple-Funktionen bereitstellt, im lokalen Netzwerk. Audiodaten werden zwischen den Geräten über verschlüsselte Kanäle ausgetauscht, die sowohl HomeKit- als auch AirPlay-Protokolle nutzen.

Bei aktivierter Funktion „Auf ‚Hey Siri‘ achten“ wird mit einer lokal auf dem Gerät ausgeführten Trigger-Engine überprüft, ob der Befehl gesprochen wird. Erkennt die Engine den Befehl, sendet sie das Audiomaterial mit HomeKit direkt an den gekoppelten HomePod. Der HomePod führt eine zweite Überprüfung des Audiomaterials durch und bricht die Sitzung gegebenenfalls ab, wenn der Trigger-Befehl nicht enthalten ist.

Wenn die Funktion zum Aktivieren von Siri durch Berührung eingeschaltet ist, können Benutzer eine spezielle Taste auf dem Gerät drücken, um eine Konversation mit Siri zu starten. Das Audiomaterial wird direkt an den gekoppelten HomePod gesendet.

Nachdem eine erfolgreiche Aktivierung von Siri erkannt wurde, sendet der HomePod das Audiomaterial an die Siri-Server und führt mit denselben Sicherheits-, Datenschutz- und Verschlüsselungsmaßnahmen, die der HomePod beim Verarbeiten von direkt an den HomePod gestellten Anfragen anwendet, den vom Benutzer initiierten Vorgang aus. Wenn Siri eine Antwort in Audioform hat, wird diese über einen AirPlay-Audiokanal an das Gerät gesendet. Für einige Siri-Anfragen sind zusätzliche Informationen vom Benutzer erforderlich (beispielsweise bei der Nachfrage, ob der Benutzer weitere Optionen hören möchte). In diesem Fall empfängt das Gerät den Hinweis, dass der Benutzer zu einer Eingabe aufgefordert werden sollte. Das zusätzliche Audiomaterial wird an den HomePod gestreamt.

Das Gerät muss einen visuellen Indikator besitzen, um dem Nutzer anzuzeigen, wann es aktiv auf Befehle achtet (beispielsweise eine LED). Das Gerät erhält keine Informationen über die Siri-Anfrage, sondern nur für den Zugriff auf den Audiostream. Es werden keine Benutzerdaten auf dem Gerät gespeichert.

HomeKit-Datensicherheit

Die HomeKit-Daten können via iCloud und iCloud-Schlüsselbund zwischen den iOS-, iPadOS- und macOS-Geräten eines Benutzers sicher aktualisiert werden. Während dieses Vorgangs werden die HomeKit-Daten auf Basis von Schlüsseln verschlüsselt, die aus der HomeKit-Identität und einer zufälligen Nonce hergeleitet werden; die Daten werden als nicht einsehbare Daten oder als *blob* behandelt. Die jüngsten Daten werden in iCloud gespeichert, dort aber für keinen anderen Zweck verwendet. Da die Verschlüsselung mit Codes erfolgt, die nur auf den iOS-, iPadOS- und macOS-Geräten des Benutzers verfügbar sind, kann auf den Inhalt weder bei der Übertragung noch in iCloud zugegriffen werden.

Die HomeKit-Daten werden auch zwischen mehreren Benutzern im selben Zuhause synchronisiert. Dieser Prozess verwendet die Authentifizierung und Verschlüsselung, die auch zwischen einem iOS-, iPadOS- oder macOS-Gerät und einem HomeKit-Gerät verwendet wird. Die Authentifizierung basiert auf öffentlichen Ed25519-Schlüsseln, die zwischen den Geräten ausgetauscht werden, wenn einem Zuhause ein Benutzer hinzugefügt wird. Wurde dem Zuhause ein neuer Benutzer hinzugefügt, wird jede weitere Kommunikation mit dem Station-to-Station-Protokoll und für die Sitzung erzeugten Schlüsseln authentifiziert und verschlüsselt.

Neue Benutzer kann der Benutzer, der die häusliche Umgebung in HomeKit angelegt hat, oder jeder andere Benutzer mit Bearbeitungsrechten hinzufügen. Der Eigentümer des Geräts konfiguriert das Zubehörgerät mit dem öffentlichen Schlüssel des neuen Benutzers, sodass das Zubehörgerät den neuen Benutzer authentifizieren und Befehle von ihm empfangen kann. Wenn ein Benutzer mit Bearbeitungsrechten einen neuen Benutzer hinzufügt, wird der Vorgang an eine Steuerzentrale delegiert, die das Verfahren abschließt.

HomeKit und Apple TV

Die Bereitstellung des Apple TV-Geräts für die Verwendung mit HomeKit erfolgt automatisch, wenn der Benutzer sich bei iCloud anmeldet. Für den iCloud-Account muss die Zwei-Faktor-Authentifizierung aktiviert sein. Das Apple TV-Gerät und das Gerät des Eigentümers tauschen via iCloud temporäre öffentliche Ed25519-Schlüssel aus. Wenn sich der Eigentümer des Geräts und das Apple TV-Gerät in demselben lokalen Netzwerk befinden, werden die temporären Schlüssel verwendet, um eine sichere Verbindung über das lokale Netzwerk auf der Basis des Station-to-Station-Protokolls und der für die Sitzung erzeugten Schlüssel herzustellen. Dieser Prozess verwendet die Authentifizierung und Verschlüsselung, die auch zwischen einem iOS-, iPadOS- oder macOS-Gerät und einem HomeKit-Gerät verwendet wird. Über diese sichere lokale Verbindung überträgt das Gerät des Eigentümers die öffentlich/privaten Ed25519-Schlüsselpaare des Benutzers an das Apple TV-Gerät. Diese Schlüssel werden anschließend verwendet, um die Kommunikation zwischen dem Apple TV-Gerät und dem HomeKit-Gerät sowie zwischen dem Apple TV-Gerät und weiteren iOS-, iPadOS- und macOS-Geräten zu schützen, die Teil des HomeKit-Zuhauses sind.

Wenn ein Benutzer nicht mehrere Geräte verwendet und nicht zulässt, dass weitere Benutzer Zugriff auf seine häusliche Umgebung haben, werden keine HomeKit-Daten an iCloud übermittelt.

Daten und Apps eines Zuhauses

Der Zugriff von Apps auf die Daten eines Zuhauses wird in den Datenschutzeinstellungen des Benutzers festgelegt. Ähnlich wie bei Kontakten, Fotos und anderen iOS-, iPadOS- und macOS-Datenquellen werden Benutzer gefragt, ob der Zugriff gewährt werden soll, wenn Apps auf die Daten eines Zuhauses zuzugreifen versuchen. Stimmt der Benutzer zu, können Apps auf Zimmernamen, Gerätenamen, Gerätestandorte und weitere Informationen zugreifen, die in der HomeKit-Entwicklerdokumentation unter <https://developer.apple.com/homekit/> beschrieben werden.

Lokaler Datenspeicher

HomeKit speichert Daten über das Zuhause, das Zubehör, die Szenen und die Benutzer auf dem iOS-, iPadOS- oder macOS-Gerät des Benutzers. Diese gespeicherten Daten werden mit Schlüsseln verschlüsselt, die von den Schlüsseln der HomeKit-Identität und einer zufälligen Nonce abgeleitet werden. Außerdem werden HomeKit-Daten mit der Datensicherheitsklasse „Geschützt bis zur ersten Benutzerauthentifizierung“ geschützt. HomeKit-Daten werden ausschließlich in verschlüsselten Backups gesichert. So enthalten zum Beispiel unverschlüsselte Backups mit dem Finder (macOS 10.15 oder neuer) oder iTunes (macOS 10.14 oder älter), die über USB erstellt werden, keine HomeKit-Daten.

Sichern von Routern mit HomeKit

Router, die HomeKit unterstützen, ermöglichen es Benutzern, die Sicherheit des häuslichen Netzwerks zu verbessern, indem der WLAN-Zugriff verwaltet wird, den HomeKit-Geräte für das lokale Netzwerk und das Internet verwenden. Diese Router unterstützen auch die private PSK-Authentifizierung (PPSK), sodass Geräte mit einem Schlüssel zum WLAN hinzugefügt werden können, der für das Gerät eindeutig ist und bei Bedarf zurückgenommen werden kann. Durch die PSK-Authentifizierung wird die Sicherheit verbessert, da das WLAN-Hauptpasswort den Geräten nicht preisgegeben wird. Zudem kann der Router ein Gerät somit sicher identifizieren, selbst wenn es seine MAC-Adresse ändert.

Mit der App „Home“ kann ein Benutzer die Zugriffsrechte für Gerätegruppen wie folgt konfigurieren:

- *Keine Einschränkung:* Der Zugriff auf das Internet und das lokale Netzwerk ist uneingeschränkt erlaubt.
- *Automatisch:* Dies ist die Standardeinstellung. Der Zugriff auf das Internet und das lokale Netzwerk ist basierend auf einer Liste mit Internet-Websites und lokalen Ports erlaubt, die Apple vom Gerätehersteller bereitgestellt werden. Diese Liste enthält alle Websites und Ports, die das Gerät benötigt, um ordnungsgemäß zu funktionieren. („Keine Einschränkung“ wird verwendet, bis eine solche Liste verfügbar ist.)
- *Auf Home beschränken:* Der Zugriff auf das Internet oder das lokale Netzwerk ist nicht erlaubt. Ausgenommen sind Verbindungen, die HomeKit benötigt, um Geräte des lokalen Netzwerks (inkl. der Steuerzentrale zur Unterstützung der Fernbedienung) zu erkennen.

Ein PPSK ist ein starker, gerätespezifischer WPA2 Personal-Code, der von HomeKit automatisch generiert wird. Er wird zurückgenommen, wenn das Gerät später aus der häuslichen Umgebung entfernt wird. Ein PPSK wird verwendet, wenn ein Gerät durch HomeKit zum WLAN in einem Zuhause hinzugefügt wird, das mit einem HomeKit-Router konfiguriert wurde; dies wird durch die Einstellung „Wi-Fi-Sicherheit: HomeKit-verwaltet“ in der App „Home“ des betreffenden Geräts reflektiert. Zubehörgeräte, die dem WLAN hinzugefügt wurden, bevor der Router hinzukam, werden so umkonfiguriert, dass auch sie ein PPSK verwenden, sofern das Zubehör dies unterstützt; anderenfalls behalten sie ihre vorhandenen Anmeldedaten bei.

Als weitere Sicherheitsmaßnahme müssen Benutzer den HomeKit-Router mit der App des Routerherstellers so konfigurieren, dass die App prüfen kann, ob der Benutzer auf den Router zugreifen kann und ihn zur App „Home“ hinzufügen darf.

Sicherheit von HomeKit-Kameras

Kameras, denen in HomeKit eine IP-Adresse (Internet Protocol) zugewiesen wird, senden verschlüsselte Video- und Audiostreams direkt an iPhone-, iPad-, Apple TV- und Mac-Geräte, die auf den Stream zugreifen. Die Streams werden auf dem Gerät und in der IP-Kamera mit nach dem Zufallsprinzip generierten Schlüsseln verschlüsselt, die über die sichere HomeKit-Sitzung mit der Kamera ausgetauscht werden. Wenn sich ein Gerät nicht im lokalen Netzwerk befindet, werden die verschlüsselten Streams über die Steuerzentrale (Home Hub) an das Gerät weitergeleitet. Die Steuerzentrale entschlüsselt die Streams nicht, sondern fungiert nur als Relais zwischen dem Gerät und der IP-Kamera. Wenn in einer App das Videomaterial angezeigt wird, das von der IP-Kamera von HomeKit übermittelt wird, gibt HomeKit die Videobilder sicher über einen separaten Systemprozess wieder. Auf diese Weise wird verhindert, dass die App auf den Videostream zugreifen oder ihn speichern kann. Außerdem dürfen Apps keine Bildschirmfotos aus diesem Stream speichern.

Sicheres HomeKit-Video

HomeKit bietet einen sicheren und privaten Ende-zu-Ende-Mechanismus, um Clips von HomeKit-IP-Kameras aufzunehmen, zu analysieren und anzusehen, ohne dieses Material für Apple oder Drittanbieter zugänglich zu machen. Wird von der IP-Kamera eine Bewegung festgestellt, werden die Videoclips direkt an ein Apple-Gerät gesendet, das als Steuerzentrale konfiguriert ist. Dazu wird eine dedizierte, lokale Netzwerkverbindung zwischen der Steuerzentrale und der IP-Kamera verwendet. Die lokale Netzwerkverbindung wird mit einem pro Sitzung abgeleiteten HKDF-SHA-512-Schlüsselpaar verschlüsselt, das über die HomeKit-Sitzung zwischen der Steuerzentrale und der IP-Kamera ausgehandelt wird. HomeKit entschlüsselt die Audio- und Videostreams auf der Steuerzentrale und analysiert lokal die Videobilder, um Unregelmäßigkeiten festzustellen. Falls Unregelmäßigkeiten festgestellt werden, verschlüsselt HomeKit den Videoclip mithilfe von AES-256-GCM mit einem zufällig generierten AES-256-Schlüssel. HomeKit generiert außerdem für jeden Clip Titelbilder. Diese Bilder werden mit demselben AES-256-Schlüssel verschlüsselt. Die Daten verschlüsselter Titelbilder sowie die Audio- und Videodaten werden auf iCloud-Server geladen. Die zu jedem Clip gehörenden Metadaten, inkl. Verschlüsselungsschlüssel, werden mittels iCloud-Ende-zu-Ende-Verschlüsselung in CloudKit geladen.

Im Hinblick auf die Klassifizierung von Gesichtern speichert HomeKit in CloudKit alle Daten, die zur Klassifizierung des Gesichts einer bestimmten Person herangezogen werden, wobei die Ende-zu-Ende-Verschlüsselung von iCloud zum Einsatz kommt. Die gespeicherten Daten umfassen Informationen zur jeweiligen Person, z. B. deren Namen und Bilder, die das Gesicht der Person darstellen. Diese Gesichtsbilder können aus der App „Fotos“ des Benutzers entnommen werden, sofern der Benutzer seine Zustimmung dazu gibt, oder aus bereits analysierten Videosequenzen der IP-Kamera gesammelt werden. In einer Sitzung zur Analyse eines sicheren HomeKit-Videos werden diese Klassifizierungsdaten verwendet, um Gesichter im sicheren Videostream zu identifizieren, der direkt von der IP-Kamera gesendet wird; dabei werden auch die oben erwähnten Identifikationsinformationen einbezogen, die in den Metadaten der Clips enthalten sind.

Wenn die Home-App verwendet wird, um die Clips einer Kamera anzuzeigen, werden die Daten von iCloud geladen und die Schlüssel zum Entschlüsseln der Streams werden lokal mithilfe der iCloud-Ende-zu-Ende-Verschlüsselung ausgepackt. Der verschlüsselte Videoinhalt wird vom Server gestreamt und lokal auf dem iOS-Gerät entschlüsselt, bevor er im Viewer angezeigt wird. Jede Videoclipsitzung kann unterteilt werden. Der Inhaltsstream wird dabei von jedem Teil mit einem eigenen eindeutigen Schlüssel verschlüsselt.

HomeKit-Sicherheit bei Apple TV-Geräten

HomeKit stellt sichere Verbindungen zwischen bestimmten Remote-Geräten anderer Anbieter und Apple TV her und ermöglicht es, Benutzerprofile zum Eigentümer des häuslichen Apple TV hinzuzufügen.

Verwenden von Remote-Zubehörgeräten anderer Anbieter für Apple TV

Remote-Zubehörgeräte einiger Anbieter stellen HID-Ereignisse (Human Interface Design) und Siri-Audio für ein zugeordnetes Apple TV bereit, das über die App „Home“ hinzugefügt wird. Das Remote-Gerät sendet HID-Ereignisse über eine sichere Sitzung an das Apple TV-Gerät. Eine Siri-fähige TV Remote-Fernbedienung sendet Audiodaten an Apple TV, wenn der Benutzer das Mikrofon auf der Remote-Fernbedienung explizit mit einer speziellen Siri-Taste aktiviert. Das Remote-Gerät sendet die Audio-Frames über eine dedizierte lokale Netzwerkverbindung direkt an Apple TV. Die lokale Netzwerkverbindung wird mit einem pro Sitzung hergeleiteten HKDF-SHA-512-Schlüsselpaar verschlüsselt, das über die HomeKit-Sitzung zwischen dem Apple TV- und dem TV Remote-Gerät verschlüsselt ausgehandelt wird. HomeKit entschlüsselt die Audio-Frames auf Apple TV und leitet sie an die App „Siri“ weiter, wo diese mit denselben Sicherheits- und Datenschutzmechanismen behandelt werden wie alle Siri-Audioeingaben.

Apple TV-Profil für HomeKit-Zuhause

Wenn ein Benutzer eines HomeKit-Zuhauses sein Profil zum Eigentümer des Apple TV des betreffenden Zuhauses hinzufügt, erhält der Benutzer Zugriff auf die TV-Sendungen, Musiktitel und Podcasts des Eigentümers bzw. dessen Zuhauses. Die Einstellungen jedes Benutzers, die für die Profilnutzung auf dem Apple TV relevant sind, werden über eine iCloud-Verbindung mit Ende-zu-Ende-Verschlüsselung mit dem iCloud-Account des Eigentümers geteilt. Die Daten verbleiben im Besitz des jeweiligen Benutzers und werden dem Eigentümer nur zum Lesen bereitgestellt. Jeder Benutzer des betreffenden Zuhauses kann diese Werte in der App „Home“ ändern, und das Apple TV-Gerät des Eigentümers verwendet diese Einstellungen.

Wenn eine Einstellung aktiviert wird, wird der iTunes-Account des Benutzers auf dem Apple TV-Gerät bereitgestellt. Wenn eine Einstellung deaktiviert wird, werden der Account und alle Daten des betreffenden Benutzers auf dem Apple TV-Gerät gelöscht. Die erstmalige CloudKit-Freigabe wird durch das Gerät des Benutzers initiiert und der Token, der für die Einrichtung der sicheren CloudKit-Freigabe verwendet wird, wird über denselben sicheren Kanal gesendet, der auch für die Synchronisierung von Daten zwischen den Benutzern des Zuhauses verwendet wird.

Sicherheit beim SiriKit für iOS, iPadOS und watchOS

Siri nutzt das System für App-Erweiterungen, um mit Apps anderer Anbieter zu kommunizieren. Auf dem Gerät kann Siri auf die Kontaktinformationen des jeweiligen Benutzers und den aktuellen Standort von dessen Gerät zugreifen. Bevor Siri geschützte Daten für eine App bereitstellt, werden die vom Benutzer für die App festgelegten Zugriffsrechte überprüft. Entsprechend diesen Berechtigungen übermittelt Siri an die App-Erweiterung nur das relevante Fragment der Originaläußerung des Benutzers. Wenn eine App beispielsweise keinen Zugriff auf die Kontaktinformationen hat, löst Siri keine Beziehung in einer Anfrage wie der folgenden auf: „Sende meiner Mutter 10 Euro mit der ‚Zahlungs-App‘“. In diesem Beispiel würde die App nur die Wortfolge „meiner Mutter“ sehen.

Wenn hingegen der Benutzer der App den Zugriff auf die Kontaktinformationen gewährt hat, würde die App die aufgelösten Informationen über die Mutter des Benutzers erhalten. Wenn im Haupttext einer Nachricht auf eine Beziehung abgehoben wird – zum Beispiel: „Erzähle meiner Mutter über die ‚Nachrichten-App‘, mein Bruder ist super“ – löst Siri das Fragment „mein Bruder“ unabhängig von den Berechtigungen der App nicht auf.

SiriKit unterstützende Apps können app- oder benutzerspezifisches Vokabular an Siri senden, zum Beispiel die Namen von Kontakten des Benutzers. Diese Informationen ermöglichen es den Siri-Funktionen für die Spracherkennung und das Verstehen natürlicher Sprachen, das Vokabular der betreffenden App zu erkennen. Sie werden der zufälligen Kennung zugeordnet. Die spezifischen Informationen werden vorgehalten, solange diese Kennung in Gebrauch ist oder bis zu dem Zeitpunkt, an dem der Benutzer in seinen Einstellungen die Siri-Integration für die App deaktiviert oder die SiriKit unterstützende App deinstalliert wird.

Für die Äußerung „Rufe mit der RideShareApp ein Taxi für die Fahrt zum Haus meiner Mutter“ benötigt die Anforderung Standortdaten aus den Kontakten des Benutzers. Ausschließlich für diese Anforderung stellt Siri die benötigten Informationen bereit, und zwar unabhängig von den Berechtigungen und Einstellungen, die der Benutzer für die App in Bezug auf Standort- und Kontaktinformationen festgelegt hat.

Sicherheit beim DriverKit für macOS

DriverKit ist das Framework, mit dessen Hilfe Entwickler die Gerätetreiber erstellen können, die Benutzer auf ihren Mac-Computern installieren. Mit DriverKit erstellte Treiber werden im Benutzerspeicherplatz, und nicht als Kernel-Erweiterungen ausgeführt. Dadurch erhöhen sich Sicherheit und Stabilität des Systems. Dies vereinfacht die Installation und erhöht zugleich die Stabilität und die Sicherheit von macOS.

Der Benutzer muss die App nur herunterladen – bei Verwendung von Systemerweiterungen oder DriverKit entfällt die Notwendigkeit für Installationsprogramme. Die Erweiterung wird nur aktiviert, wenn sie explizit angefordert wird. In vielen Fällen können dadurch kexts ersetzt werden, für die installierte Administratorrechte in „/System/Library“ oder „/Library“ notwendig sind.

IT-Administratoren, die Gerätetreiber, Cloud-Speicherlösungen, Netzwerkfunktionen und Sicherheits-Apps verwenden, die Kernel-Erweiterungen benötigen, wird der Umstieg auf neuere, auf Systemerweiterungen aufbauende Versionen empfohlen. Diese neuen Versionen reduzieren die Wahrscheinlichkeit von kritischen Kernel-Situationen auf dem Mac in ganz erheblichem Umfang, da sie auch die Angriffsfläche reduzieren. Diese neuen Erweiterungen werden im Benutzerspeicherbereich ausgeführt, sie erfordern für die Installation keine speziellen Berechtigungen und sie werden automatisch entfernt, wenn die zugehörige App in den Papierkorb bewegt wird.

Das DriverKit-Framework umfasst C++-Klassen für E/A-Dienste, für den Geräteabgleich, für Speicherdeskriptoren und für Dispatch-Warteschlangen. Es definiert außerdem die E/A-geeigneten Typen für Zahlungen, Sammlungen, Zeichenfolgen (Strings) und andere gebräuchliche Typen. Benutzer können sie mithilfe familienzuspezifischer Treiber-Frameworks wie USBDriverKit und HIDDriverKit verwenden. Das Framework für Systemerweiterungen kann verwendet werden, um einen Treiber oder ein Upgrade für einen Treiber zu installieren.

Sicherheit beim ReplayKit für iOS und iPadOS

ReplayKit ist ein Framework, das es Entwicklern ermöglicht, ihre Apps mit Funktionen zur Aufzeichnung und Live-Übertragung auszustatten. Darüber hinaus ermöglicht es Benutzern, ihren Aufnahmen und Übertragungen mit der Frontkamera und dem Mikrofon des Geräts Kommentare hinzuzufügen.

Filmaufnahme

Das Aufnehmen eines Films umfasst verschiedene Sicherheitsstufen:

- *Berechtigungsfenster:* Vor dem Beginn der Aufnahme fordert ReplayKit den Benutzer in einem Hinweisfenster auf zu bestätigen, dass der Bildschirm, das Mikrofon und die Frontkamera aufgezeichnet werden sollen. Dieser Hinweis erscheint einmal pro App-Prozess. Wenn die App länger als 8 Minuten im Hintergrund verbleibt, wird der Hinweis erneut angezeigt.
- *Bildschirm- und Audioaufzeichnung:* Bildschirm- und Audioaufzeichnungen erfolgen außerhalb des App-Prozesses im ReplayKit-Daemon „replayd“. Auf diese Weise soll gewährleistet werden, dass der aufgezeichnete Inhalt für den App-Prozess stets unzugänglich bleibt.
- *Bildschirm- und Audioaufzeichnung in Apps:* Dies ermöglicht einer App den Abruf von Video- und Sample-Puffern unter dem Schutz des Dialogfensters „Berechtigungen“.
- *Filmerstellung und -sicherung:* Die Filmdatei wird in ein Verzeichnis geschrieben, das ausschließlich für ReplayKit-Subsysteme zugänglich ist und zu dem Apps keinerlei Zugang haben. Dadurch soll verhindert werden, dass Aufnahmen ohne Einwilligung des Benutzers von Dritten verwendet werden können.
- *Vorschau und Freigabe durch den Endbenutzer:* Mit der von ReplayKit bereitgestellten Benutzeroberfläche kann der Benutzer den Film in einer Vorschau ansehen und ihn mit anderen teilen. Die Benutzeroberfläche wird außerhalb des Prozesses über die Infrastruktur für iOS-Erweiterungen angezeigt und hat Zugriff auf die generierte Filmdatei.

ReplayKit-Übertragung

Die Übertragung eines Films umfasst verschiedene Sicherheitsstufen:

- *Bildschirm- und Audioaufzeichnung:* Der Mechanismus der Bildschirm- und Audioaufzeichnung während der Übertragung ist identisch mit der Filmaufnahme und erfolgt im Daemon replayd.
- *Erweiterungen für die Übertragung:* Damit Dienste anderer Anbieter an der ReplayKit-Übertragung teilnehmen können, müssen sie zwei neue Erweiterungen erstellen, die mit dem Endpunkt com.apple.broadcast-services konfiguriert wurden:
 - Eine Erweiterung der Benutzeroberfläche, die es dem Benutzer ermöglicht, seine Übertragung einzurichten.
 - Eine Upload-Erweiterung für das Hochladen von Video- und Audiodaten an die Back-End-Server des Dienstes.

Die Architektur stellt sicher, dass die bereitstellenden Apps keine Berechtigungen an den übertragenen Video- und Audioinhalten erhalten. Zugriff haben nur ReplayKit und die für die Übertragung genutzten Erweiterungen der anderen Anbieter.

- *Auswahl des Übertragungsdienstes:* Der Übertragungsdienst ermöglicht es Benutzern, Systemübertragungen direkt aus der App zu starten und dabei die gleiche systemdefinierte Benutzeroberfläche zu verwenden, die über das Kontrollzentrum zugänglich ist. Die Benutzeroberfläche wird mit einer privaten API implementiert und ist eine Erweiterung, die innerhalb des ReplayKit-Frameworks verbleibt. Sie läuft außerhalb des Prozesses der bereitstellenden App.
- *Upload-Erweiterung:* Dies ist die Erweiterung, die Übertragungsdienste anderer Anbieter implementieren, um die Video- und Audioinhalte während der Übertragung zu verarbeiten. Sie verwendet Sample-Puffer vom Typ „RAW uncodiert“. Während dieses Handhabungsmodus werden Video- und Audiodaten serialisiert und über eine direkte XPC-Verbindung in Echtzeit an die Upload-Erweiterung des anderen Anbieters übergeben. Die Videodaten werden codiert, indem das IOSurface-Objekt aus dem Video-Sample-Puffer extrahiert, als XPC-Objekt sicher codiert, via XPC an die Erweiterung des anderen Anbieters gesendet und schließlich wieder sicher in ein IOSurface-Objekt decodiert wird.

Sicherheit beim ARKit für iOS und iPadOS

ARKit ist ein Framework, das es Entwicklern ermöglicht, ihre Apps und Spiele mit AR-Erlebnissen (Augmented Reality [Erweiterte Realität]) auszustatten. Entwickler können mithilfe der vorderen oder hinteren Kamera eines iOS- oder iPadOS-Geräts 2D- und 3D-Elemente hinzufügen.

Apple konzipiert Kameras unter Berücksichtigung des Datenschutzes. Apps anderer Anbieter müssen die Zustimmung des jeweiligen Benutzers einholen, wenn sie auf die Kamera zugreifen wollen. Bei iOS und iPadOS kann eine App, der der Benutzer den Zugriff auf die Kamera gestattet hat, auf die Echtzeitbilder der vorderen und der hinteren Kamera zugreifen. Apps bleibt es verwehrt, die Kamera zu nutzen, ohne dass für den Benutzer offensichtlich ist, dass die Kamera in Gebrauch ist.

Fotos und Videos, die mit der Kamera aufgenommen werden, können weitere Informationen enthalten, zum Beispiel Zeitpunkt und Ort der Aufnahme, Schärfen- oder Fokustiefe und OverCapture. Wenn Benutzer nicht möchten, dass in Fotos und Videos, die mit der App „Kamera“ aufgenommen werden, der Aufnahmeort vermerkt wird, können sie dies in den Einstellungen im Bereich „Datenschutz“ > „Ortungsdienste“ > „Kamera“ angeben. Wenn Benutzer vermeiden wollen, dass mit den Fotos und Videos, die sie teilen, die jeweiligen Aufnahmeorte bereitgestellt werden, können sie im Menü „Optionen“ der Freigabeseite die Option für den Standort bzw. Aufnahmeort deaktivieren.

Apps, die ARKit nutzen, können Informationen zu World Tracking und Gesichtserfassung verwenden, die von der anderen Kamera bereitgestellt werden, um das AR-Erlebnis für den Benutzer zu optimieren. Für das World Tracking kommen Algorithmen auf dem Gerät des Benutzers zum Einsatz, um die Informationen dieser Sensoren zu verarbeiten und auf dieser Grundlage die relative Position im physischen Raum zu bestimmen. Durch World Tracking werden Funktionen wie „Optische Orientierung“ in der App „Karten“ ermöglicht.

Sichere Geräteverwaltung

Sichere Geräteverwaltung – Übersicht

iOS, iPadOS, macOS und tvOS unterstützen flexible Sicherheitsrichtlinien und Konfigurationen, die einfach umgesetzt und verwaltet werden können. Dadurch ist es Organisationen möglich, interne Informationen zu schützen und sicherzustellen, dass Mitarbeiter die Vorgaben der Organisation einhalten, selbst wenn sie ihre eigenen Geräte verwenden, z. B. im Rahmen eines BYOD-Programms („Bring Your Own Device“).

Organisationen können Methoden wie Codesicherheit, Konfigurationsprofile, Fernlöschung und Lösungen für die Mobilgeräteverwaltung (Mobile Device Management, MDM) anderer Anbieter verwenden, um den Gerätebestand zu verwalten und Firmendaten zu schützen, selbst wenn Mitarbeiter über persönliche Geräte auf diese Daten zugreifen.

Apple-Geräte mit iOS 13 (oder neuer), iPadOS 13.1 (oder neuer) und macOS 10.15 (oder neuer) unterstützen eine neue Option „Benutzerregistrierung“, die gezielt für BYOD-Szenarien konzipiert ist. Die Benutzerregistrierung gestattet Benutzern mehr Autonomie auf ihren Geräten, während zugleich Schutz und Sicherheit von Firmendaten optimiert werden, da die Daten auf separaten, kryptografisch geschützten APFS (Apple File System)-Volumes gespeichert werden. Dadurch werden bei BYOD-Szenarien Sicherheit, Datenschutz und Benutzererlebnis in ein ausgewogenes Verhältnis gebracht.

Sicherheit beim Kopplungsmodell für iPhone und iPad

iOS und iPadOS verwenden ein Kopplungsmodell, um den Zugriff auf ein Gerät durch einen Host-Computer zu steuern. Die Kopplung stellt über den Austausch der öffentlichen Schlüssel eine vertrauenswürdige Verbindung zwischen dem Gerät und dem verbundenen Host her. iOS und iPadOS verwenden diese vertrauenswürdige Verbindung, um zusätzliche Funktionen mit dem verbundenen Host (z. B. die Synchronisierung von Daten) zu aktivieren. Bei iOS 9 (und neueren Versionen) gilt Folgendes für Dienste:

- Sie können, wenn das Koppeln erforderlich ist, erst gestartet werden, nachdem die Gerätesperre durch den Benutzer aufgehoben wurde.
- Sie werden nur gestartet, wenn das Gerät kurz davor entsperrt wurde.
- Sie werden möglicherweise erst gestartet, wenn das Gerät entsperrt wird (dies gilt zum Beispiel für Dienste für die Fotosynchronisierung).

Für die Kopplung muss der Benutzer das Gerät entsperren und die Anfrage des Hosts annehmen. Bei iOS 9 (und neueren Versionen) muss der Benutzer außerdem seinen Code eingeben. Daraufhin tauschen Host und Gerät öffentliche 2048-Bit RSA-Schlüssel aus und sichern sie. Der Host erhält daraufhin einen 256-Bit-Schlüssel, mit dem ein auf dem Gerät gespeicherter Escrow-Keybag entsperrt werden kann. Die ausgetauschten Schlüssel werden zum Starten einer verschlüsselten SSL-Sitzung verwendet, die das Gerät benötigt, damit es geschützte Daten an den Host senden oder einen Dienst starten kann (iTunes- oder Finder-Synchronisierung, Dateiübertragung, Xcode-Entwicklung usw.). Das Gerät benötigt eine Verbindung von einem Host über WLAN, um diese verschlüsselte Sitzung für die gesamte Kommunikation zu verwenden. Es muss daher zuvor über USB gekoppelt worden sein. Das Koppeln ermöglicht auch mehrere Diagnosefunktionen. Bei iOS 9 verfällt ein für die Kopplung genutzter Datensatz, wenn er länger als 6 Monate nicht verwendet wurde. Bei iOS 11 (und neueren Versionen) ist dieses Zeitfenster auf 30 Tage verkürzt.

Bestimmte Diagnosedienste (z. B. `com.apple.mobile.pcapd`) sind auf USB-Verbindungen beschränkt. Der Dienst `com.apple.file_relay` benötigt außerdem ein von Apple signiertes Konfigurationsprofil, um installiert werden zu können. Bei iOS 11 (und neueren Versionen) kann Apple TV das SRP-Protokoll (Secure Remote Password) verwenden, um drahtlos eine Kopplungsbeziehung zu erstellen.

Der Benutzer kann die Liste vertrauenswürdiger Hosts mit der Option „Netzwerkeinstellungen zurücksetzen“ oder „Standort & Datenschutz zurücksetzen“ löschen.

Mobilgeräteverwaltung (Mobile Device Management)

Sicherheit bei der Mobilgeräteverwaltung – Übersicht

Betriebssysteme von Apple unterstützen die Mobilgeräteverwaltung (Mobile Device Management, MDM), die es Organisationen ermöglicht, skalierte Implementierungen von Apple-Geräten sicher zu konfigurieren und zu verwalten.

Sichere Funktionsweise von MDM

MDM-Funktionen basieren auf vorhandenen Technologien des Betriebssystems (z. B. Konfigurationsprofile, drahtlose Registrierung und [APNS \(Apple Push Notification Service\)](#), [dem Apple-Dienst für Push-Benachrichtigungen oder Push-Mitteilungen](#)). APNS wird beispielsweise verwendet, um den Ruhezustand eines Geräts zu beenden, damit es über eine sichere Verbindung direkt mit der zugehörigen MDM-Lösung kommunizieren kann. Über APNS werden keine vertraulichen oder organisationseigenen Informationen übertragen.

MDM eröffnet IT-Abteilungen die Möglichkeit, Apple-Geräte sicher in einer Unternehmensumgebung zu registrieren, drahtlos Einstellungen zu konfigurieren und zu aktualisieren, die Einhaltung von Unternehmensrichtlinien zu überwachen, Richtlinien für Softwareaktualisierungen zu verwalten und verwaltete Geräte per Fernzugriff zu löschen oder zu sperren.

Zusätzlich zu den von iOS, iPadOS, macOS und tvOS bisher schon unterstützten Geräteregistrierungen wurde in iOS 13, iPadOS 13.1 und macOS 10.15 ein Registrierungstyp hinzugefügt: die Benutzerregistrierung. Benutzerregistrierungen sind Registrierungen bei einer MDM-Lösung, die gezielt für BYOD-Szenarien („Bring Your Own Device“) konzipiert sind, d. h. für Szenarien, in denen Benutzer in ihrem Besitz befindliche Geräte in einer betreuten Umgebung verwenden. Im Vergleich zu Registrierungen betreuter Geräte gewähren Benutzerregistrierungen der MDM-Lösung weniger Berechtigungen. Sie sorgen aber für die kryptografisch basierte Trennung von privaten Daten des Benutzers und Firmendaten.

Registrierungstypen

- *Automatische Geräteregistrierung:* Die automatische Geräteregistrierung bietet Organisationen die Möglichkeit, Geräte ab dem Moment, in dem sie aus ihrer Verpackung genommen werden, zu konfigurieren und zu verwalten; (dieser Prozess wird als *Auto Advance-Implementierung* bezeichnet). Diese Geräte werden als *betreute Geräte* bezeichnet. In diesem Fall kann verhindert werden, dass das MDM-Profil vom Benutzer entfernt werden kann. Die automatische Geräteregistrierung ist für Geräte konzipiert, die Eigentum der jeweiligen Organisation sind.

- *Geräteregistrierung:* Die Geräteregistrierung ermöglicht es Organisationen, die Geräte manuell durch die Benutzer registrieren zu lassen und danach verschiedene Aspekte der Gerätenutzung zu verwalten – auch die Möglichkeit, das Gerät zu löschen. Bei der Geräteregistrierung wird auch eine größere Anzahl an Payloads und Einschränkungen unterstützt, die auf das jeweilige Gerät angewendet werden können. Wenn ein Benutzer ein Registrierungsprofil entfernt, werden automatisch auch alle Konfigurationsprofile und deren Einstellungen sowie alle Apps entfernt, die auf Basis des betreffenden Registrierungsprofils verwaltet werden.
- *Benutzerregistrierung:* Die Benutzerregistrierung ist für benutzereigene Geräte konzipiert und mit verwalteten Apple-IDs integriert, um eine Benutzeridentität auf dem Gerät zu etablieren. Die verwalteten Apple-IDs sind Bestandteil des Profils für die Benutzerregistrierung. Der Benutzer muss sich erfolgreich authentifizieren, damit die Registrierung zu Ende geführt wird. Verwaltete Apple-IDs können parallel zur privaten/ persönlichen Apple-ID verwendet werden, mit der sich der Benutzer bereits angemeldet hat. Verwaltete Apps und Accounts verwenden eine verwaltete Apple-ID, wohingegen persönliche Apps und Accounts eine persönliche Apple-ID verwenden.

Geräteeinschränkungen

Einschränkungen können aktiviert – oder in einigen Fällen deaktiviert – werden, um zu verhindern, dass ein Benutzer auf einem iPhone-, iPad-, Mac- oder Apple TV-Gerät, das in einer Lösung für die Mobilgeräteverwaltung (MDM) registriert ist, auf eine bestimmte App, einen bestimmten Dienst oder eine bestimmte Funktion zugreift. Die Einschränkungen werden in einer spezifischen Payload, die Bestandteil eines Konfigurationsprofils ist, an die Geräte gesendet. Bestimmte Einschränkungen auf einem iPhone können auf eine gekoppelte Apple Watch gespiegelt werden.

Verwalten der Code- und Passworteinstellungen

Der Gerätecode des Benutzers ist standardmäßig eine aus vier Ziffern bestehende PIN. Auf iOS- und iPadOS-Geräten mit Face ID oder Touch ID muss der Gerätecode mindestens sechs Ziffern umfassen. Es wird empfohlen, längere und komplexere Codes zu verwenden, da sie schwerer zu erraten oder anzugreifen sind.

Administratoren können komplexe Codes und andere Richtlinien mit Lösungen für die Mobilgeräteverwaltung oder per Exchange ActiveSync durchsetzen oder alternativ die Benutzer auffordern, dass sie die Konfigurationsprofile manuell installieren. Zum Installieren der Payload mit der Richtlinie für den macOS-Code ist ein Administratorpasswort erforderlich. Für Codes relevante Richtlinien geben möglicherweise vor, dass Codes eine bestimmte Mindestlänge, bestimmte Zeichenkombinationen oder bestimmte Attribute aufweisen müssen.

Durchsetzen von Richtlinien mit Konfigurationsprofilen

Bei einer MDM-Lösung sind Konfigurationsprofile die erste Wahl, wenn es darum geht, Richtlinien und Einschränkungen zu verwalten und sie an verwaltete Geräte zu übermitteln. Konfigurationsprofile sind eine sichere und verlässliche Methode, um eine große Anzahl von Geräten zu konfigurieren oder spezifische E-Mail-Einstellungen, Netzwerkeinstellungen oder Zertifikate auf einer großen Anzahl von Geräten bereitzustellen.

Konfigurationsprofile

Ein *Konfigurationsprofil* ist eine XML-Datei (mit der Endung `.mobileconfig`), die sich aus Payloads zusammensetzt, mit denen Einstellungen und Berechtigungsinformationen auf Apple-Geräte geladen werden. Mit Konfigurationsprofilen kann die Konfiguration von Einstellungen, Accounts, Einschränkungen und Anmeldedaten automatisiert werden. Diese Dateien können von einer MDM-Lösung oder von Apple Configurator für den Mac erstellt werden. Darüber hinaus ist es möglich, sie manuell zu erstellen. Damit ein Konfigurationsprofil an ein Apple-Gerät gesendet werden kann, muss das Gerät vorab mithilfe eines Registrierungsprofils in der MDM-Lösung registriert werden.

Registrierungsprofile

Ein *Registrierungsprofil* ist ein Konfigurationsprofil mit einer MDM-Payload, mit der ein Gerät in der MDM-Lösung registriert wird, die für das Gerät angegeben wurde. Danach ist die MDM-Lösung in der Lage, Befehle und Konfigurationsprofile an das Gerät zu senden und bestimmte Informationen und Aspekte vom Gerät abzurufen. Wenn ein Benutzer ein Registrierungsprofil entfernt, werden automatisch auch alle Konfigurationsprofile und deren Einstellungen sowie alle Apps entfernt, die auf Basis des betreffenden Registrierungsprofils verwaltet werden. Auf einem Gerät kann stets nur ein Registrierungsprofil installiert sein.

Einstellungen in Konfigurationsprofilen

Ein Konfigurationsprofil enthält eine Reihe von Einstellungen in spezifischen Payloads, mit denen u. a. Folgendes festgelegt werden kann (ohne darauf beschränkt zu sein):

- Richtlinien für Codes und Passwörter
- Einschränkungen hinsichtlich Gerätefunktionen (z. B. Deaktivieren der Kamera)
- Netzwerk- und VPN-Einstellungen
- Einstellungen für Microsoft Exchange
- Einstellungen für Mail
- Accounteinstellungen
- Einstellungen für LDAP-Verzeichnisdienst
- Einstellungen für CalDAV-Kalenderdienst
- Anmeldedaten und Schlüssel
- Softwareaktualisierungen

Signieren und Verschlüsseln von Profilen

Konfigurationsprofile können signiert werden, um ihre Quelle zu validieren, und verschlüsselt werden, um ihre Integrität zu wahren und ihren Inhalt zu schützen. Konfigurationsprofile für iOS und iPadOS werden gemäß RFC 5652 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc5652.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt) unter Verwendung von Cryptographic Message Syntax (CMS) verschlüsselt, sodass 3DES und AES-128 unterstützt werden.

Installieren von Profilen

Benutzer können Konfigurationsprofile mithilfe von Apple Configurator für den Mac direkt auf ihren Geräten installieren. Profile können alternativ auch in Safari geladen, als Anhang einer E-Mail gesendet und empfangen, via AirDrop oder über die App „Dateien“ auf iOS- und iPadOS-Geräte transferiert oder drahtlos mithilfe einer Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM) gesendet werden. Wenn ein Benutzer ein Gerät in [Apple School Manager](#) oder [Apple Business Manager](#) einrichtet, lädt und installiert das Gerät ein Profil für die Registrierung in der MDM-Lösung. Weitere Informationen zum Entfernen von Profilen sind unter [Einführung in MDM \(Mobile Device Management\)](#) auf der Website für die Implementierung von Apple-Plattformen zu finden.

Hinweis: Bei betreuten Geräten können Konfigurationsprofile auch gesperrt und fest mit einem Gerät verknüpft werden. So soll verhindert werden, dass sie überhaupt nicht oder nur bei Eingabe eines Codes entfernt werden können. Da vielfach Organisationen im Besitz der iOS- und iPadOS-Geräte sind, können Konfigurationsprofile, die ein Gerät fest an eine MDM-Lösung binden, entfernt werden. Allerdings werden dabei zugleich auch alle verwalteten Konfigurationsinformationen, Daten und Apps entfernt.

Automatische Geräteregistrierung

Organisationen können iOS-, iPadOS-, macOS- und tvOS-Geräte automatisch in ihrer Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM) registrieren, ohne sie vor der Übergabe an die Benutzer in die Hand nehmen oder vorbereiten zu müssen. Nach der Registrierung für einen der Dienste melden sich Administratoren auf der Website des Dienstes an, um das Programm mit ihrer MDM-Lösung zu verknüpfen. Die von der Organisation erworbenen Geräte können danach über die MDM-Lösung den Benutzern zugewiesen werden. Sicherheit und Schutz sensibler Daten können im Zuge der Gerätekonfiguration erhöht werden, indem geeignete Schutz- und Sicherheitsvorkehrungen eingerichtet werden. Zum Beispiel:

- Benutzer können gezwungen werden, sich im Zuge der Erstkonfiguration bei der Aktivierung des Apple-Geräts im Systemassistenten zu authentifizieren.
- Es kann eine Anfangskonfiguration mit eingeschränktem Zugriff bereitgestellt werden, die für den Zugriff auf sensible Daten eine nachträgliche erweiterte Gerätekonfiguration erforderlich macht.

Nachdem die Zuweisung zu einem Benutzer erfolgte, werden von der MDM spezifizierte Konfigurationen, Einschränkungen und Steuerungen automatisch installiert. Die gesamte Kommunikation zwischen den Geräten und den Apple-Servern wird mit HTTPS (TLS) verschlüsselt.

Darüber hinaus lässt sich der Konfigurationsprozess für Benutzer weiter vereinfachen, indem bestimmte Schritte im Systemassistenten für Geräte entfernt werden, sodass die Benutzer die Einrichtung schnell abschließen und ihre Arbeit aufnehmen können. Administratoren können auch steuern, ob Benutzer das MDM-Profil von ihren Geräten löschen können. Außerdem können sie sicherstellen, dass die Geräteeinschränkungen vom ersten Moment an greifen und über die gesamte Lebensdauer des Geräts verbindlich bleiben. Nachdem ein Gerät aus der Verpackung genommen und aktiviert wurde, kann es in der MDM-Lösung der Organisation registriert werden. Daraufhin werden alle verwalteten Einstellungen, Apps und Bücher in der vom MDM-Administrator vorgegebenen Form installiert.

Apple School Manager, Apple Business Manager und Apple Business Essentials

Apple School Manager, Apple Business Manager und Apple Business Essentials sind für IT-Admins konzipierte Dienste, mit denen sie Apple-Geräte implementieren können, die ihre Organisationen direkt von Apple oder von teilnehmenden, autorisierten Apple-Vertriebspartnern und Anbietern erworben haben.

Mithilfe der MDM-Lösung können Admins den Konfigurationsprozess für Benutzer vereinfachen, Geräteeinstellungen konfigurieren und Apps und Bücher verteilen, die mit diesen drei Diensten erworben werden. Apple School Manager lässt sich mit einem Studierendeninformationssystem (SIS, Student Information System) direkt oder mittels SFTP integrieren. Die Integration der drei Dienste mit Microsoft Azure Active Directory (Azure AD) kann durch die föderierte Anmeldung oder mittels SCIM (System for Cross-domain Identity Management) erfolgen, sodass Admins schnell und einfach Accounts erstellen können.

Apple unterhält Zertifizierungen in Übereinstimmung mit den Standards ISO/IEC 27001 und 27018, um Apple-Kunden zu ermöglichen, ihre gesetzlichen und vertraglichen Verpflichtungen zu erfüllen. Diese Zertifizierungen bieten unseren Kunden eine unabhängige Bescheinigung für Datenschutz- und Informationssicherheitsverfahren von Apple für Systeme im Anwendungsbereich dieser Zertifizierungen. Weitere Informationen sind unter [Sicherheitszertifizierungen für Apple-Internetdienste](#) auf der Website für Zertifizierungen der Apple-Plattformen zu finden.

Hinweis: Weitere Informationen dazu, ob ein Apple-Programm in einem bestimmten Land oder einer bestimmten Region verfügbar ist, enthält der Apple Support-Artikel [Verfügbarkeit von Apple-Programmen und Zahlungsmethoden für Bildungseinrichtungen und Unternehmen](#).

Gerätebetreuung

Betreuung impliziert im Allgemeinen, dass ein Gerät im Besitz der Organisation ist, was die erweiterte Kontrolle und Steuerung der Konfiguration und der Einschränkungen für das Gerät zulässt. Weitere Informationen sind unter [Apple-Gerätebetreuung](#) auf der Website für die Implementierung von Apple-Plattformen zu finden.

Sicherheit bei der Aktivierungssperre

Wie Apple die Aktivierungssperre durchsetzt, hängt davon ab, ob es sich bei dem Gerät um ein iPhone oder iPad, einen Mac mit Apple Chips oder einen Intel-basierten Mac mit Apple T2 Security Chip handelt.

Verhalten bei iPhone und iPad

Bei iPhone- und iPad-Geräten wird die Aktivierungssperre im Zuge des Aktivierungsprozesses mit dem iOS- bzw. iPadOS-Systemassistenten im Anschluss an den Bildschirm für die WLAN-Auswahl erzwungen. Wenn ein Gerät signalisiert, dass es aktiviert wird, sendet es eine Anfrage an einen Apple-Server, um ein Aktivierungszertifikat zu erhalten. Auf Geräten, die durch die Aktivierungssperre gesperrt sind, wird der Benutzer aufgefordert, die iCloud-Anmeldedaten des Benutzers einzugeben, von dem die aktuelle Aktivierungssperre ausgelöst wurde. Der Systemassistent von iOS bzw. iPadOS wird erst fortgesetzt, wenn ein gültiges Zertifikat empfangen wird.

Auf einem Mac mit Apple Chips

Bei einem Mac mit Apple Chips stellt LBB sicher, dass eine gültige LocalPolicy-Datei – eine gültige lokale Richtlinie – für das Gerät vorhanden ist und dass die Nonce-Werte dieser lokalen Richtlinie mit den Werten übereinstimmen, die in der Secure Storage-Komponente gespeichert sind. LLB startet recoveryOS in folgenden Fällen:

- Es gibt keine lokale Richtlinie (LocalPolicy) für das aktuelle macOS.
- Die lokale Richtlinie (LocalPolicy) ist für das jeweilige macOS ungültig.
- Die Nonce-Hash-Werte der lokalen Richtlinie stimmen nicht mit den Hash-Werten überein, die in der Secure Storage-Komponente gespeichert sind.

recoveryOS erkennt, dass der Mac-Computer nicht aktiviert wurde, und stellt die Verbindung zum Aktivierungsserver her, um ein Aktivierungszertifikat zu erhalten. Wenn das Gerät durch die Aktivierungssperre gesperrt ist, wird der Benutzer aufgefordert, die iCloud-Anmeldedaten des Benutzers einzugeben, von dem die aktuelle Aktivierungssperre ausgelöst wurde. Nachdem ein gültiges Aktivierungszertifikat empfangen wurde, wird der Schlüssel dieses Aktivierungszertifikats verwendet, um ein RemotePolicy-Zertifikat, d. h. ein Zertifikat für die „Remote-Richtlinie“ abzurufen. Der Mac-Computer verwendet den LocalPolicy-Schlüssel und das RemotePolicy-Zertifikat, um eine gültige lokale Richtlinie zu erstellen. LLB lässt das Starten von macOS nur zu, wenn eine gültige lokale Richtlinie vorliegt.

Verhalten bei Intel-basierten Mac-Computern

Bei einem Intel-basierten Mac mit T2-Chip verifiziert die Firmware dieses Chips, dass ein gültiges Aktivierungszertifikat vorliegt, bevor das Starten von macOS auf dem Computer erlaubt wird. Die UEFI-Firmware, die durch den T2-Chip geladen wird, ist verantwortlich dafür, den Aktivierungsstatus des Geräts beim T2-Chip zu erfragen und recoveryOS anstelle von macOS zu starten, wenn kein gültiges Aktivierungszertifikat vorliegt. recoveryOS erkennt daraufhin, dass der Mac nicht aktiviert wurde und stellt die Verbindung zum Aktivierungsserver her, um ein Aktivierungszertifikat zu erhalten. Wenn das Gerät durch die Aktivierungssperre gesperrt ist, wird der Benutzer aufgefordert, die iCloud-Anmeldedaten des Benutzers einzugeben, von dem die aktuelle Aktivierungssperre ausgelöst wurde. Die UEFI-Firmware lässt das Starten von macOS nur zu, wenn ein gültiges Aktivierungszertifikat vorliegt.

Verwalteter Modus „Verloren“ und Fernlöschung

Der Modus „Verloren“ wird verwendet, um ein betreutes Gerät, das entwendet wurde, zu orten. Nachdem es geortet wurde, kann es per Fernzugriff gesperrt oder gelöscht werden.

Verwalteter Modus „Verloren“

Wenn ein betreutes iOS- oder iPadOS-Gerät mit iOS 9 (oder neuer) verloren geht oder gestohlen wird, kann der Administrator einer MDM-Lösung (Mobile Device Management) auf dem Gerät per Fernzugriff den Modus „Verloren“ (verwalteter Modus „Verloren“ genannt) aktivieren. Wenn der verwaltete Modus „Verloren“ aktiviert wird, wird der aktuelle Benutzer abgemeldet und das Gerät kann nicht entsperrt werden. Auf dem Bildschirm erscheint eine Mitteilung, die vom Administrator angepasst werden kann. Sie kann beispielsweise eine Telefonnummer für den Finder des Geräts zeigen. Der Administrator kann das Gerät anweisen, seinen aktuellen Standort mitzuteilen (selbst wenn die Ortungsdienste ausgeschaltet sind), und optional einen Ton auf dem Gerät abspielen lassen. Der verwaltete Modus „Verloren“ kann nur durch Admins deaktiviert und beendet werden. Ist dies der Fall, wird der Benutzer mit einer entsprechenden Mitteilung auf dem Sperrbildschirm oder einem Hinweis auf dem Home-Bildschirm darüber informiert.

Fernlöschung

iOS-, iPadOS- und macOS-Geräte können von einem Administrator oder einem Benutzer per Fernzugriff gelöscht werden. (Das sofortige Löschen per Fernzugriff ist nur möglich, wenn FileVault auf dem Mac aktiviert ist.) Das sofortige Löschen per Fernzugriff wird erreicht, indem der Medienschlüssel aus Effaceable Storage (Auslöschbarer Speicher) gelöscht wird, sodass der gesamte Datenbestand nicht mehr gelesen werden kann. Beim Fernlöschen über Microsoft Exchange ActiveSync rückversichert sich das Gerät beim Microsoft Exchange Server, bevor es den Löschvorgang startet.

Wenn die Fernlöschung durch die MDM-Lösung oder über iCloud veranlasst wird, sendet das iPhone-, iPad-, iPod touch- oder Mac-Gerät eine Bestätigung, ehe es den Löschvorgang durchführt.

In den folgenden Situationen ist das Löschen per Fernzugriff nicht möglich:

- Bei Benutzerregistrierung
- Bei Verwendung von Microsoft Exchange ActiveSync, wenn der Account im Zuge der Benutzerregistrierung erstellt wurde
- Bei einem betreuten Gerät, wenn Microsoft Exchange ActiveSync verwendet wird

Benutzer können iOS- und iPadOS-Geräte, die ihnen persönlich gehören, mithilfe der App „Einstellungen“ löschen. Wie bereits erwähnt, können iOS- und iPadOS-Geräte auch so konfiguriert werden, dass sie nach einer bestimmten Anzahl fehlgeschlagener Code-Eingaben den Löschvorgang automatisch einleiten.

Geteiltes iPad – Sicherheit in iPadOS

Die Option „Geteiltes iPad“ ist ein Mehr-Benutzer-Modus für die Implementierung von iPad-Geräten. Sie ermöglicht Benutzern, ein iPad geteilt zu nutzen und die Dokumente und Daten der einzelnen Benutzer zu trennen. Jeder Benutzer erhält einen privaten, reservierten Speicherort, der als APFS (Apple File System)-Volume erstellt und durch die Benutzerdaten geschützt wird. Für die Option „Geteiltes iPad“ ist eine verwaltete Apple-ID erforderlich, die von der Organisation ausgegeben wird und ihrer Kontrolle unterliegt.

Mit „Geteiltes iPad“ haben Benutzer die Möglichkeit, sich an jedem Gerät im Besitz der Organisation anzumelden, das für die Nutzung durch mehrere Personen konfiguriert ist. Die Daten der Benutzer werden in getrennten Verzeichnissen in jeweils eigenen Datensicherheits-Domains gespeichert und durch UNIX-Zugriffsrechte und Sandboxing geschützt. Benutzer von iPadOS 13.4 (oder neuer) können sich für temporäre Sitzungen anmelden. Wenn Benutzer sich von einer temporären Sitzung abmelden, wird ihr APFS-Volume gelöscht und der reservierte Speicherplatz an das System zurückgeführt.

Anmeldung an geteilten iPad-Geräten

Für die Anmeldung an einem geteilten iPad werden native und föderierte verwaltete Apple-IDs unterstützt. Beim erstmaligen Verwenden eines föderierten Accounts wird der Benutzer zum Anmeldeportal des Identitätsanbieters (IdP, Identity Provider) umgeleitet. Nach erfolgreicher Authentifizierung wird ein nur für kurze Zeit gültiger Zugangstoken für die im Hintergrund stehende verwaltete Apple-ID ausgestellt. Danach wird der Anmeldevorgang in ähnlicher Weise wie bei der Anmeldung mittels einer nativen verwalteten Apple-ID fortgesetzt. Nach der Anmeldung wird der Benutzer auf dem geteilten iPad vom Systemassistenten aufgefordert, einen Code (Anmeldedaten) festzulegen. Dieser Code wird für den Schutz der lokalen Daten auf dem Gerät und für die künftige Authentifizierung im Anmeldebildschirm des Geräts verwendet. Wie bei einem Gerät, das nur von einer Person benutzt wird und bei dem sich der Benutzer nur einmal unter Verwendung seiner verwalteten Apple-ID und seines föderierten Accounts anmeldet und sein Gerät anschließend mithilfe seines Codes sperrt, meldet sich bei einem geteilten iPad der Benutzer nur einmal unter Verwendung des föderierten Accounts an und verwendet danach den von ihm festgelegten Code.

Wenn sich Benutzer ohne föderierte Authentifizierung anmelden, wird die verwaltete Apple-ID mithilfe des SRP-Protokolls durch den Apple Identity Service (IDS) authentifiziert. Ist die Authentifizierung erfolgreich, wird ihnen ein nur für kurze Zeit gültiger, gerätespezifischer Zugangstoken zugewiesen. Wenn ein Benutzer das gleiche Gerät bereits zuvor einmal verwendet hat, gibt es für ihn bereits einen lokalen Benutzeraccount, der mit denselben Anmeldedaten entsperrt wird.

Für den Fall, dass ein Benutzer das Gerät noch nie verwendet hat oder für eine temporäre Sitzung angemeldet ist, stellt die Option „Geteiltes iPad“ eine neue UNIX-Benutzerkennung, ein APFS-Volume zum Speichern der persönlichen Benutzerdaten und einen lokalen Schlüsselbund bereit. Da der Speicherplatz für den Benutzer bei der Erstellung des APFS-Volumens zugewiesen (reserviert) wird, steht möglicherweise nicht genügend Speicherplatz zum Erstellen eines neuen Volumens zur Verfügung. In diesem Fall sucht das System nach einem vorhandenen Benutzer, dessen Daten bereits mit der Cloud synchronisiert wurden, und entfernt diesen Benutzer von dem Gerät, damit der neue Benutzer sich anmelden kann. In dem unwahrscheinlichen Fall, dass das Hochladen der Cloud-Daten aller vorhandenen Benutzer noch nicht abgeschlossen ist, schlägt die neue Benutzeranmeldung fehl. Um sich anzumelden, muss der neue Benutzer warten, bis die Datensynchronisierung eines anderen Benutzers abgeschlossen ist oder ein Administrator einen vorhandenen Benutzer-Account zwangsweise löscht, wobei das Risiko eines Datenverlusts besteht.

Wenn das Gerät nicht mit dem Internet verbunden ist (z. B. wenn dem Benutzer kein WLAN-Zugangspunkt zur Verfügung steht), kann die Authentifizierung während einer begrenzten Anzahl von Tagen auf der Basis des lokalen Accounts ausgeführt werden. In diesem Fall können sich aber nur Benutzer anmelden, die bereits einen lokalen Account haben oder eine temporäre Sitzung nutzen. Nach dem Ablauf der Frist müssen sich Benutzer online authentifizieren, auch wenn bereits lokale Accounts für sie vorhanden sind.

Nachdem der lokale Account für einen Benutzer entsperrt oder erstellt und per Fernzugriff authentifiziert wurde, wird der von den Apple-Servern ausgegebene, kurzzeitig gültige Token in einen iCloud-Token umgewandelt, der die Anmeldung bei iCloud erlaubt. Als Nächstes werden die Einstellungen des Benutzers wiederhergestellt und seine Dokumente und Daten von iCloud synchronisiert.

Solange die Sitzung aktiv ist und das Gerät online bleibt, werden erstellte oder geänderte Dokumente und Daten in iCloud gespeichert. Ein Synchronisierungsmechanismus, der im Hintergrund abläuft, sorgt zudem dafür, dass nach dem Abmelden des Benutzers die Änderungen in iCloud oder per NSURLSession-Hintergrundsitzungen in einen anderen Webdienst gepusht werden. Nachdem die Synchronisierung im Hintergrund für den Benutzer abgeschlossen ist, wird der APFS-Datenträger des Benutzers deaktiviert und kann ohne erneute Anmeldung des Benutzers nicht mehr aktiviert werden.

Temporäre Sitzungen werden nicht mit iCloud synchronisiert. Auch wenn Benutzer sich in einer temporären Sitzung bei einem Synchronisierungsdienst anderer Anbieter wie Box oder Google Drive anmelden können, besteht nicht die Möglichkeit, die Synchronisierung nach dem Ende der temporären Sitzung fortzusetzen.

Abmelden von geteilten iPad-Geräten

Wenn sich ein Benutzer auf einem geteilten iPad abmeldet, wird der zugehörige Keybag sofort gesperrt und alle Apps werden geschlossen. Damit sich ein neuer Benutzer schnell anmelden kann, stellt iPadOS einige reguläre Abmeldeaktionen vorübergehend zurück und zeigt dem neuen Benutzer das Anmeldefenster an. Wenn sich ein Benutzer während dieser Zeit (etwa 30 Sekunden) anmeldet, führt das geteilte iPad die zurückgestellte Bereinigung im Zuge der Anmeldung beim Account des nachfolgenden Benutzers durch. Bleibt das geteilte iPad hingegen ungenutzt, löst es die zurückgestellte Bereinigung aus. Während der Bereinigungsphase wird das Anmeldefenster erneut so gestartet, als hätte eine weitere Abmeldung stattgefunden.

Wenn eine temporäre Sitzung beendet wird, führt das geteilte iPad die vollständige Abmeldungssequenz durch und löscht direkt das APFS-Volumen der temporären Sitzung.

Sicherheit bei Apple Configurator

Mit dem flexiblen, sicheren und gerätekonzentrierten Design von Apple Configurator für den Mac können Admins einzelne oder Dutzende iOS-, iPadOS- oder tvOS-Geräte, die über USB an einen Mac angeschlossen sind (oder über Bonjour gekoppelte tvOS-Geräte), vor der Übergabe an die Benutzer schnell und unkompliziert konfigurieren. Apple Configurator für den Mac bietet Admins die Möglichkeit, Software zu aktualisieren, Apps und Konfigurationsprofile zu installieren, Umbenennungen vorzunehmen und den Hintergrund von Geräten zu ändern, Geräteinformationen und Dokumente zu exportieren und viele andere Aktionen auszuführen.

Darüber hinaus kann Apple Configurator für den Mac Mac-Computer mit Apple Chips und mit Apple T2 Security Chip reparieren oder wiederherstellen. Wenn ein Mac auf diese Weise repariert oder wiederhergestellt wird, wird die Datei mit den aktuellsten Updates für die Betriebssysteme (macOS, recoveryOS für Apple Chips oder sepOS für T2) sicher von Apple-Servern geladen und direkt auf dem Mac installiert. Nach der erfolgreichen Reparatur oder Wiederherstellung wird die Datei auf dem Mac, auf dem Apple Configurator ausgeführt wird, gelöscht. Der Benutzer kann zu keiner Zeit die Datei außerhalb von Apple Configurator untersuchen oder verwenden.

Admins haben außerdem die Möglichkeit, mithilfe von Apple Configurator für den Mac oder Apple Configurator für das iPhone Geräte zu Apple School Manager, Apple Business Manager oder Apple Business Essentials hinzuzufügen, selbst wenn diese nicht direkt von Apple, einem von Apple autorisierten Vertriebspartner oder einem autorisierten Mobilfunkanbieter erworben wurden. Wenn Admins ein Gerät konfigurieren, das manuell registriert wurde, verhält sich dieses wie jedes andere Gerät in einem der Dienste, d. h. mit obligatorischer Betreuung und Registrierung in der MDM-Lösung. Bei Geräten, die nicht direkt erworben wurden, wird dem Benutzer eine Frist von 30 Tagen zugestanden, innerhalb der er das Gerät aus einem der Dienste, der Betreuung und der Mobilgeräteverwaltung entfernen kann.

Organisationen können Apple Configurator für den Mac auch verwenden, um iOS-, iPadOS- und tvOS-Geräte, die selbst keinen Zugang zum Internet haben, zu aktivieren. Hierfür müssen die Geräte bei der Konfiguration mit einem als Host fungierenden Mac-Computer mit Internetzugang verbunden werden. Admins können auf diese Weise Geräte wiederherstellen, aktivieren und mit der benötigten Konfiguration (inklusive Apps, Profile und Dokumente) vorbereiten, ohne dass eine Verbindung zum WLAN oder zum Mobilfunknetz hergestellt werden muss. Bei dieser Vorgehensweise ist es Administratoren nicht möglich, vorhandene Anforderungen an die Aktivierungssperre zu umgehen, die bei einer nicht über Kabel erfolgenden Aktivierung üblicherweise erforderlich sind.

Sicherheit bei der Bildschirmzeit

Die Option „Bildschirmzeit“ ist eine integrierte Funktion für die Anzeige und Verwaltung der Zeit, die Erwachsene und Kinder unter anderem mit Apps und Websites verbringen. Es werden zwei Benutzertypen unterschieden: Erwachsene und (verwaltete) Kinder.

Die Option „Bildschirmzeit“ ist keine neue Sicherheitsfunktion des Systems. Dennoch ist es wichtig, zu wissen, wie die Option zum Schutz und zur Sicherheit der Daten beiträgt, die erfasst, gesammelt und zwischen Geräten geteilt werden. Die Option „Bildschirmzeit“ ist in iOS 12 (oder neuer), iPadOS 13.1 (oder neuer), macOS 10.15 (oder neuer) und für einige Funktionen in watchOS 6 (oder neuer) verfügbar.

Die Tabelle unten beschreibt die Hauptfunktionen der Option „Bildschirmzeit“.

Funktion	Unterstütztes Betriebssystem
Nutzungsdaten anzeigen	iOS iPadOS macOS
Zusätzliche Beschränkungen durchsetzen	iOS iPadOS macOS watchOS
Beschränkungen für die Internetnutzung festlegen	iOS iPadOS macOS
Beschränkungen für Apps festlegen	iOS iPadOS macOS watchOS
Auszeiten konfigurieren	iOS iPadOS macOS watchOS

Für Benutzer, die die Nutzung ihrer Geräte selbst verwalten, lassen sich die Steuerung und die Nutzungsdaten der Funktion „Bildschirmzeit“ geräteübergreifend auf allen Geräten, die demselben iCloud-Account zugewiesen sind, mit der Ende-zu-Ende-Verschlüsselung von CloudKit synchronisieren. Voraussetzung hierfür ist, dass für den betreffenden Benutzeraccount die Zwei-Faktor-Authentifizierung aktiviert ist (die Synchronisierung ist standardmäßig ausgeschaltet). „Bildschirmzeit“ ersetzt die Funktionen der Einschränkungen in früheren Versionen von iOS und iPadOS und die Funktion der Kindersicherung in früheren Versionen von macOS.

In iOS 13 oder neuer, iPadOS 13.1 oder neuer und macOS 10.15 oder neuer werden für die Benutzer der Option „Bildschirmzeit“ und die von ihnen verwalteten Kinder die jeweiligen Angaben zur Nutzung über alle Geräte automatisch geteilt, sofern für den jeweiligen iCloud-Account die Zwei-Faktor-Authentifizierung aktiviert ist. Wenn ein Benutzer seinen Safari-Verlauf oder eine App löscht, werden die zugehörigen Nutzungsdaten vom Gerät und von allen synchronisierten Geräten entfernt.

Eltern und Bildschirmzeit

Eltern können die Funktion „Bildschirmzeit“ auf iOS-, iPadOS- und macOS-Geräten einsetzen, um das Nutzungsverhalten ihrer Kinder zu steuern. Der Organisator einer Familie (in der iCloud-Familienfreigabe) kann die Nutzungsdaten der Kinder anzeigen und ihre für die Bildschirmzeit relevanten Einstellungen verwalten. Die Kinder werden informiert, wenn ihre Eltern die Funktion „Bildschirmzeit“ aktivieren, und sie können ihr Nutzungsverhalten auch selbst überwachen. Wenn Eltern die Bildschirmzeit für ihre Kinder aktivieren, legen sie einen Code fest, sodass die Kinder keine Änderungen vornehmen können. Sobald Kinder volljährig werden (wobei das entsprechende Alter abhängig von Land oder Region variiert), können sie diese Überwachung deaktivieren.

Die Nutzungsdaten und Konfigurationseinstellungen werden zwischen den Geräten von Eltern und Kindern unter Verwendung des Protokolls Apple Identity Service (IDS) mit einer Ende-zu-Ende-Verschlüsselung übertragen. Die verschlüsselten Daten können kurzfristig auf den IDS-Servern abgelegt werden, bis sie vom Empfängergerät gelesen wurden (z. B. sobald das iPhone oder iPad oder der iPod touch eingeschaltet wird, wenn es/er ausgeschaltet war). Diese Daten sind für Apple nicht lesbar.

Analysedaten zur Bildschirmzeit

Wenn der Benutzer die Option „iPhone- & Watch-Analyse teilen“ aktiviert, werden ausschließlich die folgenden anonymisierten Daten erfasst, damit Apple einen besseren Einblick erhält, wie die Funktion „Bildschirmzeit“ verwendet wird:

- Wurde die Bildschirmzeit im Systemassistenten oder später in den Einstellungen aktiviert
- Wurde die Nutzung für eine Kategorie geändert, nachdem eine entsprechende Beschränkung festgelegt wurde (innerhalb von 90 Tagen)
- Ist die Funktion „Bildschirmzeit“ aktiviert
- Ist die Auszeit aktiviert
- Wie oft wurde die Anfrage „Mehr Zeit anfordern“ verwendet
- Wie viele App-Beschränkungen gibt es
- Wie oft sahen sich Benutzer die Nutzungsdaten in den Einstellungen der Option „Bildschirmzeit“ an – aufgeschlüsselt nach Benutzer- und Darstellungstyp (lokal, Fernzugriff, Widget)
- Wie oft ignorierten Benutzer eine Beschränkung – aufgeschlüsselt nach Benutzertyp
- Wie oft löschten Benutzer eine Beschränkung – aufgeschlüsselt nach Benutzertyp

Apple erfasst keine speziellen Daten zur Nutzung von Apps oder dem Web. Wird einem Benutzer eine Liste von Apps in den Nutzungsinformationen der Funktion „Bildschirmzeit“ angezeigt, stammen die App-Symbole direkt aus dem App Store, d. h., es werden keinerlei Daten von diesen Anfragen gespeichert.

Glossar

AES (Advanced Encryption Standard) Ein populärer globaler Verschlüsselungsstandard, der zur Verschlüsselung von Daten verwendet wird, um diese privat zu halten.

AES-XTS Ein Modus von AES gemäß IEEE 1619-2007, der für die Verschlüsselung von Speichermedien konzipiert ist.

APFS (Apple File System) Das Standarddateisystem für iOS, iPadOS, tvOS, watchOS und Mac-Computer mit macOS 10.13 (oder neuer). APFS ermöglicht eine starke Verschlüsselung, die gemeinsame Nutzung von Plattenspeicherbereichen, Snapshots (Schnapschüsse) für die Datensicherung und allgemeine Verbesserungen des Dateisystems.

Apple Business Manager Ein webbasiertes Portal, das einfach in der Handhabung ist und IT-Administratoren die Möglichkeit bietet, Apple-Geräte, die ihre Organisation direkt bei Apple oder einem teilnehmenden autorisierten Apple-Vertriebspartner oder Anbieter erworben hat, schnell und effizient bereitzustellen. Geräte können automatisch in der organisationseigenen Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM) registriert werden, ohne dass sie vor der Übergabe an die Benutzer in die Hand genommen und vorbereitet werden müssen.

Apple Identity Service (IDS) Apple-Verzeichnis der öffentlichen iMessage-Schlüssel, APNS-Adressen, Telefonnummern und E-Mail-Adressen, die zur Überprüfung von Schlüsseln und Geräteadressen verwendet werden.

Apple Push-Benachrichtigungsdienst (APNS) Ein globaler Dienst von Apple, der Push-Benachrichtigungen an Apple-Geräte sendet.

Apple School Manager Ein webbasiertes Portal, das einfach in der Handhabung ist und IT-Administratoren die Möglichkeit bietet, Apple-Geräte, die ihre Organisation direkt bei Apple oder einem teilnehmenden autorisierten Apple-Vertriebspartner oder Anbieter erworben hat, schnell und effizient bereitzustellen. Geräte können automatisch in der organisationseigenen Lösung für die Mobilgeräteverwaltung (Mobile Device Management, MDM) registriert werden, ohne dass sie vor der Übergabe an die Benutzer in die Hand genommen und vorbereitet werden müssen.

Apple Security Bounty Eine Belohnung von Apple für Personen, die auf eine Schwachstelle in zuletzt veröffentlichten Betriebssystemen und, sofern relevant, der neuesten Hardware hinweisen.

Autorisierung der Systemsoftware Ein Prozess, der in die Hardware integrierte, kryptografische Schlüssel mit einem Online-Dienst kombiniert, um zu überprüfen, dass im Zuge eines Upgrades nur für die unterstützten Geräte geeignete und legitime Software von Apple bereitgestellt und installiert wird.

Bereitstellungsprofil Eine von Apple signierte Liste von Eigenschaften (.plist-Datei), die Entitäten und Berechtigungen enthält, mit denen Apps auf einem iOS- oder iPadOS-Gerät installiert und geprüft werden können. Ein Bereitstellungsprofil für die Entwicklung führt alle Geräte auf, die der Entwickler für die Ad-Hoc-Verteilung ausgewählt hat. Ein Bereitstellungsprofil für die Verteilung enthält die App-ID einer firmenintern entwickelten App.

Boot Camp Ein Mac-Dienstprogramm, das die Installation von Microsoft Windows auf unterstützten Mac-Computern ermöglicht.

Boot Progress Register (BPR) Eine Reihe von System on Chip (SoC)-Hardwaremarkierungen, die von der Software verwendet werden können, um die Startmodi zu verfolgen, in denen sich das Gerät befindet (z. B. DFU-Modus und Wiederherstellungsmodus). Nachdem eine BPR-Markierung gesetzt wurde, kann sie nicht mehr entfernt werden. Hierdurch erhält spätere Software einen vertrauenswürdigen Indikator für den Status des Systems.

Boot-ROM Der erste Code, den der Prozessor des Geräts am Beginn des Startvorgangs ausführt. Als integraler Bestandteil des Prozessors kann er weder von Apple noch von einem Angreifer modifiziert werden.

CKRecord Ein Verzeichnis mit Schlüssel-Wert-Paaren für Daten, die in CloudKit gespeichert oder von dort abgerufen werden.

Codebasierter Schlüssel (PDK) Der Verschlüsselungsschlüssel, der auf der Verknüpfung des Benutzerpassworts mit dem SKP-Langzeitschlüssel und der UID der Secure Enclave basiert.

Data Vault Ein Mechanismus (vom Kernel durchgesetzt) für den Schutz gegen unautorisierten Zugriff auf Daten, unabhängig davon, ob sich die anfordernde App in der Sandbox befindet oder nicht.

Dateisystemschlüssel Der Schlüssel, mit dem die Metadaten jeder Datei, einschließlich des Klassenschlüssels, verschlüsselt werden. Wird im Effaceable Storage gespeichert und ermöglicht eine schnelle Löschung und dient weniger der Vertraulichkeit.

Datensicherheit Ein Schutzmechanismus für Dateien und Schlüsselbundobjekte auf unterstützten Apple-Geräten. Kann sich auch auf APIs beziehen, die Apps zum Schutz von Dateien und Schlüsselbundobjekten verwenden.

DFU-Modus (Device Firmware Upgrade) Modus, bei dem der Boot-ROM des Geräts darauf wartet, über USB wiederhergestellt zu werden. Der Bildschirm bleibt im DFU-Modus schwarz, bis eine Verbindung zu einem Computer mit iTunes oder dem Finder hergestellt wird. Daraufhin wird die folgende Eingabeaufforderung angezeigt: „iTunes (oder der Finder) hat ein Gerät (iPhone, iPad oder iPod touch) im Wiederherstellungsmodus erkannt. Der Benutzer muss dieses Gerät (iPad, iPhone oder iPod touch) wiederherstellen, damit es mit iTunes (oder dem Finder) verwendet werden kann.“

Direct Memory Access (DMA) Eine Funktion, die Hardwaresubsystemen den direkten Zugriff auf den Hauptspeicher ermöglicht, indem die CPU umgangen wird.

Effaceable Storage (Auslöschbarer Speicher) Ein bestimmter Bereich des NAND-Speichers, in dem kryptografische Schlüssel gespeichert werden und der direkt abgerufen und sicher gelöscht werden kann. Zwar bietet er keinen Schutz, wenn ein Angreifer direkt auf das Gerät zugreifen kann, die Schlüssel im Effaceable Storage können aber als Teil einer Schlüsselhierarchie verwendet werden und so eine schnelle Löschung und Forward Secrecy ermöglichen.

Elliptic Curve Diffie-Hellman Exchange Ephemeral (ECDHE) Ein Austauschmechanismus für Schlüssel, der auf elliptischen Kurven basiert. ECDHE ermöglicht es zwei Parteien, einen geheimen Schlüssel so zu vereinbaren, dass Unbefugte, die Nachrichten zwischen zwei Parteien abfangen, diesen Schlüssel nicht entdecken.

Elliptic Curve Digital Signature Algorithm (ECDSA) Ein Algorithmus für digitale Signaturen, der auf Kryptografie mit elliptischen Kurven basiert.

Enhanced Serial Peripheral Interface (eSPI) Ein All-in-one-Bus, der für die synchrone serielle Kommunikation entwickelt wurde.

Exclusive Chip Identification (ECID) Eine eindeutige 64-Bit-Kennung für jeden Prozessor eines iOS- oder iPadOS-Geräts. Wird ein Anruf auf einem Gerät angenommen, wird das Klingeln auf über iCloud gekoppelten Geräten in der Nähe durch eine kurze Ankündigung via Bluetooth Low Energy (BLE) 4.0 beendet. Die Ankündigungsbytes werden mit derselben Methode wie Handoff-Ankündigungen verschlüsselt. Wird für die Personalisierung verwendet und ist nicht geheim.

Gatekeeper In macOS eine Technologie, die dafür entwickelt wurde, dass auf dem Mac eines Benutzers nur vertrauenswürdige Software ausgeführt wird.

Gruppen-ID (GID) Ähnlich der UID, aber für alle Prozessoren einer Klasse identisch.

Hardware Sicherheitsmodul (HSM) Ein spezieller manipulationssicherer Computer, der digitale Schlüssel schützt und verwaltet.

HMAC Ein Hash-basierter Code für die Nachrichtenauthentifizierung, der auf einer kryptografischen Hash-Funktion basiert.

iBoot Der Stufe-2-Bootloader für alle Apple-Geräte. Code, der im Rahmen des sicheren Startvorgangs XNU lädt. Abhängig von der System on Chip (SoC)-Generation kann iBoot vom Low-Level-Bootloader (LLB) oder direkt vom Boot-ROM geladen werden.

Input/Output Memory Management Unit (IOMMU) Eine Eingabe-Ausgabe-Speicherverwaltungseinheit. Das Subsystem in einem integrierten Chip, das den Zugriff von anderen Eingabe/Ausgabe- und Peripheriegeräten auf den Adressbereich steuert.

Integrierter Schaltkreis (IC) Auch als *Mikrochip* bezeichnet.

Joint Test Action Group (JTAG) Ein Standardwerkzeug für Hardwaredebugging, das von Programmierern und Schaltkreisentwicklern verwendet wird.

Key-Wrapping Verschlüsseln eines ersten Schlüssels mit einem zweiten Schlüssel. iOS und iPadOS verwenden die Verschlüsselungsmethode NIST AES Key Wrapping gemäß RFC 3394 (weitere Informationen sind hier verfügbar <https://www.ietf.org/rfc/rfc3394.txt>; bitte beachte, dass es sich dabei um eine von Dritten betriebene Webseite handelt).

Keybag Eine Datenstruktur, die zum Speichern von Klassenschlüsselsammlungen verwendet wird. Alle Keybag-Arten (User, Device, Backup, Escrow oder iCloud Backup) weisen dasselbe Format auf.

Ein Header mit: Version (eingestellt auf vier in iOS 12 und neueren Versionen), Typ (System, Backup, Escrow oder iCloud Backup), Keybag UUID, HMAC im Falle eines signierten Keybag, zum Verpacken der Klassenschlüssel verwendete Methode – verknüpft mit UID oder PBKDF2 und Anzahl der Salt- und Iteration-Parameter.

Eine Liste der Klassenschlüssel: Schlüssel-UUID; Klasse (Sicherheitsklasse der Datei- oder Schlüsselbunddaten), Verpackungstyp (nur UID-basierter Schlüssel; UID- und codebasierter Schlüssel), verpackter Klassenschlüssel und öffentlicher Schlüssel im Falle asymmetrischer Klassen.

Kryptografische AES-Engine Eine dedizierte Hardwarekomponente zur Implementierung von AES.

Low-Level-Bootloader (LLB) Code, der auf Mac-Computern mit einer zweiphasigen Boot-Architektur beim sicheren Starten aus dem Boot-ROM abgerufen wird und seinerseits iBoot lädt.

Medienschlüssel Teil der Schlüsselhierarchie von Verschlüsselungen, der zum sicheren und sofortigen Löschen beiträgt. Bei iOS, iPadOS, tvOS und watchOS verpackt der Medienschlüssel die Metadaten auf dem Datenvolume (somit ist der Zugriff ohne diesen Schlüssel auf alle pro Datei erzeugten Schlüssel nicht möglich, und es ist auch nicht möglich, auf Dateien zuzugreifen, die mit dieser Datensicherheitstechnologie geschützt werden). Unter macOS verpackt der Medienschlüssel das Verschlüsselungsmaterial, alle Metadaten und die Daten auf dem FileVault-geschützten Volume. In beiden Fällen kann nach dem Löschen des Medienschlüssels nicht mehr auf verschlüsselte Daten zugegriffen werden.

Mobilgeräteverwaltung (Mobile Device Management, MDM) Dienst, mit dem ein Administrator registrierte Geräte per Fernzugriff (remote) verwalten kann. Nach dem Registrieren eines Geräts können Admins den MDM-Dienst über das Netzwerk verwenden, um ohne Benutzerinteraktion Einstellungen zu konfigurieren und andere Aufgaben auf dem Gerät auszuführen.

NAND Nicht flüchtiger Flash-Speicher.

Nonce Eine eindeutige, einmal verwendete Nummer, die in verschiedenen Sicherheitsprotokollen verwendet wird.

Pro Datei erzeugter Schlüssel Der Schlüssel wird von der Datensicherheit verwendet, um eine Datei im Dateisystem zu verschlüsseln. Der pro Datei erzeugte Schlüssel wird mit einem Klassenschlüssel verpackt und in den Metadaten der Datei gespeichert.

Ridge Flow Angle Mapping Mathematische Darstellung der Ausrichtung und Breite der aus einem Fingerabdruck extrahierten Papillarleisten.

Schlüsselbund Die Infrastruktur und eine Sammlung von APIs, die von Apple-Betriebssystemen und Apps anderer Anbieter genutzt werden, um Passwörter, Schlüssel und andere vertrauliche Anmeldedaten zu sichern und abzurufen.

Sealed Key Protection (SKP) Eine Technologie der Datensicherheit, die Verschlüsselungsschlüssel mit Kennzahlen der auf dem System installierten Software und mit Schlüsseln schützt oder *versiegelt*, die nur in der Hardware verfügbar sind (wie beispielsweise die UID der Secure Enclave).

Secure Storage-Komponente Dieser Chip ist mit einem unveränderlichem RO-Code, einem Hardwaregenerator für Zufallszahlen, kryptografischen Engines und einem Mechanismus zum Schutz vor physischer Manipulation ausgestattet. Auf unterstützten Geräten wird die Secure Enclave mit einer Secure Storage-Komponente gekoppelt, um die Anti-Replay-Nonce zu speichern. Zum Lesen und Aktualisieren von Nonces nutzen die Secure Enclave und der Secure-Storage-Chip ein sicheres Protokoll, das den exklusiven Zugriff auf die Nonces sicherstellt. Von dieser Technologie existieren mehrere Generationen mit unterschiedlichen Sicherheitsgarantien.

sepOS Die Firmware der Secure Enclave basiert auf einer von Apple angepassten Version des L4-Mikrokernels.

Software-Seed-Bits Dedizierte Bits in der AES-Engine der Secure Enclave, die an die UID angefügt werden, wenn von der UID Schlüssel generiert werden. Jedes Software-Seed-Bit verfügt über ein entsprechendes Sperrbit. Boot-ROM und Betriebssystem der Secure Enclave können den Wert jedes Software-Seed-Bits unabhängig ändern, sofern nicht das zugehörige Sperrbit gesetzt wurde. Wurde das Sperrbit gesetzt, können weder das Software-Seed-Bit noch das Sperrbit geändert werden. Die Software-Seed-Bits und die zugehörigen Sperrbits werden zurückgesetzt, wenn die Secure Enclave neu gestartet wird.

Speichercontroller Das Subsystem in einem System on Chip (SoC), das die Schnittstelle zwischen dem SoC und seinem Hauptspeicher steuert.

Speicherverwürfelung (Address Space Layout Randomization, ASLR) Eine von Betriebssystemen verwendete Technik, die das erfolgreiche Ausnutzen von Softwarebugs erschwert. Da Speicheradressen und Segmentierung nicht vorhergesagt werden können, kann Exploit-Code diese Werte nicht hart codieren.

SSD-Controller Ein Hardware subsystem zur Verwaltung der Speichermedien (Solid-State-Laufwerk).

System on Chip (SoC) Integrierter Schaltkreis (IC), der mehrere Komponenten in einem einzigen Chip zusammenfasst. Der Anwendungsprozessor, die Secure Enclave und andere Coprozessoren sind Komponenten des SoC.

System-Coprozessor-Integritätsschutz (SCIP) Ein Mechanismus von Apple, der Modifikationen der Coprozessor-Firmware verhindert.

Unified Extensible Firmware Interface (UEFI) Firmware Eine technologische Verbesserung für BIOS, die bewirkt, dass die Firmware mit dem Betriebssystem eines Computers verbunden wird.

Uniform Resource Identifier (URI) Eine Zeichenkette, mit der webbasierte Ressourcen identifiziert werden können.

Unique ID (UID) Ein AES 256-Bit-Schlüssel, der bei der Fertigung in den Prozessor eingebrannt wird. Er kann weder von der Firmware noch von der Software gelesen werden und wird nur von der AES-Engine der Prozessorhardware verwendet. Um den eigentlichen Schlüssel zu erhalten, müsste ein potenzieller Angreifer einen extrem komplexen und kostenintensiven physischen Angriff auf den Chip des Prozessors ausführen. Die UID ist mit keiner anderen Kennung des Geräts verbunden. Dazu zählt auch die UDID, ohne darauf beschränkt zu sein.

Verknüpfung Verfahren, mit dem der Code eines Benutzers in einen kryptografischen Schlüssel konvertiert wird, der mit der UID des Geräts zusätzlich gesichert wird. Dieser Prozess stellt sicher, dass Brute-Force-Angriffe auf dem jeweiligen Gerät durchgeführt werden müssen, sodass sie nicht von mehreren Geräten gleichzeitig ausgeführt werden können. Für die Verknüpfung wird der Algorithmus PBKDF2 benutzt, der AES mit der UID des Geräts als pseudozufällige Funktion (PRF) für jede Iteration verwendet.

Wiederherstellungsmodus Dieser Modus wird für die Wiederherstellung vieler Apple-Geräte genutzt, damit der Benutzer das Betriebssystem neu installieren kann, falls es nicht erkannt wird.

xART Eine Abkürzung für eXtended Anti-Replay Technology. Eine Reihe von Diensten, die der Secure Enclave verschlüsselten, permanenten Speicher mit Anti-Replay-Funktionen auf Basis der physischen Speicherarchitektur bereitstellen. Weitere Informationen findest du im Abschnitt der Secure Storage-Komponente.

XNU Der Kernel im Zentrum der Apple-Betriebssysteme. Er wird als vertrauenswürdig eingestuft und setzt Sicherheitsmaßnahmen wie die Codesignierung, das Ausführen in einer Sandbox, das Überprüfen von Berechtigungen und Address Space Layout Randomization (ASLR) durch.

XProtect In macOS eine integrierte Antivirustechnologie für die signaturbasierte Erkennung und Entfernung von Schadprogrammen.

Dokumentrevisionen

Dokumentrevisionen

Datum	Zusammenfassung
Dezember 2022	<p>Hinzugefügte Themen:</p> <ul style="list-style-type: none">• Erweiterter Datenschutz für iCloud <p>Aktualisierte Themen:</p> <ul style="list-style-type: none">• iCloud-Sicherheit – Übersicht• iCloud-Verschlüsselung• Sicherheit bei iCloud Backup• Sicherheit beim Kontakt für die Accountwiederherstellung• Sicherheit beim Nachlasskontakt

Datum	Zusammenfassung
Mai 2022	<p>Aktualisiert für:</p> <ul style="list-style-type: none"> • iOS 15.4 • iPadOS 15.4 • macOS 12,3 • tvOS 15.4 • watchOS 8.5 <p>Hinzugefügte Themen:</p> <ul style="list-style-type: none"> • Einschränkungen für das gekoppelte recoveryOS • Local Operating System Version (love) • Teilen mit Health • Sicherheit beim Kontakt für die Accountwiederherstellung • Sicherheit beim Nachlasskontakt • Sicherheit der Funktion „Tap to Pay“ auf dem iPhone • Zugriff mit Apple Wallet • Zugangsdatentypen • Ausweisdokumente in Apple Wallet • Siri-fähige HomeKit-Geräte <p>Aktualisierte Themen:</p> <ul style="list-style-type: none"> • Magic Keyboard mit Touch ID • Face ID, Touch ID, Codes und Passwörter • Sicherheit beim Gesichtsabgleich • Express-Karten mit Energiereserve • Startmodi eines Mac mit Apple Chips • Inhalt einer LocalPolicy-Datei für Mac-Computer mit Apple Chips • Sicherheit des Signed System Volume in iOS, iPadOS und macOS • Systemsicherheit bei watchOS • Apple Security Research Device • Rolle von APFS (Apple File System) • App-Zugriff auf Benutzerdaten schützen • Sicherheit bei Apps für macOS – Einführung • Schutz vor Malware bei macOS • iCloud-Sicherheit – Übersicht • Sichere Schlüsselbundsynchronisierung • Sichere Wiederherstellung des iCloud-Schlüsselbunds • Kartenzahlung mit Apple Pay • Kontaktlose Ausweise in Apple Pay • Unbrauchbarmachung von Karten mit Apple Pay • Sicherheit bei der Beantragung einer Apple Card • Sicherheit bei Apple Cash • Hinzufügen von ÖPNV- und eMoney-Karten zu Apple Wallet • Sicherheit bei Apple Messages for Business • FaceTime-Sicherheit • Sicherheit der Funktion „Autoschlüssel“ in iOS • Sicherheit bei Apple Configurator <p>Entfernte Themen:</p> <ul style="list-style-type: none"> • HomeKit-Zubehör und iCloud

Datum	Zusammenfassung
Mai 2021	<p data-bbox="812 212 954 233">Aktualisiert für:</p> <ul data-bbox="812 243 948 405" style="list-style-type: none"><li data-bbox="812 243 911 264">• iOS 14.5<li data-bbox="812 275 948 296">• iPadOS 14.5<li data-bbox="812 306 943 327">• macOS 11,3<li data-bbox="812 338 927 359">• tvOS 14.5<li data-bbox="812 369 948 390">• watchOS 7.4 <p data-bbox="812 415 1024 436">Hinzugefügte Themen:</p> <ul data-bbox="812 447 1373 575" style="list-style-type: none"><li data-bbox="812 447 1110 468">• Magic Keyboard mit Touch ID.<li data-bbox="812 478 1373 499">• Gesicherte Absicht und Verbindungen zur Secure Enclave.<li data-bbox="812 510 1279 531">• Automatisches Entsperrn mit der Apple Watch.<li data-bbox="812 541 1211 562">• CustomOS Image4 Manifest Hash (coih). <p data-bbox="812 585 1003 606">Bearbeitete Themen:</p> <ul data-bbox="812 617 1451 793" style="list-style-type: none"><li data-bbox="812 617 1451 667">• Zwei neue Transaktionen für den Expressmodus in Express-Karten mit Energiereserve hinzugefügt.<li data-bbox="812 678 1284 699">• Secure Enclave – Funktionsübersicht bearbeitet.<li data-bbox="812 709 1393 760">• Softwareaktualisierungsinhalte zu Secure Multi-Boot (smb3) hinzugefügt.<li data-bbox="812 770 1435 791">• Zusätzliche Inhalte für Sealed Key Protection (SKP) hinzugefügt.

Datum	Zusammenfassung
Februar 2021	<p>Aktualisiert für:</p> <ul style="list-style-type: none"> • iOS 14.3 • iPadOS 14.3 • macOS 11,1 • tvOS 14.3 • watchOS 7.2 <p>Hinzugefügte Themen:</p> <ul style="list-style-type: none"> • Speichersichere iBoot-Implementierung • Startvorgang von einem Mac mit Apple Chips • Startmodi eines Mac mit Apple Chips • Sicherheitsrichtlinie „Startup Disk“ bei Mac-Computern mit Apple Chips • Erstellung und Verwaltung von Signaturschlüsseln für die lokale Richtlinie (LocalPolicy) • Inhalt einer LocalPolicy-Datei für Mac-Computer mit Apple Chips • Sicherheit des Signed System Volume in iOS, iPadOS und macOS • Apple Security Research Device • Passwortüberwachung • IPv6-Sicherheit • Sicherheit der Funktion „Autoschlüssel“ in iOS <p>Aktualisierte Themen:</p> <ul style="list-style-type: none"> • Secure Enclave • Hardwareabschaltung des Mikrofons • recoveryOS und Diagnoseumgebungen bei Intel-basierten Mac-Computern • DMA-Schutz bei Mac-Computern • Kernel-Erweiterungen bei macOS • Systemintegritätsschutz • Systemsicherheit bei watchOS • Verwalten von FileVault in macOS • Zugriff durch Apps auf gesicherte Passwörter • Sicherheitsempfehlungen für Passwörter • Sicherheit bei Apple Cash • Sicherheit bei Apple Messages for Business • Datenschutz bei WLAN-Verbindungen • Sicherheit bei der Aktivierungssperre • Sicherheit bei Apple Configurator

Datum	Zusammenfassung
April 2020	<p>Aktualisiert für:</p> <ul style="list-style-type: none"> • iOS 13.4 • iPadOS 13.4 • macOS 10.15.4 • tvOS 13.4 • watchOS 6.2 <p>Aktualisierungen:</p> <ul style="list-style-type: none"> • iPad-Mikrofonabschaltung zu Hardwareabschaltung des Mikrofons hinzugefügt. • Data Vaults zu App-Zugriff auf Benutzerdaten schützen hinzugefügt. • Aktualisierungen in Verwalten von FileVault in macOS und an Befehlszeilenprogrammen vorgenommen. • MRT (Werkzeug zum Entfernen von Malware) in Schutz vor Malware bei macOS ergänzt. • Aktualisierungen in Sicherheit in iPadOS für geteilte iPad-Geräte vorgenommen.
Dezember 2019	<p>Zusammenführung der Dokumente „iOS-Sicherheit – Handbuch“, „macOS-Sicherheit – Übersicht“ und „T2 Security Chip – Übersicht“.</p> <p>Aktualisiert für:</p> <ul style="list-style-type: none"> • iOS 13.3 • iPadOS 13.3 • macOS 10.15.2 • tvOS 13.3 • watchOS 6.1.1 <p>Datenschutzeinstellungen, Siri und Siri-Vorschläge sowie ITP in Safari (Intelligent Tracking Prevention) wurden entfernt. Aktuelle Informationen über diese Funktionen sind unter https://www.apple.com/de/privacy/ zu finden.</p>
Mai 2019	<p>Aktualisiert für iOS 12.3</p> <ul style="list-style-type: none"> • Unterstützung für TLS 1.3 • Überarbeitete Beschreibung der Sicherheit bei AirDrop • DFU-Modus und Wiederherstellungsmodus • Codeanforderungen für Zubehörverbindungen
November 2018	<p>Aktualisiert für iOS 12.1</p> <ul style="list-style-type: none"> • FaceTime-Gruppe

Datum	Zusammenfassung
September 2018	Aktualisiert für iOS 12 Secure Enclave <ul style="list-style-type: none"> • OS-Integritätsschutz • Express-Karten mit Energiereserve • DFU-Modus und Wiederherstellungsmodus • HomeKit TV Remote-Zubehör • Kontaktlose Ausweise • Studierendenausweise • Siri-Vorschläge • Kurzbefehle in Siri • App „Kurzbefehle“ • Verwaltung von Benutzerpasswörtern • Bildschirmzeit • Sicherheitszertifizierungen und -programme
Juli 2018	Aktualisiert für iOS 11.4 <ul style="list-style-type: none"> • Biometrische Richtlinien • HomeKit • Apple Pay • Geschäftschat • Nachrichten in iCloud • Apple Business Manager
Dezember 2017	Aktualisiert für iOS 11.2 <ul style="list-style-type: none"> • Apple Pay Cash
Oktober 2017	Aktualisiert für iOS 11.1 <ul style="list-style-type: none"> • Sicherheitszertifizierungen und -programme • Touch ID/Face ID • Geteilte Notizen • Ende-zu-Ende-Verschlüsselung in CloudKit • TLS-Aktualisierung • Apple Pay, Mit Apple Pay im Web zahlen • Siri-Vorschläge • Geteiltes iPad
Juli 2017	Aktualisiert für iOS 10.3 <ul style="list-style-type: none"> • Secure Enclave • Sicherheit von Dateidaten • Keybags • Sicherheitszertifizierungen und -programme • SiriKit • HealthKit • Netzwerksicherheit • Bluetooth • Geteiltes iPad • Modus „Verloren“ • Aktivierungssperre • Datenschutzeinstellungen

Datum	Zusammenfassung
März 2017	Aktualisiert für iOS 10 Systemsicherheit <ul style="list-style-type: none"> • Datensicherheitsklassen • Sicherheitszertifizierungen und -programme • HomeKit, ReplayKit, SiriKit • Apple Watch • WLAN, VPN • Gesamtauthentifizierung (Single Sign-On) • Apple Pay, Mit Apple Pay im Web zahlen • Bereitstellung von Kredit-, Debit- und Prepaid-Karten • Safari-Vorschläge
Mai 2016	Aktualisiert für iOS 9.3 <ul style="list-style-type: none"> • Verwaltete Apple-ID • Zwei-Faktor-Authentifizierung für Apple-IDs • Keybags • Sicherheitszertifizierungen • Modus „Verloren“, Aktivierungssperre • Geschützte Notizen • Apple School Manager • Geteiltes iPad
September 2015	Aktualisiert für iOS 9 Aktivierungssperre für Apple Watch <ul style="list-style-type: none"> • Coderichtlinien • API-Unterstützung für Touch ID • Datenschutz bei A8 per AES-XTS • Keybags für unbeaufsichtigte Softwareaktualisierungen • Aktualisierungen für Zertifizierung • Modell der Vertrauenswürdigkeit von Unternehmens-Apps • Datenschutz für Safari-Lesezeichen • Transportsicherheit für Apps • VPN-Spezifikation • iCloud Remote-Zugriff für HomeKit • Kunden-/Bonuskarten für Apple Pay, App des Kartenausstellers für Apple Pay • Gerätebasierte Indexierung für Spotlight • iOS-Kopplungsmodell • Apple Configurator 2 • Einschränkungen

© 2022 Apple Inc. Alle Rechte vorbehalten.

Die Verwendung des über die Tastatur erzeugten Apple-Logos (Option-Umschalt-+) für kommerzielle Zwecke ohne vorherige schriftliche Einwilligung von Apple kann als Markenmissbrauch und unlauterer Wettbewerb gerichtlich verfolgt werden.

Apple, das Apple-Logo, AirDrop, AirPlay, Apple Books, Apple Card, Apple Music, Apple Pay, Apple TV, Apple Wallet, Apple Watch, AppleScript, ARKit, Bonjour, Boot Camp, CarPlay, Face ID, FaceTime, FileVault, Finder, FireWire, Handoff, HealthKit, HomeKit, HomePod, HomePod mini, iMac, iMac Pro, iMessage, iPad, iPadOS, iPad Air, iPad Pro, iPhone, iPod touch, iTunes, Keychain, Lightning, Mac, Mac Catalyst, Mac mini, Mac Pro, MacBook, MacBook Air, MacBook Pro, macOS, Magic Keyboard, Objective-C, OS X, QuickType, Retina, Rosetta, Safari, Siri, Siri Remote, SiriKit, Swift, Spotlight, Touch ID, TrueDepth, tvOS, watchOS und Xcode sind Marken der Apple Inc., die in den USA und weiteren Ländern und Regionen eingetragen sind.

App Clips, Find My und Touch Bar sind Marken der Apple Inc.

App Store, AppleCare, CloudKit, iCloud, iCloud Drive, iCloud Keychain und iTunes Store sind Dienstleistungsmarken der Apple Inc., die in den USA und weiteren Ländern und Regionen eingetragen sind.

Apple Messages for Business ist eine Dienstleistungsmarke der Apple Inc.

Apple
One Apple Park Way
Cupertino, CA 95014
apple.com

IOS ist eine Marke oder eingetragene Marke von Cisco in den USA und weiteren Ländern und wird in Lizenz verwendet.

Die Bluetooth®-Wortmarke und -Logos sind eingetragene Marken der Bluetooth SIG, Inc. und die Verwendung solcher Marken durch Apple erfolgt in Lizenz.

Java ist eine eingetragene Marke von Oracle und/oder ihren Tochtergesellschaften.

UNIX® ist eine eingetragene Marke von The Open Group.

Andere hier genannte Produkt- und Herstellernamen sind möglicherweise Marken ihrer jeweiligen Rechtsinhaber.

Es wurden alle Anstrengungen unternommen, um die Richtigkeit der Informationen in diesem Handbuch sicherzustellen. Apple übernimmt keine Verantwortung für Druck- oder Schreibfehler.

Einige Apps sind nicht in allen Regionen verfügbar. Änderungen bei der Verfügbarkeit von Apps sind vorbehalten.

D028-00625