Apple 平台安全保护

2022 年 5 月

目录

Apple 平台安全保护

Apple 平台安全保护介绍

Apple 平台以安全性为设计核心。Apple 利用在打造优秀移动操作系统方面所积累的丰富经验，创建了可满足移动设备、手表、台式设备以及家居设备独特要求的安全性架构。

硬件、**软件**和**服务**在每台 Apple 设备上紧密联系、共同协作，旨在为用户提供最高的安全性和透明的使用体验，从而实现保护个人信息安全这一终极目标。例如，Apple 设计的芯片和安全性硬件为关键的安全性功能提供了支持。其次，软件保护为操作系统和第三方 App 提供了安全保障。最后，服务提供了安全且及时的软件更新机制，帮助建立了受保护的 App 生态系统，并保障了通信和支付的安全。因此，Apple 设备不仅保护设备及其数据，还保护整个生态系统，包括用户在本地、网络上以及使用互联网核心服务执行的所有操作。

我们设计的产品在保持简单、直观和强大的同时，也很安全。而基于硬件的设备加密等关键的安全性功能不会因为误操作而被停用。面容 ID 和触控 ID 等其他功能让设备安全防护变得更简单直观，从而增强了用户体验。由于很多安全性功能在默认情况下均处于启用状态，因此用户或 IT 部门无需执行大量的配置操作。

本文详细介绍了安全性技术和功能如何在 Apple 平台中得以实现，同时帮助组织将 Apple 平台安全保护技术和功能与其自身的政策和规程相结合，从而满足组织的特定安全性需求。

内容主要分为以下几个主题：

- **硬件安全性和生物识别：**即构成 Apple 设备安全性基础的芯片和硬件，包括 Apple 芯片、安全隔区、加密引擎、面容 ID 和触控 ID
- **系统安全性：**为 Apple 操作系统的安全启动、更新和持续运行而提供的集成硬件和软件功能
- **加密和数据保护：**在设备丢失或被盗，或有未授权人员或进程尝试使用或修改用户数据时，能够保护设备上用户数据的架构和设计
- **App 安全性：**提供安全的 App 生态系统并确保 App 安全运行且不会破坏平台完整性的软件和服务
- **服务安全性：**用于身份认证、密码管理、支付、通信以及查找丢失设备的 Apple 服务
- **网络安全性：**针对传输中的数据提供安全认证和加密的行业标准联网协议
- **开发者套件安全性：**供第三方 App 安全私密地管理家庭和健康以及扩展 Apple 设备和服务功能的框架“套件”
- **安全设备管理：**允许对 Apple 设备进行管理、帮助防止未经授权的使用以及在设备丢失或被盗时启用远程擦除的方法

安全性承诺

Apple 致力于使用领先的隐私和安全性技术来帮助客户保护其个人信息，以及采用全面的方法来保护商业环境中企业的数据。Apple 会以发放 Apple 安全性奖金的方式奖励发现漏洞的研究人员。有关计划和奖金类别的详细信息，请访问 https://developer.apple.com/security-bounty/。

我们拥有一支专门的安全团队，负责为所有 Apple 产品提供支持。该团队为开发中和已发布的产品提供安全审核和测试。这个 Apple 团队还提供安全工具和培训，并主动监控新增安全问题的威胁和报告。Apple 是事件响应与安全组织论坛 (FIRST) 的成员。

Apple 会在安全性和隐私领域继续探索更多可能。Apple 在多条产品线中使用定制芯片，从 Apple Watch、iPhone 和 iPad 到搭载 T2 安全芯片和 Apple 芯片的 Mac，在提供高效计算的同时，又兼顾了安全性。例如，Apple 芯片是安全启动、面容 ID 和触控 ID 以及数据保护的基础。另外，设备上由 Apple 芯片驱动的安全功能（如内核完整性保护、指针认证代码和快速权限访问限制）有助于阻止常见类型的网络攻击。因此，即使攻击者代码通过某些方式得到执行，其可造成的损害也会显著降低。

为了充分利用 Apple 平台内置的大量安全性功能，我们鼓励组织审视自身的 IT 和安全策略，以确保充分利用这些平台提供的多重安全技术。

要进一步了解如何向 Apple 报告问题以及如何订阅安全通知，请参阅报告安全性或隐私漏洞。

Apple 坚信隐私是一项基本人权，因而内建了一系列控制和选项，可让用户决定 App 如何使用其信息，何时使用其信息以及使用何种信息。若要进一步了解 Apple 针对隐私所采取的措施、Apple 设备的隐私控制以及 Apple 隐私政策，请访问 https://www.apple.com/cn/privacy**。**

【注】除非另有说明，本文涵盖以下操作系统版本：iOS 15.4、iPadOS 15.4、macOS 12.3、Apple tvOS 15.4 和 watchOS 8.5。

硬件安全性和生物识别

硬件安全性概览

为了确保安全性，软件必须安装在具有内建安全性的硬件上。正因为如此，运行 iOS、iPadOS、macOS、Apple tvOS 和 watchOS 的 Apple 设备的芯片内设安全性功能，包括实现系统安全性功能的 CPU 以及专用于提供安全性功能的其他芯片。注重安全性的硬件遵循支持有限且单独定义的功能这一准则，以使攻击面最小化。此类组件包括形成安全启动硬件信任根的 Boot ROM、用于高效且安全加密和解密的专用 AES 引擎以及安全隔区。**安全隔区**是一个片上系统 (SoC)，内置于所有新款的 iPhone、iPad、Apple Watch、Apple TV 和 HomePod 设备以及搭载 Apple 芯片和搭载 Apple T2 安全芯片的 Mac 上。安全隔区本身遵循与 SoC 相同的设计准则，自身拥有独立的 Boot ROM 和 AES 引擎。安全隔区还为加密静态数据所需密钥的安全生成和储存提供了基础，同时还保护和评估面容 ID 和触控 ID 的生物识别数据。

储存加密必须快速高效，但与此同时不能透露其用于建立加密密钥关系的数据 (或**密钥材料**)。AES 硬件引擎通过在**读写文件时**执行快速的嵌入式加密和解密解决了这个问题。安全隔区中的特殊通道向 AES 引擎提供所需的密钥材料，而不将此信息透露给应用程序处理器 (或 CPU) 或者整个操作系统。这有助于确保 Apple 的数据保护和文件保险箱技术在保护用户文件的同时，不会透露长期有效的加密密钥。

Apple 所设计的安全启动可保护软件的最底层不被篡改，并且仅允许来自 Apple 的受信任操作系统软件在启动时加载。安全启动在不可更改的代码 (称为 Boot ROM，在 Apple SoC 制造阶段植入，也称为**硬件信任根**) 中开始执行。在搭载 T2 芯片的 Mac 电脑上，macOS 安全启动的信任植根在 T2 中。(T2 芯片和安全隔区也均会使用各自独立的 Boot ROM 执行自己的安全启动进程，这准确地模拟了 A 系列芯片和 M1 系列芯片安全启动的方式。)

安全隔区还会处理 Apple 设备中分别来自面容 ID 和触控 ID 传感器的面部和指纹数据。这样做既提供了安全认证，同时又保护了用户生物识别数据的隐私和安全，还可让用户享有较长及较复杂密码带来的安全性，以及在许多情形下访问或购买时快速认证的便利性。

Apple SoC 安全性

Apple 设计的芯片构建了适用于所有 Apple 产品的通用架构，现搭载于 Mac 以及 iPhone、iPad、Apple TV 和 Apple Watch 上。十多年来，Apple 的世界级芯片设计团队一直致力于构建并完善 Apple 片上系统 (SoC)。最终为所有设备设计出了可扩展式架构，其安全性功能处于行业领先水平。只有能够自行研制芯片来与自身研发的软件相协同的公司，才能为各项安全性功能构建这一共同基础。

Apple 芯片经过专门设计和制造，用于实现以下详述的这些系统安全性功能：

功能	A10	A11、S3	A12、S4	A13、S5	A14、A15、S6、S7	M1 系列
内核完整性保护	✓	✓	✓	✓	✓	✓
快速权限访问限制		✓	✓	✓	✓	✓
系统协处理器完整性保护			✓	✓	✓	✓
指针认证代码			✓	✓	✓	✓
页面保护层		✓	✓	✓	✓	请参阅下方备注。

【注】页面保护层 (PPL) 要求平台**仅**执行受信任的签名代码；这种安全模型在 macOS 中不适用。

Apple 设计的芯片还专门实现了以下详述的这些“数据保护”功能。

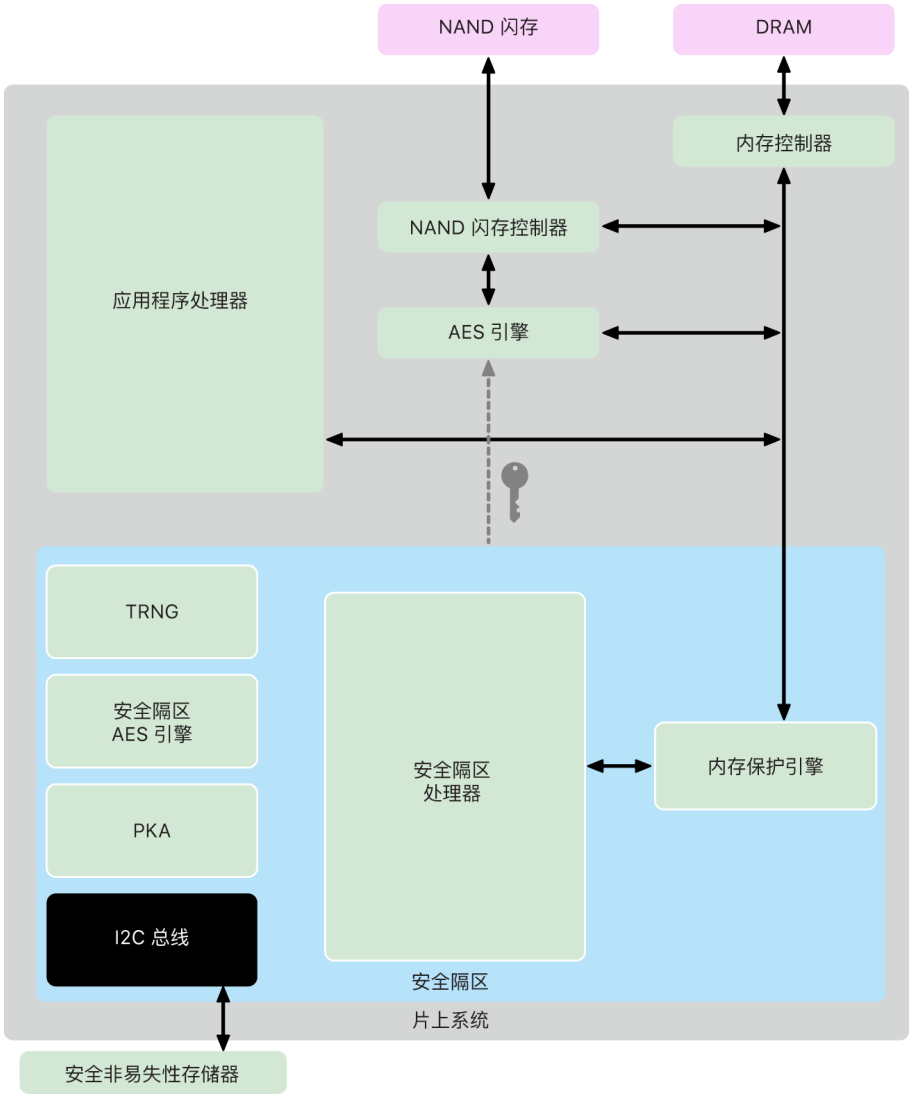
功能	A10	A11、S3	A12、S4	A13、S5	A14、A15、S6、S7、M1 系列
密封密钥保护 (SKP)	✓	✓	✓	✓	✓
recoveryOS - 所有数据保护类受保护	✓	✓	✓	✓	✓
DFU 备选启动模式、诊断和更新 - A 类、B 类和 C 类数据受保护			✓	✓	✓

安全隔区

安全隔区是最新版本的 iPhone、iPad、iPod touch、Mac、Apple TV、Apple Watch 和 HomePod 的专用安全子系统。

概览

安全隔区是集成到 Apple 片上系统 (SoC) 的专用安全子系统。安全隔区独立于主处理器，可提供额外的安全保护，即使应用程序处理器内核遭到入侵，也可保护敏感用户数据的安全。其遵循与 SoC 相同的设计准则：Boot ROM 用于建立硬件信任根，AES 引擎用于高效安全的加密操作以及内存受保护。虽然安全隔区不含储存设备，但它拥有一套将信息安全储存在所连接储存设备上的机制，该储存设备与应用程序处理器和操作系统使用的 NAND 闪存互相独立。



安全隔区是大多数版本的 iPhone、iPad、Mac、Apple TV、Apple Watch 和 HomePod 的硬件功能，支持的版本包括：

- iPhone 5s 或后续机型
- iPad Air 或后续机型
- 配备触控栏且搭载 Apple T1 芯片的 MacBook Pro 电脑（2016 年款和 2017 年款）
- 基于 Intel 芯片且搭载 Apple T2 安全芯片的 Mac 电脑
- 搭载 Apple 芯片的 Mac 电脑
- Apple TV HD 或后续机型
- Apple Watch Series 1 或后续机型
- HomePod 和 HomePod mini

安全隔区处理器

安全隔区处理器为安全隔区提供了主要的计算能力。为了提供最高级别的隔离，安全隔区处理器专供安全隔区使用。这可帮助阻止旁路攻击的发生，这种攻击依赖于恶意软件使用与被攻击目标软件相同的执行核。

安全隔区处理器运行 Apple 定制版的 L4 微内核。安全隔区处理器设计为以较低时钟速度高效运行，这有助于防范时钟攻击和功耗攻击。从 A11 和 S4 开始，安全隔区处理器包括内存受保护的引擎和具备反重放功能的加密内存、安全启动、专用的随机数生成器和独有的 AES 引擎。

内存保护引擎

安全隔区从设备 DRAM 内存的专用区域运行。多层保护将由安全隔区保护的内存与应用程序处理器隔离。

当设备启动时，安全隔区 Boot ROM 会为内存保护引擎生成随机临时内存保护密钥。每当安全隔区写入到其专用内存区域时，内存保护引擎就会在 Mac XEX (xor-encrypt-xor) 模式中使用 AES 加密内存块，并为内存计算基于密码的消息认证码 (CMAC) 认证标签。内存保护引擎会将认证标签与加密内存一同储存。当安全隔区读取内存时，内存保护引擎会验证认证标签。如果认证标签匹配，内存保护引擎就会解密内存块。如果标签不匹配，内存保护引擎会向安全隔区报告错误。发生内存认证错误后，在系统重新启动前安全隔区会停止接受请求。

从 Apple A11 和 S4 SoC 开始，内存保护引擎为安全隔区内存增加了重放保护。为了帮助防止重放安全性要求极高的数据，内存保护引擎会将内存块的唯一的一次性数字（称为**随机数**）与认证标签一同储存。随机数为 CMAC 认证标签提供了额外保护。所有内存块的随机数受植根于安全隔区内专用 SRAM 中完整性树的保护。发生写入操作时，内存保护引擎会**更新**随机数以及完整性树自 SRAM 起向下的的每一层。发生读取操作时，内存保护引擎会**验证**随机数以及完整性树自 SRAM 起向下的的每一层。随机数不匹配的处理方法与认证标签不匹配的处理方法类似。

在 Apple A14、A15、M1 系列及后续型号的 SoC 上，内存保护引擎支持两组临时内存保护密钥。第一组密钥用于安全隔区独有的数据，第二组密钥用于与安全神经网络引擎共享的数据。

内存保护引擎以内联方式运行且对安全隔区透明。安全隔区将内存当作普通的未加密 DRAM 一样进行读取和写入，而安全隔区外的观察程序只能看到加密和认证版本的内存。这样做的结果是既提供了强大的内存保护，又未牺牲性能或增加软件复杂度。

安全隔区 Boot ROM

安全隔区包括一个专用的安全隔区 Boot ROM。与应用程序处理器 Boot ROM 类似，安全隔区 Boot ROM 也是不可更改的代码，用于为安全隔区建立硬件信任根。

系统启动时，iBoot 会给安全隔区分配一个专用的内存区域。在使用内存前，安全隔区 Boot ROM 就会初始化内存保护引擎，为由安全隔区保护的内存提供加密保护。

应用程序处理器随后将 sepOS 映像发送给安全隔区 Boot ROM。将 sepOS 映像拷贝到由安全隔区保护的内存后，安全隔区 Boot ROM 会检查映像的加密哈希值和签名，以验证 sepOS 已获授权来在设备上运行。如果 sepOS 映像已正确签名以在设备上运行，安全隔区 Boot ROM 会将控制转移给 sepOS。如果签名无效，安全隔区 Boot ROM 会设计为在下一次芯片还原前阻止进一步使用安全隔区。

在 Apple A10 及后续型号的 SoC 上，安全隔区 Boot ROM 将 sepOS 的哈希值锁定到专用于此目的的寄存器。公钥加速器会为操作系统绑定 (OS-bound) 密钥使用此哈希值。

安全隔区启动监视器

在 Apple A13 及后续型号的 SoC 上，安全隔区包括一个启动监视器，旨在确保所启动 sepOS 的哈希值具有更强的完整性。

系统启动时，安全隔区处理器的系统协处理器完整性保护 (SCIP) 配置会帮助阻止安全隔区处理器执行安全隔区 Boot ROM 以外的任何其他代码。启动监视器会帮助阻止安全隔区直接修改 SCIP 配置。为了使加载的 sepOS 具有可执行性，安全隔区 Boot ROM 会向启动监视器发送一个包含所加载 sepOS 的地址和大小的请求。启动监视器在收到请求后会重设安全隔区处理器、对加载的 sepOS 执行哈希算法、更新 SCIP 设置以允许执行加载的 sepOS，并在新加载的代码内开始执行。随着系统继续启动，只要有可供执行的新代码，就会使用这一相同过程。启动监视器每次都会更新启动过程的运行哈希值。启动监视器还会在运行哈希值中包括关键的安全性参数。

启动完成时，启动监视器会确定最终的运行哈希值，并将它发送到公钥加速器以用于操作系统绑定密钥。此过程旨在确保即使安全隔区 Boot ROM 中存在漏洞，也不可绕过操作系统密钥绑定。

真随机数生成器

真随机数生成器 (TRNG) 用于生成安全的随机数据。安全隔区在每次生成随机加密密钥、随机密钥种子或其他熵时都会使用 TRNG。TRNG 基于多个环形振荡器并经过 CTR_DRBG（基于计数器模式中块密码的算法）后处理。

根加密密钥

安全隔区包括一个唯一 ID (UID) 根加密密钥。UID 对于每台设备来说都是唯一的，且与设备上的任何其他标识符都无关。

随机生成的 UID 在制造过程中便被固化到 SoC 中。从 A9 SoC 开始，UID 在制造过程中由安全隔区 TRNG 生成，并使用完全在安全隔区中运行的软件进程写入到熔丝中。这一过程可以防止 UID 在制造过程中于设备之外可见，因此不可被 Apple 或其任何供应商访问或储存。

sepOS 使用 UID 来保护设备特定的密钥。有了 UID，就可以通过加密方式将数据与特定设备捆绑起来。例如，用于保护文件系统的密钥层级就包括 UID，因此如果将内置 SSD 储存设备从一台设备整个移至另一台设备，文件将不可访问。其他受保护的设备特定密钥包括面容 ID 或触控 ID 数据。在 Mac 上，只有链接到 AES 引擎的完全内置储存设备才会获得这个级别的加密。例如，通过 USB 连接的外置储存设备或者添加到 2019 年款 Mac Pro 的基于 PCIe 的储存设备都无法获得这种方式的加密。

安全隔区还有设备组 ID (GID)，它是使用特定 SoC 的所有设备共用的 ID（例如，所有使用 Apple A15 SoC 的设备共用同一个 GID）。

UID 和 GID 不可以通过联合测试行动小组 (JTAG) 或其他调试接口使用。

安全隔区 AES 引擎

安全隔区 AES 引擎是一个硬件块，用于基于 AES 密码来执行对称加密。AES 引擎设计用于防范通过时序和静态功耗分析 (SPA) 泄露信息。从 A9 SoC 开始，AES 引擎还包括动态功耗分析 (DPA) 对策。

AES 引擎支持硬件密钥和软件密钥。硬件密钥派生自安全隔区 UID 或 GID。这些密钥保存在 AES 引擎内，即使对 sepOS 软件也不可见。虽然软件可以请求通过硬件密钥进行加密和解密操作，但不能提取密钥。

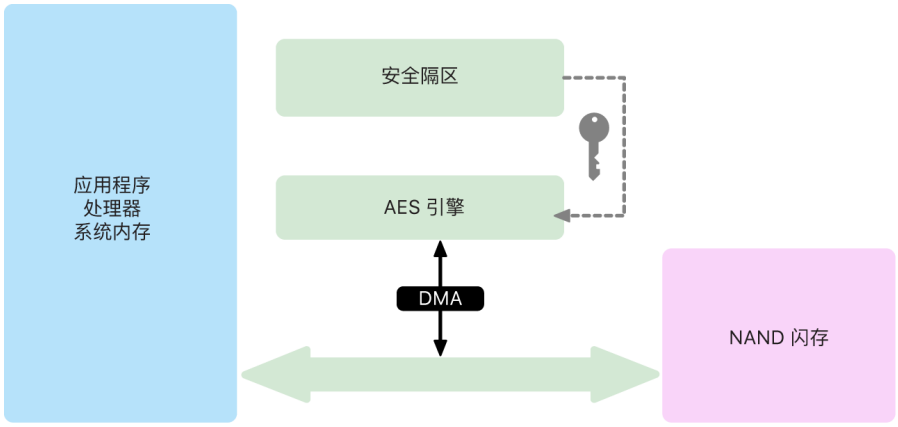
在 Apple A10 及后续型号的 SoC 上，AES 引擎包括可锁定的种子位，它可使派生自 UID 或 GID 的密钥多样化。这允许基于设备的操作模式有条件地访问数据。例如，可锁定的种子位用于从设备固件更新 (DFU) 模式启动时拒绝对受密码保护的数据进行访问。有关更多信息，请参阅密码。

AES 引擎

每台包含安全隔区的 Apple 设备还具有专用的 AES256 加密引擎（“AES 引擎”），它内置于 NAND（非易失性）闪存与主系统内存之间的直接内存访问 (DMA) 路径中，可以实现高效的文件加密。在 A9 或后续型号的 A 系列处理器上，闪存子系统位于隔离的总线上，该总线被授权只能通过 DMA 加密引擎访问包含用户数据的内存。

启动时，sepOS 会使用 TRNG 生成一个临时封装密钥。安全隔区使用专用线将此密钥传输到 AES 引擎，旨在防止它被安全隔区外的任何软件访问。sepOS 随后可以使用临时封装密钥来封装文件密钥，供应用程序处理器文件系统驱动程序使用。当文件系统驱动程序读取或写入文件时，它会将封装的密钥发送到 AES 引擎以解封密钥。AES 引擎绝不会将未封装的密钥透露给软件。

【注】AES 引擎是一个独立于安全隔区和安全隔区 AES 引擎的组件，但其操作与安全隔区紧密相连，具体如下所示。



公钥加速器

公钥加速器 (PKA) 是一个硬件块，用于执行非对称加密操作。PKA 支持 RSA 和 ECC（椭圆曲线加密）签名和加密算法。PKA 设计用于防范使用时序攻击及 SPA 和 DPA 等旁路攻击来泄露信息。

PKA 支持软件密钥和硬件密钥。硬件密钥派生自安全隔区 UID 或 GID。这些密钥保存在 PKA 内，即使对 sepOS 软件也不可见。

从 A13 SoC 开始，PKA 的加密实现方法已通过形式化验证技术被证明在数学方面是正确的。

在 Apple A10 及后续型号的 SoC 上，PKA 支持操作系统绑定密钥，也称为密封密钥保护 (SKP)。这些密钥基于设备 UID 和设备上所运行 sepOS 的哈希值组合生成。哈希值由安全隔区 Boot ROM 提供，在 Apple A13 及后续型号的 SoC 上，则由安全隔区启动监视器提供。这些密钥还用于在请求特定 Apple 服务时验证 sepOS 版本，以及用于在系统发生重大更改而未获得用户授权时通过协助阻止访问密钥材料来提高受密码保护数据的安全性。

安全非易失性存储器

安全隔区配备了专用的安全非易失性存储器设备。安全非易失性存储器通过专用的 I2C 总线与安全隔区连接，因此它仅可被安全隔区访问。所有用户数据加密密钥植根于储存在安全隔区非易失性存储器中的熵内。

搭载 A12、S4 及后续型号 SoC 的设备并用了安全隔区与安全储存组件来储存熵。安全储存组件本身设计为使用不可更改的 ROM 代码、硬件随机数生成器、每个设备唯一的加密密钥、加密引擎和物理篡改检测。安全隔区和安全储存组件使用加密且认证的协议通信以提供对熵的独有访问权限。

2020 年秋季或之后首次发布的设备配备了第二代安全储存组件。第二代安全储存组件增加了计数器加密箱。每个计数器加密箱储存一个 128 位盐、一个 128 位密码验证器、一个 8 位计数器，以及一个 8 位最大尝试值。对计数器加密箱的访问通过加密且认证的协议来实现。

计数器加密箱中含有所需用于解锁受密码保护用户数据的熵。若要访问用户数据，配对的安全隔区必须从用户的密码和安全隔区的 UID 中派生出正确的密码熵值。从除配对安全隔区之外其他来源发送的解锁尝试均无法获知用户的密码。如果密码的尝试次数超过限制（例如，在 iPhone 上为 10 次），安全储存组件就会完全抹掉受密码保护的数据。

为了创建计数器加密箱，安全隔区会向安全储存组件发送密码熵值和最大尝试次数值。安全储存组件会使用其随机数生成器生成盐值。之后通过提供的密码熵、安全储存组件的唯一加密密钥和盐值派生出密码验证器值和加密箱熵值。安全储存组件使用计数 0、提供的最大尝试次数值、派生的密码验证器值和盐值来初始化计数器加密箱。之后安全储存组件将生成的加密箱熵值返回到安全隔区。

为了稍后从计数器加密箱中取回加密箱熵值，安全隔区会向安全储存组件发送密码熵。安全储存组件会先为加密箱递增计数器。如果递增后的计数器超过最大尝试次数值，安全储存组件就会完全抹掉计数器加密箱。如果尚未达到最大尝试次数，安全储存组件会尝试通过与用于创建计数器加密箱相同的算法来派生出密码验证器值和加密箱熵值。如果派生的密码验证器值匹配所储存的密码验证器值，安全储存组件会将加密箱熵值返回到安全隔区并将计数器重设为 0。

用于访问受密码保护数据的密钥植根于计数器加密箱所储存的熵中。有关更多信息，请参阅数据保护概览。

安全隔区中的所有反重放服务均使用了安全非易失性存储器。安全隔区上的反重放服务用于在发生以下标记反重放边界的事件时撤销数据，这些事件包括但不限于：

- 更改密码
- 启用或停用面容 ID 或触控 ID
- 添加或移除面容 ID 面容或触控 ID 指纹
- 重设面容 ID 或触控 ID
- 添加或移除 Apple Pay 卡片
- 抹掉所有内容和设置

在未配备安全储存组件的架构上，EEPROM（电可擦除可编程只读存储器）被用于为安全隔区提供安全储存服务。跟安全储存组件类似，EEPROM 连接到安全隔区并仅可从安全隔区访问，但其不包含专用的硬件安全性功能，不能确保对熵的独有访问权限（除了其物理连接特性），也不具备计数器加密箱功能。

安全神经网络引擎

在配备面容 ID 的设备上，安全神经网络引擎将 2D 图像和深度图转化为用户脸部的数学表达式。

在 A11 到 A13 SoC 上，安全神经网络引擎已集成到安全隔区中。安全神经网络引擎采用直接内存访问 (DMA) 以实现高性能。由 sepOS 内核控制的输入输出内存管理单元 (IOMMU) 将此直接访问的范围限制在经授权的内存区域。

从 A14 和 M1 系列开始，安全神经网络引擎在应用程序处理器的神经网络引擎中作为安全模式实现。一个专用的硬件安全性控制器会在应用程序处理器和安全隔区的任务间切换，每次转换时神经网络引擎状态均会被重设以保持面容 ID 数据的安全。一个专用的引擎会应用内存加密、认证和访问控制。同时，它使用单独的加密密钥和内存范围，以将安全神经网络引擎限制在经授权的内存区域。

功耗和时钟监视器

所有的电子设备都被设计为在一定的电压和频率包络内运行。如果在此包络外运行，电子设备可能会发生故障，然后安全性控制就可能被绕过。为了帮助确保电压和频率保持在安全的范围内，安全隔区中设计了监视电路。这些监视电路被设计为具有比安全隔区其余部分更大的运行包络。如果监视器检测到非法运行点，安全隔区中的时钟会自动停止，在下一次 SoC 重设前不会重新开始运行。

安全隔区功能摘要

【注】2020 年秋季首次发布的 A12、A13、S4 和 S5 产品搭载了第二代安全储存组件，而基于这些 SoC 的更旧款产品搭载了第一代安全储存组件。

SoC	内存保护引擎	安全储存区	AES 引擎	PKA
A8	加密和认证	EEPROM	是	否
A9	加密和认证	EEPROM	DPA 保护	是
A10	加密和认证	EEPROM	DPA 保护和可锁定的种子位	操作系统绑定密钥
A11	加密、认证和重放预防	EEPROM	DPA 保护和可锁定的种子位	操作系统绑定密钥
A12（2020 秋季前发布的 Apple 设备）	加密、认证和重放预防	安全储存组件第 1 代	DPA 保护和可锁定的种子位	操作系统绑定密钥
A12（2020 秋季后发布的 Apple 设备）	加密、认证和重放预防	安全储存组件第 2 代	DPA 保护和可锁定的种子位	操作系统绑定密钥
A13（2020 秋季前发布的 Apple 设备）	加密、认证和重放预防	安全储存组件第 1 代	DPA 保护和可锁定的种子位	操作系统绑定密钥和启动监视器
A13（2020 秋季后发布的 Apple 设备）	加密、认证和重放预防	安全储存组件第 2 代	DPA 保护和可锁定的种子位	操作系统绑定密钥和启动监视器
A14、A15	加密、认证和重放预防	安全储存组件第 2 代	DPA 保护和可锁定的种子位	操作系统绑定密钥和启动监视器
S3	加密和认证	EEPROM	DPA 保护和可锁定的种子位	是
S4	加密、认证和重放预防	安全储存组件第 1 代	DPA 保护和可锁定的种子位	操作系统绑定密钥
S5（2020 秋季前发布的 Apple 设备）	加密、认证和重放预防	安全储存组件第 1 代	DPA 保护和可锁定的种子位	操作系统绑定密钥
S5（2020 秋季后发布的 Apple 设备）	加密、认证和重放预防	安全储存组件第 2 代	DPA 保护和可锁定的种子位	操作系统绑定密钥
S6、S7	加密、认证和重放预防	安全储存组件第 2 代	DPA 保护和可锁定的种子位	操作系统绑定密钥
T2	加密和认证	EEPROM	DPA 保护和可锁定的种子位	操作系统绑定密钥
M1 系列	加密、认证和重放预防	安全储存组件第 2 代	DPA 保护和可锁定的种子位	操作系统绑定密钥和启动监视器

面容 ID 和触控 ID

面容 ID 和触控 ID 安全性

密码是 Apple 设备安全性的基础。同时，用户需要能便捷访问自己的设备，通常一天会访问上百次。生物认证技术兼具强密码的安全性（甚至强化了密码，因为不需要手动输入），同时提供了便利性，只需手指轻轻一按或者看一眼便能快速解锁。面容 ID 和触控 ID 不会取代密码，而是在大多数情况下实现更快和更简单的访问。

Apple 的生物识别安全性架构依赖于生物识别传感器和安全隔区之间严格的职责独立性以及此二者之间安全的关联性。传感器会捕捉生物识别图像并将其安全传输到安全隔区。在注册期间，安全隔区会处理、加密和储存对应的面容 ID 和触控 ID 模板数据。匹配时，安全隔区会将从生物识别传感器传入的数据与已储存的模板对比，以决定是否解锁设备或回应匹配是否有效（用于 Apple Pay、App 内以及面容 ID 和触控 ID 的其他用途）。该架构支持包含传感器和安全隔区的设备（如 iPhone、iPad 和许多 Mac 系统），并且支持将传感器在物理上独立于外围设备中，然后与搭载 Apple 芯片的 Mac 中的安全隔区安全配对。

面容 ID 安全性

只需简单看一眼，面容 ID 就会安全地解锁受支持的 Apple 设备。它借助于原深感摄像头系统所使用的先进技术来准确绘制用户脸部的几何特征，从而提供直观而安全的认证方法。面容 ID 使用神经网络来确认屏幕注视、匹配和反欺诈，用户只需看一眼即可解锁手机，使用支持的设备时，甚至戴着口罩也能解锁。面容 ID 自动适应外貌的变化，并谨慎地保护用户生物识别数据的隐私和安全。

面容 ID 旨在确认用户的屏幕注视，提供一种匹配错误率低的可靠认证方法，并减少数字和实体诈骗。

无论是用户拿起具有面容 ID 功能的 Apple 设备或轻点屏幕来唤醒它时，还是此类设备尝试对用户进行认证以显示收到的通知时，或者是支持的 App 要求进行面容 ID 认证时，原深感摄像头都会自动查找用户的脸部。检测到脸部后，面容 ID 通过检测到用户双眼睁开且注视着设备，来确认屏幕注视和解锁意图；对于辅助功能，当“旁白”激活时，面容 ID 注视检测会停用，并且可在需要时单独停用。戴口罩使用面容 ID 时始终需要进行注视检测。

原深感摄像头确认存在注视着设备的脸部后，会投影并读取数千个红外点以绘制脸部的深度图和 2D 红外图像。此数据用于创建一个 2D 图像和深度图序列，经过数字签名后发送到安全隔区。为抵制数字和实体诈骗，原深感摄像头会随机化捕捉到的 2D 图像和深度图序列，并投影出特定设备的随机图案。神经网络引擎（在安全隔区中受到保护）中的一部分会将此数据转换为数学表达式，并将该表达式与注册的脸部数据进行对比。此注册的脸部数据本身就是捕捉自用户脸部各种姿态转换而成的数学表达式。

触控 ID 安全性

触控 ID 是指纹感应系统，有助于更快、更轻松地安全访问受支持的 Apple 设备。此技术可从任意角度读取指纹数据，随着传感器每次使用时识别出更多重叠的节点而不断扩大指纹图，逐步提高对用户指纹识别的能力。

配备触控 ID 传感器的 Apple 设备可以使用指纹解锁。触控 ID 不会取代使用设备密码或用户密码的需要，在设备启动、重启或退出登录（在 Mac 上）后仍然需要密码。在部分 App 中，触控 ID 还可以用于代替设备密码或用户密码使用，例如，在“备忘录” App 中解锁受密码保护的备忘录，解锁受钥匙串保护的网站，以及解锁支持的 App 密码。但在部分使用场景中始终需要设备密码或用户密码（例如，更改现有的设备密码或用户密码，或者移除现有的指纹注册或创建新的指纹）。

指纹传感器检测到指纹接触后，会触发先进的成像阵列来扫描手指，然后将扫描结果发送至安全隔区。用于保护这种连接的通道各有不同，具体取决于触控 ID 传感器是否内建于带安全隔区的设备中或者是否处于独立的外围设备中。

指纹扫描被向量化处理以进行分析的同时，光栅扫描结果会临时储存在安全隔区的加密内存中，之后便会被丢弃。此分析采用皮下纹路走向角度映射，这是一种有损过程，会在分析完成后丢弃所需用于重建用户实际指纹的“指纹详细数据”。在注册期间，生成的节点图以一种只能由安全隔区读取的加密格式作为模板储存，但不包含任何身份信息，以用于对比将来的匹配对象。此数据绝对不会离开设备，不会发送给 Apple，也不会包括在设备备份中。

内建触控 ID 的通道安全性

安全隔区和内建触控 ID 传感器之间的通信通过串行外设接口总线实现。处理器将数据转发到安全隔区，但其本身无法读取这些数据。数据通过会话密钥进行加密和认证，该密钥通过为每个触控 ID 传感器及其对应的安全隔区出厂预置的共享密钥进行协商。对于每个触控 ID 传感器，共享密码都是随机且不同的强密码。会话密钥交换使用 AES 密钥封装，其中双方会提供一个随机密钥，用于建立会话密钥和使用兼具认证和保密的传输加密（通过 AES-CCM）。

配备触控 ID 的妙控键盘

配备触控 ID 的妙控键盘（以及配备触控 ID 和数字小键盘的妙控键盘）在可与任何搭载 Apple 芯片的 Mac 配合使用的外接键盘中提供了触控 ID 传感器。配备触控 ID 的妙控键盘担当生物识别传感器的角色；其不会储存生物识别模板、执行生物识别匹配或强制实施安全性策略（例如，在 48 小时内未解锁后必须输入密码）。配备触控 ID 的妙控键盘中的触控 ID 传感器在可使用前必须与 Mac 上的安全隔区安全配对，然后安全隔区会以与对待内建触控 ID 传感器相同的方式来执行注册和匹配操作以及强制实施安全性策略。Apple 会在出厂前对 Mac 附带的配备触控 ID 的妙控键盘执行配对过程。配对还可由用户按需执行。一个配备触控 ID 的妙控键盘一次仅可与一台 Mac 安全配对，但一台 Mac 最多可与五个不同的配备触控 ID 的妙控键盘保持安全配对。

配备触控 ID 的妙控键盘与内建触控 ID 传感器相兼容。如果已在 Mac 内建触控 ID 传感器上注册的指纹出现在配备触控 ID 的妙控键盘上，则 Mac 中的安全隔区会成功通过匹配，反之亦然。

为支持 Mac 安全隔区与配备触控 ID 的妙控键盘安全配对和通信，键盘配备了公钥加速器 (PKA) 硬件块以提供证明，并配备了基于硬件的密钥以执行必要的加密过程。

安全配对

在可用于触控 ID 操作前，配备触控 ID 的妙控键盘需要与 Mac 安全配对。为了配对，Mac 上的安全隔区与配备触控 ID 的妙控键盘中的 PKA 块交换公钥（植根于受信任的 Apple CA），然后它们使用由硬件持有的证明密钥和临时 ECDH 来安全证明其身份。在 Mac 上，此数据受安全隔区保护；在配备触控 ID 的妙控键盘上，此数据受 PKA 块保护。安全配对后，Mac 与配备触控 ID 的妙控键盘之间交换的所有触控 ID 数据都会使用 AES-GCM 加密（密钥长度为 256 位），并且临时 ECDH 密钥使用基于已储存身份的 NIST P-256 曲线。（普通按键使用蓝牙安全性进行交换，与任何其他蓝牙键盘交换的方式一样。）

配对的安全意图

若要首次执行某些触控 ID 操作，如注册新指纹，用户必须手动确认其要将配备触控 ID 的妙控键盘和 Mac 搭配使用的意图。若要手动确认意图，可在用户界面提示时按下两次 Mac 电源键，或使用之前已通过 Mac 注册的指纹成功匹配。有关更多信息，请参阅安全意图和与安全隔区的连接。

Apple Pay 交易可通过触控 ID 匹配进行认证，还可以通过输入 macOS 用户密码和以及在配备触控 ID 的妙控键盘上按下两次触控 ID 按钮进行认证。后者可让用户手动确认意图，甚至无需触控 ID 匹配。

配备触控 ID 的妙控键盘的通道安全性

为帮助确保配备触控 ID 的妙控键盘中的触控 ID 传感器与配对的 Mac 上的安全隔区之间通信通道的安全性，需要以下条件：

- 配备触控 ID 的妙控键盘的 PKA 块与安全隔区之间的安全配对，如上所述
- 配备触控 ID 的妙控键盘的传感器与其 PKA 块之间的安全通道

在出厂时，配备触控 ID 的妙控键盘的传感器与其 PKA 块之间的安全通道通过此二者之间共享的独有密钥建立。（这种技术同样用于配备内建触控 ID 的 Mac 电脑，以创建 Mac 上的安全隔区与其内建传感器之间的安全通道。）

面容 ID、触控 ID 和密码

若要使用面容 ID 或触控 ID，用户必须对设备设置密码解锁。当面容 ID 或触控 ID 检测到匹配成功后，用户的设备便会解锁，且不会要求输入设备密码。这让使用更长、更复杂的密码变得更加实际，因为用户无需频繁地输入这样的密码。面容 ID 和触控 ID 不会取代用户的密码，而是在精心设计的范围和时间限制内方便用户轻松访问设备。这一点十分重要，因为强密码是用户的 iPhone、iPad、Mac 或 Apple Watch 通过加密方式保护该用户数据的根基。

需要设备密码时

用户可以随时使用密码来代替面容 ID 或触控 ID，但是有些情况下不允许使用生物识别技术。在以下注重安全性的操作中，始终需要输入密码：

- 更新软件
- 抹掉设备
- 查看或更改密码设置
- 安装配置描述文件
- 在 Mac 上的“系统偏好设置”中解锁“安全性与隐私”面板
- 在 Mac 上的“系统偏好设置”中解锁“用户与群组”面板（如果文件保险箱已打开）

如果设备处于以下任一状态，也需要使用密码：

- 设备刚刚开机或重新启动。
- 用户已退出登录其 Mac 帐户（或尚未登录）。
- 用户已超过 48 小时未解锁自己的设备。
- 用户在过去 156 个小时（六天半）内未使用密码解锁自己的设备，且在过去 4 小时内未使用生物识别解锁设备。
- 设备已收到远程锁定命令。
- 用户退出关机/SOS 紧急联络（同时按住任一音量按钮和睡眠/唤醒按钮 2 秒后按下“取消”）。
- 五次生物识别尝试后未能成功匹配（但出于可用性考虑，设备可能会在少于五次失败尝试后提示输入密码而不是使用生物识别）。

在 iPhone 上启用“戴口罩使用面容 ID”后，如果用户完成了以下任一项操作，该功能将在此后的 6.5 小时内可用：

- 面容 ID 匹配成功（戴或不戴口罩）
- 设备密码验证
- 设备通过 Apple Watch 解锁

顺利完成以上任一项操作都可将有效时间再延长 6.5 小时。

在 iPhone 或 iPad 上启用面容 ID 或触控 ID 后，按下睡眠/唤醒按钮时，设备会立即锁定，且设备每次进入睡眠状态后都会锁定。每次唤醒时，需要成功匹配面容 ID 和触控 ID，或者选择使用密码。

人群中随机一个人使用面容 ID（包括“戴口罩使用面容 ID”打开时）解锁某位用户的 iPhone 或 iPad 的成功概率小于一百万分之一。对于用户的配备触控 ID 的 iPhone、iPad、Mac 机型，以及与妙控键盘配对的机型，此概率小于五万分之一。如果注册多个指纹或面容，此概率会随之增长。注册五个指纹或两个面容时概率分别达到一万分之一或五十万分之一。为了提供额外的保护，面容 ID 和触控 ID 都只允许五次不成功的匹配尝试，之后便会要求输入密码来获得用户的设备或帐户的访问权限。以下人群使用面容 ID 时，错误匹配概率更高：

- 与用户长得像的双胞胎和兄弟姐妹
- 13 岁以下的儿童（因为其脸部特征可能尚未完全成型）

这两类人群戴口罩使用面容 ID 时，此概率会进一步提高。如果用户对错误匹配感到担忧，Apple 建议使用密码来认证。

脸部匹配安全性

脸部匹配在安全隔区内执行，通过专为此目的而训练的神经网络实现。Apple 开发的脸部匹配神经网络使用了超过十亿张图像，其中包括在参与者知情同意的条件下所开展的研究中收集的红外 (IR) 图像和深度图像。Apple 随后与全世界的参与者开展合作，力争包括不同性别、年龄、种族和其他因素的各类人群代表。此项研究根据需要进行了扩增，从而为各种不同范围的用户提供高度的准确性。即使用户戴有帽子、围巾、眼镜、隐形眼镜和许多种类的太阳眼镜，面容 ID 也会确保正常使用。对于自 iPhone 12 起且运行 iOS 15.4 或更高版本的 iPhone 设备，面容 ID 还支持戴口罩解锁。此外，在室内、室外，甚至完全漆黑的情况下都可以使用。训练用来发现和抵制诈骗的额外神经网络可以抵御尝试使用照片或面具来解锁设备的行为。面容 ID 数据，其中包括用户脸部的数学表达式，经过加密且仅供安全隔区使用。此数据绝对不会离开设备，不会发送给 Apple，也不会包括在设备备份中。在日常操作中，以下面容 ID 数据会存储和加密，仅供安全隔区使用：

- 注册时，计算出的用户脸部数学表达式
- 在某些解锁尝试过程中计算出的用户脸部数学表达式（前提是面容 ID 认为这些表达式有助于增强日后的匹配能力）

在日常操作中所捕捉的脸部图像不会存储，在注册时或与注册的面容 ID 数据进行对比时，会在计算出数学表达式后立即丢弃。

改善面容 ID 匹配

为了改善匹配性能并紧跟脸部和外观的自然变化，面容 ID 会随着时间扩增其储存的数学表达式。成功匹配后，面容 ID 会在有限的次数内使用新计算的数学表达式（如果其质量够好）来匹配，然后丢弃该数据。相反，如果面容 ID 未能识别脸部，但匹配质量较某些阈值要高且用户在失败后立即输入了密码，面容 ID 会再次捕捉并使用新计算的数学表达式来扩增其注册的面容 ID 数据。如果用户停止与之匹配或在有限的匹配次数后，此新的面容 ID 数据会丢弃；选择还原面容 ID 的选项后，也会丢弃此新数据。这些扩增过程让面容 ID 能够紧跟用户脸部毛发的显著改变或识别化妆品的使用，同时尽量减少错误接受率。

面容 ID 和触控 ID 的用途

解锁设备或用户帐户

如果面容 ID 或触控 ID 已关闭，则当设备或帐户锁定时，安全隔区中保存的用于实现最高类数据保护水平的密钥将被丢弃。除非用户输入密码来解锁设备或帐户，否则不允许访问该类中的文件和钥匙串项。

如果面容 ID 或触控 ID 已打开，当设备或帐户锁定时，这些密钥不会被丢弃，而是通过提供给安全隔区中的面容 ID 或触控 ID 子系统的密钥进行封装。当用户尝试解锁设备或帐户时，如果设备检测到匹配成功，它将提供用于解封数据保护密钥的密钥，从而解锁设备或帐户。此过程要求数据保护和面容 ID 或触控 ID 子系统相互配合以解锁设备，因此提供了额外的保护。

设备重新启动后，面容 ID 或触控 ID 用来解锁设备或帐户所需的密钥会丢失；如果出现任何需要输入密码的情况，安全隔区会丢弃密钥。

使用 Apple Pay 安全购物

用户也可以搭配 Apple Pay 使用面容 ID 和触控 ID 在商店、App 和网上轻松安全地购物：

- **在商店中使用面容 ID：**要使用面容 ID 授权店内支付，用户必须先连按侧边按钮两下来确认支付意图。此连按操作使用直接链接到安全隔区的物理手势来捕捉用户的意图，从而防范恶意进程的伪造。用户然后使用面容 ID 进行认证，之后将设备靠近免接触式支付读卡器。用户可以在面容 ID 认证后选择其他 Apple Pay 支付方式，此操作需要重新认证，但无需再连按两下侧边按钮。
- **在 App 内和网上使用面容 ID：**若要在 App 内和网上进行支付，用户需要连按两下侧边按钮来确认其支付意图，然后使用面容 ID 认证来授权此次付款。如果在连按两下侧边按钮后的 60 秒内没有完成 Apple Pay 交易，用户必须再次连按两下侧边按钮来重新确认支付意图。
- **使用触控 ID：**使用触控 ID 时，系统通过使用手势激活触控 ID 传感器并成功匹配用户的指纹来确认支付意图。

使用系统提供的 API

第三方 App 可以使用系统提供的 API 要求用户使用面容 ID、触控 ID 或密码进行认证，支持触控 ID 的 App 无需任何更改即可自动支持面容 ID。使用面容 ID 或触控 ID 时，App 只会收到认证是否成功的通知，而无法访问面容 ID、触控 ID 或与已注册用户关联的数据。

保护钥匙串项

钥匙串项也可使用面容 ID 或触控 ID 进行保护，使安全隔区仅当匹配成功或设备密码或帐户密码正确时才将其释出。在要求使用面容 ID、触控 ID 或密码解锁钥匙串项前，App 开发者会调用 API 确认用户已设置密码。App 开发者可以执行以下任一项操作：

- 要求认证 API 操作不回退到 App 密码或者设备密码。开发者可以查询用户是否进行了注册，从而允许在注重安全性的 App 中将面容 ID 或触控 ID 用作第二重身份。
- 在安全隔区内生成和使用受面容 ID 或触控 ID 保护的椭圆曲线加密 (ECC) 密钥。涉及这些密钥的操作始终在安全隔区授权使用后在安全隔区内执行。

购买和批准购买

用户还可以对面容 ID 或触控 ID 进行配置，以便批准 iTunes Store、App Store 和 Apple Books 等商店中的购买行为，省去每次都要输入 Apple ID 密码的麻烦。进行购买时，安全隔区会先验证发生了生物识别授权，然后才会释放 ECC 密钥以用于给商店请求签名。

安全意图和与安全隔区的连接

安全意图为确认用户意图提供了一种无需与操作系统或应用程序处理器交互的方式。实体按钮和安全隔区以物理方式连接，该连接在以下机型中可用：

- iPhone X 或后续机型
- Apple Watch Series 1 或后续机型
- iPad Pro（所有机型）
- iPad Air（2020 年）
- 搭载 Apple 芯片的 Mac 电脑

通过此连接，用户可以一种让软件（即使是拥有 root 权限或在内核中运行的软件）无法欺骗的方式确认其完成某操作的意图。

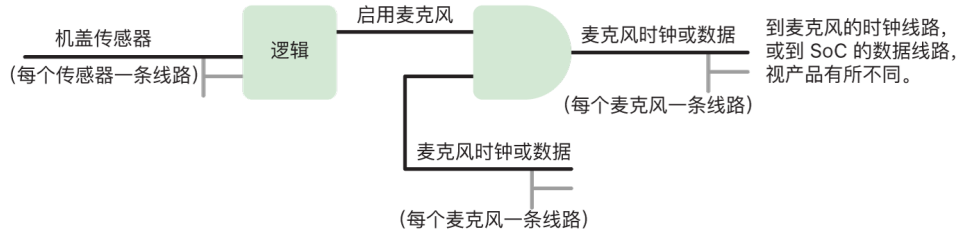
此功能用于确认用户意图，在 Apple Pay 交易期间和在最终确认将配备触控 ID 的妙控键盘与搭载 Apple 芯片的 Mac 配对时。在用户界面提示时按下两次相应的按钮（针对面容 ID）或者进行指纹扫描（针对触控 ID），表示确认用户意图。有关更多信息，请参阅使用 Apple Pay 安全购物。基于安全隔区和 T2 固件的类似机制，在搭载 Apple T2 安全芯片且未配备触控栏的 Mac 上获得支持。

硬件麦克风断联

所有基于 Apple 芯片的 Mac 笔记本电脑和所有基于 Intel 芯片且搭载 Apple T2 安全芯片的 Mac 笔记本电脑都具备硬件断联功能，该功能会在机盖合上后停用麦克风。在所有搭载 T2 芯片的 13 英寸 MacBook Pro 和 MacBook Air 笔记本电脑、所有搭载 T2 芯片的 2019 年款 MacBook 笔记本电脑或后续机型以及搭载 Apple 芯片的 Mac 笔记本电脑上，此断联功能在硬件中单独实现。这旨在防止任何软件（甚至是拥有 macOS root 或内核权限的软件以及 T2 芯片或其他固件上的软件）在机盖合上后启用麦克风。（摄像头不会在硬件中断联，因为其可视区域在机盖合上后会被完全遮挡。）

2020 年及之后推出的 iPad 机型也支持硬件麦克风断联功能。将符合 MFi 标准的保护壳（包括 Apple 销售的保护壳）连接到 iPad 并合上之后，麦克风会在硬件层面断联。这旨在防止任何软件使用麦克风音频数据（即使软件在 iPadOS 或任何设备固件中拥有 root 或内核权限）。

此部分的保护直接通过硬件逻辑按照如下电路图实施：



每款支持硬件切断麦克风的产品都有一个或多个机盖传感器，可通过用户交互的某些物理属性（如霍尔效应传感器或铰链角度传感器）检测到机盖或保护壳以物理方式盖上的行为。对于需要校准的传感器，其参数会在设备的生产过程中设定，校准过程还会包括不可逆的硬件锁定，以阻止对传感器上敏感参数的任何后续更改。这些传感器会发出直接的硬件信号，该信号会通过一组简单的不可重新编程的硬件逻辑。此逻辑会在停用麦克风前提供防抖、滞后和/或最高 500 毫秒的延迟。根据产品的不同，此信号可通过停用麦克风和片上系统 (SoC) 之间用于传输数据的线路来实现，或停用连接到麦克风模块上允许其启用的其中一个输入线路来实现，如时钟线路或类似的有效控制。

通过备用电量使用快捷卡

如果 iOS 由于 iPhone 需要充电而无法运行，电池中可能仍有足够的电量来支持快捷卡交易。支持的 iPhone 设备会自动配合以下卡片使用此功能：

- 指定为快捷交通卡的付款卡或交通卡
- 打开了“快捷模式”的学生证
- 打开了“快捷模式”的车钥匙
- 打开了“快捷模式”的家庭钥匙
- 打开了“快捷模式”的酒店或公司门禁卡

按下侧边按钮（或者在第 2 代 iPhone SE 上按下主屏幕按钮）后，屏幕显示低电量图标和可使用快捷卡的文本。NFC 控制器执行快捷卡交易的条件与 iOS 运行时的情况相同，除了此时交易仅以触感通知来表示（无可见通知显示）。在第 2 代 iPhone SE 上，交易在完成后可能需要经过几秒钟才会显示在屏幕上。当用户发起的标准关机已执行时，此功能不可用。

系统安全性

系统安全性概览

在 Apple 硬件独特功能的基础上，系统安全性负责控制对 Apple 设备中系统资源的访问，同时不影响可用性。系统安全性覆盖了启动过程、软件更新以及对 CPU、内存、磁盘、软件程序和已储存数据等电脑系统资源的保护。

最新版本的 Apple 操作系统最为安全。Apple 安全性的一个重要组成部分是**安全启动**，它可保护系统在启动时不受恶意软件的感染。安全启动始于硬件并通过软件构建信任链，这个过程中的每一步都设计为确保下一步能够正常运行之后才转移控制权。这种安全模型不仅支持 Apple 设备的默认启动，还支持 Apple 设备上的各种恢复模式和及时更新。T2 芯片和安全隔区等子组件还会执行自己的安全启动过程，以帮助确保只启动来自 Apple 的已知安全代码。更新系统的设计旨在阻止降级攻击，以使设备不能回滚到较旧版本的操作系统（攻击者知道如何入侵该版本），从而防止用户数据遭窃。

Apple 设备还包括启动保护和运行时保护，因此在操作过程中会保持其完整性。iPhone、iPad、Apple Watch、Apple TV、HomePod 和搭载 Apple 芯片的 Mac 上由 Apple 设计的芯片为保护操作系统完整性提供了通用架构。macOS 还具有一系列扩展的可配置保护功能（以支持其不同的计算模型）以及所有 Mac 硬件平台上都支持的功能。

安全启动

iOS 和 iPadOS 设备启动过程

启动过程每个步骤包含的组件都经 Apple 加密签名以启用完整性检查，因此只有在验证信任链后，启动才能继续。这些组件包括引导载入程序、内核、内核扩展项和蜂窝网络基带固件。这一安全启动链的设计旨在验证软件的最底层不被篡改。

iOS 或 iPadOS 设备开机后，其应用程序处理器会立即执行只读内存（称为 Boot ROM）中的代码。这些不可更改的代码（称为**硬件信任根**）是在制造芯片时设定的隐式受信任代码。Boot ROM 代码包含 Apple 根证书颁发机构 (CA) 公钥，该公钥用于验证 iBoot 引导载入程序是否经过 Apple 签名，以决定是否允许其载入。这是信任链中的第一步，信任链中的每个步骤都会检查下一步骤是否已经过 Apple 的签名。iBoot 完成任务后，会验证和运行 iOS 或 iPadOS 内核。对于搭载 A9 或更早 A 系列处理器的设备，Boot ROM 还会载入和验证底层引导载入程序 (LLB)，之后会依次载入和验证 iBoot。

无法载入或验证以下阶段时，处理方式因硬件而异：

- **Boot ROM 无法载入 LLB（较旧的设备）：**设备固件升级 (DFU) 模式
- **LLB 或 iBoot：**恢复模式

出现任一情况时，设备都必须通过 USB 连接到“访达”（macOS 10.15 或更高版本）或 iTunes（macOS 10.14 或更低版本），并恢复为出厂默认设置。

安全隔区使用启动进程寄存器 (BPR) 来限制不同模式中对用户数据的访问，在进入以下模式前会对 BPR 进行更新：

- **DFU 模式：**由搭载 Apple A12 或后续型号 SoC 的设备上的 Boot ROM 设定
- **恢复模式：**由搭载 Apple A10、S2 或后续型号 SoC 的设备上的 iBoot 设定

对于可接入蜂窝网络的设备，蜂窝网络基带子系统使用已签名的软件以及由基带处理器验证的密钥来执行额外的安全启动过程。

安全隔区还会执行安全启动过程来检查其软件 (sepOS) 是否已经过 Apple 验证和签名。

内存安全 iBoot 实施

在 iOS 14 和 iPadOS 14 中，Apple 修改了用于构建 iBoot 引导载入程序的 C 编译器工具链，以提高其安全性。修改后的工具链会运行旨在阻止内存和类型安全性问题的代码，此类问题通常在 C 程序中遇到。例如，代码可帮助防止以下类中的大多数漏洞：

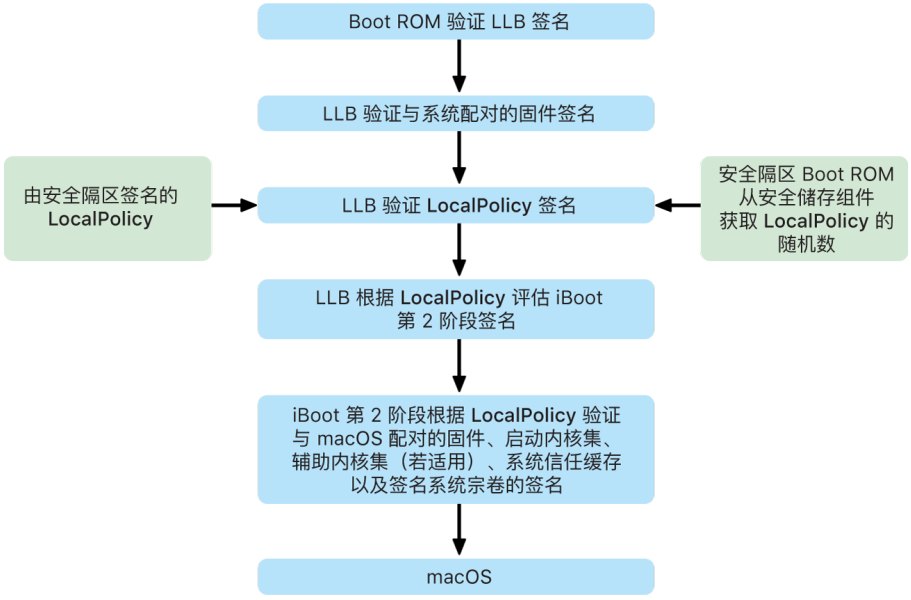
- 缓冲区溢出（通过确保所有指针都携带在访问内存时经过验证的边界信息）
- 堆利用（通过将堆数据和其元数据分开，以及准确检测诸如双重释放错误等错误状态）
- 类型混淆（通过确保所有指针都携带在指针转换操作期间经过验证的运行时类型信息）
- 由释放后重用错误导致的类型混淆（通过使用静态类型来分离所有动态内存分配）

此技术在搭载 Apple A13 仿生或更高版本的 iPhone 和搭载 A14 仿生芯片的 iPad 上可用。

搭载 Apple 芯片的 Mac 电脑

搭载 Apple 芯片的 Mac 的启动过程

搭载 Apple 芯片的 Mac 开机后，会执行与 iPhone 和 iPad 非常相似的启动过程。



芯片会在信任链的第一步从 Boot ROM 执行代码。搭载 Apple 芯片的 Mac 上的 macOS 安全启动不仅会验证操作系统代码本身，还会验证安全性策略甚至由授权用户配置的 Kext（支持但不建议）。

LLB（即底层引导载入程序）启动后，会验证签名并为存储、显示、系统管理和雷雳控制器等 SoC 内的核心载入与系统配对的固件。LLB 还负责载入 LocalPolicy，LocalPolicy 是一个由安全隔区处理器签名的文件。LocalPolicy 文件描述了用户已为系统启动和运行时安全性策略选取的配置。LocalPolicy 的数据结构格式与所有其他启动对象相同，但它是由仅在特定电脑的安全隔区内提供的私钥在本地签名，而不是由中央 Apple 服务器签名（类似于软件更新）。

为帮助防止任何之前的 LocalPolicy 重放，LLB 必须从与安全隔区连接的安全储存组件查找随机数。为执行此操作，LLB 使用安全隔区 Boot ROM 并确保 LocalPolicy 中的随机数与安全储存组件中的随机数匹配。这有助于防止安全性升级后将可能具有更低安全性配置的旧 LocalPolicy 重新应用到系统中。因此，搭载 Apple 芯片的 Mac 上的安全启动不仅可帮助防范操作系统版本回滚，还能防范安全性策略降级。

LocalPolicy 文件会捕捉操作系统是配置为“完整”、“降低”还是“宽松”安全性。

- **完整安全性：**系统运行方式与 iOS 和 iPadOS 相似，且允许仅启动在安装时已知为最新可用版本的软件。
- **降低安全性：**LLB 被要求信任与操作系统捆绑的“全局”签名。这允许系统运行 macOS 较旧版本。macOS 较旧版本不可避免地具有未修补的漏洞，所以此安全性模式称为**“降低”**安全性。这也是支持启动内核扩展 (Kext) 所需的策略等级。
- **宽松安全性：**与“降低安全性”类似，系统也为 iBoot 及其后的过程使用全局签名验证，但它还通知 iBoot 应当接受由安全隔区使用给 LocalPolicy 签名的相同密钥来签名的部分启动对象。此策略等级支持用户构建、签名和启动自己的自定 XNU 内核。

如果 LocalPolicy 告知 LLB 所选操作系统以“完整安全性”运行，LLB 会为 iBoot 评估定制化签名。如果 LLB 以“降低安全性”或“宽松安全性”运行，它会评估全局签名。任何签名验证错误都会导致系统启动进入 recoveryOS，以提供修复选项。

LLB 转交给 iBoot 后，它会为安全神经网络引擎、始终在线处理器等载入与 macOS 配对的固件以及载入其他固件。iBoot 还会查看由 LLB 交给它的 LocalPolicy 的相关信息。如果 LocalPolicy 表明应当具有辅助内核集 (AuxKC)，iBoot 会在文件系统上查找 AuxKC，验证它是由安全隔区使用给 LocalPolicy 签名的相同密钥进行签名，并验证其哈希值与 LocalPolicy 中储存的哈希值匹配。如果 AuxKC 通过验证，iBoot 会将其放到包含启动内核集的内存中，然后使用系统协处理器完整性保护 (SCIP) 锁定包含启动内核集和 AuxKC 的整个内存区域。如果策略表明 AuxKC 应当存在，但无法找到，系统会在缺少 AuxKC 的情况下继续启动进入 macOS。iBoot 还负责验证签名系统宗卷 (SSV) 的根哈希值，以检查内核将要装载的文件系统的完整性是否经过完全验证。

搭载 Apple 芯片的 Mac 的启动模式

搭载 Apple 芯片的 Mac 有如下所述的启动模式。

模式	组合键	描述
macOS	在关机状态下，按下电源按钮并**松开**。	1. Boot ROM 转交给 LLB。 2. LLB 为所选 macOS 载入与系统配对的固件和 LocalPolicy。 3. LLB 将正在启动进入 macOS 的指示锁定到启动进程寄存器 (BPR) 中，并转交给 iBoot。 4. iBoot 载入与 macOS 配对的固件、静态信任缓存、设备树和启动内核集。 5. 如果 LocalPolicy 允许，iBoot 会载入第三方 Kext 的辅助内核集 (AuxKC)。 6. 如果 LocalPolicy 未停用签名系统宗卷 (SSV)，iBoot 会验证其根签名哈希值。
配对 recoveryOS	在关机状态下，**按住**电源按钮。	1. Boot ROM 转交给 LLB。 2. LLB 为 recoveryOS 载入与系统配对的固件和 LocalPolicy。 3. LLB 将正在启动进入配对 recoveryOS 的指示锁定到启动进程寄存器中，并转交给配对 recoveryOS 的 iBoot。 4. iBoot 载入与 macOS 配对的固件、信任缓存、设备树和启动内核集。 5. 如果配对 recoveryOS 启动失败，将尝试启动进入回退 recoveryOS。 **【注】**配对 recoveryOS LocalPolicy 上不允许安全性降级。
回退 recoveryOS	在关机状态下，**按两下并按住**电源按钮。	1. Boot ROM 转交给 LLB。 2. LLB 为 recoveryOS 载入与系统配对的固件和 LocalPolicy。 3. LLB 将正在启动进入配对 recoveryOS 的指示锁定到启动进程寄存器中，并转交给 recoveryOS 的 iBoot。 4. iBoot 载入与 macOS 配对的固件、信任缓存、设备树和启动内核集。 **【注】**配对 recoveryOS LocalPolicy 上不允许安全性降级。
“安全”模式	按照上述过程启动进入 recoveryOS，然后按住 **Shift** 键选择启动宗卷。	1. 按照上述过程启动到 recoveryOS。 2. 按住 Shift 键选择宗卷会导致 BootPicker App 照常批准该 macOS 启动，同时也会设定 `nvram` 变量告知 iBoot 在下次启动时不要载入 AuxKC。 3. 系统重新启动到目标宗卷，但 iBoot 不会载入 AuxKC。

配对 recoveryOS 的访问限制

在 macOS 12.0.1 或更高版本中，每次安装新 macOS 时，也会将配对版本的 recoveryOS 安装到对应的 APFS 宗卷组。基于 Intel 的 Mac 电脑用户非常熟悉此设计，但在搭载 Apple 芯片的 Mac 上，其提供了额外的安全性和兼容性保障。因为现在每次 macOS 安装都有专用的配对 recoveryOS，这有助于确保只有该专用的配对 recoveryOS 可执行安全性降级操作。此设计有助于保护新版本 macOS 的安装免受旧版本 macOS 发起的篡改，反之亦然。

配对访问限制按照以下方式强制执行：

- 所有 macOS 11 安装都与 recoveryOS 配对。在搭载 Apple 芯片的 Mac 上，如果选择默认启动 macOS 11 安装，可在启动时按住电源按钮来启动 recoveryOS。recoveryOS 可降级任何 macOS 11 安装的安全性设置，但不可对任何 macOS 12.0.1 安装执行此操作。
- 如果选择默认启动 macOS 12.0.1 或更高版本安装，可在 Mac 开机时按住电源按钮来启动进入其配对 recoveryOS。配对 recoveryOS 可降级配对 macOS 安装的安全性设置，但不可对任何其他 macOS 安装执行此操作。

若要启动进入任何 macOS 安装的配对 recoveryOS，需要将该安装选择为默认，方法是使用“系统偏好设置”中的“启动磁盘”，或者通过启动任何 recoveryOS 并在选择宗卷时按住 Option 键。

【注】回退 recoveryOS 无法为任何 macOS 安装执行降级。

搭载 Apple 芯片的 Mac 的启动磁盘安全性策略控制

概览

与基于 Intel 的 Mac 的安全性策略不同，搭载 Apple 芯片的 Mac 上的安全性策略针对的是安装的每个操作系统。这意味着同一 Mac 上支持安装版本和安全性策略不同的多个 macOS 实例。为此，“启动安全性实用工具”中添加了**操作系统选择器**。

在搭载 Apple 芯片的 Mac 上，“系统安全性实用工具”会指明用户配置的 macOS 整体安全性状态，如 Kext 的启动或系统完整性保护 (SIP) 配置。如果更改安全性设置会显著降低安全性或使系统更易遭到入侵，则用户必须通过按住电源按钮进入 recoveryOS（使恶意软件无法触发信号，只有用户通过实际接触才能触发）才能进行更改。因此基于 Apple 芯片的 Mac 也不要求提供（亦不支持）固件密码，所有关键更改已经由用户授权保护。有关 SIP 的更多信息，请参阅系统完整性保护。

“完整安全性”和“降低安全性”可在 recoveryOS 中使用“启动安全性实用工具”设定。但“宽松安全性”只能通过命令行工具访问，适用于接受大大降低 Mac 安全性的风险的用户。

“完整安全性”策略

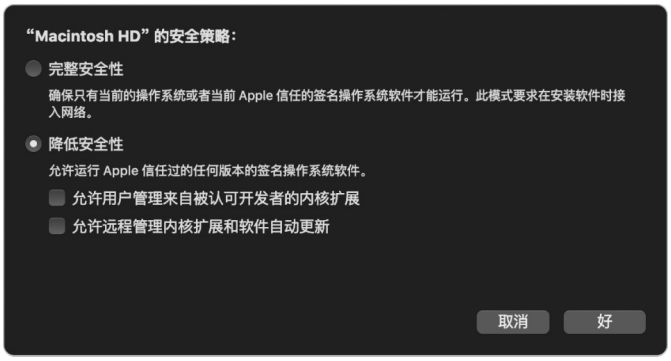
“完整安全性”是默认的安全启动策略，其运行方式与 iOS 和 iPadOS 相似。下载好软件并准备安装时，macOS 会联系用于 iOS 和 iPadOS 的相同 Apple 签名服务器并请求新的“定制化”签名，而不是使用软件附带的全局签名。当签名包括作为签名请求一部分的专有芯片 ID (ECID) (此处指 Apple CPU 特定的唯一 ID) 时，即属于定制化签名。此时签名服务器返回的签名是唯一的且只能供该特定的 Apple CPU 使用。“完整安全性”策略生效时，Boot ROM 和 LLB 帮助确保给定的签名不仅只由 Apple 签名，而且只针对这台特定的 Mac 签名，从而实际上将该版本的 macOS 与此 Mac 绑定。

相对于传统的全局签名方法，使用在线签名服务器还能更好地防范回滚攻击。在全局签名系统中，安全时间点可能多次更迭，但是一直未安装最新固件的系统无法得知。例如，当前认为处于安全时间点 1 的电脑会接受来自安全时间点 2 的软件，即使当前实际的安全时间点为 5。通过 Apple 芯片在线签名系统，签名服务器可以拒绝为不是处于最新安全时间点的软件创建签名。

另外，如果攻击者在安全时间点更迭后发现漏洞，他们无法简单地利用前一个时间点系统 A 中存在漏洞的软件并将其应用于系统 B 以实施攻击。来自较旧时间点的存在漏洞的软件是专为系统 A 量身定制的，这有助于防止其转移，也就无法将其用于攻击系统 B。所有这些机制协同工作，更有力确保攻击者无法故意在 Mac 上安装存在漏洞的软件以绕过最新软件提供的保护。但是拥有 Mac 管理员用户名和密码的用户可以根据自己的使用情况，始终选择最适合的安全性策略。

“降低安全性”策略

“降低安全性”与基于 Intel 且搭载 T2 芯片的 Mac 上的“中等安全性”行为类似：供应商 (此处指 Apple) 生成代码的数字签名以声明代码来自供应商。此设计可帮助防止攻击者插入未签名的代码。Apple 将这种签名称为“全局”签名，因为该签名可以在任何 Mac 上不限时长使用，只要 Mac 当前的策略设为“降低安全性”。“降低安全性”本身并不能防范回滚攻击 (虽然未经授权的操作系统更改可导致用户数据无法访问)。有关更多信息，请参阅搭载 Apple 芯片的 Mac 中的内核扩展。



除了可让用户运行较旧版本的 macOS，其他可能为用户的系统安全带来风险的操作也需要“降低安全性”，如引入第三方内核扩展 (Kext)。Kext 与内核权限相同，因此第三方 Kext 中的任何漏洞都可能导致整个操作系统遭到入侵。因此，在未来搭载 Apple 芯片的 Mac 电脑上的 macOS 不再支持 Kext 之前，我们强烈建议开发者采用系统扩展。即使第三方 Kext 已启用，也不能按需将其载入内核。这些 Kext 会被合并到辅助内核集 (AuxKC)，AuxKC 的哈希值储存在 LocalPolicy 中，因此需要重新启动。有关 AuxKC 生成的更多信息，请参阅 macOS 中的内核扩展。

“宽松安全性”策略

如果用户接受使其 Mac 进入非常不安全状态的风险，则适合使用“宽松安全性”。此模式与基于 Intel 且搭载 T2 芯片的 Mac 上的“无安全性”模式不同。使用“宽松安全性”时，整个安全启动链上仍然会执行签名验证，但将策略设为“宽松”会发送信号告知 iBoot 应当接受由安全隔区在本地签名的启动对象，如用户生成的从自定 XNU 内核中构建的启动内核集。通过这种方式，“宽松安全性”也为运行任意“完全不受信任操作系统”内核提供了架构层面的功能。自定启动内核集或完全不受信任的操作系统载入到系统上时，部分解密密钥会变为不可用。此设计旨在防止完全不受信任的操作系统访问受信任操作系统的数据。

【重要事项】Apple 不提供也不支持自定 XNU 内核。

“宽松安全性”与基于 Intel 且搭载 T2 芯片的 Mac 上“无安全性”还有一点不同：“宽松安全性”是某些过去可独立控制的安全性降级的先决条件。最明显的表现是，若要在搭载 Apple 芯片的 Mac 上停用系统完整性保护 (SIP)，用户必须确认要为系统启用“宽松安全性”。需要用户确认是因为停用 SIP 始终会将系统置于内核更易遭到入侵的状态。尤其是在搭载 Apple 芯片的 Mac 上停用 SIP 会在 AuxKC 生成期间停用 Kext 签名实施，从而允许将任意 Kext 载入内核内存。在搭载 Apple 芯片的 Mac 上，针对 SIP 的另一改进是将策略储存移出 NVRAM 并移入 LocalPolicy。因此，现在需要具有 LocalPolicy 签名密钥访问权限的用户进入 recoveryOS（通过按住电源按钮），在其中进行认证后才能停用 SIP。此功能使仅软件层面的攻击者（甚至处于电脑前的攻击者）停用 SIP 的难度显著提高。

从“启动安全性实用工具” App 中无法降级到“宽松安全性”。用户只能通过 recoveryOS 中的“终端”运行命令行工具（如 `csrutil` 以停用 SIP）来降级。用户降级后，已降级的状态会反映在“启动安全性实用工具”中，使用户可以轻松将安全性设为更安全的模式。

【注】搭载 Apple 芯片的 Mac 不要求（也不支持）特定介质启动策略，因为从技术上来说所有启动都在本地执行。如果用户选择从外部介质启动，该操作系统版本必须先使用 recoveryOS 中已认证的重新启动操作进行定制化。此重新启动操作会在内部驱动器上创建 LocalPolicy 文件，该内部驱动器用于从储存在外部介质上的操作系统执行受信任启动。这意味着从外部介质启动的配置对每个操作系统始终明确启用，且已经要求用户授权，因此无需额外的安全配置。

LocalPolicy 签名密钥创建和管理

创建

出厂前首次安装 macOS 时，或执行联机抹掉-安装时，Mac 会运行来自临时恢复 RAM 磁盘的代码，以初始化默认状态。在此过程中，恢复环境会创建一对新的公钥和私钥，保存在安全隔区中。私钥称为**所有者身份密钥 (OIK)**。如果已存在任何 OIK，此过程中会将其销毁。恢复环境还会初始化用于激活锁的密钥，即**用户身份密钥 (UIK)**。该过程中有一部分是搭载 Apple 芯片的 Mac 所独有：当激活锁需要 UIK 认证时，过程中会包括一系列需要在验证时应用于 LocalPolicy 的限制。如果设备无法获取针对激活锁进行认证的 UIK（例如，由于设备当前与“查找我的 Mac”帐户关联且报告为丢失），则无法继续进行 LocalPolicy 创建。如果已为设备颁发了**用户身份证书 (ucrt)**，该 ucrt 中会包含由服务器实施的策略限制以及用户请求的策略限制（位于 X.509 v3 扩展项中）。

激活锁/`ucrt` 成功获取后，会储存在服务器端的数据库中，同时也会返回给设备。设备有了 ucrt 后，针对与 OIK 对应的公钥的认证请求会发送到**基础证明机构 (BAA)** 服务器。BAA 会使用来自 ucrt 的公钥来验证 OIK 认证请求，BAA 具有对储存该 ucrt 的数据库的访问权限。如果 BAA 能够验证此认证，它会通过返回由 BAA 签名且包含储存在 ucrt 中的限制的**所有者身份证书 (OIC)** 来认证该公钥。OIC 会被发送回安全隔区。从此以后，只要安全隔区为新的 LocalPolicy 进行签名，就会将 OIC 附加到 Image4 中。LLB 内建了对 BAA 根证书的信任，因此也会信任该 OIC，从而也会信任整个 LocalPolicy 签名。

RemotePolicy 限制

所有 Image4 文件（不只是 LocalPolicy）都包含 Image4 清单评估限制。这些限制使用叶证书中的特殊对象标识符 (OID) 编码。Image4 验证库在签名评估期间从证书中查找特殊证书限制 OID，然后通过机制评估在其中指定的限制。限制具有以下形式：

- X 必须存在
- X 必须不存在
- X 必须具有特定值

因此，例如对于“定制化”签名，证书限制将包括“ECID 必须存在”；对于“全局”签名，限制将包括“ECID 必须不存在”。这些限制旨在确保给定密钥签名的所有 Image4 文件必须符合特定要求，以避免生成错误的已签名 Image4 清单。

对于每个 LocalPolicy，这些 Image4 证书限制称为 **RemotePolicy**。不同启动环境的 LocalPolicy 可存在不同的 RemotePolicy。RemotePolicy 用于限制 recoveryOS LocalPolicy，以确保在 recoveryOS 启动时只能像以“完整安全性”启动时一样的方式运行。由于可在 recoveryOS 启动环境中更改策略，这种方式提高了其完整性的可信度。RemotePolicy 会将 LocalPolicy 限制为包含生成 LocalPolicy 的 Mac 的 ECID，以及该 Mac 上安全储存组件中储存的特定远程策略随机数哈希值 (`rpnh`)。只有在对“查找我的 Mac”和激活锁执行注册、取消注册、远程锁定和远程擦除等操作时，`rpnh` 才会发生更改，因此 RemotePolicy 也只能在这些情况下随之更改。RemotePolicy 限制在用户身份密钥 (UIK) 认证时确定和指定，并且登记到已颁发的用户身份证书 (ucrt) 中。某些 RemotePolicy 限制由服务器确定，如 ECID、ChipID 和 BoardID。此设计旨在防止一台设备为另一台设备的 LocalPolicy 文件签名。其他 RemotePolicy 限制可能由设备指定，以帮助防止在未同时提供访问当前 OIK 所需的本地认证以及与设备激活锁关联的帐户的远程认证的情况下，对 LocalPolicy 进行安全性降级。

搭载 Apple 芯片的 Mac 的 LocalPolicy 文件内容

LocalPolicy 是一个经过安全隔区签名的 Image4 文件。Image4 是经 ASN.1（抽象语法标记一）DER 编码的数据结构格式，用于描述 Apple 平台上有关安全启动链对象的信息。在基于 Image4 的安全启动模型中，当针对中央 Apple 签名服务器的签名请求发起软件安装时，会要求采用安全性策略。如果策略可接受，签名服务器会返回已签名的 Image4 文件，其中包含各种四字符代码 (4CC) 序列。这些已签名的 Image4 文件和 4CC 在启动时会经过 Boot ROM 或 LLB 等软件的评估。

操作系统之间的所有权交接

对所有者身份密钥 (OIK) 的访问权限称为“所有权”。进行策略或软件更改后，需要所有权才能允许用户丢弃 LocalPolicy。OIK 使用封装密钥保护 (SKP) 中所述的相同密钥层级进行保护，同时也受到与宗卷加密密钥 (VEK) 相同的密钥加密密钥 (KEK) 的保护。这意味着 OIK 通常同时受到用户密码以及操作系统和策略的测量值的保护。Mac 上所有操作系统只有一个 OIK。因此，安装第二个操作系统时，需要第一个操作系统上用户的明确同意才能将所有权交接给第二个操作系统上的用户。但是，从第一个操作系统上运行安装器时，第二个操作系统上尚不存在用户。通常情况下，直到操作系统启动且“设置助理”运行时，才会生成用户。因此，在搭载 Apple 芯片的 Mac 上安装第二个操作系统时，需要执行两项新操作：

- 为第二个操作系统创建 LocalPolicy
- 为所有权交接准备“安装用户”

运行“安装助理”并将安装指向次级空白宗卷时，一则提示会询问用户是否要拷贝当前宗卷的用户以作为第二个宗卷的首个用户。如果用户选择是，就会创建“安装用户”，它实际上是一个由所选用户的密码和硬件密钥派生的 KEK，然后该 KEK 会用于在 OIK 被转交给第二个操作系统时对其进行加密。之后，在第二个操作系统的“安装助理”中，系统会提示输入该用户的密码，以允许其访问安全隔区中新操作系统的 OIK。如果用户选择不拷贝用户，“安装用户”仍会以同样的方式创建，但会使用空密码而非用户的密码。这第二个流程存在于某些系统管理场景中。但是，如果用户想进行多宗卷安装并且以最安全的方式执行所有权交接，应当始终选择将用户从第一个操作系统拷贝到第二个操作系统。

搭载 Apple 芯片的 Mac 上的 LocalPolicy

在搭载 Apple 芯片的 Mac 上，本地安全性策略控制已委托给安全隔区中运行的一个应用程序。此软件可利用用户凭证和主 CPU 启动模式来确定谁可以更改安全性策略，以及从什么启动环境中更改。这有助于防范恶意软件通过降级安全性策略控制来获得更多权限，从而攻击用户。

LocalPolicy 清单属性

LocalPolicy 文件包含可在几乎所有 Image4 文件中找到的某些架构 4CC，如电路板或型号 ID (BORD)，用于指示特定 Apple 芯片 (CHIP) 或专有芯片 ID (ECID)。但以下 4CC 只专注于用户可配置的安全性策略。

【注】Apple 使用术语**第一真正配对 recoveryOS (1TR)** 来表示通过按住一次实体电源按钮来启动进入配对 recoveryOS。它与普通 recoveryOS 启动不同，后者可通过 NVRAM 实现、通过按两下并按住电源按钮实现，或者可能发生在启动出错时。按照特定方式按实体按钮会提高信任，使已经攻入 macOS 的仅软件层面攻击者无法进入启动环境。

LocalPolicy 随机数哈希值 (lpnh)

- **类型：**OctetString (48)
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**`lpnh` 用于 LocalPolicy 反重放。这是 LocalPolicy 随机数 (LPN) 的 SHA384 哈希值，其中 LPN 储存在安全储存组件中，并可通过安全隔区 Boot ROM 或安全隔区访问。原始随机数对应用程序处理器永不可见，仅对 sepOS 可见。如果攻击者想让 LLB 相信其捕获的某个之前的 LocalPolicy 有效，则需要将与希望重放的 LocalPolicy 中找到的 `lpnh` 值相同的一个哈希值放入安全储存组件。系统上通常只有一个 LPN 有效，除了在软件更新期间会有两个 LPN 同时有效，以在更新出错时允许回退并启动旧版本软件。任何操作系统的任何 LocalPolicy 更改时，所有策略会使用与安全储存组件中找到的新 LPN 对应的新 `lpnh` 值重新签名。用户更改安全性设置，或者创建多个新操作系统且每个操作系统都有一个新的 LocalPolicy 时，此值就会发生更改。

远程策略随机数哈希值 (rpnh)

- **类型：**OctetString (48)
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**`rpnh` 行为与 `lpnh` 相同，但 rpnh 只在远程策略更新时才更新，如更改“查找”注册的状态时。用户在其 Mac 上更改“查找”状态时，会发生此更改。

recoveryOS 随机数哈希值 (ronh)

- **类型：**OctetString (48)
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**ronh 行为与 lpnh 相同，但 ronh 只能在系统 recoveryOS 的 LocalPolicy 中找到。它在系统 recoveryOS 更新时更新，如在软件更新时。系统还会使用一个独立于 lpnh 和 rpnh 的随机数，以便设备被“查找”置于停用状态时，现有操作系统可被停用（通过将其 LPN 和 RPN 从安全储存组件中移除）的同时仍保持系统 recoveryOS 可启动。通过这种方式，在系统所有者通过输入其用于“查找”帐户的 iCloud 密码来证明对系统享有控制权时，这些操作系统可被重新启用。用户更新系统 recoveryOS 或者创建新操作系统时，此值就会发生更改。

下一阶段 Image4 清单哈希值 (nsih)

- **类型：**OctetString (48)
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**nsih 字段代表 Image4 清单数据结构（描述已启动的 macOS）的 SHA384 哈希值。macOS Image4 清单包含对 iBoot、静态信任缓存、设备树、启动内核集和签名系统宗卷 (SSV) 的宗卷根哈希值等所有启动对象的测量值。LLB 被要求启动给定的 macOS 时，其设计旨在确保附于 iBoot 的 macOS Image4 清单哈希值与 LocalPolicy 的 nsih 字段中捕获的值匹配。通过这种方式，nsih 将得知用户已为哪个操作系统创建了 LocalPolicy。用户执行软件更新时，就间接更改了 nsih 值。

辅助内核集 (AuxKC) 策略哈希值 (auxp)

- **类型：**OctetString (48)
- **可变环境：**macOS
- **描述：**auxp 是用户授权 Kext 列表 (UAKL) 策略的 SHA384 哈希值。在 AuxKC 生成期间，它用于帮助确保 AuxKC 中只包括用户授权的 Kext。smb2 是设定此字段的先决条件。用户在“系统偏好设置”的“安全性与隐私”面板中批准 Kext 以更改 UAKL 时，就间接更改了 auxp 值。

辅助内核集 (AuxKC) Image4 清单哈希值 (auxi)

- **类型：**OctetString (48)
- **可变环境：**macOS
- **描述：**系统验证 UAKL 哈希值与 LocalPolicy 的 auxp 字段中找到的值匹配后，会请求负责 LocalPolicy 签名的安全隔区处理器应用程序为 AuxKC 签名。接下来，AuxKC Image4 清单签名的 SHA384 哈希值会被放入 LocalPolicy，以避免在启动时错将之前签名的 AuxKC 与操作系统匹配。如果 iBoot 在 LocalPolicy 中找到 auxi 字段，会尝试从储存空间中载入 AuxKC 并验证其签名。iBoot 还会验证附于 AuxKC 的 Image4 清单哈希值与 auxi 字段中找到的值匹配。如果 AuxKC 出于任何原因未能载入，系统会在没有此启动对象的情况下继续启动，因此也不载入任何第三方 Kext。auxp 字段是设定 LocalPolicy 中 auxi 字段的先决条件。用户在“系统偏好设置”的“安全性与隐私”面板中批准 Kext 以更改 UAKL 时，就间接更改了 auxi 值。

辅助内核集 (AuxKC) 接收项哈希值 (auxr)

- **类型：**OctetString (48)
- **可变环境：**macOS
- **描述：**auxr 是 AuxKC 接收项的 SHA384 哈希值，AuxKC 接收项表示包括在 AuxKC 中确切的 Kext 集。AuxKC 接收项可以是 UAKL 的子集，因为如果 Kext 已知被用于攻击，即使它们经过用户授权，也可被排除在 AuxKC 之外。另外，可用于破坏用户内核边界的某些 Kext 可能会导致功能受损，例如不能使用 Apple Pay 或播放 4K 和 HDR 内容。希望拥有这些功能的用户可选择使用更严格的 AuxKC 包括条件。auxp 字段是设定 LocalPolicy 中 auxr 字段的先决条件。用户在“系统偏好设置”的“安全性与隐私”面板中构建新的 AuxKC 时，就间接更改了 auxr 值。

CustomOS Image4 清单哈希值 (coih)

- **类型：**OctetString (48)
- **可变环境：**1TR
- **描述：**`coih` 是 CustomOS Image4 清单的 SHA384 哈希值。iBoot（而非 XNU 内核）使用该清单的有效负载来传输控制。用户在 1TR 中使用 `kmutil configure-boot` 命令行工具时，就间接更改了 `coih` 值。

APFS 宗卷组 UUID (vuid)

- **类型：**OctetString (16)
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**`vuid` 表示应当被内核用作根的宗卷组。此字段主要提供信息，不用于安全性限制。此 `vuid` 由用户在创建新操作系统安装时间接设定。

密钥加密密钥 (KEK) 组 UUID (kuid)

- **类型：**OctetString (16)
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**`kuid` 表示已启动的宗卷。密钥加密密钥通常用于数据保护。对于 LocalPolicy，它被用于保护 LocalPolicy 签名密钥。`kuid` 由用户在创建新操作系统安装时间接设定。

配对 recoveryOS 受信任启动策略测量值 (prot)

- **类型：**OctetString (48)
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**配对 recoveryOS 受信任启动策略测量值 (TBPM) 是一种对 LocalPolicy Image4 清单的特殊迭代 SHA384 哈希计算值，计算中不包含随机数，以随时间进行恒定的测量（因为像 `lpnh` 这样的随机数会频繁更新）。`prot` 字段只能在每个 macOS LocalPolicy 中找到，它提供配对功能来指示与该 macOS LocalPolicy 对应的 recoveryOS LocalPolicy。

有安全隔区签名的 recoveryOS LocalPolicy (hrlp)

- **类型：**布尔值
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**`hrlp` 表示以上 `prot` 值是否为经过安全隔区签名的 recoveryOS LocalPolicy 的测量值。如果不是，则 recoveryOS LocalPolicy 是由 Apple 在线签名服务器签名，该服务器为 macOS Image4 文件等内容签名。

本地操作系统版本 (love)

- **类型：**布尔值
- **可变环境：**1TR、recoveryOS、macOS
- **描述：**`love` 表示为其创建 LocalPolicy 的操作系统版本。该版本在 LocalPolicy 创建期间从下一个状态清单中获取，用于强制执行 recoveryOS 配对访问限制。

安全 Multi-Boot (smb0)

- **类型：**布尔值
- **可变环境：**1TR、recoveryOS
- **描述：**如果 `smb0` 存在且为真，LLB 将允许对下一阶段 Image4 清单进行全局签名，而不是要求定制化签名。用户可通过“启动安全性实用工具”或 `bputil` 更改此字段，以降级到“降低安全性”。

安全 Multi-Boot (smb1)

- **类型：**布尔值
- **可变环境：**1TR
- **描述：**如果 `smb1` 存在且为真，iBoot 将允许安全隔区使用与 LocalPolicy 相同的密钥给自定内核集等对象签名。`smb0` 的存在是 `smb1` 存在的先决条件。用户可使用 `csrutil` 或 `bputil` 等命令行工具更改此字段，以降级到“宽松安全性”。

安全 Multi-Boot (smb2)

- **类型：**布尔值
- **可变环境：**1TR
- **描述：**如果 `smb2` 存在且为真，iBoot 将允许安全隔区使用与 LocalPolicy 相同的密钥给辅助内核集签名。`smb0` 的存在是 `smb2` 存在的先决条件。用户可使用“启动安全性实用工具”或 `bputil` 更改此字段，以降级到“降低安全性”并启用第三方 Kext。

安全 Multi-Boot (smb3)

- **类型：**布尔值
- **可变环境：**1TR
- **描述：**如果 `smb3` 存在且为真，表示设备用户已选择为其系统启用移动设备管理 (MDM) 控制。此字段的存在使 LocalPolicy 控制安全隔区处理器应用程序接受 MDM 认证，而不是要求本地用户认证。用户可使用“启动安全性实用工具”或 `bputil` 更改此字段，以启用针对第三方 Kext 和软件更新的受管理控制。（在 macOS 11.2 或更高版本中，如果当前安全性模式设为“完整安全性”，MDM 还可以启动更新到最新版本的 macOS。）

安全 Multi-Boot (smb4)

- **类型：**布尔值
- **可变环境：**macOS
- **描述：**如果 `smb4` 存在且为真，表示设备已选择通过“Apple 校园教务管理”、“Apple 商务管理”或“Apple 商务必备”为操作系统启用 MDM 控制。此字段的存在使 LocalPolicy 控制安全隔区应用程序接受 MDM 认证，而不是要求本地用户认证。此字段由 MDM 解决方案在检测到设备序列号出现在这三项服务中的任一项中时更改。

系统完整性保护 (sip0)

- **类型：**64 位未签名整数
- **可变环境：**1TR
- **描述：**`sip0` 保存现有系统完整性保护 (SIP) 策略位，这些位之前储存在 NVRAM 中。新 SIP 策略位在此处添加（而不是使用如下 LocalPolicy 字段），前提是这些位只在 macOS 中使用，且不被 LLB 使用。用户可从 1TR 使用 `csrutil` 更改此字段以停用 SIP，并降级到“宽松安全性”。

系统完整性保护 (sip1)

- **类型：**布尔值
- **可变环境：**1TR
- **描述：**如果 `sip1` 存在且为真，iBoot 将允许 SSV 宗卷根哈希值验证失败。用户可从 1TR 使用 `csrutil` 或 `bputil` 更改此字段。

系统完整性保护 (sip2)

- **类型：**布尔值
- **可变环境：**1TR
- **描述：**如果 sip2 存在且为真，iBoot 将不会锁定将内核内存标记为不可写入的**可配置文本只读区域 (CTRR)** 硬件寄存器。用户可从 1TR 使用 `csrutil` 或 `bputil` 更改此字段。

系统完整性保护 (sip3)

- **类型：**布尔值
- **可变环境：**1TR
- **描述：**如果 `sip3` 存在且为真，iBoot 将不会为 boot-args NVRAM 变量实施其内建允许列表，否则该列表将过滤传递到内核的选项。用户可从 1TR 使用 `csrutil` 或 `bputil` 更改此字段。

证书和 RemotePolicy

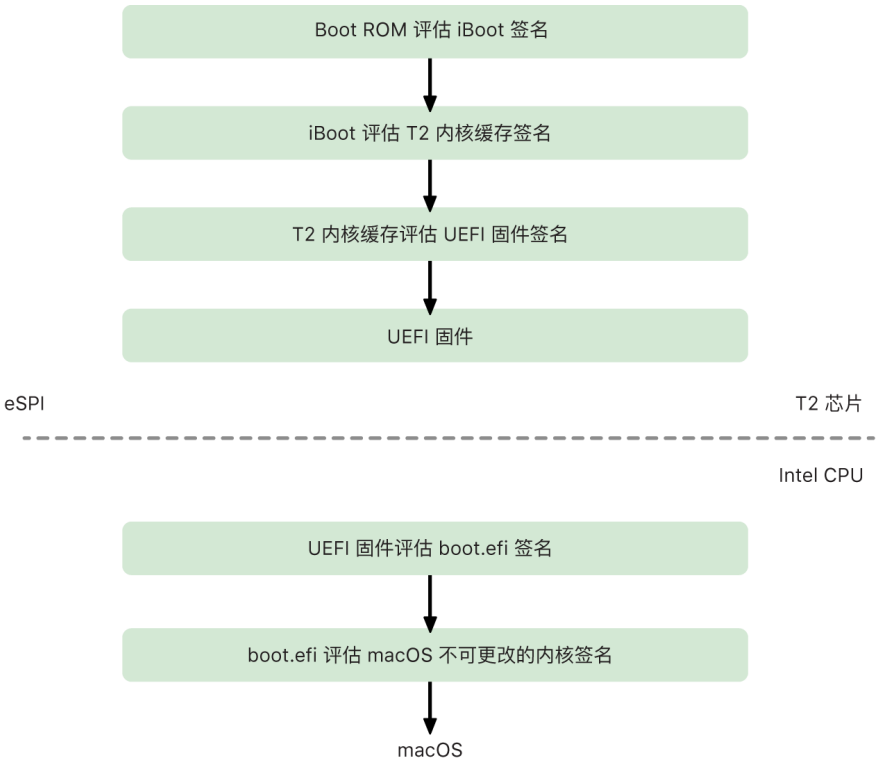
如 LocalPolicy 签名密钥创建和管理中所述，LocalPolicy Image4 中还包含所有者身份证书 (OIC) 和嵌入的 RemotePolicy。

基于 Intel 的 Mac 电脑

基于 Intel 的 Mac 的启动过程

基于 Intel 且搭载 Apple T2 安全芯片的 Mac

基于 Intel 且搭载 Apple T2 安全芯片的 Mac 电脑开机后，芯片会通过与 iPhone、iPad 和搭载 Apple 芯片的 Mac 相同的方式从 Boot ROM 执行安全启动。此步骤会验证 iBoot 引导载入程序，是信任链的第一步。iBoot 会检查 T2 芯片上的内核和内核扩展代码，之后 T2 芯片会检查 Intel UEFI 固件。UEFI 固件和相关的签名最初只能被 T2 芯片使用。



通过验证后，UEFI 固件映像会映射到 T2 芯片内存的某个部分。此内存通过增强的串行外设接口 (eSPI) 供 Intel CPU 访问。Intel CPU 首次启动时，它会通过 eSPI 从 T2 芯片上进行了完整性检查和内存映射的固件副本中获取 UEFI 固件。

信任链的评估会在 Intel CPU 中继续：UEFI 固件会评估作为 macOS 引导载入程序的 boot.efi 的签名。Intel 芯片中驻留的 macOS 安全启动签名使用 Image4 格式储存，该格式与用于 iOS、iPadOS 和 T2 芯片安全启动的格式相同；而解析 Image4 文件的代码与当前实施 iOS 和 iPadOS 安全启动的强化代码相同。Boot.efi 又会验证称为 immutablekernel 的新文件的签名。安全启动启用后，immutablekernel 文件表示启动 macOS 所需的一系列完整的 Apple 内核扩展。进入 immutablekernel 阶段后，安全启动策略便会告一段落；之后 macOS 安全性策略（如“系统完整性保护”和签名的内核扩展）便会实施。

如果在这个过程中发生了任何错误或失败，Mac 会进入恢复模式、Apple T2 安全芯片恢复模式或 Apple T2 安全芯片设备固件升级 (DFU) 模式。

基于 Intel 且搭载 T2 芯片的 Mac 上的 Microsoft Windows

基于 Intel 且支持安全启动的 Mac 默认只信任由 Apple 签名的内容。但是为了提高“启动转换”安装的安全性，Apple 也支持 Windows 安全启动。统一可扩展固件接口 (UEFi) 固件包括了用于认证 Microsoft 引导载入程序的 Microsoft Windows Production CA 2011 证书副本。

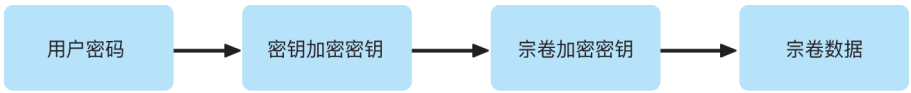
【注】目前尚未提供针对 Microsoft Corporation UEFI CA 2011 的信任，该信任允许对由 Microsoft 合作伙伴签名的代码进行验证。此 UEFI CA 通常用于验证其他操作系统（如 Linux 变体）中引导载入程序的真实性。

对 Windows 安全启动的支持默认处于未启用状态，需要通过“启动转换助理”(BCA) 启用。用户运行 BCA 时，macOS 会重新配置以在启动过程中信任 Microsoft 第一方签名的代码。BCA 完成后，如果 macOS 在安全启动过程中未能通过 Apple 第一方信任评估，UEFI 固件会尝试根据 UEFI 安全启动格式评估对于对象的信任。如果信任评估成功，Mac 会继续并启动 Windows。如果失败，Mac 会进入 recoveryOS 并通知用户信任评估失败。

基于 Intel 且不搭载 T2 芯片的 Mac 电脑

基于 Intel 且不搭载 T2 芯片的 Mac 不支持安全启动。因此统一可扩展固件接口 (UEFi) 固件会在未经验证的情况下从文件系统载入 macOS 引导程序 (boot.efi)，而引导程序会在未经验证的情况下从文件系统载入内核 (prelinkedkernel)。为了保护启动链的完整性，用户应该启用以下所有安全性机制：

- **系统完整性保护 (SIP)：**默认启用；此功能会保护引导程序和内核免受来自正在运行的 macOS 的恶意写入。
- **文件保险箱：**可通过两种方式启用：由用户启用或由移动设备管理 (MDM) 管理员启用。此功能可防止处于电脑前的攻击者使用“目标磁盘模式”改写引导程序。
- **固件密码：**可通过两种方式启用：由用户启用或由 MDM 管理员启用。此功能可帮助防止处于电脑前的攻击者启动备选启动模式，如 recoveryOS、“单用户模式”或者“目标磁盘模式”，在这些模式下，引导程序可被改写。它还可帮助防止从备选介质启动；攻击者可以通过这种方式运行代码来改写引导程序。



基于 Intel 且搭载 Apple T2 安全芯片的 Mac 的启动模式

基于 Intel 且搭载 Apple T2 安全芯片的 Mac 启动时，通过按下可被 UEFI 固件或引导程序识别的组合键可以进入各种启动模式。部分启动模式，如“单用户模式”，只有在“启动安全性实用工具”中将安全策略更改为“无安全性”时才会正常工作。

模式	组合键	描述
macOS 启动	无	验证会由 UEFI 固件转交给 macOS 引导程序 (UEFI 应用程序)，而后者又会将验证转交给 macOS 内核。正常启动启用了文件保险箱的 Mac 时，macOS 引导程序会显示登录窗口界面，在其中输入密码可解密储存设备。
启动管理器	Option (⌥)	UEFI 固件会启动内建的 UEFI 应用程序，该应用程序会向用户显示启动设备选择界面。
目标磁盘模式 (TDM)	T	UEFI 固件会启动内建的 UEFI 应用程序，该应用程序会将内置储存设备显示为基于块的无格式储存设备，且该设备通过 FireWire、雷雳、USB 或这三种端口的任意组合连接（具体取决于 Mac 机型）。
单用户模式	Command (⌘)-S	macOS 内核传递 launchd 参数向量中的 –s 标志位，之后 launchd 会在“控制台” App 的 tty 中创建单用户 shell。 **【注】**如果用户退出 shell，macOS 会继续启动进入登录窗口。
recoveryOS	Command (⌘)-R	UEFI 固件会从内置储存设备上的签名磁盘映像 (.dmg) 文件中载入最简化版 macOS。

模式	组合键	描述
互联网 recoveryOS	Option (⌥)-Command (⌘)-R	使用 HTTP 从互联网下载签名的磁盘映像。
诊断	D	UEFI 固件会从内置储存设备上的签名磁盘映像文件中载入最简化版 UEFI 诊断环境。
互联网诊断	Option (⌥)-D	使用 HTTP 从互联网下载签名的磁盘映像。
Windows 启动	无	如果使用“启动转换”安装了 Windows，验证会由 UEFI 固件转交给 Windows 引导程序，而后者又会将验证转交给 Windows 内核。

搭载 Apple T2 安全芯片的 Mac 上的启动安全性实用工具

概览

在基于 Intel 且搭载 Apple T2 安全芯片的 Mac 上，“启动安全性实用工具”负责处理许多安全性策略设置。若要访问此实用工具，请启动进入 recoveryOS，然后从“实用工具”菜单中选择“启动安全性实用工具”；它可防止支持的安全性设置被攻击者轻易操纵。

即使处于“恢复模式”，对关键策略进行更改也要求认证。“启动安全性实用工具”首次打开后，用户需要从与当前启动的 recoveryOS 关联的主要 macOS 安装中输入管理员密码。如果不存在管理员，必须创建管理员才能更改策略。T2 芯片要求 Mac 电脑当前已启动进入 recoveryOS，且已使用安全隔区支持的凭证进行认证，才能更改此类策略。安全性策略更改有两个隐性要求。recoveryOS 必须：

- 从与 T2 芯片直接相连的储存设备启动，因为其他设备上的分区没有绑定到内置储存设备且受安全隔区支持的凭证。
- 位于基于 APFS 的宗卷上，因为只支持将发送到安全隔区的“恢复中认证”凭证储存在驱动器的“预启动” APFS 宗卷上。HFS+ 格式的宗卷无法使用安全启动。

此策略仅显示在基于 Intel 且搭载 T2 芯片的 Mac 上的“启动安全性实用工具”中。虽然在大多数使用情况下应该不需要更改安全启动策略，但用户的设备设置最终由用户控制，并且用户可根据自己的需要在 Mac 上停用或者降级安全启动功能。

在此 App 中对安全启动策略进行的更改仅应用于正在 Intel 处理器上进行验证的信任链评估。“T2 芯片安全启动”选项始终有效。

你可以将安全启动策略配置为以下三种设置中的其中一种：完整安全性、中等安全性、无安全性。“无安全性”完全停用 Intel 处理器上的安全启动评估，允许用户启动任何系统。

“完整安全性”启动策略

“完整安全性”是默认的启动策略，其行为与 iOS 和 iPadOS，或者搭载 Apple 芯片的 Mac 上的“完整安全性”类似。软件已下载并准备好安装时，会使用签名进行定制化，该签名包括作为签名请求一部分的专有芯片 ID (ECID)（此处指 T2 芯片特定的唯一 ID）。此时签名服务器返回的签名是唯一的且只能由该特定 T2 芯片使用。统一可扩展固件接口 (UEFI) 固件旨在确保“完整安全性”策略生效时给定的签名不仅只由 Apple 签名，而且只针对这台特定的 Mac 签名，从而实际上将该版本的 macOS 与此 Mac 绑定。这有助于防范搭载 Apple 芯片的 Mac 上的“完整安全性”中所述的回滚攻击。

“中等安全性”启动策略

“中等安全性”启动策略与传统的 UEFI 安全启动情况有些类似：供应商（此处为 Apple）生成代码的数字签名以声明代码来自供应商。这样攻击者就无法插入未签名的代码。我们将这种签名称为“全局”签名，因为该签名可以在任何 Mac 上不限时长使用，只要 Mac 当前的启动策略设为“中等安全性”。iOS、iPadOS 和 T2 芯片本身都不支持全局签名。此设置不会尝试阻止回滚攻击。

介质启动策略

介质启动策略只存在于基于 Intel 且搭载 T2 芯片的 Mac 上，且独立于安全启动策略。因此，即使用户停用安全启动，也不会更改只能从与 T2 芯片直接相连的储存设备启动 Mac 这一默认行为。（搭载 Apple 芯片的 Mac 无需介质启动策略。有关更多信息，请参阅启动磁盘安全性策略控制。）

基于 Intel 的 Mac 中的固件密码保护

基于 Intel 且搭载 Apple T2 安全芯片的 Mac 电脑上的 macOS 支持使用固件密码，以帮助防止意外修改特定 Mac 上的固件设置。固件密码旨在阻止选择备选启动模式，如启动进入 recoveryOS 或“单用户模式”、从未经授权的宗卷启动或者启动进入“目标磁盘模式”。

【注】搭载 Apple 芯片的 Mac 无需固件密码，因为它所限制的关键固件功能已移到 recoveryOS 中，并且（文件保险箱启用时）recoveryOS 需要用户认证才能允许访问其关键功能。

在基于 Intel 且**不搭载** T2 芯片的 Mac 上，可以通过 recoveryOS 中的“固件密码实用工具”来访问固件密码的最基本模式；而在基于 Intel 且**搭载** T2 芯片的 Mac 上，可以从“启动安全性实用工具”进行访问。在 macOS 中，可以使用 `firmwarepasswd` 命令行工具来访问高级选项（如在每次启动时都提示输入密码）。

在基于 Intel 且不搭载 T2 芯片的 Mac 电脑上设置固件密码尤为重要，这可以降低处于电脑前的攻击者进行攻击的风险。固件密码可以帮助阻止攻击者启动进入 recoveryOS，否则攻击者可能在其中停用系统完整性保护 (SIP)。通过限制从备选介质启动，攻击者无法执行其他操作系统中的特权代码来攻击外围固件。

固件密码重设机制的存在对忘记密码的用户很有帮助。用户在启动时按下组合键后，系统会显示机型特定的字符串以提供给 AppleCare。AppleCare 在使用统一资源标识符 (URI) 检查资源的签名后，会对资源进行数字签名。如果签名经过验证且内容是针对该特定 Mac，UEFI 固件会移除固件密码。

对于不想让除自己外的其他任何人通过软件方式移除固件密码的用户，macOS 10.15 为 `firmwarepasswd` 命令行工具增加了 `-disable-reset-capability` 选项。在设置此选项前，用户必须确认在忘记密码且需要移除该密码的情况下，自行承担移除密码带来的逻辑板更换费用。对于想保护其 Mac 电脑不受外部攻击者和员工攻击的组织，必须在组织拥有的系统上设定固件密码。此操作可通过以下任一种方式在设备上完成：

- 预置期间，手动使用 `firmwarepasswd` 命令行工具
- 通过使用 `firmwarepasswd` 命令行工具的第三方管理工具
- 使用移动设备管理 (MDM)

基于 Intel 的 Mac 的 recoveryOS 和诊断环境

recoveryOS

recoveryOS 完全独立于主 macOS，且所有内容储存在名为 BaseSystem.dmg 的磁盘映像文件中。与之关联的还有一个 BaseSystem.chunklist，用于验证 BaseSystem.dmg 的完整性。chunklist 是 BaseSystem.dmg 中大小为 10 MB 区块的一系列哈希值。统一可扩展固件接口 (UEFi) 固件评估 chunklist 文件的签名，然后评估 BaseSystem.dmg 中区块的哈希值（一次评估一个区块）。这可帮助确保其匹配 chunklist 中所包含的签名内容。如果有任何哈希值不匹配，则会中止从本地 recoveryOS 启动；UEFI 固件会转而尝试从互联网 recoveryOS 启动。

如果成功完成验证，UEFI 固件会将 BaseSystem.dmg 作为内存磁盘装载并启动其中包含的 boot.efi。UEFI 固件无需对 boot.efi 执行特定检查，boot.efi 也无需对内核进行检查，因为操作系统中所完成的内容（这些元素只是其中的一个子集）已经经过了完整性检查。

Apple 诊断

启动本地诊断环境的进程与启动 recoveryOS 的进程大致相同。启动过程中会使用单独的 AppleDiagnostics.dmg 和 AppleDiagnostics.chunklist 文件，但它们的验证方式与 BaseSystem 文件相同。UEFI 固件会启动磁盘映像（.dmg 文件）内名为 diags.efi 的文件，而不是 boot.efi；该文件接下来会负责调用各种其他的 UEFI 驱动程序来与硬件通过接口进行通信并检查错误。

互联网 recoveryOS 和诊断环境

如果在启动本地恢复或诊断环境时出错，UEFI 固件会转而尝试从互联网下载映像。（用户还可以在启动时以特定顺序按下按键来特别要求从互联网获取映像。）对于从操作系统恢复服务器下载的磁盘映像和 chunklist，其完整性验证方式与从储存设备获取的映像的验证方式相同。

通过 HTTP 连接到操作系统恢复服务器完成后，仍会按照前述方式对完成下载的内容进行完整性检查，以此来防范控制了网络的攻击者的操纵。如果单个区块的完整性验证失败，在放弃并显示错误之前，会重新向操作系统恢复服务器再次请求该区块 11 次。

在 2011 年将互联网恢复和诊断模式添加到 Mac 电脑时，我们确定比较好的方案是使用更简单的 HTTP 传输方式并使用 chunklist 机制处理内容认证，而不是在 UEFI 固件中实施更复杂的 HTTPS 功能，否则会扩大固件的攻击面。

iOS、iPadOS 和 macOS 中的签名系统宗卷安全性

Apple 在 macOS 10.15 中引入了只读系统宗卷，这是一个专用于系统内容的独立宗卷。macOS 11 或更高版本通过**签名系统宗卷 (SSV)** 为系统内容新增了强大的加密保护措施。SSV 具有的内核机制会在运行时验证系统内容的完整性，并拒绝不含来自 Apple 的有效加密签名的任何代码和非代码数据。自 iOS 15 和 iPadOS 15 起，iOS 和 iPadOS 设备上的系统宗卷也获得了签名系统宗卷的加密保护。

SSV 不仅有助于防止对作为操作系统一部分的任何 Apple 软件的篡改，还使 macOS 软件更新更可靠、更安全。由于 SSV 使用 APFS（Apple 文件系统）快照，如果某项更新无法执行，无需重新安装即可恢复到旧系统版本。

引入 SSV 以来，APFS 已在内部储存设备上使用非加密校验和来提供文件系统元数据完整性。SSV 新增加密哈希值以增强完整性机制，从而将其扩展到覆盖文件数据的每个字节。来自内部储存设备的数据（包括文件系统元数据）在读取路径中通过加密方式生成哈希值，然后该哈希值会与文件系统元数据中的预期值比较。如果不匹配，系统将假定数据已遭到篡改，不会将其返回给发起请求的软件。

每个 SSV SHA256 哈希值储存在主文件系统元数据树中，该树自身也经过哈希化。与二进制哈希（默克尔）树类似，该树的每个节点会以递归方式验证其子节点哈希值的完整性，因此根节点的哈希值（称为**封章**）覆盖了 SSV 中数据的每个字节，也意味着加密签名覆盖了整个系统宗卷。

macOS 安装和更新期间，会在设备端从文件系统重新计算该封章，并将该测量值与 Apple 签名的测量值进行对比验证。在搭载 Apple 芯片的 Mac 上，引导载入程序会先验证封章，然后将控制转交给内核。在基于 Intel 且搭载 Apple T2 安全芯片的 Mac 上，引导载入程序会将测量值和签名转发到内核，内核接着会直接验证封章，然后装载根文件系统。无论哪种情况，如果验证失败，启动过程将暂停，用户将收到重新安装 macOS 的提示。此流程会在每次启动时重复运行，除非用户已选择进入更低的安全性模式并且已单独选择停用签名系统宗卷。

iOS 和 iPadOS 软件更新期间，会以类似的方式对系统宗卷进行准备和重新计算。iOS 和 iPadOS 引导载入程序会验证封章完整且与 Apple 签名的值匹配，然后才允许设备启动内核。如果启动时发生不匹配，会提示用户更新设备上的系统软件。用户无法停用 iOS 和 iPadOS 上签名系统宗卷的保护。

SSV 和代码签名

代码签名仍存在并由内核执行。签名系统宗卷在从内部储存设备上读取任何字节时提供保护。相反，代码签名在 Mach 对象在内存中映射为可执行时提供保护。所有读取和执行路径上的可执行代码均会受到 SSV 和代码签名的保护。

SSV 和文件保险箱

在 macOS 11 中，SSV 为系统内容提供对等的静息状态保护，因此系统宗卷不再需要加密。读取到对静息状态文件系统进行的任何修改时，文件系统会检测到这些修改。如果用户已启用文件保险箱，数据宗卷上的用户内容仍通过用户提供的密钥加密。

如果用户选择停用 SSV，处于静息状态的系统会变得易于篡改，此篡改可使攻击者能够在系统下次启动时提取加密的用户数据。因此，如果文件保险箱已启用，系统不会允许用户停用 SSV。静息状态保护必须同时为两个宗卷启用或停用。

在 macOS 10.15 或更低版本中，文件保险箱在静息状态时通过加密用户和系统内容来保护操作系统软件，加密所使用的密钥受用户提供的密钥保护。此功能防范了可实际接触设备的攻击者访问或有效修改包含系统软件的文件系统。

SSV 和搭载 Apple T2 安全芯片的 Mac

在搭载 Apple T2 安全芯片的 Mac 上，只有 macOS 本身受 SSV 保护。在 T2 芯片上运行并验证 macOS 的软件受安全启动保护。

安全软件更新

安全保护是一个过程，可靠启动出厂安装的操作系统版本是不够的，还必须存在一个可快速安全获取最新安全性更新的机制。Apple 会定期发布软件更新，以解决新出现的安全性问题。iOS 和 iPadOS 设备的用户会在设备上收到更新通知。Mac 用户可在“系统偏好设置”中找到可用更新。更新通过无线方式发送，目的在于尽快应用最新的安全性修复。

更新过程

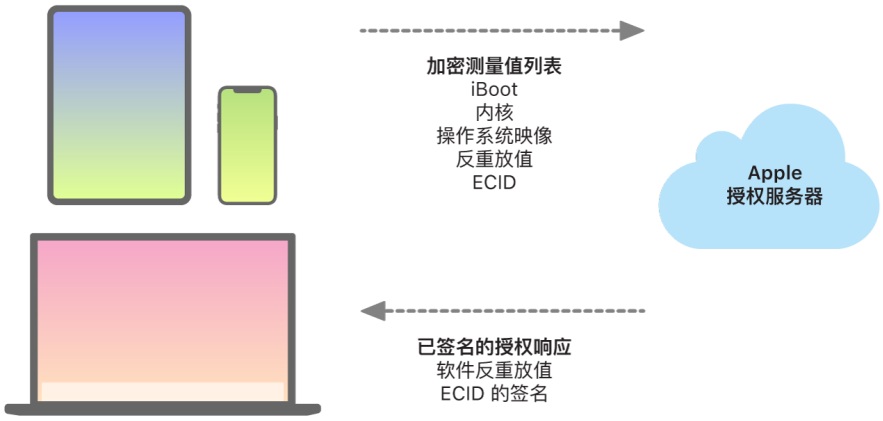
更新过程使用与安全启动过程相同的基于硬件的信任根，其设计旨在仅安装 Apple 签名的代码。更新过程还使用系统软件授权来检查是否只有经 Apple 动态签名的操作系统版本的副本才能安装在 iOS 和 iPadOS 设备上，或者安装在“启动安全性实用工具”中将“完整安全性”设置配置为安全启动策略的 Mac 电脑上。有了这些安全过程后，Apple 便可停止为存在已知漏洞的较旧版本操作系统签名，以及帮助阻止降级攻击。

为提高软件更新的安全性，要升级的设备通过物理方式连接到 Mac 时，系统会下载并安装 iOS 或 iPadOS 的完整副本。但是如果采用无线 (OTA) 方式安装软件更新，系统将**仅下载完成更新所需的组件**，而不是下载整个操作系统，这样可有效提升网络效率。此外，软件更新可以缓存到运行 macOS 10.13 或更高版本且打开了“内容缓存”的 Mac 上，这样 iOS 和 iPadOS 设备便无需通过互联网重新下载必要的更新。(设备仍需联系 Apple 服务器来完成更新过程。)

定制化更新过程

升级和更新过程中会连接到 Apple 安装授权服务器，该服务器包括要安装的安装包各部分 (例如 iBoot、内核及操作系统映像) 的加密测量值列表、一个随机的反重放值 (随机数) 以及设备的唯一专有芯片 ID (ECID)。

授权服务器将提供的测量值列表与允许安装的版本进行比较，如果找到匹配项，就会将 ECID 添加到测量值并对结果进行签名。作为升级过程的一部分，服务器会将完整的一组已签名数据传递给设备。添加 ECID 可为请求设备“定制化”授权。通过仅对已知测量值授权和签名，服务器可帮助确保更新的内容即为 Apple 所提供的内容。



启动时信任链评估会验证签名是否来自 Apple，并结合设备的 ECID 验证从储存设备载入的项目测量值是否与该签名认可的内容相匹配。这些步骤旨在确保在支持定制化的设备上授权只针对特定设备，且较旧的操作系统或固件版本无法从一台设备拷贝到另一台。随机数可帮助防止攻击者存储服务器的响应和利用该响应来破坏设备或通过其他方式篡改系统软件。

由于定制化过程的存在，更新搭载 Apple 设计的芯片的任何设备 (包括基于 Intel 且搭载 Apple T2 安全芯片的 Mac) 时都始终需要通过网络连接到 Apple。

最后，软件更新过程中绝不会装载用户数据宗卷，以帮助防止更新期间在该宗卷中读取或写入任何内容。

在搭载安全隔区的设备上，该硬件会以类似的方式使用系统软件授权来检查其软件的完整性，其设计还可防止降级安装。

操作系统完整性

Apple 的操作系统软件在设计时以安全性为核心。此设计包括硬件信任根（可用于启用安全启动），以及快速安全的安全软件更新过程。Apple 操作系统还使用为特定目的构建的基于芯片的硬件功能，以在系统运行时帮助阻止恶意利用。这些运行时功能可以保护受信任的代码在执行时的完整性。简而言之，Apple 的操作系统软件可帮助减少攻击和对技术的恶意利用，无论攻击是来自恶意 App、网页还是通过任何其他渠道。此处列出的保护措施在搭载 Apple 设计的受支持 SoC 的设备上可用，此类 SoC 包括 iOS、iPadOS、Apple tvOS 和 watchOS，现在还包括搭载 Apple 芯片的 Mac 上的 macOS。

功能	A10	A11、S3	A12、S4	A13、S5	A14、A15、S6、S7	M1 系列
内核完整性保护	✓	✓	✓	✓	✓	✓
快速权限访问限制		✓	✓	✓	✓	✓
系统协处理器完整性保护			✓	✓	✓	✓
指针认证代码			✓	✓	✓	✓
页面保护层		✓	✓	✓	✓	请参阅下方备注。

【注】页面保护层 (PPL) 要求平台**仅**执行受信任的签名代码；这种安全模型在 macOS 上不适用。

内核完整性保护

操作系统内核初始化完成后，会启用内核完整性保护 (KIP) 来帮助防止对内核和驱动器代码进行修改。内存控制器提供了受保护的物理内存区域，供 iBoot 用于载入内核和内核扩展。启动完成后，内存控制器会拒绝对受保护的物理内存区域的写入。应用程序处理器的内存管理单元 (MMU) 被配置为帮助防止从受保护内存区域之外的物理内存中映射权限代码，并帮助防止对内核内存区域内的物理内存进行可写入映射。

为防止重新配置，用于启用 KIP 的硬件会在启动过程完成后锁定。

快速权限访问限制

自 Apple A11 仿生和 S3 后的 SoC 新增了硬件原语。此“快速权限访问限制”原语中包括可基于每个线程快速限制访问权限的 CPU 寄存器。通过快速权限访问限制（也称为 APRR 寄存器），支持的操作系统可从内存移除执行权限，无需通过系统调用和页表寻访或刷新。这些寄存器更进一步减少了来自网页的攻击，尤其是经过运行时编译（即时编译）的代码，因为内存无法在读写的同时有效执行。

系统协处理器完整性保护

协处理器固件会处理许多关键系统任务，例如安全隔区、图像感应处理器和运动协处理器，因此其安全性是整个系统安全性中的关键部分。Apple 使用一种叫做**系统协处理器完整性保护 (SCIP)** 的机制来阻止协处理器固件修改。

SCIP 的工作方式与内核完整性保护 (KIP) 类似：启动时，iBoot 将每个协处理器的固件载入到独立于 KIP 区域并且受保护的保留内存区域。iBoot 会配置每个协处理器的内存单元，以帮助防止：

- 协处理器的受保护内存区域部分外的可执行映射
- 协处理器的受保护内存区域部分内的可写入映射

同时，在启动期间，安全隔区操作系统会用于为安全隔区配置 SCIP。启动过程完成后，用于启用 SCIP 的硬件会锁定。此设计旨在防止重新配置。

指针认证代码

指针认证代码 (PAC) 用来防止对内存损坏错误的利用。系统软件和内建 App 使用 PAC 来帮助防止修改函数指针和返回地址（代码指针）。PAC 使用五种 128 位密钥值来签名内核指令和数据，并且每个用户空间进程都有其自己的 B 类密钥。项目按照如下所示加盐和签名。

项目	键	加盐
函数返回地址	IB	存储地址
函数指针	IA	0
块调用函数	IA	存储地址
Objective-C 方法缓存	IB	存储地址 + 类 + 选择器
C++ 虚函数表条目	IA	存储地址 + 哈希值（损坏的方法名称）
计算的 Goto 标签	IA	哈希值（函数名称）
内核线程状态	GA	•
用户线程状态寄存器	IA	存储地址
C++ 虚函数表指针	DA	0

签名值储存在 64 位指针顶部未使用的填充位中。在使用签名前会进行验证，且填充会恢复以帮助确保指针地址正常工作。验证失败会导致中止使用签名。这种验证提高了许多攻击的难度，如试图通过操纵储存在堆栈中的函数返回地址来欺骗设备恶意执行现有代码的面向返回编程 (ROP) 攻击。

页面保护层

iOS、iPadOS 和 watchOS 中的页面保护层 (PPL) 旨在防止代码签名验证完成后对用户空间代码进行修改。PPL 以内核完整性保护和快速权限访问限制为基础，通过管理页表权限覆盖来确保只有 PPL 才能修改包含用户代码和页表的受保护页面。系统支持实施系统层面代码完整性检查（即使在内核遭到入侵的情况下），从而大大减小了攻击面。此保护在 macOS 上不提供，因为 PPL 只在所有执行代码都必须签名的系统上适用。

其他 macOS 系统安全性功能

其他 macOS 系统安全性功能

macOS 可在更大范围的一系列硬件上运行（例如基于 Intel 的 CPU、基于 Intel 的 CPU 和 Apple T2 安全芯片的组合以及基于 Apple 芯片的 SoC），且支持各种各样的通用计算使用场景。部分用户仅使用基础的预装 App 或 App Store 中提供的 App，另一些用户则是内核专家，他们需要停用几乎所有平台保护措施，从而以最高信任级别运行和测试自己的执行代码。大部分用户都处于这两者之间，且许多用户拥有需要不同访问级别的外围设备和软件。Apple 在设计 macOS 平台时采用的是硬件、软件和服务相集成的方法，该平台在设计上安全可靠，并可方便地进行配置、部署和管理，但也保留了用户需要的可配置性。macOS 还具备 IT 专业人员所需的关键安全技术，方便他们保护公司数据以及集成到安全的企业网络环境中。

以下功能可支持且有助于保护 macOS 用户的各种需求，其中包括：

- 签名系统宗卷安全性
- 系统完整性保护
- 信任缓存
- 外围设备保护
- 搭载 Apple 芯片的 Mac 上的 Rosetta 2（自动转译）支持和安全性
- DMA 支持和保护
- 内核扩展 (Kext) 支持和安全性
- Option ROM 支持和安全性
- 基于 Intel 的 Mac 电脑的 UEFI 固件安全性

系统完整性保护

macOS 使用内核权限并通过叫做**系统完整性保护 (SIP)** 的功能来限制关键系统文件的可写性。这是一项独立功能，与搭载 Apple 芯片的 Mac 上提供的基于硬件的内核完整性保护 (KIP) 不同，后者用于防范内存中的内核修改。系统使用强制访问控制技术提供此功能以及许多其他内核级保护措施，包括沙盒化和数据保险箱。

强制访问控制

macOS 使用强制访问控制，强制访问控制是一系列设定安全访问限制的策略，由开发者创建且无法被覆盖。该方法与自定义访问控制有所不同，后者允许用户根据自己的偏好覆盖安全策略。

强制访问控制对用户不可见，但是这种底层技术有助于实现多项重要的功能，例如沙盒化、家长控制、受管理的偏好设置、扩展以及系统完整性保护。

系统完整性保护

系统完整性保护可对特定关键文件系统位置中的组件进行只读限制，以帮助防止恶意代码修改此类组件。系统完整性保护是特定于电脑的一项设置，用户升级至 OS X 10.11 或更高版本后默认处于开启状态。在基于 Intel 的 Mac 上，停用此设置将移除对物理存储设备上所有分区的保护。macOS 将此安全性策略应用于系统上运行的每个进程，无论进程是在沙盒模式下运行还是使用管理权限运行。

信任缓存

静态信任缓存是安全启动链中包括的一个对象，它是所有在签名系统宗卷中管理的 Mach-O 二进制文件的受信任记录。每个 Mach-O 由一个代码目录哈希值表示。为了进行高效搜索，这些哈希值会在插入信任缓存之前进行排序。代码目录是 `codesign(1)` 执行的签名操作的结果。若要执行信任缓存，SIP 必须保持启用。若要在搭载 Apple 芯片的 Mac 上停用信任缓存执行，安全启动必须配置为“宽松安全性”。

执行二进制文件时（无论是在生成新进程的过程中，还是在将可执行代码映射到现有进程的过程中），会提取其代码目录并计算哈希值。如果生成的哈希值可在信任缓存中找到，则为该二进制文件创建的可执行映射会被授予平台权限；也就是说，这些映射可能拥有任何授权，而且可在无需对签名的真实性进行进一步验证的情况下执行。与之相反，在基于 Intel 的 Mac 上，平台权限由为二进制文件签名的 Apple 证书传递给操作系统内容。（此证书不限制二进制文件可拥有的授权。）

非平台二进制文件（例如已公证的第三方代码）必须拥有有效的证书链才能执行，它们可拥有的授权受 Apple 开发者计划向开发者签发的签名描述文件限制。

macOS 中附带的所有二进制文件都使用**平台标识符**签名。在搭载 Apple 芯片的 Mac 上，此标识符用于指示即使二进制文件由 Apple 签名，但仅当其代码目录哈希值在信任缓存中存在时才能执行。在基于 Intel 的 Mac 上，平台标识符用于执行将二进制文件从较旧的 macOS 版本中定向撤销的操作；这种定向撤销有助于防止这些二进制文件在较新的版本上执行。

静态信任缓存将一组二进制文件完全锁定到给定的 macOS 版本中。此行为可帮助防止较旧版本操作系统中由 Apple 合法签名的二进制文件被引入较新版本的操作系统，进而被攻击者利用。

操作系统之外提供的平台代码

Apple 会提供一些未使用平台标识符签名的二进制文件，例如 Xcode 和开发工具堆栈。即便如此，搭载 Apple 芯片的 Mac 和搭载 T2 芯片的 Mac 上仍然允许这些二进制文件以平台权限执行。由于此平台软件独立于 macOS 提供，因此不受静态信任缓存实施的撤销行为的制约。

可载入信任缓存

Apple 会通过**可载入信任缓存**来提供某些软件包。这些缓存与静态信任缓存的数据结构相同。但是，虽然静态信任缓存只有一个，且其内容在内核早期初始化完成后就始终保证锁定到只读范围，系统中还增加了运行时的可载入信任缓存。

这些信任缓存通过与启动固件认证相同的机制进行认证（使用 Apple 信任的签名服务进行定制化），或者作为全局签名对象进行认证（这些信任缓存的签名不将其绑定到特定设备）。

定制化信任缓存的一个示例是随磁盘映像提供的缓存，用来在搭载 Apple 芯片的 Mac 上执行实地诊断。此信任缓存和磁盘映像经过定制，并在目标 Mac 电脑启动进入诊断模式时载入其内核中。该信任缓存允许磁盘映像内的软件以平台权限运行。

全局签名信任缓存的一个示例是 macOS 软件更新附带的信任缓存。此信任缓存允许软件更新内的一个代码块（**更新主脑**）以平台权限运行。该更新主脑负责任何主机系统无法在不同版本之间一致执行的软件更新规划工作。

Mac 电脑中的外围处理器安全性

所有现代计算机系统都内建了很多外围处理器，专门用于处理联网、图形、电源管理等各种任务。这些外围处理器通常用途单一，远没有主 CPU 性能强大。未实施充分安全保护的内建外围处理器更容易成为被攻击者利用来持续感染操作系统的目标。在感染了外围处理器固件后，攻击者能够以主 CPU 上的软件为攻击目标，或者直接捕获敏感数据（例如，以太网设备可以查看未加密的数据包内容）。

Apple 致力于尽可能减少必要外围处理器的数量，并尽量避免采用需要固件的设计。但在需要使用自带固件的独立处理器时，Apple 采取措施帮助确保攻击者无法持续攻击该处理器。可以通过以下两种方式之一来验证处理器以防范攻击：

- 运行处理器，使其在启动时从主 CPU 下载经过验证的固件
- 让外围处理器实施其独有的安全启动链，以在 Mac 每次启动时验证外围处理器固件

Apple 与供应商合作以审核他们的实施情况，并改善他们的设计以包括必要的属性，例如：

- 确保最低的加密强度
- 确保坚决撤销已知存在漏洞的固件
- 停用调试接口
- 使用储存在受 Apple 控制的硬件安全模块 (HSM) 中的加密密钥对固件签名

近年来 Apple 携手部分外部供应商，使其采用与 Apple 芯片所使用的相同“Image4”数据结构、验证码和签名基础架构。

如果无法进行不涉及储存区的操作或者无法实现储存区安全启动，该设计会强制要求在更新永久储存区前，需要对固件更新进行加密签名并验证。

搭载 Apple 芯片的 Mac 上的 Rosetta 2

搭载 Apple 芯片的 Mac 可通过叫做 **Rosetta 2** 的转译机制来运行针对 x86_64 指令集编译的代码。提供的转译类型有两种：即时和提前。

即时转译

在即时 (JIT) 转译流程中，x86_64 Mach 对象早在映像执行路径中就会被识别。遇到这些映像时，内核会将控制转交给特殊的 Rosetta 转译存根，而不是动态链接编辑器 `dyld(1)`。然后转译存根会在映像执行期间转译 x86_64 页。此转译完全在该过程中进行。内核仍会根据页出错所在二进制文件中所附的代码签名来验证每个 x86_64 页的代码哈希值。如果某个哈希值不匹配，内核会实施适用于该过程的修复策略。

提前转译

在提前 (AOT) 转译流程中，x86_64 二进制文件会在系统认定对该代码响应能力最有益的时间从储存空间中被读取出来。转译后的成品会作为特殊类型的 Mach 对象文件写入储存空间。该文件类似于可执行映像，但会被标记以指示这是另一个映像的转译结果。

在此模型中，AOT 成品会根据原始 x86_64 可执行映像派生出其全部身份信息。为了实施此绑定，有权限的用户空间实体会使用安全隔区管理的设备特定密钥给转译成品签名。此密钥仅对该有权限的用户空间实体可见，该实体使用受限授权进行识别。为该转译成品创建的代码目录包括原始 x86_64 可执行映像的代码目录哈希值。转译成品本身的签名称为**补充签名**。

AOT 流程的开始与 JIT 流程类似，内核将控制转交给 Rosetta 运行环境，而不是动态链接编辑器 `dyld(1)`。但 Rosetta 运行环境随后会向 Rosetta 系统服务发送进程间通信 (IPC) 查询，询问当前可执行映像是否有可用的 AOT 转译。如果找到，Rosetta 服务会向该转译提供一个句柄，以将其映射到过程中并执行。执行期间，内核为转译成品计算代码目录哈希值，该哈希值经过植根于设备特定签名密钥的签名认证。此过程不涉及原始 x86_64 映像的代码目录哈希值。

转译成品会储存在数据保险箱中，运行时除 Rosetta 服务外任何实体都无法访问。Rosetta 服务通过向单个转译成品分发只读文件描述符来管理其缓存的访问权限；这限制了对 AOT 成品缓存的访问。此服务的进程间通信和从属足迹有意保持在非常窄的范围，以限制其攻击面。

如果原始 x86_64 映像的代码目录哈希值与编码到 AOT 转译成品签名中的哈希值不匹配，此结果会被视为等同于无效代码签名，此时会执行适当的实施操作。

如果远程过程向内核查询经 AOT 转译的可执行内容的授权或其他代码身份属性，内核会向其返回原始 x86_64 映像的身份属性。

静态信任缓存内容

macOS 11 或更高版本附带 Mach “胖”二进制文件，其中包含 x86_64 和 arm64 电脑代码的分段。在搭载 Apple 芯片的 Mac 上，用户可能会决定通过 Rosetta 流程执行系统二进制文件的 x86_64 分段，例如要载入没有原生 arm64 变体的插件时。为了支持这一方法，macOS 附带的静态信任缓存一般会为每个 Mach 对象文件包含三个代码目录哈希值：

- arm64 分段的代码目录哈希值
- x86_64 分段的代码目录哈希值
- x86_64 分段 AOT 转译的代码目录哈希值

Rosetta AOT 转译过程是确定的，针对任何给定输入都会重现相同的输出，无论什么时间或在什么设备上执行转译。

macOS 构建期间，每个 Mach 对象文件都通过与被构建 macOS 版本关联的 Rosetta AOT 转译流程运行，生成的代码目录哈希值会记录到信任缓存中。为提高效率，实际转译结果不会附带在操作系统中，而是在用户请求时按需重新整合。

在搭载 Apple 芯片的 Mac 上执行 x86_64 映像时，如果该映像的代码目录哈希值位于静态信任缓存中，则生成的 AOT 成品的代码目录哈希值**也**应该在静态信任缓存中。此类结果不会由设备特定密钥签名，因为签名权限植根于 Apple 安全启动链中。

未签名 x86_64 代码

搭载 Apple 芯片的 Mac 不允许原生 arm64 代码执行，除非其附带有效的签名。此签名可以像 ad hoc 代码签名（请参阅 `codesign(1)`）一样简单，该类代码签名中不包含来自非对称密钥对中私钥部分的任何实际身份（而只是二进制文件的一个未认证测量值）。

为提高二进制文件兼容性，转译后的 x86_64 代码被允许在没有任何签名信息的情况下通过 Rosetta 执行。设备特定安全隔区签名过程中不会将特定身份信息传递到此代码中，代码执行的限制条件与原生未签名代码在基于 Intel 的 Mac 上执行时完全相同。

Mac 电脑的直接内存访问保护

为了在诸如 PCIe、FireWire、雷雳和 USB 等高速接口上实现高吞吐量，电脑必须支持外围设备的直接内存访问 (DMA)。也就是说，外围设备必须能够在没有 CPU 持续参与的情况下读取和写入 RAM。自 2012 年以来，Mac 电脑采用了多项技术来保护 DMA，实现了 PC 领域内最好且最为全面的 DMA 保护。

搭载 Apple 芯片的 Mac 的直接内存访问保护

Apple 片上系统针对系统中的每个 DMA 代理（包括 PCIe 和雷雳端口）包含一个输入/输出内存管理单元 (IOMMU)。由于每个 IOMMU 有自己的一套地址转换表以转换 DMA 请求，通过 PCIe 或雷雳连接的外围设备只能访问明确映射给其使用的内存。外围设备无法访问属于系统其他部分（如内核或固件）的内存或者分配给其他外围设备的内存。如果 IOMMU 检测到外围设备试图访问未映射给该外围设备使用的内存，会触发内核崩溃。

基于 Intel 的 Mac 的直接内存访问保护

基于 Intel 且支持 Intel 定向 I/O 虚拟化技术 (VT-d) 的 Mac 电脑在启动过程的非常早期阶段会初始化 IOMMU（启用 DMA 重映射）然后中断重映射，以减少各种类别的安全性漏洞。Apple IOMMU 硬件开始以默认拒绝策略运行，因此系统接通电源时，IOMMU 会立即自动开始阻止来自外围设备的 DMA 请求。软件初始化 IOMMU 后，IOMMU 将开始允许外围设备想要访问明确映射给其使用的内存区域的 DMA 请求。

【注】中断 PCIe 重映射在搭载 Apple 芯片的 Mac 上并非必要，因为每个 IOMMU 只处理自己外围设备的 MSI。

自 macOS 11 起，如果辅助 DMA 的 UEFI 驱动器与外部设备配对，所有搭载 Apple T2 安全芯片的 Mac 电脑都会在受限的 Ring 3 环境中运行此类固件。此属性有助于减少恶意设备在启动时以非预期方式与 UEFI 驱动器交互可能导致的安全性漏洞。它尤其减少了驱动器处理 DMA 缓冲时导致的漏洞的影响。

macOS 中的内核扩展

自 macOS 11 起，如果第三方内核扩展 (Kext) 启用，就不能按需将其载入内核中。Kext 会转而合并到启动过程中载入的**辅助内核集 (AuxKC)** 中。在搭载 Apple 芯片的 Mac 上，AuxKC 的测量值会签名到 LocalPolicy 中（在之前的硬件中，AuxKC 位于数据宗卷上）。重建 AuxKC 需要用户批准、重新启动 macOS 以将更改载入内核，以及将安全启动配置为“降低安全性”。

【重要事项】macOS 中不再推荐使用 Kext。Kext 会威胁到操作系统的完整性和可靠性，Apple 建议用户选择不需要扩展内核的解决方案。

搭载 Apple 芯片的 Mac 中的内核扩展

在搭载 Apple 芯片的 Mac 上，必须通过在启动时按住电源按钮进入“第一真正 recoveryOS”（1TR）模式，然后降级到“降低安全性”并勾选启用内核扩展的复选框，才能明确启用 Kext。此操作还需要输入管理员密码以授权降级。1TR 和密码要求的结合使从 macOS 内部开始的仅软件层面攻击者难以将 Kext 插入 macOS，然后也就无法利用这些 KEXT 来获取内核特权。

用户授权 Kext 载入后，以上“用户批准的内核扩展载入”流程将用于授权 Kext 安装。用于以上流程的授权还用于捕捉 LocalPolicy 中用户授权 Kext 列表 (UAKL) 的 SHA384 哈希值。然后，内核管理监控程序 (`kmd`) 负责验证 Kext（仅限 UAKL 中找到的 Kext），以将其包括在 AuxKC 中。

- 如果系统完整性保护 (SIP) 已启用，在将每个 Kext 包括在 AuxKC 中前还会验证其签名。
- 如果 SIP 停用，Kext 签名不会实施。

此方法可让不是 Apple 开发者计划成员的开发者或用户在“宽松安全性”流程中测试 Kext，然后再为其签名。

AuxKC 创建后，其测量值会发送到安全隔区进行签名，并添加到启动时可由 iBoot 进行评估的 Image4 数据结构中。生成 Kext 接收项也是 AuxKC 构建的一部分。此接收项内含实际包括在 AuxKC 中的 Kext 列表，因为如果遇到了被禁止的 Kext，这一组 Kext 可能是 UAKL 的子集。LocalPolicy 中包括 AuxKC Image4 数据结构的 SHA384 哈希值和 Kext 接收项。iBoot 使用 AuxKC Image4 哈希值在启动时进行额外验证，以帮助确保不可能通过较新的 LocalPolicy 来启动由安全隔区签名的 AuxKC Image4 文件较旧版本。Apple Pay 等子系统使用 Kext 接收项确定当前是否载入了可能干扰 macOS 可信度的任何 Kext。如果存在此类 Kext，Apple Pay 功能可能会停用。

Kext 替代项（macOS 10.15 或更高版本）

macOS 10.15 允许开发者通过安装和管理在用户空间而非内核层面运行的系统扩展来扩展 macOS 的功能。让系统扩展运行在用户空间提高了 macOS 的稳定性和安全性。尽管 Kext 本身拥有对整个操作系统的完全访问权限，但运行在用户空间的扩展只拥有执行其特定功能所需的权限。

开发者可以使用各种框架（包括 DriverKit、EndpointSecurity 和 NetworkExtension）来写入 USB 和人机界面驱动程序、端点安全工具（如数据丢失预防或其他端点代理）以及 VPN 和网络工具，这一切都不需要编写 Kext。只有当第三方安全代理使用了这些 API 或者拥有过渡到这些 API 并不再使用内核扩展的清晰路线图，才应当使用这些第三方安全代理。

用户批准的内核扩展载入

为了提高安全性，需要得到用户同意才能载入随 macOS 10.13 安装或在其后安装的内核扩展。此过程称为**用户批准的内核扩展载入**。批准内核扩展需要管理员授权。满足以下条件的内核扩展不需要授权：

- 在 Mac 运行 macOS 10.12 或更低版本时安装
- 替换之前批准的扩展
- 允许无需用户同意便可载入（通过使用 Mac 从 recoveryOS 启动时提供的 `spctl` 命令行工具实现）
- 被允许使用移动设备管理 (MDM) 配置载入

  从 macOS 10.13.2 开始，用户可以使用 MDM 来指定无需用户同意便可载入的内核扩展列表。此选项要求 Mac 运行 macOS 10.13.2，且通过“Apple 校园教务管理”或“Apple 商务管理”在 MDM 中注册，或者由用户自行在 MDM 中注册。

macOS 中的 Option ROM 安全性

【注】搭载 Apple 芯片的 Mac 当前不支持 Option ROM。

搭载 Apple T2 安全芯片的 Mac 中的 Option ROM 安全性

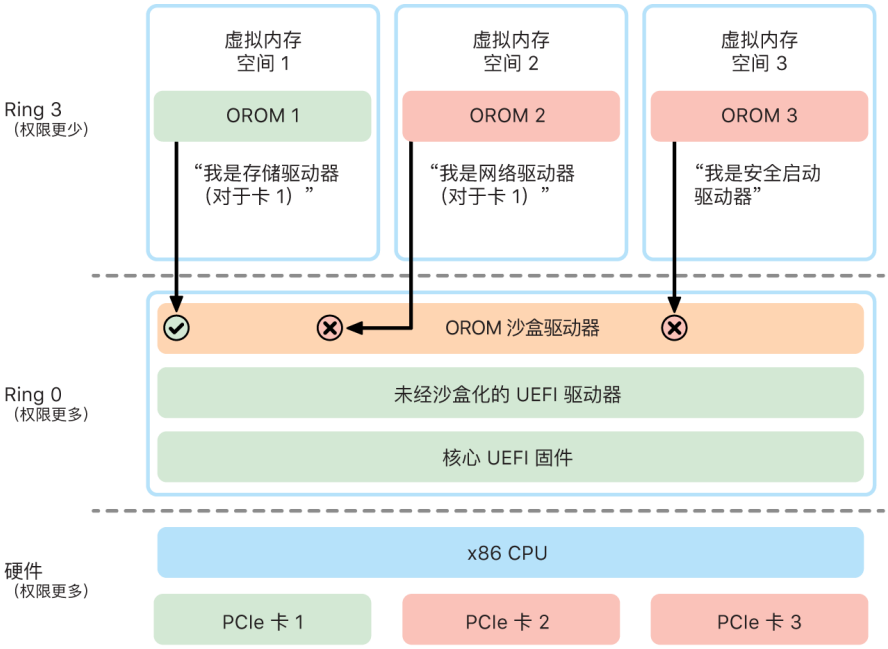
雷雳和 PCIe 设备都可以通过物理方式连接“Option ROM (OROM)”。（这通常不是真正的 ROM，而是储存固件的可重写芯片。）在基于 UEFI 的系统中，该固件通常是 UEFI 驱动程序，由 UEFI 固件读入并执行。执行的代码应该初始化并配置从中获取代码的硬件，以使硬件供其他固件使用。专门的第三方硬件需要借助此功能在启动的最初阶段载入并运行，例如从外置 RAID 阵列启动。

但是由于 OROM 通常可重写，如果攻击者改写了合法外围设备的 OROM，攻击者的代码就可在启动过程的初期执行，从而能够篡改执行环境并破坏后续载入软件的完整性。与之类似，如果攻击者将自己的恶意设备引入到系统中，他们也能够执行恶意代码。

在 macOS 10.12.3 中，2011 年之后销售的 Mac 电脑在启动时默认不会执行 OROM，除非按下特殊的组合键。这个组合键可以防范在 macOS 启动序列中无意引入恶意 OROM。而“固件密码实用工具”的默认行为也发生了更改，当用户设定了固件密码后，即使按下组合键也不能执行 OROM。这可以防范处于电脑前的攻击者故意引入恶意 OROM。对于设定了固件密码但仍需运行 OROM 的用户，可以在 macOS 中使用 `firmwarepasswd` 命令行工具来配置非默认选项。

OROM 沙盒安全性

macOS 10.15 更新了 UEFI 固件以包含在沙盒中运行 OROM 并取消 OROM 特权的机制。UEFI 固件通常以 CPU 最高特权等级（称为 Ring 0）执行包括 OROM 在内的所有代码，其所有代码和数据还具有单独的共享虚拟内存空间。Ring 0 是 macOS 内核运行的特权等级，而 Ring 3 则是 App 运行的较低特权等级。OROM 沙盒通过像内核一样利用单独的虚拟内存来降低 OROM 的特权，使其以 Ring 3 特权运行。



沙盒进一步大幅限制 OROM 可以调用的接口（与内核中的系统调用过滤非常类似）以及 OROM 可以注册为的设备类型（与 App 批准非常类似）。这样设计的好处是恶意 OROM 不再能够在 Ring 0 内存中的任何位置直接写入。它们会被限制到定义完备的非常窄的沙盒接口。这种受限的接口极大地减小了攻击面，迫使攻击者先逃离沙盒并提升特权。

基于 Intel 的 Mac 中的 UEFI 固件安全性

基于 Intel 且搭载 Apple T2 安全芯片的 Mac 通过 UEFI (Intel) 固件提供安全保护。

概览

自 2006 年起，配备基于 Intel CPU 的 Mac 电脑使用 Intel 固件，这些固件基于可扩展固件接口 (EFI) 开发工具包 (EDK) v1 或 v2。基于 EDK2 的代码符合统一可扩展固件接口 (UEFI) 技术规范。这一部分中的 **UEFI 固件**指的是 Intel 固件。UEFI 固件曾是 Intel 芯片上执行的第一段代码。

在基于 Intel 且不搭载 Apple T2 安全芯片的 Mac 上，UEFI 固件的信任根是储存固件的芯片。Apple 会对 UEFI 固件更新进行数字签名，并会在更新储存区之前经过固件的验证。为帮助防止回滚攻击，更新的版本必须始终比现有的版本高。但是，如果攻击者可以实际接触 Mac，便可能使用硬件连接到固件储存芯片并更新芯片以包含恶意内容。与之类似，如果在 UEFI 固件的启动初期发现了漏洞（限制写入储存芯片之前），还可能会导致 UEFI 固件的持续感染。这种硬件架构缺陷常见于大多数基于 Intel 的 PC 中，在基于 Intel 且不搭载 T2 芯片的所有 Mac 电脑上也存在。

为了帮助防止破坏 UEFI 固件的物理攻击，Mac 电脑经过重新架构以将 UEFI 固件中的信任植根于 T2 芯片中。在这些 Mac 电脑上，UEFI 固件的信任根专指 T2 固件；请参阅基于 Intel 的 Mac 的启动过程。

Intel 管理引擎 (ME) 子组件

Intel 管理引擎 (ME) 固件是储存在 UEFI 固件中的一个子组件。ME 是 Intel 芯片内的一个独立处理器和子系统，在只搭载了基于 Intel 显卡的 Mac 上主要用于音频和视频版权保护。为减小此子组件的攻击面，基于 Intel 的 Mac 运行的是自定 ME 固件，其中大部分组件已被移除。由此形成的 Mac ME 固件比 Intel 提供的默认最小版本还要小，因此过去作为安全研究员公开攻击对象的许多组件都不再存在。

系统管理模式 (SMM)

Intel 处理器有一种有别于正常操作的特殊执行模式。这种模式称为**系统管理模式 (SMM)**，其最初推出是为了处理具有强时效性的操作，如电源管理。但是，为了执行此类操作，Mac 电脑此前一直使用称为**系统管理控制器 (SMC)** 的独立微控制器。现在，SMC 不再是一个单独的微控制器，而是集成到了 T2 芯片中。

watchOS 系统安全性

Apple Watch 使用与 iOS 和 iPadOS 相同的许多基于硬件的平台安全性功能。例如，Apple Watch：

- 执行安全启动和安全软件更新
- 维持操作系统完整性
- 帮助保护设备上以及与配对的 iPhone 和互联网通信期间的数据

支持的技术包括“系统安全性”中所列技术（例如 KIP、SKP 和 SCIP）以及数据保护、钥匙串和网络技术。

更新 watchOS

watchOS 可配置为在夜间更新。有关 Apple Watch 密码是如何储存以及用于更新过程的更多信息，请参阅密钥包。

手腕检测

如果启用了手腕检测，设备从用户的手腕取下后会立即自动锁定。如果停用了手腕检测，控制中心会提供锁定 Apple Watch 的选项。Apple Watch 锁定后，只有通过在 Apple Watch 上输入密码才能使用 Apple Pay。用户可以在 iPhone 上的 Apple Watch App 中关闭手腕检测，也可以使用移动设备管理 (MDM) 解决方案来强制实施此设置。

激活锁

在 iPhone 上打开“查找”后，其配对的 Apple Watch 可以使用激活锁。激活锁使任何人都难以使用或出售丢失或被盗的 Apple Watch。激活锁需要用户的 Apple ID 和密码来取消配对、抹掉或重新激活 Apple Watch。

与 iPhone 安全配对

Apple Watch 一次只能与一台 iPhone 配对。取消配对 Apple Watch 时，iPhone 会发送抹掉 Apple Watch 所有内容和数据的指令。

Apple Watch 和 iPhone 的配对通过带外处理交换公钥和低功耗蓝牙 (BLE) 链接共享密钥进行保护。Apple Watch 显示一幅动画图案供 iPhone 摄像头捕捉。该图案包含加密的密钥，用于 BLE 4.1 带外配对。如果需要，Apple Watch 会使用标准 BLE 万能钥匙进入模式作为备用配对方法。

BLE 会话建立且使用《蓝牙核心规范》中可用的最高级安全协议加密后，iPhone 和 Apple Watch 会通过以下任一过程交换密钥：

- 改编自 Apple 身份识别服务 (IDS) 的过程（如 iMessage 信息安全性概览中所述）。
- 使用 IKEv2/IPsec 的密钥交换过程。初始密钥交换使用蓝牙会话密钥（用于配对场景）或 IDS 密钥（用于操作系统更新场景）认证。每台设备生成一个随机 256 位 Ed25519 公私密钥对，初始密钥交换过程中会交换公钥。

【注】密钥交换和加密所使用的机制有所不同，具体取决于 iPhone 和 Apple Watch 的操作系统版本。运行 iOS 13 或更高版本的 iPhone 设备与运行 watchOS 6 或更高版本的 Apple Watch 配对时仅使用 IKEv2/IPsec 来进行密钥交换和加密。

密钥交换后：

- 蓝牙会话密钥被丢弃，iPhone 和 Apple Watch 间的所有通信（通过加密蓝牙、无线局域网和提供二级加密层的蜂窝链路）使用以上列出的一种方法加密。
- （仅限 IKEv2/IPsec）密钥储存在“系统”钥匙串中，用于认证以后设备之间的 IKEv2/IPsec 会话。在与 Apple Watch Series 4 或后续机型（运行 watchOS 8 或更高版本）配对的 iPhone 设备（运行 iOS 15 或更高版本）上，这些设备间的进一步通信通过 AES-256-GCM 或 ChaCha20-Poly1305（256 位密钥）加密并保护完整性。

蓝牙低功耗设备地址每隔 15 分钟更新一次，以降低有人广播永久标识符导致设备被本地跟踪的风险。

为支持需要流传输数据的 App，加密采用了 FaceTime 通话安全性中所描述的方法，即使用配对 iPhone 所提供的 Apple 身份识别服务 (IDS) 或直接互联网连接。

Apple Watch 对文件和钥匙串项采用硬件加密储存方式以及基于类的保护。同时对钥匙串项还使用了访问控制密钥包。Apple Watch 和 iPhone 间通信使用的密钥也采用了基于类的保护进行加密。有关更多信息，请参阅用于数据保护的密钥包。

自动解锁和 Apple Watch

为了在使用多台 Apple 设备时更加方便，部分设备在某些情况下可以自动解锁其他设备。自动解锁支持以下三种用途：

- iPhone 可以解锁 Apple Watch。
- Apple Watch 可以解锁 Mac。
- 当检测到用户的鼻子和嘴巴被遮挡时，Apple Watch 可以解锁 iPhone。

所有这三种使用场景都建立在相同的基础上：相互认证的站对站 (STS) 协议，它会在启用功能时交换长期密钥，并为每个请求协商唯一的临时会话密钥。无论使用哪种底层通信通道，STS 隧道都是在两台设备中的安全隔区之间直接协商，且所有加密材料都保留在该安全域内（未搭载安全隔区的 Mac 电脑除外，此类 Mac 电脑会在内核中终止 STS 隧道）。

解锁

完整的解锁序列可以分为两个阶段。首先，被解锁的设备（“目标”）会生成一个加密的解锁密钥，并将其发送到执行解锁的设备（“发起端”）。之后，发起端会使用先前生成的密钥执行解锁。

若要准备自动解锁，设备需要使用 BLE 连接相互连接。然后，目标设备随机生成的 32 字节解锁密钥将通过 STS 隧道发送给发起端。在下一次生物识别或密码解锁期间，目标设备会使用解锁密钥封装其密码派生密钥 (PDK)，并丢弃其内存中的解锁密钥。

为了执行解锁，设备会发起新的 BLE 连接，然后使用点对点无线局域网来安全估计彼此之间的距离。如果设备在指定范围内且满足所需的安全性策略，发起端会通过 STS 隧道将其解锁密钥发送给目标设备。然后，目标设备会生成一个新的 32 字节解锁密钥并将其返回给发起端。如果由发起端发送的当前解锁密钥成功解密了解锁记录，则目标设备将被解锁，PDK 将使用新的解锁密钥重新进行封装。最后，目标设备内存中新的解锁密钥和 PDK 将被丢弃。

Apple Watch 自动解锁安全性策略

为了增加便利，iPhone 可在 Apple Watch 首次启动后直接解锁它，无需用户先在 Apple Watch 上输入密码。为此，会使用随机解锁密钥（在启用该功能后的第一个解锁序列期间生成）来创建长期托管记录，该记录储存在 Apple Watch 钥匙包中。托管记录密钥储存在 iPhone 钥匙串中，并在每次 Apple Watch 重新启动后用于引导新会话。

iPhone 自动解锁安全性策略

通过 Apple Watch 自动解锁 iPhone 还应用了其他安全性策略。Apple Watch 无法用于代替 iPhone 上的面容 ID 进行其他操作，如 Apple Pay 或 App 授权。当 Apple Watch 成功解锁配对的 iPhone 时，手表会显示一条通知并播放关联的触感。如果用户轻点通知中的“锁定 iPhone”按钮，手表会通过 BLE 向 iPhone 发送锁定命令。iPhone 收到锁定命令时，它会锁定并停用面容 ID 和使用 Apple Watch 解锁。下一次 iPhone 解锁必须使用 iPhone 密码执行。

使用 Apple Watch 成功解锁配对的 iPhone（启用时）需要满足以下条件：

- 戴上关联的 Apple Watch 并解锁后，必须至少使用另一种方法解锁 iPhone 一次。
- 传感器必须能够检测到鼻子和嘴巴被遮挡。
- 测量距离必须在 2–3 米或更短
- Apple Watch 不得处于就寝模式。
- Apple Watch 或 iPhone 最近必须解锁过，或者 Apple Watch 必须已检测到肢体运动，表示佩戴者处于活动状态（例如，未入睡）。
- iPhone 必须在过去 6.5 个小时内至少解锁一次。
- iPhone 必须处于允许面容 ID 执行设备解锁的状态。（有关更多信息，请参阅面容 ID、触控 ID 和密码。）

使用 Apple Watch 在 macOS 中批准

启用使用 Apple Watch 自动解锁后，Apple Watch 可替代或者配合触控 ID 来批准以下授权和认证提示：

- macOS 和 Apple App 要求授权
- 第三方 App 要求认证
- 存储的 Safari 浏览器密码
- 安全备忘录

安全使用无线局域网、蜂窝网络、iCloud 和 Gmail

当 Apple Watch 不在蓝牙通信范围内时，可以转而使用无线局域网或蜂窝网络。如果配对 iPhone 已经加入过某个无线局域网，且其网络凭证在两台设备处于相互覆盖范围内时已经同步到Apple Watch，Apple Watch 会自动加入此网络。此“自动加入”行为可稍后在 Apple Watch “设置” App 中的“无线局域网”部分对每个网络进行配置。两台设备从未加入过的无线局域网可在 Apple Watch “设置” App 中的“无线局域网”部分手动加入。

Apple Watch 和 iPhone 未在通信范围内时，Apple Watch 会直接接入 iCloud 和 Gmail 服务器来取回邮件，而不是通过互联网与配对 iPhone 同步邮件数据。对于 Gmail 帐户，用户必须在 iPhone 上 Watch App 中的“邮件”部分进行谷歌认证。从谷歌收到的 OAuth 令牌会以加密格式通过 Apple 身份识别服务 (IDS) 发送到 Apple Watch，以用于取回邮件。此 OAuth 令牌绝不会用于从配对 iPhone 连接 Gmail 服务器。

随机数生成

密码学伪随机数生成器 (CPRNG) 是安全软件的重要基石。Apple 为此提供了运行在 iOS、iPadOS、macOS、Apple tvOS 和 watchOS 内核中的受信任软件 CPRNG。它负责聚合系统中的原始熵并为内核和用户空间的使用者提供安全的随机数。

熵源

内核 CPRNG 源自启动过程中的多个熵源并存在于设备的整个生命周期。这些来源包括（取决于可用性）：

- 安全隔区硬件 TRNG
- 启动过程中所收集基于时序的时间误差
- 从硬件中断收集的熵
- 用于启动过程中保持熵的种子文件
- Intel 随机指令，例如 RDSEED 和 RDRAND（仅限基于 Intel 的 Mac）

内核 CPRNG

内核 CPRNG 的设计源自 Fortuna 算法，旨在满足 256 位安全级别。它使用以下 API 为用户空间使用者提供高质量的随机数：

- `getentropy`(2) 系统调用
- 随机设备 (/dev/random)

内核 CPRNG 通过写入随机设备接受用户提供的熵。

Apple 安全性研究设备

Apple 安全性研究设备是一台接线特殊的 iPhone，可让安全研究员在 iOS 上执行研究，而无需破坏或停用 iPhone 的平台安全性功能。通过此设备，研究员可侧载需要平台对等权限来运行的内容，从而在与生产设备上的平台更接近的平台上执行研究。

为帮助确保用户设备不受安全性研究设备执行策略的影响，策略更改会在 iBoot 变体和启动内核集中实施。它们无法在用户硬件上启动。研究 iBoot 会检查新的接线状态，如果在出于非研究目的接线的硬件上运行，该 iBoot 会进入错误循环。

Cryptex 子系统可让研究员载入定制化信任缓存和包含对应内容的磁盘映像。为确保不允许此子系统在用户设备上执行，系统已实施多种深入防护措施：

- 如果检测到普通用户设备，`launchd` 不会载入 `cryptexd` launchd 属性列表。
- 如果检测到普通用户设备，`cryptexd` 会中止运行。
- `AppleImage4` 不会在普通用户设备上发布用于验证研究 cryptex 的随机数。
- 签名服务器会拒绝为不在明确允许列表上的设备定制 cryptex 磁盘映像。

为尊重安全研究员的隐私，只有可执行内容或内核缓存的测量值（例如哈希值）以及安全性研究设备标识符会在定制化期间发送给 Apple。Apple 不会收到正在载入到设备上的 cryptex 内容。

为避免恶意方试图将研究设备伪装成用户设备以欺骗目标方在日常生活中使用该设备，安全性研究设备有以下不同之处：

- 安全性研究设备只能在充电期间启动。充电可使用闪电线缆或兼容 Qi 的充电器。如果设备启动期间未在充电，会进入恢复模式。如果用户开始为设备充电并重新启动，设备会正常启动。只要 XNU 启动，设备无需充电即可继续运行。
- iBoot 启动期间，文字**“安全性研究设备”**会显示在 Apple 标志下方。
- XNU 内核以详细模式启动。
- 设备侧面蚀刻了一条信息：“Property of Apple.Confidential and Proprietary.Call +1 877 595 1125.”

以下是启动后显示的软件中实施的其他措施：

- 设备设置期间显示文字**“安全性研究设备”**。
- 锁定屏幕上和“设置” App 中显示文字**“安全性研究设备”**。

安全性研究设备为研究员提供用户设备不具备的以下功能。研究员可以：

- 通过与 Apple 操作系统组件相同权限级别的任意授权来将可执行代码侧载到设备上
- 启动时开始服务
- 重新启动后保留内容
- 使用 `research.com.apple.license-to-operate` 授权允许进程调试系统上的任何其他进程，包括系统进程。

  只有 AppleMobileFileIntegrity 内核扩展的 RESEARCH 变体遵守 `research.` 命名空间；用户设备上签名验证期间任何带有此授权的进程都会被终止。
- 定制和恢复自定内核缓存

加密和数据保护

加密和数据保护概览

安全启动链、系统安全性和 App 安全性功能都有助于验证只有受信任的代码及 App 可以在设备上运行。Apple 设备还有更多加密功能来保护用户数据的安全，即使安全性基础架构的其他部分遭到入侵（例如设备丢失或运行了不受信任的代码）。所有这些功能对用户和 IT 管理员都大有助益，它可保护个人和企业信息，而且有办法在设备被盗或丢失时立即进行彻底的远程擦除。

iOS 和 iPadOS 设备使用称为**数据保护**的文件加密方法，基于 Intel 的 Mac 上的数据则通过称为**文件保险箱**的宗卷加密技术进行保护。搭载 Apple 芯片的 Mac 使用支持数据保护的混合模型，但请注意两点：不支持最低保护级别 (D) 类；默认级别（C 类）的作用类似于基于 Intel 的 Mac 上的文件保险箱，实际上使用宗卷密钥。在任何情况下，密钥管理层次都植根于安全隔区中的专用芯片，且专用 AES 引擎支持线速加密并帮助确保长期有效的加密密钥不会暴露给内核操作系统或 CPU（否则可能遭到入侵）。（搭配 T1 或没有安全隔区的基于 Intel 的 Mac 将不使用专用芯片保护文件保险箱加密密钥。）

除了使用数据保护和文件保险箱来帮助阻止未经授权的数据访问，Apple 还使用**操作系统内核**强制执行保护和安全措施。内核使用访问控制来沙盒化 App（限制 App 可访问的数据），还使用一种称为**数据保险箱**的机制来限制所有其他提出请求的 App 访问某一 App 的数据（而不是限制某个 App 可进行的调用）。

密码

为了保护用户数据不受恶意攻击，Apple 在 iOS、iPadOS 和 macOS 中使用了密码。密码越长，强度越高，便更容易阻止暴力破解攻击。为了进一步阻止攻击，Apple 强制执行时间延迟（针对 iOS 和 iPadOS）和密码尝试次数限制（针对 Mac）。

在 iOS 和 iPadOS 中，用户设置一个设备密码，即自动启用了数据保护。在具有 Apple 片上系统 (SoC) 的其他设备（如搭载 Apple 芯片的 Mac、Apple TV 和 Apple Watch）上也会启用数据保护。在 macOS 中，Apple 使用内建宗卷加密程序**文件保险箱**。

强密码如何提高安全性

iOS 和 iPadOS 支持 6 位、4 位和任意长度的字母数字密码。除了用于给设备解锁，密码还为特定的加密密钥提供熵。这意味着攻击者即使拿到设备，在没有密码的情况下也无法访问某些特定保护类的数据。

密码与设备的 UID 配合使用，因此在受到攻击的设备上只能进行暴力尝试。为此，系统使用较大的迭代次数来延缓每次尝试。迭代次数已经校准过，使得每次尝试约耗时 80 毫秒。事实上，尝试 6 位字符（包含小写字母和数字）字母数字密码的全部组合将耗时超过五年半。

用户密码的强度越大，加密密钥的强度就越高。而且通过使用面容 ID 和触控 ID，用户可创建一个比实际密码安全性高很多的密码。此强密码对数据保护所用加密密钥提供保护的有效熵的数量得以增加，而且不会对一天中多次解锁设备的用户体验产生负面影响。

如果输入较长的纯数字密码，锁定屏幕上会显示数字小键盘，而非全键盘。与较短的字母数字密码相比，较长的数字密码可能更容易输入，而且可以提供类似的安全性。

用户可以指定更长的字母数字密码，方法是在“设置”>“触控 ID 与密码”或“面容 ID 与密码”的“密码选项”中选择“自定字母数字密码”。

逐步增加延迟时间可如何阻止暴力破解攻击（iOS、iPadOS）

在 iOS 和 iPadOS 中，为了进一步阻止对密码的暴力破解攻击，在锁定屏幕上输入无效密码后的延迟时间会逐步增加，如下表所示。

尝试次数	执行的延迟
1–4	无
5	1 分钟
6	5 分钟
7–8	15 分钟
9	1 小时

如果（在“设置”>“触控 ID 与密码”中）打开了“抹掉数据”选项，则当连续 10 次尝试输入错误的密码后，储存的所有内容和设置会移除。连续尝试同一错误密码不计入次数限制内。此设置还可作为管理策略通过支持此功能的移动设备管理 (MDM) 解决方案和通过 Microsoft Exchange ActiveSync 提供，而且可设置为较低的阈值。

在搭载安全隔区的设备上，延迟由安全隔区执行。如果设备在定时延迟期间重新启动，延迟仍然执行，且定时器从当期重新计时。

逐步增加延迟时间可如何阻止暴力破解攻击 (macOS)

为了帮助阻止暴力破解攻击，当 Mac 启动时，登录窗口中或使用目标磁盘模式时最多允许 10 次密码尝试，并且在错误密码的输入达到一定次数后，延迟时间会逐步增加。延迟由安全隔区执行。如果 Mac 在定时延迟期间重新启动，延迟仍然执行，且定时器从当期重新计时。

下表显示了搭载 Apple 芯片和搭载 T2 芯片的 Mac 上的密码尝试次数间的延迟。

尝试次数	执行的延迟
5	1 分钟
6	5 分钟
7	15 分钟
8	15 分钟
9	1 小时
10	已停用

为了帮助防止恶意软件通过尝试攻击用户密码来导致永久数据丢失，这些限制在用户成功登录 Mac 后不会执行，但会在重启后重新执行。如果 10 次尝试失败，启动进入 recoveryOS 后会增加 10 次尝试。如果仍失败，每个文件保险箱恢复机制 (iCloud 恢复、文件保险箱恢复密钥和机构密钥) 将各增加 10 次额外尝试，总的新增次数最多为 30 次。这些额外尝试用尽后，安全隔区将不再处理任何解密宗卷或验证密码的请求，驱动器上的数据变为不可恢复。

若要帮助保护企业设置中的数据，IT 应使用 MDM 解决方案定义和实施文件保险箱配置策略。组织有若干加密宗卷管理选项，包括机构恢复密钥、个人恢复密钥 (可选择由 MDM 托管储存) 或两者结合。密钥循环也可以设为 MDM 中的一种策略。

在搭载 Apple T2 安全芯片的 Mac 上，密码提供类似的功能，区别在于生成的密钥用于文件保险箱加密，而不是数据保护。macOS 还提供其他密码恢复选项：

- iCloud 恢复
- 文件保险箱恢复
- 文件保险箱机构密钥

数据保护

数据保护概览

Apple 使用称为数据保护的技术来保护储存在搭载 Apple SoC 的设备（如 iPhone、iPad、Apple Watch、Apple TV 和搭载 Apple 芯片的 Mac）上闪存中的数据。通过数据保护，设备可以响应来电等常见事件，同时针对用户数据提供高级别加密。某些系统 App（例如“信息”、“邮件”、“日历”、“通讯录”、“照片”）和“健康”数据值默认使用数据保护。第三方 App 自动受到此类保护。

实施

数据保护通过构建和管理密钥层级来实施，并建立在 Apple 设备内建的硬件加密技术基础上。它通过将某个类分配给每个文件来实现对文件的逐个控制；可访问性根据该类密钥是否已解锁确定。APFS（Apple 文件系统）使文件系统可进一步以各个范围为基础对密钥进行细分（文件的各个部分可以拥有不同的密钥）。

每次在数据宗卷中创建文件时，数据保护都会创建一个新的 256 位密钥（**文件独有密钥**），并将其提供给硬件 AES 引擎，此引擎会使用该密钥在文件写入闪存时对其进行加密。在搭载 A14、A15 和 M1 系列的设备上，加密在 XTS 模式中使用 AES-256，其中 256 位文件独有密钥通过密钥派生功能 (NIST Special Publication 800-108) 派生出一个 256 位 tweak 密钥和一个 256 位 cipher 密钥。采用 A9 到 A13、S5、S6 和 S7 的每一代硬件在 XTS 模式中使用 AES-128，其中 256 位文件独有密钥会被拆分，以提供一个 128 位 tweak 密钥和一个 128 位 cipher 密钥。

在搭载 Apple 芯片的 Mac 上，数据保护默认为 C 类（请参阅数据保护类），但使用宗卷密钥，而非范围独有密钥或文件独有密钥，可为用户数据高效重建文件保险箱安全模型。用户仍须选择使用文件保险箱，以获得加密密钥层级搭配用户密码的全面保护。开发者也可以选择使用文件独有密钥或范围独有密钥的更高保护类。

Apple 设备中的数据保护

在支持数据保护的 Apple 设备上，每个文件通过唯一的文件独有密钥（或范围独有密钥）进行保护。该密钥使用 NIST AED 密钥封装算法封装，之后会进一步使用多个类密钥中的一个进行封装，具体取决于计划如何访问该文件。随后封装的文件独有密钥储存在文件的元数据中。

使用 APFS 格式运行的设备可能支持文件克隆（使用写入时拷贝技术的零损耗拷贝）。如果文件被克隆，克隆的每一半都会得到一个新的密钥以接受传入的数据写入，这样新数据会使用新密钥写入媒介。久而久之，文件可能会由不同的范围（或片段）组成，每个映射到不同的密钥。但是，组成文件的所有范围受同一类密钥保护。

当打开一个文件时，系统会使用文件系统密钥解密文件的元数据，以显露出封装的文件独有密钥和表示它受哪个类保护的记号。文件独有（或范围独有）密钥使用类密钥解封，然后提供给硬件 AES 引擎，该引擎会在从闪存中读取文件时对文件进行解密。所有封装文件密钥的处理发生在安全隔区中；文件密钥绝不会直接透露给应用程序处理器。启动时，安全隔区与 AES 引擎协商得到一个临时密钥。当安全隔区解开文件密钥时，它们又通过该临时密钥再次封装，然后发送回应用程序处理器。

数据宗卷文件系统中所有文件的元数据都使用随机宗卷密钥进行加密，该密钥在首次安装操作系统或用户擦除设备时创建。此密钥由密钥封装密钥加密和封装，密钥封装密钥由安全隔区长期储存，只在安全隔区中可见。每次用户抹掉设备时，它都会发生变化。在 A9（及后续型号）SoC 上，安全隔区依靠由反重放系统支持的熵来实现可擦除性，以及保护其他资源中的密钥封装密钥。有关更多信息，请参阅安全非易失性存储器。

正如文件独有密钥或范围独有密钥一样，数据宗卷的元数据密钥绝不会直接透露给应用程序处理器；相反，安全隔区会提供一个临时的启动独有的版本。储存后，加密的文件系统密钥还会使用储存在可擦除存储器中的“可擦除密钥”封装或者使用受安全隔区反重放机制保护的媒介密钥封装密钥进行封装。此密钥不会提供数据的额外机密性。相反，它可以根据需要快速抹掉（由用户使用“抹掉所有内容和设置”选项来抹掉，或者由用户或管理员通过从移动设备管理 (MDM) 解决方案、Microsoft Exchange ActiveSync 或 iCloud 发出远程擦除命令来抹掉）。以这种方式抹掉密钥将导致所有文件因存在加密而不可访问。

文件的内容可能使用文件独有（或范围独有）的一个或多个密钥进行加密，密钥使用类密钥封装并储存在文件的元数据中，文件元数据又使用文件系统密钥进行加密。类密钥通过硬件 UID 获得保护，而某些类的类密钥则通过用户密码获得保护。此层次结构既可提供灵活性，又可保证性能。例如，更改文件的类只需重新封装其文件独有密钥，更改密码只需重新封装类密钥。

数据保护类

在支持数据保护的设备上创建新文件时，创建它的 App 会为其分配一个类。每个类使用不同的策略来确定数据何时可被访问。基本的类和策略信息请见下文的描述。基于 Apple 芯片的 Mac 电脑不支持 D 类：无保护，且在登录和退出登录（不包括在 iPhone、iPad 和 iPod touch 上锁定或解锁）时会建立安全边界。

类	保护类型
A 类：全面保护	NSFileProtectionComplete
B 类：未打开文件的保护	NSFileProtectionCompleteUnlessOpen
C 类：首次用户认证前保护 **【注】** macOS 使用宗卷密钥重新创建文件保险箱保护特性。	NSFileProtectionCompleteUntilFirstUserAuthentication
D 类：无保护 **【注】** macOS 中不支持。	NSFileProtectionNone

全面保护

NSFileProtectionComplete：该类密钥通过从用户密码和设备 UID 派生的密钥得到保护。用户锁定设备后不久（如果“需要密码”设置为“立即”，则为 10 秒钟），解密的类密钥会被丢弃，此类的所有数据都无法访问，除非用户再次输入密码或使用面容 ID 或触控 ID 解锁（登录）设备。

在 macOS 中，上一个用户退出登录不久后，解密的类密钥会被丢弃，此类的所有数据都无法访问，直到某位用户再次输入密码或使用触控 ID 登录设备。

未打开文件的保护

NSFileProtectionCompleteUnlessOpen：设备锁定或用户退出登录时，可能需要写入部分文件。如邮件附件在后台下载。此行为通过使用非对称椭圆曲线加密技术（基于 Curve25519 的 ECDH）实现。普通的文件独有密钥通过使用一次性迪菲-赫尔曼密钥交换协议（One-Pass Diffie-Hellman Key Agreement，如 NIST SP 800-56A 中所述）派生的密钥进行保护。

该协议的临时公钥与封装的文件独有密钥一起储存。KDF 是串联密钥导出函数 (Approved Alternative 1)，如 NIST SP 800-56A 中 5.8.1 所述。AlgorithmID 已忽略。PartyUInfo 和 PartyVInfo 分别是临时公钥和静态公钥。SHA256 被用作哈希函数。一旦文件关闭，文件独有密钥就会从内存中擦除。要再次打开该文件，系统会使用“未打开文件的保护”类的私钥和文件的临时公钥重新创建共享密钥，用来解开文件独有密钥的封装，然后用文件独有密钥来解密文件。

在 macOS 中，只要系统上的任何用户已登录或认证即可访问 NSFileProtectionCompleteUnlessOpen 的私有部分。

首次用户认证前保护

NSFileProtectionCompleteUntilFirstUserAuthentication：此类和“全面保护”类的行为方式相同，只不过在设备锁定或用户退出登录时已解密的类密钥不会从内存中删除。此类中的保护与桌面电脑全宗卷加密有类似的属性，可防止数据受到涉及重新启动的攻击。这是未分配给数据保护类的所有第三方 App 数据的默认类。

在 macOS 中，此类的作用类似于文件保险箱，且使用只要宗卷装载即可访问的宗卷密钥。

无保护

NSFileProtectionNone：此类密钥仅受 UID 的保护，并且存储在可擦除存储器中。由于解密该类中的文件所需的所有密钥都储存在设备上，因此采用该类加密的唯一好处就是可以进行快速远程擦除。即使未向文件分配数据保护类，此文件仍会以加密形式储存（就像 iOS 和 iPadOS 设备上的所有数据那样）。

macOS 不支持该类。

【注】在 macOS 中，对于未对应已启动操作系统的宗卷，只要该宗卷已装载，即可访问所有数据保护类。默认数据保护类为 NSFileProtectionCompleteUntilFirstUserAuthentication。范围独有密钥功能适用于 Rosetta 2 和原生 App。

用于数据保护的密钥包

在 iOS、iPadOS、watchOS 和 Apple tvOS 上，文件和钥匙串数据保护类的密钥通过密钥包进行收集和管理。这些操作系统使用以下密钥包：用户、设备、备份、托管和 iCloud 云备份。

用户密钥包

用户密钥包是设备常规操作中使用的封装类密钥的储存位置。例如，输入密码后，NSFileProtectionComplete 会从用户密钥包中载入并解封。它是储存在“无保护”类中的二进制属性列表 (.plist) 文件。

对于搭载 A9 之前的 SoC 的设备，该 .plist 文件的内容通过保存在可擦除存储器中的密钥加密。为了给密钥包提供前向安全性，用户每次更改密码时，系统都会擦除并重新生成此密钥。

对于搭载 A9 或后续型号 SoC 的设备，该 .plist 文件包含一个密钥，表示密钥包储存在受反重放随机数（由安全隔区控制）保护的有锁储存库中。

安全隔区管理用户密钥包并且可用于查询设备的锁定状态。仅当用户密钥包中的所有类密钥均可访问且成功解封时，它才会报告设备已解锁。

设备密钥包

设备密钥包用来储存用于涉及设备特定操作数据的封装类密钥。配置为共用的 iPadOS 设备有时需要在用户登录前访问凭证；因此，需要一个不受用户密码保护的密钥包。

iOS 和 iPadOS 不支持对用户独有的文件系统内容进行单独加密，这就意味着系统使用来自设备密钥包的类密钥，对文件独有密钥进行封装。而钥匙串则使用来自用户密钥包中的类密钥来保护用户钥匙串中的项目。在配置为单用户使用（默认配置）的 iOS 和 iPadOS 设备中，设备密钥包和用户密钥包是同一个，并受用户的密码保护。

备份密钥包

备份密钥包在“访达”（macOS 10.15 或更高版本）或 iTunes（macOS 10.14 或更低版本）进行加密备份时创建，并储存在设备被备份到的电脑中。新密钥包是通过一组新密钥创建的，备份的数据会使用这些新密钥重新加密。如前所述，不可迁移钥匙串项仍使用 UID 派生密钥封装，以使其可以恢复到最初备份它们的设备，但在其他设备上不可访问。

密钥包通过设置的密码加以保护，且加密过程会运行一千万次 PBKDF2 密钥派生函数的迭代。虽然迭代次数非常多，但是密钥包并未捆绑特定设备，因此在理论上，尝试在多台电脑上对备份密钥包并行展开暴力破解是可行的。而安全性足够高的密码可以减小这一威胁。

如果用户选择不加密备份，那么不管备份文件属于哪一种数据保护类，备份文件都不加密，但钥匙串仍使用 UID 派生密钥获得保护。这就是只有设置备份密码才能将钥匙串项迁移到新设备的原因。

托管密钥包

托管密钥包用于通过 USB 与“访达”(macOS 10.15 或更高版本)或 iTunes(macOS 10.14 或更低版本)进行同步,还用于移动设备管理 (MDM)。此密钥包允许“访达”或 iTunes 执行备份和同步,而无需用户输入密码,它还允许 MDM 解决方案远程清除用户密码。它储存在用于与“访达”或 iTunes 进行同步的电脑上,或者在远程管理设备的 MDM 解决方案上。

托管密钥包改善了设备同步过程中的用户体验,期间可能需要访问所有类别的数据。当使用密码锁定的设备首次连接到“访达”或 iTunes 时,会提示用户输入密码。然后设备创建托管密钥包,其中包含的类密钥与设备上使用的完全相同,该密钥包由新生成的密钥进行保护。托管密钥包及用于保护它的密钥划分到设备和主机或服务器上,其数据以“首次用户认证前保护”类储存在设备上。这就是重新启动后,用户首次使用“访达”或 iTunes 进行备份之前必须输入设备密码的原因。

在无线 (OTA) 软件更新的情况下,开始更新时系统会提示用户输入密码。这被用来安全地创建一个一次性解锁令牌,该令牌在更新后解锁用户密钥包。此令牌只有在输入用户密码后才能生成,且如果更改了用户密码,则任何此前生成的令牌都将失效。

一次性解锁令牌用于有人值守式或无人值守式的软件更新安装。安全隔区中单调计数器的当前值、密钥包的 UUID 和安全隔区 UID 会生成一个密钥,用来加密一次性解锁令牌。

在 A9(及后续型号)SoC 上,一次性解锁令牌不再依赖计数器或可擦除存储器,而是受到由安全隔区控制的反重放随机数的保护。

对于有人值守式的软件更新,一次性解锁令牌会在 20 分钟后过期。在 iOS 13 和 iPadOS 13.1 或更高版本中,令牌储存在由安全隔区保护的有锁储存库中。在 iOS 13 之前的版本中,此令牌从安全隔区导出,并被写入可擦除存储器或者受安全隔区反重放机制保护。如果设备未在 20 分钟内重新启动,策略定时器会使计数器增值。

系统检测到可用更新且满足以下条件时,会进行无人值守式软件更新:

- iOS 12 或更高版本中配置了自动更新。
- 用户收到更新通知时选取了“稍后安装”。

用户输入密码后会生成一次性解锁令牌,可在安全隔区中保持有效长达 8 小时。如果还未进行更新,此一次性解锁令牌会在每次锁定时销毁,并在下一次解锁时重新创建。每一次解锁都会使 8 小时的时间窗口重新开始。8 小时后,策略定时器会使一次性解锁令牌失效。

iCloud 云备份密钥包

iCloud 云备份密钥包与备份密钥包类似。该密钥包中的所有类密钥都是非对称的(与“未打开文件的保护”数据保护类一样,使用 Curve25519)。非对称密钥包还可用于“iCloud 钥匙串”钥匙串恢复中的备份。

在备选启动模式中保护密钥

数据保护被设计为仅在认证成功后才仅向授权用户提供对用户数据的访问权限。数据保护类被设计为支持各种使用情况，如即使在设备锁定时（但需要其首次解锁后）也支持读取和写入某些数据。备选启动模式期间还有其他保护用户数据访问的措施，如用于设备固件升级 (DFU) 模式、恢复模式、Apple 诊断或甚至是软件更新期间的措施。这些功能以硬件和软件功能的结合为基础，并随着 Apple 设计的芯片的发展而扩展。

功能	A10	A11、S3	A12、S4	A13、S5	A14、A15、S6、S7、M1 系列
恢复：所有数据保护类受保护	✓	✓	✓	✓	✓
DFU 备选启动模式、恢复和软件更新：A 类、B 类和 C 类数据受保护		✓	✓	✓	✓

安全隔区 AES 引擎配备了可锁定的软件种子位。从 UID 创建密钥时，这些种子位会包括在密钥导出函数中，以创建额外的密钥层次。种子位的使用方式因片上系统而异：

- 从 Apple A10 和 S3 开始的 SoC 上，种子位专用于区分受用户密码保护的密钥。需要用户密码的密钥（包括数据保护 A 类、B 类和 C 类密钥）会设定种子位，不需要用户密码的密钥（包括文件系统元数据密钥和 D 类密钥）会清除种子位。
- 在搭载 A10 或后续型号且运行 iOS 13 或更高版本和 iPadOS 13.1 或更高版本的设备上，当设备启动进入诊断模式时，所有用户数据都因存在加密而不可访问。此功能通过引入一个附加种子位实现，该种子位的设置管理媒介密钥的访问权限，访问通过数据保护加密的数据宗卷上的元数据（因而所有文件内容）时需要该媒介密钥。此保护涵盖所有类（A、B、C 和 D）中受保护的文件，不只是需要用户密码的文件。
- 在 A12 SoC 上，如果应用程序处理器已进入设备固件升级 (DFU) 模式或恢复模式，安全隔区 Boot ROM 会锁定密码种子位。密码种子位锁定时不允许对其进行任何更改操作。这旨在阻止对受用户密码保护的数据的访问。

进入 DFU 模式后恢复设备，可使设备恢复到已知的正常状态，该状态下只存在未修改的 Apple 签名的代码。可通过以下方式手动进入 DFU 模式。

请参阅以下 Apple 支持文章以了解如何使设备进入 DFU 模式：

设备	文章
iPhone、iPad、iPod touch	如果你忘记了 iPhone 密码
Apple TV	如果在 Apple TV 上看到警告符号
搭载 Apple 芯片的 Mac	修复或恢复搭载 Apple 芯片的 Mac

保护用户数据抵抗攻击

试图提取用户数据的攻击者通常会尝试使用各种技术：将加密数据提取到其他媒介以用于暴力破解攻击、操纵操作系统版本或者以其他方式更改或弱化设备的安全性策略以方便实施攻击。攻击设备上的数据通常需要使用物理接口（如闪电或 USB）与设备通信。Apple 设备具备帮助防御此类攻击的功能。

Apple 设备支持一项称为**密封密钥保护 (SKP)** 的技术，其旨在确保加密材料在这些情况下不可用：脱离设备时，或者对操作系统版本或安全性设置存在未经用户正确授权的操纵时。此功能**不是**由安全隔区提供，而是由位于更底层的硬件寄存器支持，目的是针对解密用户数据所需的密钥提供独立于安全隔区的额外保护层。

【注】SKP 仅在搭载 Apple 设计的 SoC 的设备上提供。

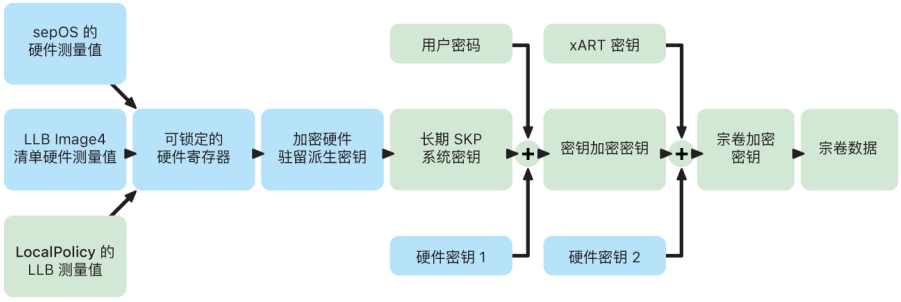
功能	A10	A11、S3	A12、S4	A13、S5	A14、A15、S6、S7、M1 系列
密封密钥保护	✓	✓	✓	✓	✓

当 iPhone 和 iPad 可能仍受授权持有者的物理控制时，设备还可配置为仅激活数据连接。

密封密钥保护 (SKP)

在支持数据保护的 Apple 设备上，密钥加密密钥 (KEK) 既受系统上软件测量值的保护（或密封），又与只能从安全隔区获得的 UID 绑定。在搭载 Apple 芯片的 Mac 上，对 KEK 的保护通过整合有关系统安全性策略的信息进一步得到了加强，因为 macOS 支持其他平台不支持的关键安全性策略更改（例如，停用安全启动或 SIP）。在搭载 Apple 芯片的 Mac 上，由于文件保险箱的实施使用数据保护（C 类），此保护涵盖文件保险箱密钥。

由用户密码与长期 SKP 密钥和硬件密钥 1（安全隔区的 UID）配合使用而生成的密钥称为**密码派生密钥**。此密钥用于保护用户密钥包（在所有支持的平台上）和 KEK（仅限在 macOS 中），然后启用生物识别解锁或使用其他设备（如 Apple Watch）自动解锁。



安全隔区启动监视器会捕获所加载的安全隔区操作系统的测量值。当应用程序处理器 Boot ROM 测量附于 LLB 的 Image4 清单时，该清单也包含所有其他已加载的系统配对固件的测量值。LocalPolicy 包含已加载 macOS 的核心安全性配置。还包含 `nsih` 字段，它是 macOS Image4 清单的哈希值。macOS Image4 清单包含所有与 macOS 配对的固件和 macOS 核心启动对象（如启动内核集或签名系统宗卷 (SSV) 根哈希值）的测量值。

如果攻击者能够意外地更改任何上述测量的固件、软件或安全性配置组件，则也会修改储存在硬件寄存器中的测量值。测量值的修改会导致从加密硬件派生的**系统测量根密钥 (SMRK)** 派生出不同的值，从而有效破坏密钥层级的封章。这将导致无法访问**系统测量设备密钥 (SMDK)**，从而导致无法访问 KEK，因此也无法访问数据。

但是，系统在未受到攻击时，必须容纳合法的软件更新，这些更新会更改固件测量值和 LocalPolicy 中的 `nsih` 字段，以指向新的 macOS 测量值。在其他尝试整合固件测量值但没有已知真实来源的系统中，用户将被要求停用安全性，更新固件后重新启用安全性，以便捕获新的测量基线。这大大增加了攻击者在软件更新期间篡改固件的风险。Image4 清单包含所需的所有测量值，这对系统很有帮助。正常启动期间如果测量值匹配，使用 SMRK 解密 SMDK 的硬件也可以将 SMDK 加密为所建议的将来的 SMRK。通过指定软件更新后预期的测量值，硬件可以加密在当前操作系统中可访问的 SMDK，以便在将来的操作系统中仍可访问。同样地，当客户在 LocalPolicy 中合法更改其安全性设置时，必须根据 LLB 下次重新启动时计算的 LocalPolicy 测量值，将 SMDK 加密为将来的 SMRK。

在 iOS 和 iPadOS 中安全激活数据连接

在 iOS 或 iPadOS 设备上，如果近期没有数据连接建立，用户必须使用面容 ID、触控 ID 或密码来激活通过闪电、USB 或智能接点接口建立的数据连接。此功能限制了物理连接设备（如恶意充电器）的攻击面，同时仍在合理的时间限制内支持其他配件的使用。如果 iOS 或 iPadOS 设备锁定或配件的数据连接终止超过一小时，则解锁设备前不允许建立任何新的数据连接。在这一个小时内，只允许来自之前设备处于解锁状态时已连接过的配件的数据连接。这些配件会在上次连接后的 30 天内被记住。如果在此期间发生未知配件尝试打开数据连接的情况，则会停用通过闪电接口、USB 接口和智能接点接口建立的所有配件数据连接，直到设备被重新解锁。这一个小时的意义：

- 帮助确保经常连接 Mac 或 PC、配件或通过线缆连接 CarPlay 车载的用户无需在每次连接其设备时都输入密码
- 十分必要，因为配件生态系统不提供在数据连接建立前识别配件的可靠加密方式

此外，如果一个数据连接通过配件建立了 3 天以上，设备从锁定时刻起将不会再允许新的数据连接。这也增强了对不经常使用此类配件的用户的保护性。如果设备处于需要密码来重新启用生物认证的状态，通过闪电接口、USB 接口和智能接点接口建立的数据连接也会停用。

用户可以选择在“设置”中重新启用始终打开数据连接（设置某些辅助设备时会自动启用）。

Apple 文件系统作用

Apple 文件系统 (APFS) 是一种专有文件系统，其设计过程中始终将加密放在重要位置。APFS 广泛应用于 Apple 的所有平台：iPhone、iPad、iPod touch、Mac、Apple TV 和 Apple Watch。APFS 针对闪存/SSD 储存进行了优化，具有强加密、写入时拷贝元数据、空间共享、文件和目录克隆、快照、快速目录大小统计功能以及原子级安全存储元和改进的文件系统基础，还具有独有的写入时拷贝设计，可使用 I/O 合并来达到最高性能，同时确保数据可靠性。

空间共享

APFS 根据需求分配储存空间。单个 APFS 容器有多个宗卷时，容器的可用空间会共享，并且可按需分配到任意单独的宗卷。每个宗卷仅使用整体容器的一部分，这样一来，可用空间即容器的总大小减去该容器中所有宗卷已使用空间的大小。

多个宗卷

在 macOS 10.15 或更高版本中，用于启动 Mac 的 APFS 容器必须包含至少五个宗卷，其中前三个宗卷对用户隐藏：

- **预启动宗卷：**此宗卷未加密，且包含启动容器中每个系统宗卷所需的数据。
- **VM 宗卷：**此宗卷未加密，被 macOS 用来储存加密的交换文件。
- **恢复宗卷：**此宗卷未加密，且必须在未解锁系统宗卷的情况下可用才能在 recoveryOS 中启动。
- **系统宗卷：**包含以下内容：
  - 用于启动 Mac 的所有必要文件
  - macOS 原生安装的所有 App（之前位于“/应用程序”文件夹中的 App 现在位于“/系统/应用程序”中）

  【注】系统宗卷中默认不能写入进程，甚至 Apple 系统进程也不能写入。
- **数据宗卷：**包含可能发生更改的数据，如：
  - 用户文件夹中的任何数据，包括照片、音乐、视频和文稿
  - 用户安装的 App，包括 AppleScript 和“自动操作”应用程序
  - 用户、组织或第三方 App 安装的自定框架和监控程序
  - 用户拥有且能够写入的其他位置，如“/应用程序”、“/资源库”、“/用户”、“/Volumes”、“/usr/local”、“/private”、“/var”和“/tmp”

每增加一个系统宗卷，便会创建一个数据宗卷。预启动宗卷、VM 宗卷和恢复宗卷全为共享宗卷且无法复制。

在 macOS 11 或更高版本中，系统宗卷通过快照捕捉。操作系统从系统宗卷的快照启动，而不仅仅是从可变系统宗卷的只读装载进行启动。

在 iOS 和 iPadOS 中，储存空间分为至少两个 APFS 宗卷：

- 系统宗卷
- 数据宗卷

钥匙串数据保护

许多 App 都需要处理密码和其他一些简短但比较敏感的数据，如密钥和登录令牌。钥匙串提供了储存这些项的安全方式。不同的 Apple 操作系统采用不同机制实施与各钥匙串保护类关联的保障。在 macOS（包括搭载 Apple 芯片的 Mac）中，数据保护不直接用于实施此类保障。

概览

钥匙串项使用两种不同的 AES-256-GCM 密钥加密：表格密钥（元数据）和行独有密钥（私密密钥）。钥匙串元数据（除 kSecValue 外的所有属性）使用元数据密钥加密以加速搜索，私密值 (kSecValueData) 使用私密密钥进行加密。元数据密钥受安全隔区保护，但会缓存在应用程序处理器中以便进行钥匙串快速查询。私密密钥则始终需要通过安全隔区进行往返处理。

钥匙串以储存在文件系统中的 SQLite 数据库的形式实现，而且数据库只有一个，`securityd` 监控程序决定每个进程或 App 可以访问哪些钥匙串项。钥匙串访问 API 将生成对监控程序的调用，从而查询 App 的“keychain-access-groups”、“application-identifier”和“application-group”权限。访问组允许在 App 之间共享钥匙串项，而非将访问权限限制于单个进程。

钥匙串项只能在来自同一开发者的 App 之间共享。若要共享钥匙串项，要求第三方 App 在其应用程序组中使用包含前缀的访问组，该前缀由 Apple Developer Program（Apple 开发者计划）分配。通过代码签名、预置描述文件和 Apple Developer Program（Apple 开发者计划）来强制实施对前缀的要求和应用程序组唯一性。

系统用于保护钥匙串数据的类结构与文件数据保护中使用的类结构相似。这些类具有与文件数据保护类对等的行为，但使用的密钥和函数不同。

可用性	文件数据保护	钥匙串数据保护
未锁定状态下	NSFileProtectionComplete	kSecAttrAccessibleWhenUnlocked
锁定状态下	NSFileProtectionCompleteUnlessOpen	暂无
首次解锁后	NSFileProtectionCompleteUntilFirstUserAuthentication	kSecAttrAccessibleAfterFirstUnlock
始终	NSFileProtectionNone	kSecAttrAccessibleAlways
密码启用状态下	暂无	kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly

使用后台刷新服务的 App 可将 kSecAttrAccessibleAfterFirstUnlock 用于后台更新过程中需要访问的钥匙串项。

类 kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly 与 kSecAttrAccessibleWhenUnlocked 行为方式相同；但前者仅当设备配置了密码时可用。此类仅存在于系统密钥包中，且它们：

- 不同步到 iCloud 钥匙串
- 不会备份
- 不包括在托管密钥包中

如果密码被移除或重设，类密钥便会丢弃，这些项目也变得无法使用。

其他钥匙串类都有对应的“仅限本设备”项目，后者在备份期间从设备拷贝时始终通过 UID 加以保护，因此如果恢复至其他设备将无法使用。Apple 根据所保护信息的类型以及 iOS 和 iPadOS 需要这些信息的时间来选择钥匙串类，妥善平衡了安全性和可用性。例如，VPN 证书必须始终可用，这样设备才能保持连接，但它归类为“不可迁移”，因此不能将其移至另一台设备。

钥匙串数据类保护

以下列出的类保护针对钥匙串项执行。

项目	可访问
无线局域网密码	首次解锁后
邮件帐户	首次解锁后
Microsoft Exchange ActiveSync 帐户	首次解锁后
VPN 密码	首次解锁后
LDAP、CalDAV、CardDAV	首次解锁后
社交网络帐户令牌	首次解锁后
“接力”广播加密密钥	首次解锁后
iCloud 令牌	首次解锁后
iMessage 信息密钥	首次解锁后
家庭共享密码	未锁定状态下
Safari 浏览器密码	未锁定状态下
Safari 浏览器书签	未锁定状态下
访达/iTunes 备份	未锁定状态下, 不可迁移
由配置描述文件安装的私钥	未锁定状态下, 不可迁移
VPN 证书	始终, 不可迁移
Bluetooth® 密钥	始终, 不可迁移
Apple 推送通知服务 (APNs) 令牌	始终, 不可迁移
iCloud 证书和私钥	始终, 不可迁移
SIM 卡 PIN 码	始终, 不可迁移
由配置描述文件安装的证书	始终
“查找”令牌	始终
语音信箱	始终

钥匙串访问控制

钥匙串可以使用访问控制列表 (ACL) 以设定可访问性和认证要求的策略。钥匙串项可以设立条件，要求用户指定使用面容 ID、触控 ID 或输入设备密码进行认证，否则不能访问。对钥匙串项的访问也可以限制为在该钥匙串项添加以后，面容 ID 或触控 ID 注册未发生更改。此限制有助于防止攻击者通过添加自己的指纹来访问钥匙串项。ACL 在安全隔区内部进行评估，只有符合其指定的限制条件时，才会释放到内核。

macOS 中的钥匙串架构

macOS 还提供了对钥匙串的访问，用于方便安全地储存用户名和密码、数码身份、加密密钥和安全备忘录。你可以打开“/应用程序/实用工具”中的“钥匙串访问” App 来访问它。使用钥匙串可让你无需输入每个资源的凭证，甚至无需记住。系统会为每个 Mac 用户创建初始默认钥匙串，而用户可为特定目的创建其他钥匙串。

除了用户钥匙串，macOS 还依靠多个系统级钥匙串来维护非用户特定的认证资源，如网络凭证和公用密钥基础设施 (PKI) 身份。其中一个钥匙串 System Roots 不可更改，它储存了互联网 PKI 根证书颁发机构 (CA) 证书，以辅助网上银行和电子商务等常见任务。用户可以类似的方式将内部预置的 CA 证书部署到被管理的 Mac 电脑，以帮助验证内部站点和服务。

文件保险箱

在 macOS 中使用文件保险箱加密宗卷

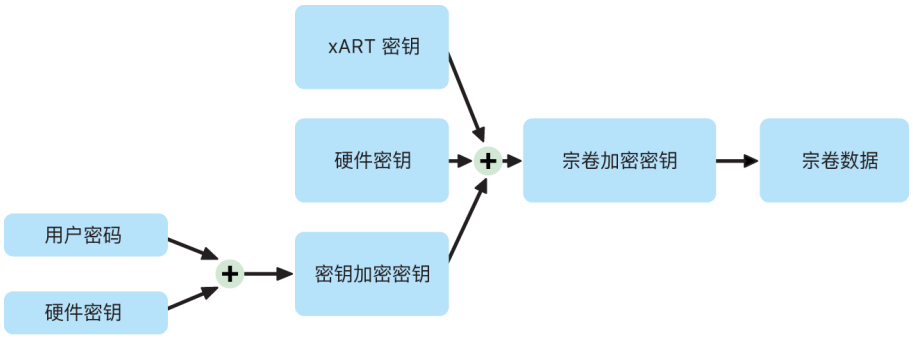
Mac 电脑会提供文件保险箱，这是一项内建加密功能，用于保护所有静态数据安全。文件保险箱使用 AES-XTS 数据加密算法保护内部和可移除储存设备上的完整宗卷。

搭载 Apple 芯片的 Mac 上的文件保险箱通过使用宗卷密钥的数据保护 C 类来实施。在搭载 Apple T2 安全芯片的 Mac 和搭载 Apple 芯片的 Mac 上，直接连接到安全隔区的加密内部储存设备会使用安全隔区的硬件安全性功能和 AES 引擎的功能。用户在 Mac 上启用文件保险箱后，启动过程中将需要其凭证。

文件保险箱已打开的内部储存设备

如果没有有效的登录凭证或加密恢复密钥，即使物理储存设备被移除并连接到其他电脑，内置 APFS 宗卷仍保持加密状态，以防止未经授权的访问。在 macOS 10.15 中，此类宗卷同时包括系统宗卷和数据宗卷。从 macOS 11 开始，系统宗卷通过签名系统宗卷 (SSV) 功能进行保护，而数据宗卷仍通过加密进行保护。搭载 Apple 芯片的 Mac 以及搭载 T2 芯片的 Mac 通过构建和管理密钥层级实施内部宗卷加密，基于芯片内建的硬件加密技术而构建。此密钥层级的设计旨在同时实现四个目标：

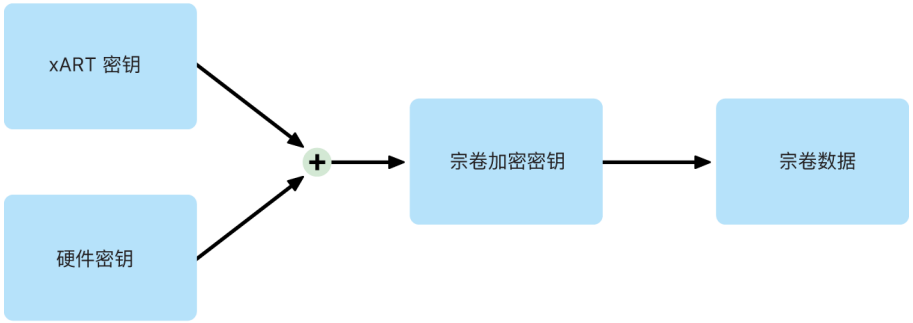
- 请求用户密码用于加密
- 保护系统免受针对从 Mac 上移除的储存媒介的直接暴力破解攻击
- 提供擦除内容的快速安全的方法，即通过删除必要的加密材料
- 让用户无需重新加密整个宗卷即可更改其密码（同时也会更改用于保护其文件的加密密钥）



在搭载 Apple 芯片的 Mac 以及搭载 T2 芯片的 Mac 上，所有文件保险箱密钥的处理都发生在安全隔区中；加密密钥绝不会直接透露给 Intel CPU。所有 APFS 宗卷默认使用宗卷加密密钥创建。宗卷和元数据内容使用此宗卷加密密钥加密，此宗卷加密密钥使用类密钥封装。文件保险箱启用时，类密钥受用户密码和硬件 UID 共同保护。

文件保险箱已关闭的内部储存设备

在搭载 Apple 芯片或搭载 T2 芯片的 Mac 上，如果在“设置助理”初次运行过程中未启用文件保险箱，宗卷仍会加密，但宗卷加密密钥仅由安全隔区中的硬件 UID 保护。



如果稍后启用了文件保险箱（由于数据已加密，该过程可立即完成），反重放机制会帮助阻止旧密钥（仅基于硬件 UID）被用于解密宗卷。然后宗卷将受用户密码和硬件 UID 共同保护（如前文所述）。

删除文件保险箱宗卷

删除宗卷时，其宗卷加密密钥由安全隔区安全删除。这有助于防止以后使用此密钥进行访问（即使是通过安全隔区）。另外，所有宗卷加密密钥都使用媒介密钥封装。媒介密钥不提供额外的数据机密性，而是旨在启用快速安全的数据删除，如果缺少了它，则不可能进行解密。

在搭载 Apple 芯片的 Mac 和搭载 T2 芯片的 Mac 上，媒介密钥一定是由受安全隔区支持的技术来抹掉，例如远程 MDM 命令。以这种方式抹掉媒介密钥将导致宗卷因存在加密而不可访问。

可移除储存设备

可移除储存设备的加密不使用安全隔区的安全性功能，而是采用与基于 Intel 的 Mac（不搭载 T2 芯片）相同的方式执行加密。

在 macOS 中管理文件保险箱

在 macOS 中，组织可以使用 SecureToken 或 Bootstrap Token 管理文件保险箱。

使用安全令牌

macOS 10.13 或更高版本中的 Apple 文件系统 (APFS) 更改了文件保险箱加密密钥的生成方式。在 CoreStorage 宗卷上的 macOS 旧版本中，文件保险箱加密过程中使用的密钥是在用户或组织在 Mac 上启用文件保险箱时创建的。在 APFS 宗卷上的 macOS 中，该类密钥在用户创建过程中、设定首位用户的密码或 Mac 用户首次登录过程中创建。加密密钥的此实施方式、生成时间和储存方式都是称为**安全令牌**这一功能的一部分。特别地，安全令牌是受用户密码保护的密钥加密密钥 (KEK) 的封装版本。

在 APFS 上部署文件保险箱时，用户可以继续：

- 使用现有工具和进程，如使用移动设备管理 (MDM) 解决方案储存以进行托管的个人恢复密钥 (PRK)
- 创建和使用机构恢复密钥 (IRK)
- 推迟文件保险箱的启用，直到用户登录或退出登录 Mac

在 macOS 11 中，为 Mac 上的首位用户设定初始密码就会授予该用户安全令牌。在部分流程中，这可能并不是预期的行为，因为在此之前授予第一个安全令牌已要求登录用户帐户。为了防止此类行为发生，在设定用户的密码前，请将 `;DisabledTags;SecureToken` 添加到以编程方式创建的用户的 `AuthenticationAuthority` 属性，如下：

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

使用 Bootstrap Token

macOS 10.15 引入了 **Bootstrap Token** 这项新功能，可帮助将安全令牌授予移动帐户和设备注册时创建的管理员帐户（“被管理的管理员”，可选项）。在 macOS 11 中，Bootstrap Token 能够将安全令牌授予给登录 Mac 电脑的任何用户，包括本地用户帐户。使用 macOS 10.15 或更高版本中的 Bootstrap Token 功能需要：

- 使用“Apple 校园教务管理”或“Apple 商务管理”在 MDM 中注册 Mac，从而让 Mac 受监督
- MDM 供应商支持

在 macOS 10.15.4 或更高版本中，启用了安全令牌的任何用户在首次登录时会生成 Bootstrap Token 并托管到 MDM，前提是 MDM 解决方案支持该功能。在需要时，还可以使用 `profiles` 命令行工具来生成 Bootstrap Token 并托管到 MDM。

在 macOS 11 中，Bootstrap Token 除了可用于向用户帐户授予安全令牌外，还可用于其他方面。在搭载 Apple 芯片的 Mac 上，通过 MDM 进行管理时，Bootstrap Token（如果可用）可用于授权安装内核扩展和软件更新。

Apple 如何保护用户的个人数据

保护 App 对用户数据的访问

除了加密静态数据外，Apple 设备还使用数据保险箱等各种技术帮助阻止 App 在未获得许可的情况下访问用户的个人信息。用户可以在 iOS 和 iPadOS 中的“设置”或者 macOS 的“系统偏好设置”中查看已批准哪些 App 访问特定信息，也可以授予或撤销未来的访问权限。强制执行访问权限的有：

- **iOS、iPadOS 和 macOS：**日历、相机、通讯录、麦克风、照片、提醒事项、语音识别
- **iOS 和 iPadOS：**蓝牙、家庭、媒体、媒体 App 及 Apple Music、运动与健身
- **iOS 和 watchOS：**健康
- **macOS：**输入监视（例如，键盘笔画）、提示、屏幕录制（例如，静态截屏和视频）、系统偏好设置

在 iOS 13.4 或更高版本和 iPadOS 13.4 或更高版本中，所有第三方 App 的数据自动受到数据保险箱的保护。数据保险箱有助于防止未经授权访问数据，甚至本身未沙盒化的进程也不可访问。iOS 15 或更高版本中的其他类包括：本地网络、附近交互、研究传感器与使用数据以及专注模式。

如果用户登录 iCloud，则会默认授予 iOS 和 iPadOS 中的 App 访问 iCloud 云盘的权限。用户可以在“设置”的 iCloud 选项中控制每个 App 的访问权限。iOS 和 iPadOS 还提供了访问限制，旨在阻止移动设备管理 (MDM) 解决方案所安装的 App 和帐户与用户所安装的 App 和帐户之间的数据移动。

保护对用户健康数据的访问

HealthKit 为 iPhone 和 Apple Watch 上的健康和健身数据提供了一个中央储存库。HealthKit 也直接与健康和健身设备配合工作，例如兼容的低功耗蓝牙 (BLE) 心率监视器和内建于许多 iOS 设备中的运动协处理器。HealthKit 与健康和健身 App、医疗保健机构以及健康和健身设备之间的所有交互都需要用户许可。此数据使用“未打开文件的保护”数据保护类储存。设备锁定 10 分钟后会丢弃数据访问权限，下一次用户输入密码或者使用面容 ID 或触控 ID 解锁设备时，数据会变为可访问。

收集和储存健康和健身数据

HealthKit 还会收集和储存管理数据，如 App 的访问权限、连接到 HealthKit 的设备名称，以及新数据可用时，用来开启 App 的计划信息。此数据储存在“首次用户认证前保护”数据保护类中。临时日志文件储存设备锁定时（例如用户锻炼时）所生成的健康记录。这些文件使用“未打开文件的保护”数据保护类储存。设备解锁后，这些临时日志文件会导入进主要健康数据库，并会在合并完成后被删除。

健康数据可以储存在 iCloud 中。“健康”数据的端对端加密要求 iOS 12 或更高版本以及双重认证。如果不满足这两个要求，用户的数据仍会在储存和传输过程中加密，但不是端对端加密。用户打开双重认证并更新到 iOS 12 或更高版本后，其健康数据会迁移为端对端加密。

如果用户使用“访达”（macOS 10.15 或更高版本）或 iTunes（macOS 10.14 或更低版本）备份其设备，只有在备份加密时才会储存健康数据。

临床健康记录

用户可在“健康” App 内登录支持的健康系统，以获取其临床健康记录的副本。当用户接入健康系统时，会通过 OAuth 2 客户端凭证进行认证。接入后，将通过受 TLS 1.3 保护的连接直接从健康机构下载临床健康记录数据。下载完成后，临床健康记录会与其他健康数据一起安全储存。

“健康”数据完整性

储存在数据库中的数据包括用于追踪每条数据记录起源的元数据。该元数据包括 App 标识符，用以识别储存了该记录的 App。此外，可选元数据项还可能包含记录的数字签名副本，从而保持受信任设备生成记录的数据完整性。数字签名采用在 RFC 5652 中描述的“密码讯息语法”(CMS) 格式。

第三方 App 访问的“健康”数据

App 只有通过授权才能访问 HealthKit API，且必须遵守数据使用方式的访问限制。例如，不允许 App 使用健康数据投放广告。App 还需要向用户提供隐私政策，详细说明其如何使用健康数据。

用户可通过“隐私”设置来控制 App 对健康数据的访问。当 App 请求访问健康数据（与请求访问“通讯录”、“照片”和其他 iOS 数据源类似）时，会要求用户授予访问权限。但对于健康数据，App 将获得读取和写入数据的单独访问权限，以及对每种健康数据类型的单独访问权限。用户可以在“设置”>“健康”>“数据访问权限与设备”中，查看和撤销已授予的访问健康数据的权限。

如果获得写入数据的权限，App 还可以读取其写入的数据。如果获得读取数据的权限，则 App 可以读取所有来源写入的数据。但 App 不能决定其他 App 所获得的访问权限。此外，App 不能完全确定其是否获得了读取健康数据的权限。如果 App 没有读取权限，所有查询都不会返回数据，如同查询空数据库。这旨在阻止 App 通过学习用户所跟踪的数据类型来推断用户的健康状况。

用户的医疗急救卡

“健康” App 可让用户在医疗急救卡表单中填写急救时可能至关重要的信息。这些信息可由用户输入或手动更新，且不会与健康数据库中的信息同步。

在锁定屏幕上轻点“紧急情况”按钮可查看医疗急救卡信息。该信息使用“无保护”数据保护类储存于设备上，因此无需输入设备密码就可进行访问。“医疗急救卡”作为一项可选功能，可让用户权衡安全性和隐私二者之间的关系。在 iOS 13 或更低版本中，此数据备份在 iCloud 云备份中。在 iOS 14 中，医疗急救卡使用 CloudKit 在设备间同步并具备与其他健康数据相同的加密特性。

健康共享

在 iOS 15 中，“健康” App 可让用户选择与其他用户共享其“健康”数据。“健康”数据通过 iCloud 端对端加密在两个用户之间共享，Apple 无法访问通过“健康”共享发送的数据。若要使用该功能，发送和接收用户的设备必须运行 iOS 15 或更高版本且启用了双重认证。

用户还可以使用“健康” App 中的“与提供者共享”功能来与其医疗服务提供者共享“健康”数据。使用此功能共享的数据通过端对端加密的方式仅提供给用户所选的健康机构，Apple 不会保留或访问加密密钥以解密、查看或以其他方式访问通过“与提供者共享”功能共享的“健康”数据。有关此服务的设计如何保护用户的“健康”数据的更多详细信息，请参阅《适用于医疗服务组织的 Apple 注册指南》中的“安全性和隐私”部分。

数字签名和加密

访问控制列表

钥匙串数据通过访问控制列表 (ACL) 进行分区和保护。因此，不同身份的 App 无法访问由第三方 App 储存的凭证，除非用户明确批准。此保护针对 Apple 设备中组织内的各种 App 和服务提供了认证凭证保护机制。

邮件

在“邮件” App 中，用户可以发送经数字签名和加密的邮件。在兼容的智能卡所附个人身份验证 (PIV) 令牌的数字签名和加密证书上，“邮件”会自动发现适当的 RFC 5322 电子邮件地址主题或主题备用名称（区分大小写）。如果已配置的电子邮件帐户符合所附 PIV 令牌上的数字签名或加密证书上的电子邮件地址，“邮件”会自动在新邮件窗口的工具栏中显示签名按钮。如果“邮件”拥有收件人的电子邮件加密证书，或者可在 Microsoft Exchange 全局地址列表 (GAL) 中找到，新邮件工具栏中会显示已解锁图标。已锁定的锁图标表示邮件将使用收件人的公钥加密后发送。

信息独有 S/MIME

iOS、iPadOS 和 macOS 支持信息独有 S/MIME。这表示 S/MIME 用户可选取始终默认签名并加密邮件，或者有选择地签名并加密单个邮件。

配合 S/MIME 使用的身份可通过配置描述文件、移动设备管理 (MDM) 解决方案、简单证书注册协议 (SCEP) 或 Microsoft Active Directory Certificate Authority 来发送到 Apple 设备。

智能卡

macOS 10.12 或更高版本包括对 PIV 卡的原生支持。这些卡片在商业和政府机构中广泛用于双重认证、数字签名和加密。

智能卡包括一个或多个数码身份，这些身份拥有一对公钥和私钥，以及关联的证书。使用个人身份号码 (PIN) 解锁智能卡时会提供用于认证、加密和签名操作的私钥的访问权限。证书决定密钥的用途、其关联的属性以及是否有证书颁发机构 (CA) 证书验证（签名）。

智能卡可用于双重认证。解锁卡片需要的双重条件是“用户拥有的”（卡片）和“用户知道的”（PIN 码）。macOS 10.12 或更高版本还原生支持对 Safari 浏览器上网站的智能卡登录窗口认证和客户端证书认证。还支持使用密钥对 (PKINIT) 进行 Kerberos 认证以单点登录到支持 Kerberos 的服务。若要了解有关智能卡和 macOS 的更多信息，请参阅**《Apple 平台部署》**中的智能卡集成介绍。

加密的磁盘映像

在 macOS 中，加密的磁盘映像用作用户储存或传输敏感文稿和其他文件的安全容器。加密的磁盘映像使用“磁盘工具”创建，该工具位于“/应用程序/实用工具”目录下。磁盘映像可使用 128 位或 256 位 AES 加密来进行加密。装载的磁盘映像会被视为连接到 Mac 的本地宗卷，因此用户可以拷贝、移动和打开其中储存的文件和文件夹。和文件保险箱一样，磁盘映像的内容是实时加密和解密的。有了加密的磁盘映像，用户就可以安全交换文稿、文件和文件夹，方法是将加密的磁盘映像存储到可移动媒介、作为邮件附件发送或者储存在远程服务器上。有关加密磁盘映像的更多信息，请参阅《磁盘工具使用手册》。

App 安全性

App 安全性概览

App 是现代安全架构中最关键的要素之一。尽管 App 可显著提高用户的工作效率，但如果处理不当，也可能对系统安全性、稳定性和用户数据产生负面影响。

鉴于此，Apple 提供了多重安全措施以帮助确保 App 不受已知恶意软件侵害以及不被篡改。其他保护措施确保谨慎处理 App 对用户数据的访问。这些安全性控制为 App 提供了安全稳定的平台，使成千上万的开发者能够在 iOS、iPadOS 和 macOS 上提供数十万款 App，而全都不会影响系统完整性。用户可以在其 Apple 设备上访问这些 App，而不必没有缘由地担心病毒、恶意软件或未经授权的攻击。

在 iPhone、iPad 和 iPod touch 上，所有 App 都从 App Store 获取且经过沙盒化，以提供最严密的控制。

在 Mac 上，许多 App 可从 App Store 获取，但 Mac 用户也可从互联网下载和使用 App。为了安全支持互联网下载，macOS 的分层保护提供了附加控制。首先，在 macOS 10.15 或更高版本中，所有 Mac App 默认需要 Apple 公证才能启动。此要求可帮助确保这些 App 不含已知的恶意软件，无需要求 App 是通过 App Store 提供。此外，macOS 包含先进的防病毒保护功能，可阻止（以及在需要时移除）恶意软件。

作为跨平台的附加控制，沙盒化可阻止 App 未经授权访问用户数据。在 macOS 中，关键区域中的数据具有自我保护功能，这可帮助确保用户仍可以控制对“桌面”、“文稿”、“下载”和所有 App 其他区域中文件的访问，无论尝试访问的 App 是否本身已沙盒化。

原生功能	第三方同等功能
未批准的插件列表、未批准的 Safari 浏览器扩展列表	病毒/恶意软件定义
文件隔离	病毒/恶意软件定义
XProtect/YARA 签名	病毒/恶意软件定义；端点保护
门禁	端点保护；对 App 强制执行代码签名以帮助确保仅运行受信任的软件
eficheck （对于不搭载 Apple T2 安全芯片的 Mac 为必要项）	端点保护；Rootkit 检测
应用程序防火墙	端点保护；防火墙
数据包过滤 (pf)	防火墙解决方案
系统完整性保护	内建于 macOS
强制访问控制	内建于 macOS
Kext 排除列表	内建于 macOS
强制性 App 代码签名	内建于 macOS
App 公证	内建于 macOS

iOS 和 iPadOS 中的 App 安全性

iOS 和 iPadOS 中的 App 安全性介绍

与其他移动平台不同，iOS 和 iPadOS 不允许用户安装来自网站的潜在恶意未签名 App 或者运行不受信任的 App。运行时，所有可执行内存页会在载入时进行代码签名检查，以帮助确保 App 自安装或上次更新之后未被修改过。

确认 App 来自批准的来源后，iOS 和 iPadOS 会强制实施相应的安全措施，以防止其危害其他 App 或系统的其余部分。

iOS 和 iPadOS 中的 App 代码签名过程

在 iOS 和 iPadOS 中，Apple 会通过强制性代码签名和严格的开发者签名等方式确保 App 安全性。

强制性代码签名

iOS 或 iPadOS 内核启动后，它将控制哪些用户进程和 App 可以运行。为帮助确保所有 App 均来自批准的已知来源并且未被篡改，iOS 和 iPadOS 要求所有可执行代码均使用 Apple 颁发的证书进行签名。设备附带的 App（如“邮件”和 Safari 浏览器）由 Apple 签名。第三方 App 也必须使用 Apple 颁发的证书进行验证和签名。强制性代码签名将信任链的概念从操作系统扩展至 App，并帮助防止第三方 App 载入未签名的代码资源或使用经 App 自身修改代码。

开发者如何给其 App 签名

开发者可通过证书验证（通过 Apple 开发者计划）给其 App 签名。他们还可以在其 App 中嵌入框架，并使用 Apple 颁发的证书（通过团队标识符字符串）对代码进行验证。

- **证书验证：**若要在 iOS 或 iPadOS 设备上开发并安装 App，开发者必须向 Apple 注册并加入 Apple Developer Program（Apple 开发者计划）。Apple 首先验证每个开发者（无论是个人还是企业）的真实身份，然后再颁发证书。开发者可使用该证书对 App 进行签名，并将其提交至 App Store 进行分发。因此，App Store 中的所有 App 都是由身份可识别的个人或组织提交的，由此防止恶意 App 的创建。此外，这些应用程序都经过 Apple 的严格审核，帮助确保它们通常可以按照所述的方式运行，并且没有明显的错误或其他明显的问题。除了已经讨论过的技术，这一处理过程还会让用户对所购 App 的品质更加放心。
- **代码签名验证：**iOS 和 iPadOS 允许开发者将框架嵌入其 App 中，使它可被 App 本身使用，也可被 App 中嵌入的扩展项使用。为保护系统并防止其他 App 在其地址空间中载入第三方代码，系统将为启动时所有链接到进程的动态资源库执行代码签名验证。此验证过程通过团队标识符 (Team ID) 完成。团队标识符提取自 Apple 颁发的证书，是由 10 个字符组成的字母数字串，例如 1A2B3C4D5F。程序可链接到系统自带的任何平台资源库，也可以链接到代码签名中具有与主可执行文件相同团队标识符的资源库。因为作为系统一部分发布的可执行文件不具有团队标识符，所以它们只能链接到随系统本身发布的资源库。

验证企业内部专有 App

符合条件的企业也可以编写供组织内部使用的企业内部专有 App，并将其分发给员工。企业和组织可以申请加入 Apple Developer Enterprise Program（ADEP，Apple 开发者企业计划）。有关更多信息以及若要检查资格要求，请参阅 Apple Developer Enterprise Program（Apple 开发者企业计划）网站。组织成为 ADEP 的成员后，便可以注册以获得一个预置描述文件，该描述文件允许企业内部专有 App 在其授权的设备上运行。

用户必须安装预置描述文件才能运行这些 App。这有助于确保只有组织的目标用户能够将 App 载入到其 iOS 和 iPadOS 设备上。通过移动设备管理 (MDM) 安装的 App 为隐式受信任 App，因为组织与设备之间的关系已经确立。否则，用户必须在“设置”中批准 App 的预置描述文件。组织还可以限制用户批准来自未知开发者的 App。任何企业内部专有 App 首次启动时，设备必须收到 Apple 的肯定询证，表明允许该 App 运行。

iOS 和 iPadOS 中的运行时进程安全性

iOS 和 iPadOS 使用“沙盒”、声明的授权和地址空间布局随机化 (ASLR) 帮助确保运行时安全性。

沙盒化

所有第三方 App 均已经过“沙盒化”，因此它们在访问其他 App 储存的文件或对设备进行更改时会受到限制。沙盒化的设计旨在防止 App 收集或修改其他 App 储存的信息。每个 App 还拥有唯一的主目录来存放其文件，主目录是在安装 App 时随机分配的。如果第三方 App 需要访问除自身信息以外的其他信息，只能通过 iOS 和 iPadOS 明确提供的服务来实现。

系统文件和资源也会与用户的 App 保持隔离。与所有第三方 App 一样，绝大部分 iOS 和 iPadOS 系统文件和资源以非权限用户“mobile”的身份运行。整个操作系统分区以只读方式装载。不必要的工具（如远程登录服务）未包含在系统软件中，并且 API 不允许 App 提升自己的权限来修改其他 App 或者 iOS 和 iPadOS。

使用授权

iOS 使用声明的授权来控制第三方 App 对用户信息及功能（如 iCloud 和扩展功能）的访问。授权是签名到 App 的键值对，允许对运行时因素之外的内容（如 UNIX 用户 ID）进行认证。授权已经过数字签名，因此无法更改。系统 App 和监控程序广泛应用授权来进行特定权限操作，如果不使用授权，则进程需要以 root 用户身份运行才能进行这些操作。这极大降低了遭入侵的系统 App 或监控程序提升权限的可能性。

此外，App 只能通过系统提供的 API 来执行后台处理。这就使 App 能够继续运行，而不会降低性能或显著影响电池续航能力。

地址空间布局随机化

地址空间布局随机化 (ASLR) 有助于防止对内存损坏错误的利用。内置 App 使用 ASLR 有助于在启动时随机安排所有内存区域。除了启动时工作，ASLR 还会随机安排可执行代码、系统库和相关编程结构的内存地址，以进一步降低遭受许多攻击的可能性。例如，“return-to-libc”攻击试图通过操纵堆栈和系统库的内存地址来诱使设备执行恶意代码。随机安排内存地址增加了执行攻击的难度，尤其是对多个设备的攻击。Xcode 和 iOS 或 iPadOS 开发环境可自动编译启用了 ASLR 支持的第三方程序。

Execute Never 功能

iOS 和 iPadOS 使用 ARM 的 Execute Never (XN) 功能提供进一步的保护，该功能会将内存页标记为不可执行。只有处于严格的控制条件下，App 才能使用标记为可写入和可执行的内存页：内核会检查 Apple 专有的动态代码签名授权是否存在。即使如此，也只有单个 `mmap` 调用能用于请求一个可执行且可写入的内存页，系统为可执行且可写入的内存页分配了随机地址。Safari 浏览器对其 JavaScript 即时 (JIT) 编译器使用了此功能。

iOS、iPadOS 和 macOS 中支持的扩展项

iOS、iPadOS 和 macOS 允许 App 提供扩展项来增加其他 App 的功能。扩展项是具有特定用途的已签名可执行二进制代码，封装在 App 中。系统会在安装期间自动检测扩展项，并使用匹配系统让其他 App 使用扩展项。

扩展点

支持扩展项的系统区域称为**扩展点**。每个扩展点都提供 API，并为该区域强制执行策略。系统基于扩展点特定的匹配规则来决定哪些扩展项可用。系统自动按需启动扩展进程，并管理它们的生命周期。通过使用授权来限制特定系统 App 的扩展可用性。例如，“今天”视图小组件只显示在“通知中心”中，而共享扩展项只在“共享”面板中可用。扩展点示例有“今天”小组件、共享、操作、照片编辑、文件提供程序和自定键盘。

扩展项如何通信

扩展项在其自己的地址空间运行。App 与其激活的扩展项之间的通信采用由系统框架协调的进程间通信。它们无权访问彼此的文件或内存空间。扩展项的设计旨在将它们彼此隔离、与其包含的 App 隔离，并且与使用它们的 App 隔离。与其他第三方 App 类似，它们也经过沙盒化并且拥有的容器与含有 App 的容器隔开。但是扩展项与其容器 App 对隐私控制具有相同的访问权限。因此，如果用户给 App 授予“通讯录”的访问权限，该 App 中嵌入的扩展项也会获得此权限，但由该 App 激活的扩展项不具有该权限。

如何使用自定键盘

自定键盘是一种特殊的扩展项类型，因为它由用户启用并适用于整个系统。启用后，键盘扩展项会用于所有的文本栏，除了密码输入栏和任何安全文本视图。为限制用户数据的传输，默认情况下自定键盘运行在一个非常受限的沙盒中，该沙盒阻止网络访问、阻止代表进程执行网络操作的服务，并阻止可允许扩展项泄露键入数据的 API。自定键盘的开发者可以要求其扩展项拥有“开放存取”权限，使系统在得到用户的同意后在默认的沙盒中运行扩展项。

MDM 和扩展项

对于在移动设备管理 (MDM) 解决方案中注册的设备，文稿和键盘扩展项将遵循“被管理的打开方式”规则。例如，MDM 解决方案可帮助阻止用户将被管理的 App 中的文稿导出到未被管理的文稿提供程序，或阻止他们在被管理的 App 中使用未被管理的键盘。另外，App 开发者可阻止在其 App 中使用第三方键盘扩展项。

iOS 和 iPadOS 中的 App 保护和 App 组

在 iOS 和 iPadOS 中，组织可以使用 IOS SDK 和通过加入 Apple Developer Portal（Apple 开发者门户）上的 App 组来安全保护 App。

在 App 中采用数据保护

iOS 和 iPadOS 软件开发套件 (SDK) 提供全套 API，使第三方和企业内部开发者能够轻松地利用数据保护，帮助确保在 App 中实现最高级别的保护。数据保护适用于文件和数据库 API，包括 NSFileManager、CoreData、NSData 和 SQLite。

“邮件” App 数据库（包括附件）、被管理的图书、Safari 浏览器书签、App 启动图像和位置数据也将加密储存，加密密钥通过用户设备上的密码进行保护。“日历”（不包括附件）、“通讯录”、“提醒事项”、“备忘录”、“信息”和“照片”采用数据保护授权的“首次用户认证前保护”。

用户安装的 App 若没有选择加入某个特定数据保护类，则默认接受“首次用户认证前保护”。

加入 App 组

指定开发者帐户拥有的 App 和扩展项在配置为 App 组的一部分后可共享内容。对于是否在 Apple Developer Portal（Apple 开发者门户）上创建合适的群组并包括所需的一套 App 和扩展项，则由开发者自行决定。App 配置为 App 组的一部分后，App 可以访问以下内容：

- 宗卷上共享的存储容器，只要有 App 组内至少有一个 App 被安装，它就会一直保留在设备上
- 共享的偏好设置
- 共享的钥匙串项

Apple Developer Portal（Apple 开发者门户）有助于保证 App 组 ID (GID) 在整个 App 生态系统中的唯一性。

在 iOS 和 iPadOS 中验证配件

“Made for iPhone, iPad, and iPod touch (Mfi)”许可计划允许接受过审查的配件生产企业使用 iPod 配件协议 (iAP) 和必要的支持硬件组件。

当 MFi 配件使用闪电接口、USB-C 接口或通过蓝牙与 iOS 或 iPadOS 设备通信时，设备会要求配件使用 Apple 提供的证书进行回应，以证明其经过 Apple 授权，然后设备对此证书进行验证。然后，设备发送一个质询，配件必须使用已签名的证书来响应。这个过程完全由定制集成电路 (IC) 来处理，而且对于配件本身是透明的。执行处理操作的定制集成电路由 Apple 提供给许可配件生产企业。

配件可以请求访问不同的传输方法和功能：例如，访问通过闪电线缆或 USB-C 线缆传输的数字音频流，或者访问通过蓝牙提供的位置信息。认证集成电路的设计旨在确保只有经过批准的配件才能获得对设备的完全访问权限。如果配件不支持认证，其访问仅限于模拟音频和一小部分串行 (UART) 音频播放控件。

“隔空播放”还使用认证集成电路来验证接收器已由 Apple 批准。“隔空播放”音频流和 CarPlay 车载视频流使用 MFi-SAP（安全关联协议），此协议使用 AES128 在计数器 (CTR) 模式下对配件和设备之间的通信进行加密。作为端对端 (STS) 协议的一部分，临时密钥使用 ECDH 密钥交换 (Curve25519) 进行交换，并使用认证集成电路的 1024 位 RSA 密钥进行签名。

macOS 中的 App 安全性

macOS 中的 App 安全性介绍

macOS 中的 App 安全性包含多层重叠的保护，第一层保护是可选择仅运行来自 App Store 已签名且受信任的 App。此外，macOS 的分层保护有助于确保从互联网下载的 App 不含已知的恶意软件。macOS 提供了检测和移除恶意软件的技术以及专为防止不受信任的 App 访问用户数据的其他保护措施。公证和 XProtect 更新等 Apple 服务旨在帮助防止恶意软件安装。必要时，这些服务会定位首次检测中可能逃脱的恶意软件，然后快速有效地移除它。最重要的是，macOS 用户可以在对他们有意义的安全模型内自由操作，包括运行完全未签名和不受信任的代码。

macOS 中的 App 代码签名进程

App Store 中的所有 App 均已由 Apple 签名。此签名的设计旨在确保它们不会被篡改或修改。Apple 设备附带的所有 App 均由 Apple 签名。

在 macOS 10.15 中，不在 App Store 中分发的所有 App 必须由开发者使用 Apple 签发的开发者 ID 证书（和专用密钥）进行签名并由 Apple 公证，以便在默认的“门禁”设置中运行。企业内部开发的 App 也应使用 Apple 签发的开发者 ID 进行签名，以便用户可以验证其完整性。

在 macOS 中，代码签名和公证独立工作，并可由不同的实施者执行，以实现不同的目的。代码签名由开发者使用其开发者 ID 证书（由 Apple 签发）执行，对此签名的验证可向用户证明自开发者构建软件并签名以来，其软件未被篡改。公证可由软件分发链中的任何人执行，并证明已向 Apple 提供代码副本以检查恶意软件且未发现已知的恶意软件。公证的输出为票据，储存在 Apple 服务器上，可选择（由任何人）包含在 App 中，而不会使开发者的签名失效。

强制访问控制 (MAC) 需要代码签名以启用受系统保护的授权。例如，需要通过防火墙访问的 App 必须使用合适的 MAC 授权进行代码签名。

macOS 中的门禁和运行时保护

macOS 提供门禁技术和运行时保护以帮助确保仅受信任的软件可在用户的 Mac 上运行。

门禁

macOS 包括一项称为**门禁**的安全技术，其设计旨在帮助确保仅受信任的软件可在用户的 Mac 上运行。当用户从 App Store 之外的地方下载并打开 App、插件或安装器软件包时，门禁会验证该软件是否来自可识别的开发者、经过 Apple 公证不含已知的恶意内容且未被修改。在首次打开下载的软件之前，门禁还会请求用户批准，以确保用户没有被诱骗运行他们认为只是数据文件的可执行代码。

默认情况下，门禁会帮助确保所有下载的软件已由 App Store 签名，或者由已注册的开发者签名且经过 Apple 公证。App Store 审核过程和公证流程均旨在确保 App 不含已知的恶意软件。因此，默认情况下，**macOS 中的所有软件在首次打开时都会检查是否包含已知的恶意内容，无论它以何种方式安装到 Mac 上。**

用户和组织可以选择仅允许从 App Store 安装软件。另外，用户可以忽略门禁策略以打开任意软件，除非受到移动设备管理 (MDM) 解决方案的限制。组织可以使用 MDM 配置“门禁”设置，包括允许使用备用身份签名的软件。如果需要，也可以完全停用门禁。

门禁还可防止在分发安全 App 时包含恶意插件。此处指使用 App 时在用户不知情的情况下触发恶意插件加载。必要时，门禁会从随机的只读位置打开 App。这旨在防止自动加载与 App 一起分发的插件。

运行时保护

系统文件、资源和内核与用户的 App 空间保持隔离。App Store 中的所有 App 均已经过沙盒化，以限制访问其他 App 储存的数据。如果来自 App Store 的 App 需要访问其他 App 的数据，则只能使用 macOS 提供的 API 和服务来进行访问。

在 macOS 中防范恶意软件

Apple 提供了威胁情报流程以快速识别和阻止恶意软件。

三层防御机制

恶意软件防御机制由三层构成：

1.**阻止恶意软件的启动或执行：**App Store 或结合公证的门禁

2.**阻止恶意软件在客户系统上运行：**门禁、公证和 XProtect

3.**修复已执行的恶意软件：**XProtect

第一层防御机制设计用于阻止恶意软件的分发，并使其一次都无法启动，这是 App Store 和结合公证的门禁的目标。

下一层防御机制是为了帮助确保任何 Mac 上出现恶意软件时可快速识别及阻止，以停止恶意软件的传播并修复恶意软件已渗透的 Mac 系统。这一层防御机制中除了门禁及公证外，还加入了 XProtect。

最后，XProtect 运作以修复已设法成功执行的恶意软件。

这些保护措施（详情如下所述）结合在一起以实现最佳的病毒和恶意软件防御机制。除此之外，还有额外的保护措施（特别是在搭载 Apple 芯片的 Mac 上）可限制设法执行的恶意软件造成的潜在破坏。请参阅保护 App 对用户数据的访问以了解 macOS 可帮助保护用户数据免遭恶意软件攻击的方法，以及操作系统完整性以了解 macOS 可限制恶意软件在系统上执行操作的方法。

公证

公证是由 Apple 提供的恶意软件扫描服务。要在 App Store 之外分发 macOS 版 App 的开发者需提交其 App 以进行扫描，这是分发流程的一部分。Apple 会扫描此软件以检测已知恶意软件，如果未发现，则签发公证票据。开发者通常将此票据打包到其 App 中，这样即使处于离线状态，门禁也可验证并启动 App。

Apple 还可给已知的恶意 App 签发撤销票据，即使这些 App 之前曾通过公证。macOS 会定期检查新的撤销票据，使门禁拥有最新的信息并可阻止此类文件的启动。此流程可非常迅速地阻止恶意 App，因为更新在后台进行，并且比推送新 XProtect 签名的后台更新更加频繁。另外，此保护措施既可应用于曾通过公证的 App，也可应用于未通过公证的 App。

XProtect

macOS 内建了称为 **XProtect** 的防病毒技术，可基于签名检测和移除恶意软件。系统使用由 Apple 定期更新的 YARA 签名，YARA 是一款用来基于签名检测恶意软件的工具。Apple 会监视新的恶意软件感染和威胁，并自动更新签名（独立于系统更新），以保护 Mac 免受恶意软件侵害。XProtect 会自动检测已知恶意软件并阻止其执行。在 macOS 10.15 或更高版本中，XProtect 会在以下操作执行时检查是否含已知的恶意内容：

- App 首次启动
- App 发生更改（在文件系统中）
- XProtect 签名发生更新

XProtect 检测到已知的恶意软件时，会阻止软件并通知用户，还提供将软件移到废纸篓的选项。

【注】公证对已知文件（或文件哈希）有效，并可用在之前启动过的 App 上。XProtect 基于签名的规则比特定文件哈希更为通用，因此它可以找到 Apple 未发现过的变体。XProtect 只扫描发生更改或首次启动的 App。

如果恶意软件已侵入 Mac，XProtect 还包含修复感染的技术。例如，它包括的引擎可基于 Apple 自动发布的更新（作为系统数据文件自动更新和安全性更新的一部分）来修复感染。它还会在收到更新信息后立即移除恶意软件，并继续定期检查是否感染。XProtect 不会自动重新启动 Mac。

自动 XProtect 安全性更新

Apple 会根据最新的威胁情报自动发布 XProtect 的更新。macOS 默认每天检查这些更新。使用 CloudKit 同步分发的公证更新更加频繁。

发现新的恶意软件时 Apple 如何响应

发现新的恶意软件时可能会执行若干步骤：

- 撤销任何关联的开发者 ID 证书。
- 针对所有文件（App 和相关文件）签发公证撤销票据。
- 生成和发布 XProtect 签名。

  这些签名还会追溯应用至之前通过公证的软件，并且任何新的检测结果均会引发之前的一个或多个操作发生。

最终，恶意软件检测会在接下来几秒、几小时和几天中相继启动一系列步骤，以将可能的最佳保护措施传播给 Mac 用户。

在 macOS 中控制 App 对文件的访问

Apple 坚信用户对哪些 App 在使用其数据应完全知悉、同意和拥有控制权。在 macOS 10.15 中，系统强制执行此模型，以帮助确保所有 App 在获得用户同意后才能访问“文稿”、“下载”、“桌面”、iCloud 云盘和网络宗卷中的文件。在 macOS 10.13 或更高版本中，必须在“系统偏好设置”中明确添加需要访问整个储存设备的 App。此外，辅助功能和自动化功能需要用户许可，以帮助确保它们不会绕过其他保护措施。取决于访问策略，用户可能会收到提示或需要在“系统偏好设置”>“安全性与隐私”>“隐私”中更改设置：

项目	App 提示用户	用户必须编辑系统隐私设置
辅助功能		✓
访问完整内部储存设备		✓
文件和文件夹 **【注】**包括“桌面”、“文稿”、“下载”、网络宗卷和可移除宗卷	✓	
自动化（Apple 事件）	✓	

任何使用“完全磁盘访问权限”的 App 均无法访问用户废纸篓中的项目；用户不会收到 App 访问的提示。如果用户想要 App 访问这些文件，则必须将文件从废纸篓移到其他位置。

在 Mac 上打开了文件保险箱的用户需要提供有效的凭证才能继续启动流程和获得特殊启动模式的访问权限。如果没有有效的登录凭证或恢复密钥，即使物理储存设备被移除并连接到其他电脑，整个宗卷仍保持加密状态，以防止未经授权的访问。

若要保护企业设置中的数据，IT 应使用移动设备管理 (MDM) 定义和实施文件保险箱配置策略。组织有若干加密宗卷管理选项，包括机构恢复密钥、个人恢复密钥（可选择由 MDM 托管储存）或两者结合。密钥循环也可以设为 MDM 中的一种策略。

“备忘录”App 中的安全功能

“备忘录” App 包括安全备忘录功能，可在 iPhone、iPad、Mac 和 iCloud 网站上使用，允许用户保护特定备忘录的内容。用户还可以与他人安全共享备忘录。

安全备忘录

安全备忘录使用用户提供的密码短语进行端对端加密，需要该密码短语才能在 iOS、iPadOS 和 macOS 设备以及 iCloud 网站上查看这类备忘录。每个 iCloud 帐户（包括“我的”设备帐户）可以有单独的密码短语。

当用户对备忘录实施保护时，会基于用户的密码短语使用 PBKDF2 和 SHA256 生成一个 16 字节的密钥。备忘录及其所有附件均使用伽罗瓦/计数器模式下的 AES (AES-GCM) 进行加密。新记录会在 Core Data 和 CloudKit 中创建，用于储存加密的备忘录、附件、标记和初始化向量。新记录创建后，未加密的原始数据将被删除。支持加密的附件包括图像、速绘、表格、地图和网站。包含其他类型附件的备忘录无法加密，不支持的附件不能添加到安全备忘录。

若要查看安全备忘录，用户必须输入其密码短语或者使用面容 ID 或触控 ID 进行认证。成功认证用户后，无论是要查看还是创建安全备忘录，“备忘录”都会打开一项安全会话。安全会话打开时，用户无需额外认证即可查看或保护其他备忘录。但是，安全会话仅适用于使用提供的密码短语进行保护的备忘录。对于使用其他密码短语保护的备忘录，用户仍需要认证。安全会话在以下情况下会关闭：

- 用户轻点“备忘录”中的“现在锁定”按钮
- “备忘录”切换到后台超过 3 分钟（在 macOS 中超过 8 分钟）
- iOS 或 iPadOS 设备锁定

若要更改安全备忘录的密码短语，用户必须输入当前密码短语，因为更改密码短语时面容 ID 和触控 ID 不可用。选取新的密码短语后，“备忘录” App 会在同一帐户中重新封装所有由先前的密码短语加密的现有备忘录的密钥。

如果用户连续三次输错密码短语，“备忘录”会显示用户提供的提示（如果设置时用户有提供）。如果用户仍想不起自己的密码短语，则可以在“备忘录”设置中重设密码短语。此功能允许用户使用新的密码短语创建新的安全备忘录，但将不允许他们查看之前保护的备忘录。如果想起旧的密码短语，仍能查看之前保护的备忘录。重设密码短语需要用户的 iCloud 帐户密码短语。

共享备忘录

未使用密码短语进行端对端加密的备忘录可与他人共享。对于用户放入备忘录中的任何文本或附件，共享备忘录仍使用 CloudKit 加密的数据类型。资源始终使用在 CKRecord 中加密的密钥进行加密。诸如创建日期和修改日期之类的元数据不会加密。CloudKit 管理进程，通过此进程参与者可以加密和解密彼此的数据。

“快捷指令”App 中的安全功能

在“快捷指令” App 中，你可选择将快捷指令通过 iCloud 在 Apple 设备间同步，还可通过 iCloud 与其他用户共享快捷指令。快捷指令以加密格式储存在本地。

自定快捷指令类似于脚本或程序，其用途十分广泛。从互联网下载快捷指令时，系统会警告用户该快捷指令未经过 Apple 审核，让用户有机会检查快捷指令。为了防范恶意快捷指令，系统会在运行时下载更新的恶意软件定义以识别恶意快捷指令。

从共享表单调用自定快捷指令时，它们还可以在 Safari 浏览器中的网站上运行用户指定的 JavaScript。为了防范恶意 JavaScript（例如，欺骗用户在社交媒体网站上运行收集其数据的脚本），系统会通过上述恶意软件定义验证 JavaScript。首次在一个域中运行 JavaScript 时，系统会提示用户允许在当前网页上为该域运行包含 JavaScript 的快捷指令。

服务安全性

服务安全性概览

Apple 所打造的一系列强大服务可帮助用户更充分地使用设备并提高工作效率。它们在提供云储存、同步、密码储存、认证、支付、信息收发和通信等强大功能的同时，保护用户的隐私及其数据的安全。

本章节涵盖用于 iCloud、“通过 Apple 登录”、Apple Pay、iMessage 信息、Apple Messages for Business、FaceTime 通话、“查找”和“连续互通”的安全性技术。

【注】部分 Apple 服务和内容并非在所有国家或地区都可用。

Apple ID 和管理式 Apple ID

Apple ID 安全性概览

Apple ID 是用于登录 Apple 服务的帐户。对于用户而言，确保 Apple ID 的安全以帮助防止未经授权使用其帐户十分重要。为了达成这一目标，Apple ID 要求使用强密码：

- 长度必须至少为 8 个字符
- 必须同时包含字母和数字
- 连续的相同字符不得超过两个
- 不能为常用的密码

在此规则的基础上，用户可以通过添加更多的字符和标点符号，让密码变得更加安全。

在帐户发生重大更改时，Apple 还会发送电子邮件和/或推送通知来通知用户。例如，密码或账单信息发生更改，或者在新设备上使用 Apple ID 登录。如有异常发生，Apple 会提示用户立即更改其 Apple ID 密码。

另外，Apple 采用了多种策略和规程来保护用户帐户。这包括限制重新尝试登录和尝试重设密码的次数，采取主动式欺诈监控以帮助在发生攻击时进行识别，以及定期对策略进行检查，让 Apple 可以针对可能影响用户安全性的任何新信息作出调整。

【注】“管理式 Apple ID”密码策略由管理员在“Apple 校园教务管理”或“Apple 商务管理”中设定。

双重认证

为帮助用户进一步提高帐户的安全性，Apple 默认使用**双重认证**，为 Apple ID 提供一层额外的安全保护。它旨在确保仅帐户所有者能访问帐户，其他人即使知道密码也无法访问。双重认证启用后，只能在受信任设备（如用户的 iPhone、iPad、iPod touch 或 Mac）上访问用户的帐户，或在其他设备上通过其中一台受信任设备或受信任的电话号码完成验证后访问。在任何新设备上首次登录需要两项信息，即 Apple ID 密码和 6 位数的验证码，验证码显示在用户的受信任设备上或发送到受信任的手机号码。输入验证码即表明用户确认他们信任新设备，且在该设备上登录是安全的。双重认证意味着已经不能仅靠密码来访问用户帐户，因而提高了用户的 Apple ID 以及所有通过 Apple 储存的个人信息的安全性。它直接集成到 iOS、iPadOS、macOS、Apple tvOS、watchOS 以及 Apple 网站使用的认证系统中。

用户使用网络浏览器登录 Apple 网站时，第二重请求会发送到与用户 iCloud 帐户关联的所有受信任设备，请求批准网站会话。如果用户从受信任设备上的浏览器登录 Apple 网站，验证码会显示在用户正在使用的设备上。当用户在该设备上输入验证码时，即批准网站会话。

密码重设和帐户恢复

如果忘记了 Apple ID 帐户密码，用户可以在受信任设备上重设密码。如果用户没有受信任的设备但知道密码，则可以使用受信任的电话号码通过短信验证来进行认证。此外，若要立即恢复 Apple ID，可以结合之前曾使用过的密码和短信验证来重设。如果无法使用上述方法进行恢复，则必须按照帐户恢复流程进行操作。有关更多信息，请参阅 Apple 支持文章：无法重设 Apple ID 密码时如何使用帐户恢复。

管理式 Apple ID 安全性

“管理式 Apple ID”提供类似 Apple ID 的功能，但它由企业或教育组织拥有和控制。这些组织可以重设密码，限制购买和通信（例如 FaceTime 通话和“信息”），以及基于职务为雇员、职员、教师和学生设置权限。

对于“管理式 Apple ID”，部分服务会停用（例如，Apple Pay、iCloud 钥匙串、HomeKit 和“查找”）。

检查管理式 Apple ID

“管理式 Apple ID”也支持**检查**，使组织遵守法律和隐私规定。“Apple 校园教务管理”的管理员、管理人员或教师能够检查特定的“管理式 Apple ID”帐户。

检查员仅可监控组织内低于其职务级别的帐户。例如，教师可以监控学生，管理人员可以检查教师和学生，而管理员可以检查管理人员、教师和学生。

当使用“Apple 校园教务管理”请求检查凭证时，会分发一个特殊帐户，该帐户仅可访问要求检查的“管理式 Apple ID”。然后检查员就可以读取和修改用户储存在 iCloud 或支持 CloudKit 的 App 中的内容。“Apple 校园教务管理”会记录每个审核访问请求。日志中显示检查员的身份、检查员请求访问的“管理式 Apple ID”、请求的时间以及是否执行了检查。

管理式 Apple ID 和个人设备

“管理式 Apple ID”也可配合个人拥有的 iOS 和 iPadOS 设备以及 Mac 电脑使用。学生使用机构分发的“管理式 Apple ID”和附加的个人用密码（作为 Apple ID 双重认证过程的第二重）登录 iCloud。学生在个人设备上使用“管理式 Apple ID”时，iCloud 钥匙串不可用，同时机构可能还会限制其他功能，如 FaceTime 通话或“信息”。学生在登录状态下创建的任何 iCloud 文稿均会按照本节上文所述接受审核。

iCloud

iCloud 安全性概览

iCloud 可以储存用户的通讯录、日历、照片、文稿和其他内容，并让这些信息在其设备间自动保持最新。它也可以供第三方 App 使用，来储存和同步文稿以及由开发者所定义的 App 数据关键值。用户通过登录 Apple ID 并选取想要使用的服务来设置 iCloud。IT 管理员可以使用移动设备管理 (MDM) 配置描述文件来停用部分 iCloud 功能，例如 iCloud 云盘和 iCloud 云备份。

iCloud 使用较强的安全措施并采用严格的策略来保护用户数据。大部分 iCloud 数据首先会在用户设备上通过设备生成的 iCloud 密钥加密，然后才上传到 iCloud 服务器。对于未采用端对端加密的数据，用户设备会将这些 iCloud 密钥安全地上传到 Apple 数据中心的 iCloud 硬件安全模块。这将允许 Apple 协助用户恢复数据，并在用户需要时（例如登录新设备、从备份恢复或者通过网页访问 iCloud 数据）代表用户解密数据。在用户设备和 iCloud 服务器之间传输的数据会通过 TLS 单独加密，且 iCloud 服务器会使用额外的加密层来储存静态用户数据。

提供给 Apple 的加密密钥安全存放在 Apple 数据中心。处理储存在第三方数据中心的数据时，这些加密密钥仅可由在安全服务器上运行的 Apple 软件获取，并且仅在处理必要流程时才可获取。为了加强隐私和安全保护，许多 Apple 服务使用端对端加密，这意味着只有用户本人才能访问自己的 iCloud 数据且只能在通过其 Apple ID 登录的受信任设备上访问。

针对用户储存在 iCloud 中的数据，Apple 提供了两种加密和保护选择：

- **标准数据保护（默认设置）：**用户的 iCloud 数据会被加密，而加密密钥安全储存在 Apple 数据中心，以便 Apple 协助恢复数据和帐户。只有特定的 iCloud 数据（包括“健康”数据和 iCloud 钥匙串中的密码等 14 个数据类别）会采用端对端加密。
- **iCloud 高级数据保护：**提供 Apple 最高级别云端数据安全性的可选设置。如果用户选择打开“高级数据保护”，则只有其受信任的设备才能访问大部分 iCloud 数据的加密密钥，因此这些数据会受到端对端加密保护。打开“高级数据保护”后，采用端对端加密的数据类别会上升到 23 个，包括 iCloud 云备份、“照片”、“备忘录”等等。

Apple 支持文章 iCloud 数据安全概览中列出了采用端对端加密进行保护的特定 iCloud 数据类别。

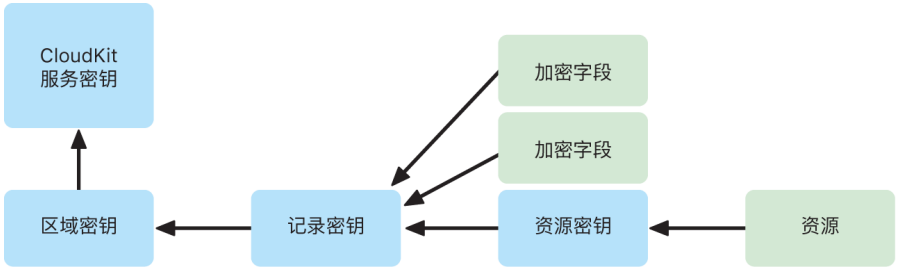
iCloud 加密

iCloud 数据加密与数据储存模型紧密相关，首先是 CloudKit 框架和 API，它们允许 App 和系统软件代表用户将数据储存在 iCloud 中并使所有内容在设备和网页上保持最新。

CloudKit 加密

CloudKit 框架允许 App 开发者将键值数据、结构性数据和资源（与数据库分开储存的大型数据，如图像或视频）储存在 iCloud 中。CloudKit 支持按容器分组的公共数据库和专用数据库。公共数据库全局共享，通常用于一般性资源，且不加密。专用数据库用于储存每位用户的 iCloud 数据。

CloudKit 使用与数据结构匹配的密钥层级。每个容器的专用数据库受密钥层级保护，该密钥层级植根于称为 **CloudKit 服务密钥**的非对称密钥。这些密钥在每位 iCloud 用户的受信任设备上生成且具有唯一性。当数据写入 CloudKit 时，所有记录密钥均会在用户受信任设备上生成并封装到对应的密钥层级，然后才会上传数据。



Apple 支持文章 iCloud 数据安全概览中列出的许多 Apple 服务使用受 iCloud 钥匙串同步保护的 CloudKit 服务密钥进行端对端加密。对于这些 CloudKit 容器，服务密钥储存在用户的 iCloud 钥匙串中并享有 iCloud 钥匙串的安全特性；服务密钥仅可在用户受信任设备上访问，Apple 或任何第三方都无法访问。如果设备丢失，用户可以使用 iCloud 钥匙串安全恢复、帐户恢复联系人或帐户恢复密钥来恢复其 iCloud 钥匙串数据。

加密密钥管理

CloudKit 中加密数据的安全性依赖对应加密密钥的安全性。CloudKit 服务密钥分为两类：端对端加密和认证后可用。

- **端对端加密服务密钥：**对于端对端加密的 iCloud 服务，相关的 CloudKit 服务私钥绝不会提供给 Apple 服务器。包括私钥在内的服务密钥对会在用户的受信任设备上本地创建并使用 iCloud 钥匙串安全性传输到用户的其他设备。虽然 Apple 服务器负责处理 iCloud 钥匙串恢复和同步流程，但加密会阻止这些服务器访问用户的任何钥匙串数据。在无法访问 iCloud 钥匙串及其所有恢复机制的最坏情况下，CloudKit 中的端对端加密数据会丢失。Apple 无法帮助恢复此数据。
- **认证后可用服务密钥：**对于“照片”和 iCloud 云盘等其他服务，服务密钥储存在 Apple 数据中心的 iCloud 硬件安全模块中，且可供部分 Apple 服务访问。当用户在新设备上登录 iCloud 并认证其 Apple ID 后，无需用户进一步交互或输入，Apple 服务器便可访问这些密钥。例如，用户登录 iCloud.com 后便可立即在线查看其照片。这类服务密钥为**认证后可用**密钥。

iCloud 高级数据保护

iCloud 高级数据保护是提供 Apple 最高级别云端数据安全性的可选设置。当用户打开“高级数据保护”后，只有其受信任的设备才能访问大部分 iCloud 数据的加密密钥，因此这些数据会受到**端对端加密**保护。对于打开“高级数据保护”的用户，采用端对端加密进行保护的数据总类别会由 14 个上升到 23 个，包括 iCloud 云备份、“照片”、“备忘录”等等。

iCloud 高级数据保护于 2022 年末向美国用户提供，并将在 2023 年初开始在世界各地陆续推出。

“高级数据保护”在概念上很容易理解：在设备上生成且随后上传到 Apple 数据中心的**认证后可用** iCloud 硬件安全模块 (HSM) 的所有 CloudKit 服务密钥会从这些 HSM 中删除，并且会完全保存在该帐户的 iCloud 钥匙串保护域中。其处理方式和现有的**端对端加密**服务密钥一样，这意味着 Apple 不可再读取或访问这些密钥。

“高级数据保护”还会自动保护第三方开发者选择标记为加密的 CloudKit 字段以及所有 CloudKit 资源。

启用“高级数据保护”

当用户打开“高级数据保护”时，其受信任设备会执行以下两项操作：首先，该设备会向用户参与端对端加密的其他设备传达用户打开“高级数据保护”的意图。具体是通过将设备本机密钥签名的新值写入其 iCloud 钥匙串设备元数据来实现。在该设备与用户的其他设备同步期间，Apple 服务器无法移除或修改此证明。

之后，该设备会发起将**认证后可用**服务密钥从 Apple 数据中心移除的操作。因为受 iCloud HSM 保护，这些密钥会立即永久删除且不可恢复。删除密钥后，Apple 不可再访问受用户服务密钥保护的**任何**数据。此时设备会开始异步密钥轮换操作，此操作会为之前密钥提供给 Apple 服务器的每项服务创建新的服务密钥。如果密钥轮换由于网络中断或任何其他错误而失败，设备会再次尝试密钥轮换直至成功。

服务密钥轮换成功后，写入服务的新数据便不能通过旧服务密钥解密。该数据受到仅由用户受信任设备控制的新密钥保护，Apple 绝对无法访问。

“高级数据保护”和 iCloud.com 网页访问

用户首次打开“高级数据保护”时，在 iCloud.com 上通过网页访问其数据会自动关闭。这是因为 iCloud 网页服务器不可再访问解密并显示用户数据所需的密钥。用户可以选择重新打开网页访问权限，并使用参与的受信任设备在网页上访问加密的 iCloud 数据。

打开网页访问权限后，在每次访问 iCloud.com 时，用户必须在其中一台受信任设备上授权网页登录。此授权“赋予”设备网页访问权限。在接下来的一小时，此设备会接受来自特定 Apple 服务器的请求以上传单个服务密钥，但仅限于通常可在 iCloud.com 上访问的服务列表所对应的密钥。换句话说，即使用户授权网页登录，服务器请求也无法诱使用户设备上传不可在 iCloud.com 上查看的数据（例如“健康”数据或 iCloud 钥匙串中的密码）的服务密钥。Apple 服务器仅会请求解密用户在网页上请求访问的特定数据所需的服务密钥。每次上传服务密钥时，该密钥会使用与用户授权的网页会话绑定的临时密钥加密，且用户设备上会出现一则通知，显示哪项 iCloud 服务的数据正临时可供 Apple 服务器访问。

保留用户的选择

“高级数据保护”和 iCloud.com 网页访问设置仅可由用户修改。这类值储存在用户的 iCloud 钥匙串设备元数据中，且只能从用户其中一台受信任设备上更改。Apple 服务器无法代表用户修改这些设置，也不能将这些设置回滚到之前的配置。

共享和协作的安全含义

大多数情况下，当用户共享内容与他人进行协作（例如，通过共享备忘录、共享提醒事项、iCloud 云盘中的共享文件夹或 iCloud 共享照片图库）且所有用户均已打开“高级数据保护”时，Apple 服务器仅会用于建立共享，但无法访问共享数据的加密密钥。共享内容仍采用端对端加密且只能在参与者的受信任设备上访问。对于每次共享操作，Apple 会采用标准数据保护来储存标题和示意缩略图以向接收用户显示预览。

启用协作时选择“任何拥有链接的用户”这一选项会采用标准数据保护来使内容可供 Apple 服务器访问，因为服务器需要能够为打开 URL 的任何用户提供访问权限。

iWork 协作和“照片”中的“共享相簿”功能不支持“高级数据保护”。当用户协作处理 iWork 文稿或者打开 iCloud 云盘中共享文件夹内的 iWork 文稿时，文稿的加密密钥会安全地上传到 Apple 数据中心的 iWork 服务器。这是因为在 iWork 中实时协作需要服务器端处理来协调参与者之间的文稿更改。添加到“共享相簿”的照片会采用标准数据保护进行储存，因为该功能允许在网上公开共享相簿。

停用“高级数据保护”

用户可以随时关闭“高级数据保护”。如果用户决定关闭，将出现以下情况：

1. 用户设备首先会在 iCloud 钥匙串参与元数据中记录其新选择，且此设置会在用户的所有设备间安全同步。

2. 用户设备会将所有**认证后可用**服务的服务密钥安全地上传到 Apple 数据中心的 iCloud HSM，但采用标准数据保护的端对端加密服务（例如 iCloud 钥匙串和“健康”）的密钥绝不会上传。

设备会同时上传在“高级数据保护”打开前生成的原始服务密钥和在用户打开该功能后生成的新服务密钥。这样会让这些服务中的所有数据在认证后可供访问，并且会将帐户恢复为采用标准数据保护，以便 Apple 可在用户无法访问其帐户时重新协助其恢复大部分数据。

“高级数据保护”未涵盖的 iCloud 数据

由于需要与全球电子邮件、通信录和日历系统进行互操作，iCloud 邮件、通讯录和日历未采用端对端加密。

即使打开“高级数据保护”，iCloud 也会储存部分不受用户特定 CloudKit 服务密钥保护的数据。在容器的模式中，CloudKit 记录字段必须明确声明为“加密”才能受到保护，而读取和写入加密字段需要使用专门的 API。文件或对象的修改日期和时间用于对用户的信息进行排序，而文件和照片数据的校验和则用于帮助 Apple 进行用户 iCloud 和设备储存空间的数据去重和优化，所有操作均不可访问文件和照片本身。有关特定数据类别所使用的加密方法的详细信息，请参阅 Apple 支持文章：iCloud 数据安全概览。

在推出 iCloud 服务时，其最初的设计决策之一便是使用校验和来进行数据去重，这项广为人知的技术称为**收敛加密**。此元数据会始终加密，但加密密钥由 Apple 采用标准数据保护进行储存。为继续提升对所有用户的数据安全保护，Apple 致力于确保在打开“高级数据保护”时，为包括此类元数据在内的更多数据提供端对端加密保护。

“高级数据保护”要求

打开 iCloud 高级数据保护的要求如下：

- 用户帐户必须支持端对端加密。端对端加密需要用户的 Apple ID 采用双重认证且在受信任的设备上设置密码。有关更多信息，请参阅 Apple 支持文章：Apple ID 的双重认证。
- 用户通过其 Apple ID 登录的设备必须更新至 iOS 16.2、iPadOS 16.2、macOS 13.1、Apple tvOS 16.2、watchOS 9.2 和最新的 Windows 版 iCloud。此要求可防止之前版本的 iOS、iPadOS、macOS、Apple tvOS 或 watchOS 将新创建的服务密钥重新上传到**认证后可用** HSM，从而错误地尝试修复帐户状态。
- 用户必须设置至少一种备用恢复方式（即设置一位或多位恢复联系人或者一个恢复密钥），以便在无法访问帐户时，通过其中一种方式来恢复其 iCloud 数据。

如果无法使用相关恢复方式，例如恢复联系人的信息过期或者用户忘记了这些信息，Apple 将无法协助恢复用户的端对端加密 iCloud 数据。

只有 Apple ID 支持打开 iCloud 高级数据保护。管理式 Apple ID 和儿童帐户（因国家或地区而异）不支持此功能。

iCloud 云备份安全性

iCloud 每天通过无线局域网备份信息，包括设备设置、App 数据、相机胶卷中的照片和视频以及“信息” App 中的对话。仅当设备处于锁定状态、接入电源且可通过无线局域网访问互联网时，iCloud 云备份才会工作。鉴于 iOS 和 iPadOS 采用的储存加密技术，iCloud 云备份的设计可在保护数据安全的同时，实现无人值守式增量备份和恢复。iCloud 云备份服务密钥默认会安全备份到 Apple 数据中心的 iCloud 硬件安全模块且属于认证后可用数据类别的一部分。对于打开 iCloud 高级数据保护的用户而言，iCloud 云备份服务密钥会采用端对端加密进行保护，且只可在受信任设备上供用户访问。

当以设备锁定时无法访问的数据保护类创建文件时，其文件独有密钥通过 iCloud 云备份密钥包中的类密钥进行加密，并以文件原始加密状态将其备份到 iCloud。所有文件在传输期间和储存时均会使用 CloudKit 加密中描述的基于帐户的密钥加密。

iCloud 云备份密钥包具有适用于数据保护类的非对称 (Curve25519) 密钥，设备锁定时便无法访问。备份集储存于用户的 iCloud 帐户中，并包括用户的文件副本和 iCloud 云备份密钥包。而随备份集一同储存的随机密钥会保护 iCloud 云备份密钥包。用户的 iCloud 密码不用于加密，因此更改 iCloud 密码不会导致现有备份失效。

恢复后，备份的文件、iCloud 云备份密钥包和密钥包的密钥将从用户的 iCloud 帐户取回。iCloud 云备份密钥包通过其密钥解密，然后密钥包中的文件独有密钥用于解密备份集中的文件，这些文件作为新文件写入文件系统中，从而根据其数据保护类对其重新加密。

以下内容会使用 iCloud 云备份来备份：

- 已购买的音乐、影片、电视节目、App 和图书的记录。用户的 iCloud 云备份包括用户设备上存在的已购内容的相关信息，但不包括已购内容本身。当用户从 iCloud 云备份恢复时，将自动从 iTunes Store、App Store、Apple TV App 或 Apple Books 下载已购内容。部分类型的内容并非在所有国家或地区都会自动下载，且之前的购买可能会变为不可用（如果已经退款或在各自的商店中不再可用）。完整的购买历史记录与用户的 Apple ID 关联。
- 用户设备上的照片和视频。请注意，如果用户在 iOS 8.1、iPadOS 13.1、OS X 10.10.3 或更高版本中打开了“iCloud 照片”，其照片和视频已经储存在 iCloud 中，因此不会包括在用户的 iCloud 云备份中。
- 联系人、日历日程、提醒事项和备忘录
- 设备设置信息
- App 数据
- 主屏幕与 App 排列方式
- HomeKit 配置
- 医疗急救卡数据
- “语音备忘录”密码（必要时，需要备份期间使用的实体 SIM 卡）
- 信息、Apple Messages for Business、短信和彩信（必要时，需要备份期间使用的实体 SIM 卡）

iCloud 云备份还会用于备份本机钥匙串，该钥匙串使用由设备安全隔区 UID 根加密密钥派生的密钥进行加密。此密钥为设备独有，Apple 也不知悉。这样使得数据库仅能恢复至生成它的同一台设备，意味着任何人（包括 Apple）均无法读取。有关更多信息，请参阅安全隔区。

iCloud 云端“信息”

iCloud 云端“信息”可让用户的全部信息历史记录在所有设备上可用并保持最新。

采用标准数据保护的 iCloud 云端“信息”会在 iCloud 云备份关闭后使用端对端加密。打开 iCloud 云备份时，备份会包括 iCloud 云端“信息”加密密钥的副本，这样即使用户无法访问 iCloud 钥匙串及其受信任设备，Apple 也能协助其恢复信息。如果用户关闭 iCloud 云备份，则设备上会生成新密钥以保护后续的 iCloud 云端“信息”。新密钥只会储存在 iCloud 钥匙串中且用户只能从其受信任设备上访问，而写入容器的新数据不能使用旧容器密钥解密。

打开“高级数据保护”后，iCloud 云端“信息”始终采用端对端加密。打开 iCloud 云备份后，包括 iCloud 云端“信息”加密密钥在内的所有云端内容均采用端对端加密。用户打开“高级数据保护”后，iCloud 云备份服务密钥以及 iCloud 云端“信息”容器密钥均会收回。有关更多信息，请参阅 Apple 支持文章：iCloud 数据安全概览。

帐户恢复联系人安全性

无论是否打开了“高级数据保护”，用户最多可将其信任的五个人添加为帐户恢复联系人，以帮助恢复其 iCloud 帐户及数据，包括所有端对端加密数据。Apple 和恢复联系人均没有可独立恢复用户端对端加密 iCloud 数据的必要信息。

恢复联系人在设计时考虑到了用户隐私。Apple 无法得知用户选择的恢复联系人。只有当用户寻求联系人的帮助且联系人实际开始协助恢复后，Apple 服务器才会在尝试恢复的后期了解恢复联系人的相关信息。恢复完成后，该信息不会被保留。

恢复联系人安全流程

当用户设置帐户恢复联系人时，用于访问用户 iCloud 数据（包括端对端加密 CloudKit 数据）的密钥会使用强随机密钥进行加密，然后该随机密钥会被拆分并分别交给恢复联系人和 Apple。在恢复过程中，只有当这两份密钥重新结合时才能恢复原始密钥和访问用户的 iCloud 数据。

为了设置帐户恢复联系人，用户设备会与 Apple 服务器通信以上传 Apple 将持有的那份密钥信息。随后设备通过恢复联系人建立端对端对端加密的 CloudKit 容器，以共享恢复联系人所需要的部分。Apple 和恢复联系人也都会收到来自用户的相同授权密钥，供稍后恢复使用。该通信通过相互认证的 IDS 通道进行，用于邀请和接受恢复联系人。恢复联系人将接收的信息自动储存在其 iCloud 钥匙串中。Apple 无法访问 CloudKit 容器内容，也无法访问储存该信息的 iCloud 钥匙串。执行共享时，Apple 服务器仅查看恢复联系人的匿名 ID。

之后，当用户需要恢复其帐户及 iCloud 数据时，可以向其恢复联系人寻求帮助。届时恢复联系人的设备会生成一个恢复验证码，随后恢复联系人可将其以非网络方式（例如当面或打电话）提供给用户。用户随后在其设备上输入恢复验证码，以使用 SPAKE2+ 协议在设备间建立安全连接，Apple 无法访问该内容。此交互过程由 Apple 服务器协调，但 Apple 无法发起恢复流程。

在建立安全连接和完成所有必需的安全检查后，恢复联系人的设备会将它那一部分的密钥信息以及之前建立的授权密钥返回给提出恢复请求的用户。用户向 Apple 服务器出示此授权密钥，以获得授权访问 Apple 所持有的密钥信息。提供授权密钥还会授权重设帐户密码，以恢复帐户访问权限。

最后，用户设备将接收自 Apple 和帐户恢复联系人的密钥信息重新结合，然后将其用于解密和恢复 iCloud 数据。

多重保护措施各司其职，用于防止恢复联系人未经用户同意而发起恢复流程，其中包括针对用户帐户的活跃性检查。如果帐户正处于活跃使用状态，使用恢复联系人的恢复流程还会要求提供最近使用的设备密码或 iCloud 安全码。

遗产联系人安全性

如果用户想让指定受益人在自己去世后可访问自己的 iCloud 数据，可以为帐户设置遗产联系人。遗产联系人受益人可访问已故者的所有 iCloud 数据，包括几乎所有端对端加密数据，但不包括帐户密码这类 iCloud 钥匙串数据。遗产联系人采用了与恢复联系人类似的底层技术，即强随机密钥会被拆分并分别交给 Apple 和遗产联系人，这样任一方都无法独自解密任何数据。无论用户是否打开了“高级数据保护”，受益人都会收到同类数据。

受益人收到的密钥信息在面向用户的文稿中称为“访问密钥”，会自动存储在支持的设备上，也可被打印和离线储存以供使用。有关更多信息，请参阅 Apple 支持文章：如何为你的 Apple ID 添加遗产联系人。

用户去世后，遗产联系人可登录请求访问的 Apple 网站以发起请求。此操作需要死亡证明，并由前文中所提到的授权密钥部分授权。在完成所有安全验证后，Apple 会对遗产联系人签发新帐户的用户名及密码并发放必要的密钥信息。

为了便于遗产联系人根据需要提供访问密钥，它会以字母数字代码及关联二维码的形式显示。输入后，对已故者 iCloud 数据的访问权限会恢复。可在设备上执行此操作，也可在线建立访问。有关更多信息，请参阅 Apple 支持文章：以遗产联系人的身份请求访问某个 Apple 帐户。

iCloud 专用代理安全性

iCloud 专用代理主要帮助在用户使用 Safari 浏览器浏览网页时进行保护，还会涉及所有 DNS 域名解析请求。这有助于确保没有任何一方（包括 Apple）可以关联用户的 IP 地址及其浏览活动。为此，它使用了不同的代理：由 Apple 管理的入口代理，以及由内容提供商管理的出口代理。若要使用 iCloud 专用代理，用户设备必须运行 iOS 15、iPadOS 15、macOS 12.0.1 或更高版本，并且使用自己的 Apple ID 登录 iCloud+ 帐户。随后用户便可在“设置”> iCloud 或“系统设置”> iCloud 中打开 iCloud 专用代理。

有关更多信息，请参阅 iCloud 专用代理概览。

密码管理

密码安全性概览

iOS、iPadOS 和 macOS 允许用户轻松认证使用密码的第三方 App 和网站。管理密码最好的方式就是不使用密码。“通过 Apple 登录”可让用户无需创建和管理额外的帐户或密码即可登录第三方 App 和网站，同时通过适用于 Apple ID 的双重认证保护登录。对于不支持“通过 Apple 登录”的站点，“自动强密码”功能可让用户的设备针对站点和 App 自动创建、同步和输入唯一的强密码。在 iOS 和 iPadOS 中，密码会存储到特殊的“自动填充密码”钥匙串中，用户可在“设置”>“密码”中控制和管理。

在 macOS 中，已存储的密码可在 Safari 浏览器的“密码”偏好设置中管理。此同步系统也可用于同步由用户手动创建的密码。

通过 Apple 登录安全性

相对于其他单点登录系统，“通过 Apple 登录”是一种能更好保护隐私的替代登录方式。“通过 Apple 登录”让用户轻点一下即可登录，对于用户而言更加便利和高效，且为用户带来了更透明的体验并赋予用户更多的个人信息控制权。

“通过 Apple 登录”允许用户使用已有的 Apple ID 设置帐户和登录 App 及网站，并且赋予用户更多控制其个人信息的权利。设置帐户时，App 只可以请求用户的名字和电子邮件地址，并且用户始终有选择权：用户可与 App 共享其个人电子邮件地址，或选择将个人电子邮件地址保密并改为使用 Apple 新的保密电子邮件中继服务。此电子邮件中继服务会与 App 共享一个唯一的匿名电子邮件地址并向用户的个人地址转发邮件，因此用户仍可收到来自开发者的有用信息，同时又可保持一定程度的隐私以及个人信息的控制权。

“通过 Apple 登录”十分注重安全性。“通过 Apple 登录”的每位用户都需要为其 Apple ID 启用双重认证。双重认证不仅可以保护用户 Apple ID 的安全，还可保护使用 App 建立的帐户的安全。另外，Apple 在“通过 Apple 登录”中开发并集成了保护隐私且反欺诈的信号。此信号让开发者确信其获得的新用户是真人，而不是机器人或脚本帐户。

自动强密码

如果 iCloud 钥匙串已启用，当用户登录或者在 Safari 浏览器网站上更改密码时，iOS、iPadOS 和 macOS 会创建唯一的随机强密码。在 iOS 和 iPadOS 中，自动强密码生成也适用于 App。用户必须自行选择不使用强密码。生成的密码存储在钥匙串中，并在启用了 iCloud 钥匙串的设备间保持更新。

iOS 和 iPadOS 默认生成的密码长度为 20 个字符。其中包含一位数字、一个大写字符、两个连字符和 16 个小写字符。这些生成的密码均为包含 71 位熵的强密码。

密码基于一种启发技术生成，这种启发技术可确定密码栏是否用于密码创建。如果启发技术未能识别创建密码时使用的特定环境密码，App 开发者可在其文本栏上设定 `UITextContentType.newPassword`，网页开发者可在其 <input> 元素中设定 `autocomplete= “new-password”`。

为帮助确保生成的密码与相关服务兼容，App 和网站可提供规则。开发者使用 `UITextInputPasswordRules` 或输入元素上的 `passwordrules` 属性来提供这些规则。设备随后会生成可以满足这些规则的最强的密码。

自动填充密码安全性

“自动填充密码”会自动填充储存在钥匙串中的凭证。iCloud 钥匙串密码管理器和“自动填充密码”提供以下功能：

- 填充 App 和网站中的凭证
- 生成强密码
- 存储 App 和 Safari 浏览器网站中的密码
- 将密码安全共享给用户的联系人
- 将密码提供给附近请求凭证的 Apple TV

在 App 中生成和存储密码以及向 Apple TV 提供密码仅适用于 iOS 和 iPadOS。

App 中的自动填充密码

iOS 和 iPadOS 允许用户在 App 中的凭证相关栏中输入已存储的用户名和密码，其工作方式与 Safari 浏览器中的“自动填充密码”类似。在 iOS 和 iPadOS 中，用户可轻点软件键盘快速输入栏中的按键直观功能。在 macOS 中，对于使用 Mac Catalyst 构建的 App，凭证相关栏下方会显示“密码”下拉菜单。

如果与 App 紧密关联的网站使用相同的 app-website 关联机制且由同一 apple-app-site-association 文件提供支持，iOS 和 iPadOS 快速输入栏和 macOS 下拉菜单会直接建议用于 App 的凭证，前提是凭证存储到“自动填充密码”钥匙串中。这允许用户选择将 Safari 浏览器所存储的凭证透露给具有相同安全性属性的 App，而这些 App 无需采用 API。

“自动填充密码”不会将凭证信息透露给 App，除非用户同意将凭证发放给 App。凭证列表的下拉或显示不包括在 App 的流程中。

当 App 和网站具有信任关系，且用户在 App 内提交了凭证时，iOS 和 iPadOS 可能会提示用户将这些凭证存储到“自动填充密码”钥匙串中供后续使用。

App 访问已存储密码的权限

iOS、iPadOS 和 macOS App 可以使用 `ASAuthorizationPasswordProvider` 和 `SecAddSharedWebCredential` 来请求“自动填充密码”钥匙串帮助用户登录。密码提供程序及其请求可与“通过 Apple 登录”一起使用，这样就可以调用相同的 API 帮助用户登录 App，不管用户的帐户是基于密码的帐户，还是使用“通过 Apple 登录”创建的帐户。

只有 App 开发者和网站管理员同时批准且用户同意后，App 才可访问已存储的密码。App 开发者通过在其 App 中加入授权，让系统获知他们需要访问 Safari 浏览器已存储的密码。授权书中列出了关联网站的完全限定域名，网站必须将文件放在其服务器上，并在其中列出已获 Apple 批准的 App 的唯一 App 标识符。

安装了带有 com.apple.developer.associated-domains 授权的 App 后，iOS 和 iPadOS 向列表中的每个网站发出 TLS 请求来请求以下其中一个文件：

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

如果文件列出被安装 App 的 App 标识符，则 iOS 和 iPadOS 会将网站和 App 标记为具有信任关系。只有在具有信任关系的情况下，调用这两个 API 时才会用户发出提示，用户同意之后，密码才能发放给 App，或者被更新或删除。

密码安全建议

iOS、iPadOS 和 macOS 中“自动填充密码”的密码列表会指示用户已存储的哪些密码在其他网站上**重复使用**、哪些密码强度**太弱**以及哪些密码在**数据泄露**中被盗。

概览

在不同服务上使用同一个密码可能会让帐户容易受到撞库攻击。如果服务被攻破并且密码外泄，攻击者可能在其他服务上尝试相同的凭证，导致更多帐户被盗。

- 如果已存储的密码在不同域间多次重复使用，则密码会被标记为**重复使用**。
- 如果密码很容易被攻击者猜到，则会标记为强度**太弱**。iOS、iPadOS 和 macOS 会检测用于创建容易记住密码的常见规律，如使用在词典里找到的字词、替换常见字母（如用“p4ssw0rd”代替“password”）、使用在键盘上找到的规律（如 QWERTY 键盘上的“q12we34r”）或重复的序列（如“123123”）。这些规律经常用于创建满足服务最低密码要求的密码，但也常被攻击者用于尝试通过暴力获取密码。

  许多服务特别要求使用 4 位或 6 位的数字 PIN 码，因此这些短密码会通过不同的规则进行评估。如果 PIN 码是最常用的 PIN 码之一，或为升序或降序的序列（如“1234”或“8765”），或者遵循重复的规律（如“123123”或“123321”），则其强度会被视为太弱。
- 如果“密码监视”功能显示密码在数据泄露中出现过，则密码会被标记为**已泄露**。有关更多信息，请参阅密码监视。

强度较弱、重复使用和已泄露的密码会在密码列表 (macOS) 中标识出来，或者显示在专门的“安全建议”界面 (iOS 和 iPadOS)。如果用户在 Safari 浏览器中使用以前存储的弱强度密码或在数据泄露中出现过的密码登录网站，则会出现提示，强烈建议用户升级到自动强密码。

在 iOS 和 iPadOS 中升级帐户认证安全性

采用帐户认证修改扩展（在“认证服务”框架中）的 App 让你轻点一下按钮，即可将基于密码的帐户轻松升级以切换为“通过 Apple 登录”或使用自动强密码。此扩展点可在 iOS 和 iPadOS 中使用。

如果 App 已经采用了该扩展点并安装在设备上，用户在“设置”的 iCloud 钥匙串密码管理器中查看与该 App 相关联凭证的“安全建议”时，会看到扩展升级选项。当用户通过存在风险的凭证登录 App 时，也会看到升级选项。App 能够告诉系统在用户登录后不显示升级选项。通过使用全新的 AuthenticationServices API，App 还可以调用自己的扩展，并在理想情况下在 App 的帐户设置或帐户管理屏幕中自行升级。

App 可以选择支持强密码升级，“通过 Apple 登录”升级或者同时支持二者。在强密码升级过程中，系统会为用户生成自动强密码。如有需要，在生成新密码时，App 可提供自定密码规则供用户遵循。如果用户将帐户从使用密码切换为使用“通过 Apple 登录”，系统会为该扩展提供新的“通过 Apple 登录”凭证以与该帐户关联。用户的 Apple ID 电子邮件不会作为凭证的一部分提供。“通过 Apple 登录”升级成功后，系统会从用户的钥匙串中删除之前所使用的密码凭证（如果凭证存储在此处）。

帐户认证修改扩展可以在执行升级前选择执行额外的用户认证。对于在密码管理器中或登录 App 后发起的升级，扩展会提供帐户的用户名和密码进行升级。对于 App 内升级，则只会提供用户名。如果扩展要求进行进一步用户认证，可以请求在升级前显示自定用户界面。显示此用户界面的预期用例是让用户输入认证的第二个因素来认证升级。

密码监视

“密码监视”功能是将储存在用户“自动填充密码”钥匙串中的密码与已知在不同在线组织的密码泄露中出现过的密码列表进行对比，该列表将持续更新并精选。如果打开了该功能，监视协议会持续将用户“自动填充密码”钥匙串中的密码与该精选列表进行对比。

监视的工作方式

用户的设备会持续轮替检查用户的密码，查询间隔不受用户密码或密码管理器使用模式影响，从而帮助确保验证状态与当前已泄露密码的精选列表保持同步更新。为了帮助防止泄露用户拥有多少个唯一密码的相关信息，查询请求会进行批处理且并行执行。每次检查时会并行验证固定数量的密码，如果用户拥有的密码数量少于此数量，则会生成随机密码并添加到查询以补齐数量差。

密码如何匹配

密码分为两个阶段进行匹配。最常见的泄露密码包含在用户设备上的本地列表中。如果用户的密码包含在此列表中，用户会立即收到通知，无需执行任何外部交互。此设计旨在确保用户所拥有的这类在数据泄露中风险最高的密码的任何相关信息不会遭到泄露。

如果密码未包含在最常见的泄露密码列表中，那么会在不常见的泄露密码列表中进行对比。

将用户密码与精选列表对比

验证密码是否包含在本地列表中的这个对比过程涉及一些与 Apple 服务器的交互。为了帮助确保合法用户的密码不会发送给 Apple，对比过程采用了加密**隐私保护集合交集**这种形式，即将用户的密码与大型泄露密码集进行对比。这样做旨在确保只会与 Apple 共享泄露风险较小的密码的少量信息，对于用户密码，此信息仅限于加密哈希值的 15 位前缀。从这个交互过程中移除最常见的泄露密码使用的是最常见泄露密码的本地列表，由此减少了使用网络服务时密码在存储桶中出现的相对频率的增量，从而使用户密码无法从这些查询中推算出来。

基础协议会将精选密码列表（编写本文时包含约 15 亿个密码）细分为 2^{15} 个不同的存储桶。密码所属的存储桶取决于密码 SHA256 哈希值的前 15 位。此外，每个泄露的密码 pw 与 NIST P256 曲线上的一个椭圆曲线点相关联：$P_{pw} = \alpha \cdot H_{SWU}(pw)$，其中 α 是仅为 Apple 所知的随机密钥，而 H_{SWU} 是一个随机 Oracle 函数，它会将密码映射到基于 Shallue-van de Woestijne-Ulas 方法的曲线点。此转换过程旨在通过计算方法隐藏密码的值并帮助防止通过“密码监视”透露最新泄露的密码。

为了计算隐私保护集合交集，用户密码所属的存储桶由用户的设备使用 λ（即 SHA256(upw) 的 15 位前缀）来确定，其中 upw 是用户的密码之一。设备生成自己的随机常量 β 并将点 $P_c = \beta \cdot H_{SWU}(upw)$ 连同与 λ 对应的存储桶查询请求发送到服务器。其中 β 会隐藏用户密码相关的信息并限制 λ 透露密码中的信息给 Apple。最终服务器会收到用户设备发送的点，计算 $\alpha P_c = \alpha\beta \cdot H_{SWU}(upw)$，并向设备返回计算结果以及点 $B\lambda = \{ P_{pw} \mid SHA256(pw)$ 以前缀 λ 开头$\}$ 对应的存储桶。

所返回的信息允许设备计算 $B'_\lambda = \{\beta \cdot P_{pw} \mid P_{pw} \in B_\lambda\}$；如果 $\alpha P_c \in B'_\lambda$，则确定用户的密码已经泄露。

将密码发送给其他用户或 Apple 设备

Apple 借助“隔空投送”和 Apple TV 将密码安全地发送到其他用户或 Apple 设备。

通过隔空投送将凭证存储到其他设备

iCloud 启用时，用户可使用“隔空投送”将存储的凭证发送到另一台设备。凭证包括用户名和密码，以及其所对应的网站。无论用户的设置如何，通过“隔空投送”发送凭证操作始终在“仅限联系人”模式下进行。在接收设备上，用户同意后，凭证会储存在用户的“自动填充密码”钥匙串中。

在 Apple TV 上的 App 中填充凭证

在 Apple TV 上的 App 中可使用“自动填充密码”填充凭证。用户在 Apple tvOS 中聚焦到用户名或密码文本栏时，Apple TV 会开始通过低功耗蓝牙 (BLE) 来广播“自动填充密码”请求。

附近的所有 iPhone、iPad 或 iPod touch 都会显示提示，邀请用户与 Apple TV 共享凭证。以下是加密方法的建立方式：

- 如果设备和 Apple TV 使用同一个 iCloud 帐户，设备间的加密会自动完成。
- 如果设备登录的 iCloud 帐户与 Apple TV 使用的不同，用户会收到通过使用 PIN 码来建立加密连接的提示。iPhone 必须解锁，并且靠近与该 Apple TV 配对的 Siri Remote 遥控器，才能收到此提示。

使用 BLE 链接加密建立加密的连接后，凭证会发送到 Apple TV，并自动填充到 App 中相关的文本栏内。

凭证提供程序扩展

在 iOS、iPadOS 和 macOS 中，用户可以在“密码”设置中的“自动填充密码”(iOS 和 iPadOS) 或“系统偏好设置”中的“扩展”设置 (macOS) 内将合作的第三方 App 指定为凭证提供程序。此机制以 App 扩展为基础构建。凭证提供程序扩展必须提供选取凭证的视图，并可选择提供有关已存储凭证的元数据，以便直接在快速输入栏 (iOS 和 iPadOS) 或自动补全建议 (macOS) 中提供凭证。元数据包括凭证针对的网站和关联的用户名，但不包括密码。用户选择将凭证填充到 App 或 Safari 浏览器网站中时，iOS、iPadOS 和 macOS 会与扩展进行通信以获取密码。凭证元数据储存在凭证提供程序 App 的容器内，App 卸载时会自动移除。

iCloud 钥匙串

iCloud 钥匙串安全性概览

iCloud 可帮助用户在 iOS 和 iPadOS 设备和 Mac 电脑之间安全地同步密码，而不会将此信息泄露给 Apple。除了强大的隐私保护和安全性，易用性和恢复钥匙串的功能对 iCloud 钥匙串的设计和架构也具有重要影响。iCloud 钥匙串包含两项服务：钥匙串同步和钥匙串恢复。

Apple 设计的 iCloud 钥匙串和钥匙串恢复可确保用户的密码在下列情况下仍然受到保护：

- 用户的 iCloud 帐户被盗。
- iCloud 遭到外部攻击者或员工的入侵。
- 第三方访问用户帐户。

密码管理器与 iCloud 钥匙串整合

iOS、iPadOS 和 macOS 可以在 Safari 浏览器中自动生成加密的随机强密码字符串作为帐户密码。iOS 和 iPadOS 还可为 App 生成强密码。生成的密码会储存在钥匙串中并同步到其他设备。钥匙串项通过 Apple 服务器在不同的设备之间传输，但会严格进行加密，Apple 和其他设备无法读取其内容。

钥匙串安全同步

当用户首次启用 iCloud 钥匙串时，设备将建立信任圈并为自己创建同步身份。同步身份包括私钥和公钥，且储存在设备的钥匙串中。同步身份的公钥放置在信任圈中，该信任圈已签名两次：第一次由同步身份的私钥签署，第二次由来自用户 iCloud 帐户密码的非对称椭圆密钥 (使用 P-256) 签署。随信任圈一起储存的还有参数 (随机盐和迭代次数)，用于创建基于用户 iCloud 密码的密钥。

对于双重认证的帐户，还会额外创建一个相似的同步信任圈并存储在 CloudKit 中。此系统中的设备身份由两对非对称椭圆密钥 (使用 P-384) 组成，且储存在钥匙串中。每台设备均维护本机信任的身份列表，且使用其中一个身份密钥为此列表签名。

iCloud 储存同步信任圈

已签名的同步信任圈储存在用户的 iCloud 键值存储区域。如果不知道用户的 iCloud 密码，就无法对其进行读取；如果没有信任圈成员同步身份的私钥，就无法对其进行有效地修改。

对于双重认证的帐户，每台设备的同步列表会存储在 CloudKit 中。如果不知道用户的 iCloud 密码，就无法读取列表；如果没有已拥有设备的私钥，就无法修改私钥。

用户的其他设备如何添加到同步信任圈

新设备登录 iCloud 时，可通过这两种方式之一加入 iCloud 钥匙串同步信任圈：与现有的 iCloud 钥匙串设备配对并获得其担保，或者使用 iCloud 钥匙串恢复。

在配对流程中，申请设备为同步信任圈和同步列表（针对双重认证帐户）新建同步身份，并将其出示给担保设备。担保设备将新成员的公钥添加至同步信任圈，并且使用其同步身份和派生自用户 iCloud 密码的密钥再次签名。新的同步信任圈放置在 iCloud 中，该信任圈的新成员也是在此处对其进行签名。在双重认证帐户中，担保设备还为加入设备提供使用其身份密钥签名的**凭证**，表明申请设备应被信任。然后，它会更新其受信任同步身份的单独列表，以包括申请设备。

现在，签名信任圈有两个成员，并且每个成员拥有另一个成员的公钥。它们现在开始通过 iCloud 键值存储交换各个钥匙串项或将其储存在 CloudKit 中，视情境而定。如果两个信任圈成员都更新了相同的项目，其中一个会被选中，最终获得一致的结果。每个同步的项目都会加密，使只有用户信任圈内的设备才能解密，任何其他设备或 Apple 都无法解密。

新设备加入同步信任圈时会重复此“加入流程”。例如，第三台设备加入时，可与其中任一台现有设备进行配对。随着新的同级设备加入，每台同级设备均与新设备进行同步。此设计旨在确保所有成员拥有相同的钥匙串项。

仅同步部分项目

部分钥匙串项特定于设备，如 iMessage 信息密钥，因而必须保留在设备上。因此，每个将要同步的项目都必须使用 `kSecAttrSynchronizable` 属性明确标记。

Apple 为 Safari 浏览器用户数据（包括用户名、密码和信用卡号）、无线局域网密码、HomeKit 加密密钥以及其他支持端对端 iCloud 加密的钥匙串项设置了该属性。

另外，在默认情况下，第三方 App 添加的钥匙串项不会进行同步。将项目添加至钥匙串时，开发者必须设置 `kSecAttrSynchronizable` 属性。

iCloud 钥匙串安全恢复

iCloud 钥匙串会将用户的钥匙串数据交予 Apple 托管，但**不允许** Apple 读取密码和钥匙串包含的其他数据。即使用户只有一台设备，钥匙串恢复也可提供一个防止数据丢失的安全网。当 Safari 浏览器用于为网络帐户生成随机强密码时，这尤其重要，因为这些密码的唯一记录在钥匙串中。

钥匙串恢复包含两大基本要素：二次身份认证和安全托管服务，后者是 Apple 专为支持此功能而创建的服务。用户的钥匙串通过强密码进行加密，只有满足一系列严格的条件，托管服务才会提供钥匙串副本。

使用双重认证

建立强密码有以下几种方法：

- 如果用户帐户启用了双重认证，则使用设备密码来恢复托管的钥匙串。
- 如果未设置双重认证，用户需要通过提供六位数字密码来创建 iCloud 安全码。除此之外，如果未设置双重认证，用户可以自行指定较长的安全码或允许其设备创建随机加密的安全码，他们可以自行记录和保存。

钥匙串托管流程

密码建立后，钥匙串由 Apple 托管。iOS、iPadOS 或 macOS 设备先导出用户的钥匙串副本，然后将其与密钥加密封装于非对称钥匙包中，并放置在用户的 iCloud 键值存储区域。密钥包被用户的 iCloud 安全码和储存托管记录的硬件安全模块 (HSM) 集群公钥所封装。它会变成用户的 **iCloud 托管记录**。对于双重认证帐户，钥匙串还储存在 CloudKit 中并封装到只能通过 iCloud 托管记录的内容才能恢复的中间密钥，因此提供了相同级别的安全保护。

托管记录的内容还允许恢复设备重新加入 iCloud 钥匙串，向所有现有设备证明恢复设备已成功执行托管流程并因此获得帐户所有者的授权。

【注】如果用户决定接受随机加密的安全码而不自行指定或使用 4 位数值，则不再需要托管记录。此时，会使用 iCloud 安全码直接封装随机密钥。

除了创建安全码，用户还必须注册电话号码。这在钥匙串恢复过程中提供了第二重认证。用户收到短信后，必须回复才能继续恢复。

iCloud 钥匙串的托管安全性

iCloud 为钥匙串托管提供了安全的基础架构，帮助确保只有经过授权的用户和设备才能执行恢复。iCloud 背后部署的是硬件安全模块 (HSM) 集群，这些集群为托管记录提供保护。如上所述，集群的每位成员都有一个密钥，用于对其监管下的托管记录进行加密。

要恢复钥匙串，用户必须使用其 iCloud 帐户和密码进行身份认证，并对发送至其注册电话号码的短信进行回复。回复完成后，用户必须输入其 iCloud 安全码。HSM 集群利用安全远程密码 (SRP) 协议验证用户是否知道其 iCloud 安全码；密码本身不会发送给 Apple。集群的每位成员单独验证用户是否未超过取回记录时所允许的最大尝试次数，如下所述。如果多数成员同意，集群会解封托管记录并将其发送至用户的设备。

下一步，设备使用 iCloud 安全码解封用于加密用户的钥匙串的随机密钥。利用该密钥，可以解密从 iCloud 键值存储和 CloudKit 取回的钥匙串并将其恢复到设备中。iOS、iPadOS 和 macOS 允许对托管记录认证和取回最多 10 次。多次尝试失败后，记录将被锁定，用户必须联系 Apple 技术支持部门才能进行更多尝试。第 10 次尝试失败后，HSM 集群将销毁托管记录，钥匙串将永久丢失。这种方式以牺牲钥匙串数据为代价，防止通过暴力破解方式来取回记录。

这些策略已编入 HSM 固件中。允许更改固件的管理访问卡已经被销毁。任何尝试更改固件或访问私钥的操作，都会导致 HSM 集群删除私钥。如果发生这种情况，受集群保护的每个钥匙串的所有者都会收到信息，告知其托管记录已经丢失。他们随后可以选择重新注册。

Apple Pay

Apple Pay 安全性概览

通过 Apple Pay，用户可以使用支持的 iPhone、iPad、Mac 和 Apple Watch 设备以方便、安全和保密的方式在商店里、App 中和使用 Safari 浏览器在网上付款。用户还可以将支持 Apple Pay 的交通卡、学生证和门禁卡添加到 Apple 钱包。Apple Pay 操作简单，且在硬件和软件中都采用了集成安全技术。

Apple Pay 的设计还可以保护用户的个人信息。Apple Pay 不会收集可追溯到用户的任何交易信息。付款交易在用户、商家和发卡机构之间发生。

Apple Pay 组件安全性

Apple Pay 使用多种软硬件功能提供安全可靠的购买方式。

安全元件

安全元件是业内公认、运行 Java Card 平台的认证芯片，它符合金融行业对电子支付的要求。安全元件 IC 和 Java Card 平台依据 EMVCo Security Evaluation 流程进行认证。成功完成安全性评估后，EMVCo 会签发唯一的 IC 和平台证书。

安全元件 IC 已基于通用标准进行认证。有关更多信息，请参阅《Apple 平台认证》中的安全隔区处理器安全认证。

NFC 控制器

NFC 控制器处理“近距离无线通信”协议，并发送应用程序处理器和安全元件之间以及安全元件和销售点终端之间的通信。

Apple 钱包

Apple 钱包 App 用于添加和管理信用卡、借记卡和商店卡，并通过 Apple Pay 进行支付。用户可以在 Apple 钱包中查看其付款卡，并可能可以查看由发卡机构提供的其他信息，如发卡机构的隐私政策、最近的交易等内容。还可以从以下位置将付款卡添加到 Apple Pay：

- iOS 和 iPadOS 上的“设置助理”和“设置”
- 针对 Apple Watch 的 Watch App
- 配备触控 ID 的 Mac 电脑上“系统偏好设置”中的“钱包与 Apple Pay”

此外，Apple 钱包还允许用户添加和管理交通卡、回馈卡、登机牌、票据、充值卡、学生证、门禁卡等。

安全隔区

在 iPhone、iPad、Apple Watch、配备触控 ID 的 Mac 电脑以及使用妙控键盘（配备触控 ID）且搭载 Apple 芯片的 Mac 电脑上，安全隔区负责管理认证流程并让支付交易得以继续。

在 Apple Watch 上，设备必须解锁且用户必须连按侧边按钮两下。检测到的连按操作会直接发送到安全元件或安全隔区（可用时），而不经过应用程序处理器。

Apple Pay 服务器

Apple Pay 服务器负责管理 Apple 钱包中信用卡、借记卡、交通卡、学生证和门禁卡的设置和预置。储存在安全元件中的设备卡号也由服务器管理。它们同时与设备和支付网络服务器或发卡机构服务器通信。Apple Pay 服务器还负责再次加密在 App 内或网上进行支付时的支付凭证。

Apple Pay 如何保护用户的购买

安全元件

安全元件包含专门设计用来管理 Apple Pay 的小程序，还包含由支付网络或发卡机构认证的小程序。加密的信用卡、借记卡或储值卡数据从支付网络或发卡机构发送到这些小程序，期间使用仅为支付网络或发卡机构和小程序的安全域所知的密钥。此数据储存在这些小程序内，并使用安全元件的安全性功能进行保护。交易期间，终端使用专门的硬件总线通过近距离无线通信 (NFC) 控制器直接与安全元件进行通信。

NFC 控制器

作为安全元件的入口，NFC 控制器可帮助确保所有免接触式支付交易都通过处于设备近距离范围内的销售点终端进行。NFC 控制器只会将来自场内终端的支付请求标记为免接触式交易。

持卡人使用面容 ID、触控 ID 或密码授权信用卡、借记卡或储值卡（包括商店卡）支付，或者在解锁的 Apple Watch 上通过连按侧边按钮两下来授权支付后，控制器会将安全元件内支付小程序准备的免接触式响应专门发送给 NFC 场。因此，免接触式支付交易的支付授权详细信息会包含在本地 NFC 场中，绝不会透露给应用程序处理器。相比之下，在 App 内和网上进行支付时，支付授权详细信息会被发送到应用程序处理器，但只有在安全元件加密后才会发送给 Apple Pay 服务器。

信用卡、借记卡和储值卡

卡片预置安全性概览

当用户将信用卡、借记卡或储值卡（包括商店卡）添加到 Apple 钱包时，Apple 会安全地将付款卡信息以及关于用户帐户和设备的其他信息，发送给发卡机构或发卡机构的授权服务商。发卡机构会使用此信息，决定是否批准将付款卡添加到 Apple 钱包。作为付款卡预置流程的一部分，Apple Pay 使用三个服务器端调用命令来发送和接收与发卡机构或网络间的通信：

- 必填栏位
- 核对付款卡
- 链接和预置

发卡机构或网络使用这些调用命令来验证、批准付款卡并将其添加到 Apple 钱包。这些客户端服务器会话使用 TLS 1.2 传输数据。

完整的付款卡卡号不会储存在设备或 Apple Pay 服务器上，而是会创建唯一的设备卡号并进行加密，然后储存在安全元件中。此唯一的设备卡号采用 Apple 无法访问的方式加密。设备卡号是独一无二的，并与大部分信用卡或借记卡卡号不同。发卡机构或支付网络可使设备卡号无法在磁条卡、电话或网站上使用。安全元件中的设备卡号绝不会储存在 Apple Pay 服务器上或备份到 iCloud，而且与 iOS、iPadOS 和 watchOS 设备以及配备触控 ID 的 Mac 隔离开来。

在 iPhone 上的 Apple Watch App 中，或在发卡机构的 iPhone App 内，用户可以为 Apple Pay 预置配合 Apple Watch 使用的付款卡。为 Apple Watch 添加付款卡时，要求手表位于蓝牙通信范围内。配合 Apple Watch 使用的付款卡会进行特别注册并拥有自己的设备卡号，该卡号储存在 Apple Watch 上的安全元件内。

添加信用卡、借记卡或储值卡（包括商店卡）后，它们会在设备（登录了相同 iCloud 帐户）的“设置助理”运行过程中显示在付款卡列表内。只要这些付款卡在至少一台设备上是活跃的，它们就会保留在列表中。付款卡从所有设备上移除的 7 天后也会从列表中移除。此功能要求相应的 iCloud 帐户启用双重认证。

将信用卡或借记卡添加到 Apple Pay

用户可将信用卡手动添加到 Apple 设备的 Apple Pay 中。

手动添加信用卡或借记卡

若要手动添加付款卡，需要使用姓名、卡号、过期日期和 CVV 码来辅助预置流程。用户可以在“设置”、Apple 钱包或 Apple Watch App 中键入或使用设备上的摄像头来输入该信息。摄像头捕获到付款卡信息后，Apple 会尝试填充姓名、卡号和过期日期。所拍摄的照片绝对不会存储到设备或储存在照片图库中。在填写好所有栏位后，“核对付款卡”流程会验证 CVV 码以外的栏位。这些信息随后会被加密并发送到 Apple Pay 服务器。

如果“核对付款卡”流程中传回了条款与条件 ID，Apple 会下载发卡机构的条款与条件并向用户显示。如果用户接受该条款与条件，Apple 会将所接受条款的 ID 以及 CVV 码发送到“链接和预置”流程。此外，作为“链接和预置”流程的一部分，Apple 会与发卡机构或网络共享设备中的信息。其中包括：(a) 有关用户 iTunes 和 App Store 帐户活动的信息（例如，用户在 iTunes 中是否有长期的交易历史记录），(b) 有关用户设备的信息（例如，电话号码、姓名、用户设备型号以及设置 Apple Pay 所需的任何配对 Apple 设备），以及 (c) 添加付款卡时用户的大致位置（如果用户启用了“定位服务”）。发卡机构会使用此信息，决定是否批准将付款卡添加到 Apple Pay。

“链接和预置”流程完成后，会发生以下操作：

- 设备开始下载代表该信用卡或借记卡的 Apple 钱包凭证文件。
- 设备开始将付款卡与安全元件绑定。

凭证文件包含用来下载付款卡图像的 URL，有关付款卡的元数据，例如联系信息、相关的发卡机构 App 以及支持的功能。它还包括凭证状态：例如安全元件是否完成了个性化设置、付款卡当前是否被发卡机构暂停使用或者在付款卡能够使用 Apple Pay 进行支付前是否需要进行额外验证。

从 iTunes Store 帐户添加信用卡或借记卡

如果使用 iTunes 存档的信用卡或借记卡，用户可能需要重新输入其 Apple ID 密码。随后会从 iTunes 取回卡号，并启动“核对付款卡”流程。如果付款卡符合 Apple Pay 的条件，设备会下载并显示条款与条件，然后将其与条款 ID 和卡片安全码一起发送到“链接和预置”流程。对于存档的 iTunes 帐户付款卡，可能会需要进行额外验证。

从发卡机构的 App 添加信用卡或借记卡

当 App 注册为使用 Apple Pay 时，将为 App 和发卡机构的服务器建立密钥。这些密钥用于加密发送到发卡机构的付款卡信息。此设计旨在阻止信息被 Apple 设备读取。预置流程与前述手动添加付款卡时类似，只有一点不同，即用一次性密码代替 CVV 码。

从发卡机构的网站添加信用卡或借记卡

某些发卡机构提供直接在其网站上发起 Apple 钱包付款卡预置流程的功能。这种情况下，用户可通过在发卡机构的网站上选择要预置的付款卡来发起任务。然后，用户会被定向至 Apple 域内的独立 Apple 登录过程，并被要求使用其 Apple ID 进行登录。成功登录后，用户选取要将付款卡预置到的一台或多台设备，并需要在每台目标设备上分别确认预置结果。

添加额外验证

发卡机构可以决定是否需要对信用卡或借记卡进行额外验证。根据发卡机构提供的功能，用户也许可以选择通过以下方式进行额外验证：短信、电子邮件、客服电话或者通过认证的第三方 App 来完成验证。用户可以选择发卡机构登记的联系信息来获取短信或电子邮件通知，并必须在 Apple 钱包、“设置”或 Apple Watch App 中输入收到的验证码。对于客服通知或使用 App 验证的方式，发卡机构有其自己的通信流程。

使用 Apple Pay 时的支付授权

对于配备安全隔区的设备，仅在收到来自安全隔区的授权后才能进行支付。在 iPhone 或 iPad 上，这涉及到确认用户已使用面容 ID、触控 ID 或设备密码进行认证。面容 ID 或触控 ID 为默认的支付方式（如果可用）；不过用户可随时使用密码来代替。如果尝试匹配指纹三次不成功或匹配面容两次不成功，会自动提供密码输入选项；五次尝试不成功，则需要输入密码。如果面容 ID 或触控 ID 尚未配置或没有为 Apple Pay 启用，用户也需要输入密码。对于在 Apple Watch 上进行的支付，用户必须使用密码解锁设备且必须连按侧边按钮两下。

使用共享的配对密钥

安全隔区和安全元件之间通过串行接口通信：安全元件连接到 NFC 控制器，NFC 控制器连接到应用程序处理器。虽然并非直接相连，但安全隔区和安全元件可以使用共享的配对密钥进行安全通信，该密钥已在设备制造过程中预置。对通信的加密和认证均基于 AES，且两方均使用加密随机数来防止重放攻击。配对密钥会使用安全隔区的 UID 密钥和安全元件唯一标识符在安全隔区内部生成。之后配对密钥会在工厂中从安全隔区安全地传输到硬件安全模块 (HSM) 中，该模块含有随后将配对密钥注入安全元件所需的密钥材料。

授权安全的交易

用户授权交易后（包括直接与安全隔区通信的物理手势），安全隔区之后会向安全元件发送认证类型签名数据以及交易类型的详细信息（免接触式或 App 中），而安全元件中绑定了授权随机 (AR) 值。用户在首次预置信用卡时，安全隔区中会生成 AR 值，只要 Apple Pay 启用，该值便会一直存在，且会受到安全隔区加密和防回滚机制的保护。AR 值会通过使用配对密钥安全地传送到安全元件中。在收到新的 AR 值后，安全元件会将之前添加的付款卡标记为删除。

使用支付密码实现动态安全性

源自支付小程序的支付交易包括支付密码和设备卡号。此密码为一次性代码，使用交易计数器和密钥计算。交易计数器值随每次新交易的产生而递增。密钥则在个性化过程中预置在支付小程序中，且为支付网络和/或发卡机构所知。根据支付方案的不同，计算过程中还可能使用其他数据，包括：

- 终端不可预知数，针对近距离无线通信 (NFC) 交易
- Apple Pay 服务器随机数，针对 App 内交易

这些安全码会被提供给支付网络和发卡机构，使发卡机构可验证每笔交易。根据交易的类型不同，这些安全码的长度也可能有所不同。

通过 Apple Pay 使用付款卡支付

Apple Pay 可用于为商店中、App 内和网站上购买的项目付款。

在商店中使用付款卡支付

如果 iPhone 或 Apple Watch 已开机且检测到了 NFC 场，它会向用户显示请求的付款卡（如果已为此卡打开自动选择）或默认付款卡，付款卡的显示可在“设置”中进行管理。用户还可以前往 Apple 钱包并选取一张付款卡，或在设备锁定时可以：

- 连按两下侧边按钮（在配备了面容 ID 的设备上）
- 连按两下主屏幕按钮（在配备了触控 ID 的设备上）
- 利用允许从锁定屏幕上使用 Apple Pay 的辅助功能

然后在传输信息前，用户必须使用面容 ID、触控 ID 或密码进行认证。Apple Watch 解锁后，连按侧边按钮两下激活默认付款卡进行支付。未经用户认证的情况下不会发送任何支付信息。

用户认证后，在处理支付时会使用设备卡号和交易专用动态安全码。Apple 和用户的设备都不会将信用卡或借记卡的完整卡号发送给商家。Apple 可能会接收诸如交易的大概时间和位置等匿名交易信息，来帮助改进 Apple Pay 和 Apple 的其他产品和服务。

在 App 内使用付款卡支付

Apple Pay 还可用来在 iPhone、iPad、Mac 和 Apple Watch App 内进行支付。当用户在 App 内使用 Apple Pay 支付时，Apple 会收到加密的交易信息。该信息在发送给开发者或商家前，Apple 会使用开发者特定的密钥对交易进行重新加密。Apple Pay 会保留大概的购买金额等匿名交易信息。该信息不会绑定到用户，也绝对不会包括用户购物的内容。

App 发起 Apple Pay 支付交易后，Apple Pay 服务器会先于商家收到来自设备的加密交易。Apple Pay 服务器随后再使用商家特定的密钥对交易进行重新加密，然后将该交易转给商家。

App 请求支付时，会调用 API 来确定设备是否支持 Apple Pay 以及用户所拥有的信用卡或借记卡能否在商家接受的支付网络中进行支付。App 会请求用来处理和实现交易所需的任意信息，例如账单和收货地址以及联系信息。然后，App 会要求 iOS、iPadOS 或 watchOS 显示 Apple Pay 表单，表单会请求 App 的信息以及其他必要信息，例如要使用的付款卡。

此时 App 会收到省市以及邮编信息来计算最终的运费。除非用户使用面容 ID、触控 ID 或设备密码进行授权支付，否则所请求的整套信息绝不会提供给 App。授权支付后，Apple Pay 表单中显示的信息会传输给商家。

App 支付授权

用户授权支付后，会向 Apple Pay 服务器发起调用来获取加密随机数，该随机数与实体店交易中 NFC 终端返回的值类似。接着会将该随机数和其他交易数据一起发送到安全元件以计算使用 Apple 密钥加密的支付凭证。加密支付凭证会返回给 Apple Pay 服务器，服务器会解密该凭证，将凭证中的随机数与 Apple Pay 服务器最初发送的随机数进行核对，然后使用与商家 ID 相关联的商家密钥对支付凭证重新加密。支付随后被返回给设备，由设备通过 API 交还给 App。App 会将凭证传递给商家系统进行处理。商家在随后处理时，会使用自己的私钥对支付凭证进行解密。该凭证和来自 Apple 服务器的签名可让商家验证该交易是否针对此特定商家。

API 会请求授权来指定受支持商家 ID。App 还可以包括用以发送到安全元件进行签名的其他数据（如订单号或客户身份），以确保交易不会转到其他客户手中。App 开发者可以通过在 PKPaymentRequest 上指定 applicationData 来实现。此数据的哈希值会包括在加密的支付数据中。商家负责验证其 applicationData 哈希值与包含在支付数据中的哈希值是否匹配。

在网站中使用付款卡支付

Apple Pay 可用于通过 iPhone、iPad、Apple Watch 和 Mac 电脑（配备触控 ID）在网站上进行支付。Apple Pay 交易还可以在 Mac 上开始，然后在使用同一个 iCloud 帐户且启用了 Apple Pay 的 iPhone 或 Apple Watch 上完成。

在网上使用 Apple Pay 要求所有合作网站在 Apple 注册。域注册后，只有在 Apple 签发了 TLS 客户端证书后才会执行域名验证。支持 Apple Pay 的网站必须通过 HTTPS 提供内容。针对每笔付款交易，网站需要使用 Apple 签发的 TLS 客户端证书来获得与 Apple 服务器之间安全、唯一的商家会话。商家会话数据由 Apple 签名。商家会话签名经过验证后，网站即可查询用户是否拥有支持 Apple Pay 的设备，以及设备上是否激活了信用卡、借记卡或储值卡。但不会共享其他详细信息。用户如果不想共享此信息，可在 iPhone、iPad 和 Mac 设备的 Safari 浏览器隐私设置中停用 Apple Pay 查询。

商家会话通过验证后，所有隐私和安全保护措施与用户在 App 内支付时一样。

如果用户正在将支付相关的信息从 Mac 传输到 iPhone 或 Apple Watch，Apple Pay 接力功能会使用端对端加密的 Apple 身份识别服务 (IDS) 协议在用户的 Mac 与进行授权的设备之间传输支付相关的信息。Mac 上的 IDS 客户端使用用户的设备密钥执行加密，因而其他设备都不能解密此信息，且 Apple 无法获得密钥。发现使用“接力”功能进行 Apple Pay 的设备时会包括用户信用卡的类型和唯一标识符以及一些元数据。用户卡片的设备特定帐号不会共享，且一直安全地储存在用户的 iPhone 或 Apple Watch 上。Apple 还会通过 iCloud 钥匙串安全地传输用户最近使用的联系人、送货和账单地址。

用户使用面容 ID、触控 ID 或密码授权支付，或者在 Apple Watch 上连按侧边按钮两下后，会有一个支付令牌针对每个网站的商家证书进行唯一加密，并安全地从用户的 iPhone 或 Apple Watch 传输到 Mac，然后传送到商家的网站。

只有彼此邻近的设备才能请求和完成支付。邻近度由低功耗蓝牙 (BLE) 广播决定。

Apple Pay 中的免接触式凭证

若要将数据从支持的凭证传输到兼容的 NFC 终端，Apple 会使用 Apple 增值服务 (Apple VAS) 协议。VAS 协议可在免接触式终端上或在 iPhone App 中实施，并使用 NFC 来与支持的 Apple 设备通信。VAS 协议仅在短距离内有效工作，可用于将免接触式凭证独立显示或显示为 Apple Pay 交易的一部分。

设备靠近 NFC 终端时，终端会通过发送凭证请求以开始接收凭证信息。如果用户的凭证具有凭证提供者的标识符，用户需要通过面容 ID、触控 ID 或密码来授权使用该凭证。凭证信息、时间戳和一次性随机 ECDH P-256 密钥将连同凭证提供者的公钥一起用来派生出一个加密密钥，以对发送到终端的凭证数据进行加密。

从 iOS 12.0.1 截止到 iOS 13，用户在将凭证显示给商家的 NFC 终端前，可手动选择凭证。在 iOS 13.1 或更高版本中，凭证提供者可将手动选择的凭证配置为要求用户认证，或无需认证即可使用。

停用 Apple Pay 付款卡

只有安全元件经过了授权（使用与添加付款卡时相同的配对密钥和授权随机 (AR) 值），才能使用添加到安全元件的信用卡、借记卡和储值卡。在收到新的 AR 值后，安全元件会将之前添加的付款卡标记为删除。在以下情形中，这可让操作系统告知安全隔区通过将 AR 副本标记为无效来停用付款卡：

方法	设备
停用密码。	iPhone、iPad、Apple Watch
停用密码。	Mac
用户退出登录 iCloud。	iPhone、iPad、Mac、Apple Watch
用户选择“抹掉所有内容和设置”。	iPhone、iPad、Mac、Apple Watch
设备从恢复模式进行恢复。	iPhone、iPad、Mac、Apple Watch
解除配对。	Apple Watch

暂停使用、移除和抹掉付款卡

通过使用“查找”将设备置于“丢失模式”，用户可以暂停使用 iPhone、iPad 和 Apple Watch 上的 Apple Pay。用户还可以使用“查找”或 iCloud.com 从 Apple Pay 移除和抹掉付款卡，或者直接在设备上使用 Apple 钱包执行此操作。在 Apple Watch 上的付款卡可使用 iCloud 设置以及 iPhone 上的 Apple Watch App 移除，或直接在手表上移除。即使设备离线且未接入蜂窝网络或无线局域网，发卡机构或者各自的支付网络也可暂停使用或移除设备上 Apple Pay 付款卡的支付功能。用户也可以致电发卡机构来暂停使用或移除 Apple Pay 中的付款卡。

当用户使用“抹掉所有内容和设置”、“查找”或恢复设备来抹掉整个设备时，iPhone、iPad、iPod touch、Mac 和 Apple Watch 会指示安全元件将所有付款卡标记为删除。这等同于立即停用付款卡，直到能够联系 Apple Pay 服务器，从安全元件中完全抹掉付款卡。安全隔区还会单独将 AR 标记为无效，因此将无法使用之前注册的付款卡进行进一步的支付授权。设备在线后会尝试联系 Apple Pay 服务器，帮助确保安全元件中的所有付款卡都已被抹掉。

Apple Card 安全性

在支持的 iPhone 和 Mac 机型上，用户可安全申请 Apple Card。

Apple Card 申请

在 iOS 12.4 或更高版本、macOS 10.14.6 或更高版本以及 watchOS 5.3 或更高版本中，Apple Card 可配合 Apple Pay 使用以在商店、App 和网上进行支付。

若要申请 Apple Card，用户必须在兼容 Apple Pay 的 iOS 或 iPadOS 设备上登录其 iCloud 帐户，并在 iCloud 帐户中设置双重认证。申请经批准后，只要用户在任何符合条件的设备上使用其 Apple ID 登录，便可在 Apple 钱包或“设置”>“钱包与 Apple Pay”中使用 Apple Card。

用户申请 Apple Card 时，Apple 的身份提供商合作伙伴会安全地验证用户的身份信息，然后与 Goldman Sachs Bank USA 共享，以用于身份和信用评估之目的。

申请过程中提供的社保号或身份文档图像等信息会安全地传输给 Apple 的身份提供商合作伙伴和/或 Goldman Sachs Bank USA 并使用其各自的密钥加密。Apple 不能解密此数据。

申请过程中提供的收入信息以及用于账单支付的银行账户信息会安全地传输给 Goldman Sachs Bank USA 并使用其密钥加密。银行账户信息会存储在钥匙串中。Apple 不能解密此数据。

将 Apple Card 添加到 Apple 钱包时，用户添加信用卡或借记卡时的相同信息可能会与 Apple 的合作银行 Goldman Sachs Bank USA 以及 Apple Payments Inc. 共享。此类信息仅用于故障诊断、避免欺诈行为与监管目的。

在 iOS 14.6 或更高版本、iPadOS 14.6 或更高版本以及 watchOS 7.5 或更高版本中，拥有 Apple Card 的 iCloud 家庭组织者可以与将其卡片与 13 岁以上的 iCloud 家庭成员共享。确认邀请需要进行用户认证。Apple 钱包使用安全隔区中的密钥来计算绑定持卡人与受邀者的签名。该签名通过 Apple 服务器进行验证。

组织者可以选择对成员设定交易限制，还可以通过 Apple 钱包随时锁定成员卡片以停止其消费。18 岁以上的共同持卡人或成员在接受邀请和申请时，所经历的申请流程与在 Apple 钱包的 Apple Card 申请部分中所定义的相同。

Apple Card 使用

实体卡可从 Apple 钱包的 Apple Card 中订购。用户收到实体卡后即可通过实体卡双折信封中的 NFC 标签激活卡片。每张卡中的标签都是唯一的，无法用于激活另一位用户的卡片。卡片还可在 Apple 钱包设置中手动激活。另外，用户也可以随时从 Apple 钱包选择锁定或解锁实体卡。

Apple Card 还款和 Apple 钱包凭证详细信息

Apple Card 账户到期的还款可从 iOS 中的 Apple 钱包通过 Apple Cash 和银行账户支付。账单还款可设为使用 Apple Cash 和银行账户分期还款，或在特定的日期一次还清。用户还款时，会向 Apple Pay 服务器发起调用以获取类似于 Apple Cash 的加密随机数。接着会将该随机数和还款设置详细信息一起发送到安全元件以计算签名。然后签名会返回给 Apple Pay 服务器。Apple Pay 服务器会通过签名和随机数来验证还款的真实性、完整性和正确性，并且订单会发送给 Goldman Sachs Bank USA 进行处理。

通过出示证书，Apple 钱包会取回 Apple Card 卡号。Apple Pay 服务器验证该证书以确认密钥在安全隔区中生成。然后在将 Apple Card 卡号返回给 Apple 钱包前，它会使用此密钥加密卡号，以使只有请求 Apple Card 卡号的 iPhone 才能进行解密。解密后，Apple Card 卡号会存储在 iCloud 钥匙串中。

在 Apple 钱包的凭证中显示 Apple Card 卡号的详细信息需要用户使用面容 ID、触控 ID 或密码进行认证。用户可以在卡片信息部分中替换该卡号并停用之前的卡号。

先进的防欺诈保护

在 iOS 15 或更高版本和 iPadOS 15 或更高版本中，Apple Card 用户可在 Apple 钱包中启用“先进的防欺诈保护”。启用时，“卡片安全码”会每几天刷新一次。

Apple Cash 安全性

在 iOS 11.2 或更高版本、iPadOS 13.1 或更高版本和 watchOS 4.2 或更高版本中，Apple Pay 可用于在 iPhone、iPad 或 Apple Watch 上与其他用户进行付款、收款和请款。收款后，款项会添加到 Apple Cash 账户。只要用户在任何符合条件的设备上使用其 Apple ID 登录，便可在 Apple 钱包或“设置”>“钱包与 Apple Pay”中使用款项。

在 iOS 14、iPadOS 14 和 watchOS 7 中，使用 Apple Cash 验证了自己身份的 iCloud 家庭组织者可以为 18 岁以下的家庭成员启用 Apple Cash。组织者可以选择限制这些用户只能向家庭成员或联系人付款。如果 18 岁以下的家庭成员执行了 Apple ID 帐户恢复，家庭组织者必须手动为该用户重新启用 Apple Cash 卡片。如果 18 岁以下的家庭成员不再是 iCloud 家庭成员，则其 Apple Cash 余额会自动转到组织者的账户。

用户设置 Apple Cash 时，用户添加信用卡或借记卡时使用的相同信息可能会共享给我们的合作伙伴银行 Green Dot Bank 和 Apple Payments Inc.。Apple Payments Inc. 是由 Apple 创立的全资子公司，它以 Apple 的其他部门无法获知的方式单独储存和处理用户信息，从而保护用户的隐私。此类信息仅用于完成故障诊断、避免欺诈行为以及满足监管目的。

在 iMessage 信息中使用 Apple Cash

若要使用个人对个人支付和 Apple Cash，用户必须在兼容 Apple Cash 的设备上登录其 iCloud 帐户，并在 iCloud 帐户中设置双重认证。用户之间可以使用“信息” App 或让 Siri 来发起请款和转账。用户尝试付款时，iMessage 信息会显示 Apple Pay 表单。Apple Cash 余额始终会优先使用。如有需要，会从用户添加到 Apple 钱包的第二张信用卡或借记卡中提取款项。

在商店、App 和网上使用 Apple Cash

Apple 钱包中的 Apple Cash 卡片可配合 Apple Pay 使用以在商店中、App 内和网上进行支付。Apple Cash 账户中的款项也可以转到银行账户。除了从其他用户处收款，也可以将 Apple 钱包中借记卡或储值卡的款项添加到 Apple Cash 账户。

交易完成后，Apple Payments Inc. 会储存用户的交易数据并可能将其用于完成故障诊断、避免欺诈行为以及满足监管目的。Apple 的其他部门无法获知用户收付款的对象或使用 Apple Cash 卡购物时的位置。

当用户使用 Apple Pay 付款、将款项添加到 Apple Cash 账户或将款项转到银行账户时，会向 Apple Pay 服务器发起调用来获取加密随机数，该随机数与 App 内为 Apple Pay 返回的值类似。接着会将该随机数和其他交易数据一起发送到安全元件以计算支付签名。签名会返回给 Apple Pay 服务器。Apple Pay 服务器会通过支付签名和随机数来验证交易的真实性、完整性和正确性。然后转账开始，交易完成后用户会收到通知。

如果交易涉及：

- 用于向 Apple Cash 充值的借记卡
- Apple Cash 余额不足时进行充值

则还会生成加密支付凭证并发送给 Apple Pay 服务器，这一过程与 Apple Pay 在 App 和网站内的使用方式类似。

Apple Cash 账户的余额超过一定金额后，或检测到异常活动时，用户便会收到要求验证其身份的提示。为验证用户身份而提供的信息，如社保号或问题答案（例如，确认用户之前住过的街道名称），会安全地传输给 Apple 合作伙伴并使用其密钥加密。Apple 不能解密此数据。如果用户执行 Apple ID 帐户恢复操作，用户会收到再次验证其身份的提示，之后才能重新使用 Apple Cash 余额。

Tap to Pay on iPhone 安全性

Tap to Pay on iPhone 可在 iOS 15.4 中使用，让美国商家可使用 iPhone 和支持合作商家的 iOS App 来接受 Apple Pay 和其他免接触式支付。通过这项服务，使用受支持 iPhone 设备的用户可以安全地接受免接触式支付和已启用 NFC 的 Apple Pay 凭证。通过 Tap to Pay on iPhone，商家无需使用额外的硬件即可接受免接触式支付。

Tap to Pay on iPhone 的设计旨在保护付款人的个人信息。此服务不会收集可追溯到付款人的交易信息。信用卡/借记卡卡号 (PAN) 等付款卡信息受安全元件的保护，且商家无法获知。付款卡信息的访问仅限商家的付款服务提供商、付款人和发卡机构。此外，Tap to Pay 服务也不会收集付款人的姓名、地址或电话号码。

Tap to Pay on iPhone 已通过由官方认可的安全实验室进行的外部测试，并获得 American Express、Discover、Mastercard 和 Visa 的认可。

免接触式支付组件安全性

- **安全元件：**安全元件 [Apple Pay 安全元件章节链接] 包含读取免接触式付款卡数据及保护其安全的支付内核。
- **NFC 控制器：**NFC 控制器处理“近距离无线通信”协议，并发送应用程序处理器和安全元件之间以及安全元件和免接触式付款卡之间的通信。
- **Tap to Pay on iPhone 服务器：**Tap to Pay on iPhone 服务器负责管理设备中支付内核的设置和预置。服务器还会监控 Tap to Pay on iPhone 设备的安全性，采取方式与由支付卡产业安全标准委员会 (PCI SSC) 公布的 Contactless Payments on COTS (CPoC) 标准兼容，并且服务器均符合 PCI DSS 的要求。

Tap to Pay 如何读取信用卡、借记卡和储值卡

预置安全性概览

通过充分满足条件的 App 首次使用 Tap to Pay on iPhone 时，Tap to Pay on iPhone 服务器会确定设备的设备型号、iOS 版本等是否达到合格标准，以及密码是否已设置。验证完成后，接受支付的小程序及相关的支付内核配置会从 Tap to Pay on iPhone 服务器下载并安装到安全元件上。此操作在 Tap to Pay on iPhone 服务器和安全元件之间安全地执行。安装前，安全元件会优先验证此数据的完整性和真实性。

卡片读取安全性概览

当 Tap to Pay on iPhone App 向 ProximityReader 框架请求读取卡片时，一个由 iOS 控制的表单会显示并提示用户刷付款卡。iOS 会初始化付款读卡器，然后请求安全元件中的支付内核开始读取卡片。

此时，安全元件在读卡器模式下控制 NFC 控制器。此模式只允许卡片数据通过 NFC 控制器在付款卡和安全元件之间交换。付款卡仅可在此模式下被读取。

安全元件上接受支付的小程序完成卡片读取后，会加密卡片数据并为其签名。卡片数据会维持加密和认证状态，直到其到达付款服务提供商处。只有 App 请求读取卡片时所使用的付款服务提供商才能解密卡片数据。付款服务提供商必须向 Tap to Pay on iPhone 服务器请求卡片数据解密密钥。当数据的完整性和真实性得到验证，且证实是在 60 秒内于设备上读取卡片后，Tap to Pay on iPhone 服务器会向付款服务提供商发出解密密钥。

此模型有助于确保卡片数据无法被除了为商家处理此交易的付款服务提供商之外的任何一方解密。

使用 Apple 钱包

使用 Apple 钱包访问

在支持的 iPhone 和 Apple Watch 设备上的 Apple 钱包中，用户可以储存家、汽车和酒店房间的钥匙。用户甚至还可以储存企业门禁卡和学生证。当用户到达门边时，正确的钥匙会自动显示，用户只需一刷即可使用“近距离无线通信”(NFC) 进门。

用户便捷

将钥匙、凭证、学生证或企业门禁卡添加到 Apple 钱包后，“快捷模式”会默认打开。启用了“快捷模式”的卡片无需使用面容 ID、触控 ID、密码认证或连按两下 Apple Watch 的侧边按钮，即可与接受的终端交互。若要停用此功能，用户可以通过在 Apple 钱包中轻点卡片正面的“更多”按钮来关闭“快捷模式”。若要重新打开“快捷模式”，用户必须使用面容 ID、触控 ID 或密码。

隐私和安全性

Apple 钱包中的钥匙充分利用了内建于 iPhone 和 Apple Watch 中的隐私和安全性功能。用户在 Apple 钱包中使用其钥匙的时间或地点将永不会与 Apple 共享，也不会在 Apple 服务器上储存，并且凭证会安全储存在受支持设备的安全元件 (SE) 内。SE 包含专门设计的小程序以安全地管理和储存访问密钥，确保密钥无法被提取。

预置任何访问密钥前，用户必须在兼容的 iPhone 上登录其 iCloud 帐户且已为该帐户打开双重认证（学生证除外，其无需打开双重认证）。

用户发起预置流程时会执行与预置信用卡和借记卡时类似的步骤，如链接和预置。交易期间，读卡器使用建立的安全通道通过近距离无线通信 (NFC) 控制器与安全元件进行通信。

可预置访问密钥的设备（包括 iPhone 和 Apple Watch）数量由各个合作伙伴规定和控制，从而可能因合作伙伴而异。该方法可让各合作伙伴根据设备类型来控制已预置访问密钥的最大数量，以适应其特定需求。为此，Apple 会向合作伙伴提供设备类型和匿名设备标识符。出于隐私和安全性原因，每个合作伙伴的标识符都不相同。

钥匙可通过以下方式停用或移除：

- 通过“查找”远程抹掉设备
- 通过“查找”启用“丢失模式”
- 接收移动设备管理 (MDM) 远程擦除命令
- 从 Apple ID 帐户页面移除所有卡片
- 从 iCloud.com 移除所有卡片
- 从 Apple 钱包移除所有卡片
- 移除发卡机构 App 中的卡片

在 iOS 15.4 或更高版本中，当用户在配备面容 ID 的 iPhone 上连按两下侧边按钮或在配备触控 ID 的 iPhone 上连按两下主屏幕按钮时，设备在对其进行认证后才会显示其凭证和访问密钥详细信息。凭证的特定信息（包括酒店预订详情）需要在使用面容 ID、触控 ID 或密码认证后才会显示在 Apple 钱包中。

访问凭证类型

Apple 钱包中涵盖了不同的访问凭证类型，例如酒店房间钥匙、企业门禁卡、学生证、家庭钥匙和汽车钥匙。

酒店

Apple 钱包中的酒店房间钥匙有助于实现从入住到退房期间轻松的免接触式体验，同时在隐私和安全性方面为住客提供了优于传统塑料酒店钥匙卡的额外优势。受支持地点的酒店住客在其兼容的 iPhone 和 Apple Watch Series 4 或后续机型上使用 Apple 钱包中的房间钥匙即可一刷开门。

Apple 钱包中的功能专为减少顾客麻烦而设计：

- 到达前从酒店的 App 预置，用于在入住前将凭证添加到 Apple 钱包
- 入住凭证板块，用于从 Apple 钱包办理入住和分房
- 预置后钥匙更新，用于支持延长或修改当前入住时间
- Apple 钱包中支持将多房间钥匙显示为单个凭证
- Apple 钱包中的过期钥匙自动归档

企业门禁卡

受支持合作企业的员工门禁卡可添加到 iPhone 和 Apple Watch 上的 Apple 钱包，让世界各地的员工都可通过免接触方式进出其工作场所。若要添加门禁卡，员工必须已为其用于登录雇主所提供 App 的帐户启用了多重认证。

员工门禁卡充分利用了 Apple 的访问功能，可让用户：

- 通过推送预置的方式将员工门禁卡自动添加到其配对的 Apple Watch，无需安装合作企业的 App
- 使用“快捷模式”无缝进入办公设施
- 即使在 iPhone 电量耗尽后也能进入工作场所

学生证

在 iOS 12 或更高版本中，合作院校中的学生、教师和职工可以将其学生证添加到支持此功能的 iPhone 和 Apple Watch 机型的 Apple 钱包中，以进入特定位置以及在可使用其证件的任何地方进行支付。

用户可通过证件发放机构或合作学校提供的 App 来将学生证添加到 Apple 钱包。此操作中发生的技术过程与从发卡机构的 App 添加信用卡或借记卡中所述的过程相同。此外，发卡机构的 App 必须针对保护其学生证访问权限的帐户提供双重认证支持。一张卡片可在最多两台支持的 Apple 设备（使用相同 Apple ID 登录）上同时设置。

多户住房

受支持合作住所的房客和人员可在 Apple 钱包中使用其家庭钥匙进入其房屋、单元及公共区域。家庭钥匙可从合作伙伴提供的 App 进行预置。对于支持无阻预置的合作伙伴，物业管理人员可使用其首选信息通道（例如电子邮件或短信）向房客发送可发起预置的链接，使房客只需点按链接即可取得钥匙。轻 App 也提供了安全的无缝式体验，使得钥匙的预置无需安装合作伙伴的 App。有关更多信息，请参阅 Apple 支持文章：在 iPhone 上使用轻 App。

家庭钥匙

Apple 钱包中的家庭钥匙可配合启用 NFC 的受支持门锁使用，只需一刷 iPhone 或 Apple Watch 即可。有关用户可如何设置和使用家庭钥匙的更多信息，请参阅 Apple 支持文章：在 iPhone 上使用家庭钥匙开门。

用户设置了家庭钥匙后，其家中的所有住户也会自动收到家庭钥匙。若要进一步共享家庭钥匙或移除共享家庭中的成员，家庭所有者可以使用“家庭” App 管理邀请和成员。当用户选择接受邀请以加入使用家庭钥匙的家庭时，此操作会开始将家庭钥匙预置到其设备上的 Apple 钱包中。如果用户选择离开家庭，或者如果家庭所有者撤销用户访问权限，此类操作也会将家庭钥匙从 Apple 钱包移除。

车钥匙

受支持的 iPhone 设备和配对的 Apple Watch 设备原生支持车钥匙以数字形式储存在 Apple 钱包中。车钥匙在 Apple 钱包中显示为凭证（由 Apple 代表汽车制造商创建），并完全遵循 Apple Pay 卡片生命周期（iCloud 丢失模式、远程擦除、本地凭证删除以及抹掉所有内容和设置）。除了标准的 Apple Pay 卡片管理外，共享的车钥匙还可从所有者的 iPhone、Apple Watch 以及车辆的人机界面 (HMI) 中删除。

车钥匙可用于解锁和锁定车辆，以及用于启动引擎或将车辆设为驾驶模式。“标准事务”提供相互认证且为引擎启动的必须条件。解锁和锁定事务可能会使用“快速事务”（当出于性能原因需要时）。

将 iPhone 与所拥有的支持此功能的车辆相配对即可创建钥匙。所有钥匙都在内嵌安全元件上基于椭圆曲线 (NIST P-256) 片上密钥生成算法 (ECC-OBKG) 创建，私钥绝对不会离开安全元件。设备和车辆之间的通信使用 NFC 或者结合使用低功耗蓝牙和 UWB，密钥管理使用具有相互认证 TLS 的 Apple 到汽车制造商服务器 API。钥匙与 iPhone 配对后，与该 iPhone 配对的任何 Apple Watch 也可以收到钥匙。在车辆中或设备上删除的钥匙无法恢复。丢失或被盗设备上的钥匙可暂停使用和恢复使用，但在新设备上进行重新预置需要重新配对或共享。

iOS 中的车钥匙安全性

开发者可为受支持的 iPhone 和配对的 Apple Watch 提供安全的无实体钥匙方式来使用车辆。

所有者配对

所有者必须先证明车辆所有权（方法具体取决于汽车制造商），才能在汽车制造商 App 中使用汽车制造商发送的电子邮件链接或在车辆菜单中开始配对流程。无论如何，所有者必须向 iPhone 展示一次性保密配对密码，该密码会通过使用 NIST P-256 曲线的 SPAKE2+ 协议生成安全配对通道。使用 App 或电子邮件链接时，密码会自动传输到 iPhone。从车辆发起配对时，必须手动在 iPhone 上输入密码。

钥匙共享

所有者在其配对的 iPhone 上使用 iMessage 信息和 Apple 身份识别服务 (IDS) 向特定设备发送邀请，即可将钥匙共享给符合条件的家庭成员和朋友的 iPhone 设备（及其配对的 Apple Watch 设备）。所有共享命令都将使用端对端加密的 IDS 功能进行交换。在共享过程中，所有者的配对 iPhone 会防止 IDS 通道发生更改，以防止邀请转发。

邀请一经接受，家庭成员或朋友的 iPhone 将创建数字密钥并将密钥创建证书链发送回所有者的配对 iPhone，以验证密钥在可信的 Apple 设备上创建。所有者的配对 iPhone 将对其他家庭成员或朋友的 iPhone 的 ECC 公钥签名，并将签名发送回家庭成员或朋友的 iPhone。所有者设备中的签名操作需要用户认证（面容 ID、触控 ID 或密码输入）和“面容 ID 和触控 ID 的用途”中所述的安全用户意图。发送邀请时需要授权，授权会储存在安全元件中以在朋友设备发回签名请求时使用。密钥授权会通过车辆 OEM 服务器在线提供或在车辆上首次使用共享密钥时提供给车辆。

钥匙删除

钥匙可在所有者设备上从钥匙包设备中删除，以及从车辆中删除。从钥匙包 iPhone 删除会立即生效，即使钥匙包正使用该钥匙。因此删除前会出现强警告。从车辆中删除钥匙的操作可能随时都可执行，也可能仅当车辆在线时才可执行。

从钥匙包设备上或从车辆中删除这两种情况都会报告给汽车制造商的钥匙库存服务器 (KIS)。出于保险目的，车辆的已颁发钥匙都会注册在该服务器中。

所有者可从所有者凭证的背面发起删除请求。请求会先发送至汽车制造商以在车辆中移除钥匙。从车辆中移除钥匙的条件由汽车制造商规定。仅当钥匙从车辆中移除时，汽车制造商服务器才向钥匙包设备发送远程终止请求。

设备中的钥匙终止使用后，管理数字车钥匙的小程序会创建加密签名的终止证明，以用于证明汽车制造商的删除操作和从 KIS 移除钥匙。

NFC 标准事务

对于使用 NFC 钥匙的车辆，读卡器和 iPhone 之间的安全通道通过在读卡器和 iPhone 上生成临时密钥对建立。通过密钥协议方法，共享密钥可在两边派生并用于使用迪菲-赫尔曼的共享对称密钥生成、密钥派生功能和来自配对时建立的长期密钥的签名。

在车辆一方生成的临时公钥使用读卡器的长期私钥签名，因此 iPhone 需要对读卡器进行认证。对于 iPhone，此协议专用于防止隐私敏感数据泄露给拦截通信的攻击者。

最后，iPhone 使用已建立的安全通道加密其公钥标识符以及根据读卡器的数据派生质询计算的签名和部分其他特定于 App 的数据。读卡器对 iPhone 签名的验证使读卡器可认证设备。

快速事务

iPhone 基于之前标准事务中共享的密钥生成密码。此密码可使车辆在需要出色性能时快速认证设备。车辆和设备之间的安全通道也可通过从之前标准事务中共享的密钥和新的临时密钥对中派生会话密钥来建立。车辆建立安全通道的功能使 iPhone 能够认证车辆。

BLE/UWB 标准事务

对于使用 UWB 钥匙的车辆，车辆和 iPhone 之间会建立低功耗蓝牙会话。与 NFC 事务相似，共享密钥在两边派生并用于建立安全会话。此会话用于后续派生和同意 URB 测距密钥 (URSK)。URSK 会提供给用户设备和车辆的 UWB 无线电，以准确定位车辆附近或车辆内部特定位置处的用户设备。然后，车辆会使用设备位置来决定允许解锁还是启动车辆。URSK 具有预定义的 TTL。若要避免在 TTL 过期时测距中断，在安全测距未启用但 BLE 已连接时，URSK 可在设备 SE 和车辆 HSM/SE 中预派生。此操作可避免标准事务在关键时期需要派生新的 URSK。预派生的 URSK 可十分快速地传输到车辆和设备的 UWB 无线电，以避免 UWB 测距中断。

隐私

汽车制造商的钥匙库存服务器 (KIS) 不会储存设备 ID、SEID 或 Apple ID，仅会储存作为可变标识符的实例 CA 标识符。此标识符不会在设备中或通过服务器绑定到任何隐私数据，并会在用户完全擦除其设备（使用“抹掉所有内容和设置”）时被删除。

将交通卡和电子货币卡添加到 Apple 钱包

在全球许多市场中，用户可在支持的 iPhone 和 Apple Watch 机型上将支持的交通卡和电子货币卡添加到 Apple 钱包。根据运营商的不同，添加方式可以是将实体卡中的余额或/和通勤车票转至其数字 Apple 钱包，也可以是从 Apple 钱包或发卡机构的 App 预置新的交通卡或电子货币卡。交通卡添加到 Apple 钱包后，用户只需将其 iPhone 或 Apple Watch 靠近交通卡读卡器，即可乘坐公共交通。部分交通卡还可用于购物付款。

交通卡和电子货币卡的工作方式

添加的交通卡和电子货币卡与用户的 iCloud 帐户相关联。如果用户在 Apple 钱包中添加了多张卡片，Apple 或发卡机构或许可以链接卡片间用户的个人信息和关联的帐户信息。交通卡和电子货币卡及交易受到一组分层加密密钥的保护。

在将实体卡的余额转至 Apple 钱包的过程中，用户需要输入卡片的特定信息，可能还需要提供个人信息来证明卡片为其所有。将车票从 iPhone 传输至 Apple Watch 的过程中，两台设备都必须保持在线。

通过 Apple 钱包或者在交通卡或电子货币卡的发卡机构 App 中，可以使用信用卡、借记卡和储值卡的款项来充值。若要了解使用 Apple Pay 时重新载入余额的安全性，请参阅在 App 内使用付款卡支付。若要了解卡片如何在发卡机构的 App 内预置，请参阅从发卡机构的 App 添加信用卡或借记卡。

如果支持从实体卡进行预置，交通卡或电子货币卡的发卡机构拥有用来认证实体卡和验证用户输入数据所需的加密密钥。数据经验证后，系统可以为安全元件创建设备卡号并在 Apple 钱包中激活包含转账余额的新增车票。对于某些卡片，从实体卡预置完成后，实体卡就会停用。

任何一种预置结束时，只要卡片余额储存在设备上，它就会加密并储存到安全元件中指定的小程序内。运营商拥有密钥，可以对卡片数据执行加密操作以实现余额交易。

默认情况下，交通卡用户可以享受无缝式快捷交通体验带来的好处，无需使用面容 ID、触控 ID 或密码即可进行支付和搭乘公交。对于附近支持“快捷模式”的任何免接触式读卡器，可能会访问诸如近期访问的站点、交易历史及附加票据等信息。用户可以通过停用快捷交通模式来在“钱包与 Apple Pay”设置中打开面容 ID、触控 ID 或密码授权要求。电子货币卡不支持使用“快捷模式”。

和其他 Apple Pay 卡片一样，用户可以通过以下方式来暂停使用或移除电子货币卡：

- 通过“查找”远程抹掉设备
- 通过“查找”启用“丢失模式”
- 输入移动设备管理 (MDM) 远程擦除命令
- 从 Apple ID 帐户页面移除所有卡片
- 从 iCloud.com 移除所有卡片
- 从 Apple 钱包移除所有卡片
- 移除发卡机构 App 中的卡片

Apple Pay 服务器会通知卡片运营商暂停使用或停用这些卡片。如果用户从在线设备移除了交通卡或电子货币卡，可通过将卡片添加回使用相同 Apple ID 登录的设备来恢复余额。如果设备已离线、关机或无法使用，则可能无法恢复余额。

将交通卡和电子货币卡添加到家庭成员的 Apple Watch

在 iOS 15 和 watchOS 8 中，iCloud 家庭组织者可以通过其 iPhone 的 Watch App 将交通卡和电子货币卡添加到家庭成员的 Apple Watch 设备。将其中一张卡片预置到家庭成员的 Apple Watch 时，该手表需要靠近组织者的 iPhone 并使用无线局域网或蓝牙与之连接。家庭成员需要为其 Apple ID 启用双重认证以执行此过程。

家庭成员可以使用 iMessage 信息发送为其 Apple Watch 上交通卡或电子货币卡充值的请求。信息内容受端对端加密保护，如 iMessage 信息安全性概览中所述。可使用无线局域网或蜂窝网络连接来远程为家庭成员 Apple Watch 上的卡片进行充值。无需近距离操作。

【注】此功能可能并非在所有国家或地区都可用。

信用卡和借记卡

在部分城市，交通卡读卡器支持使用 EMV（智能）卡支付交通费用。用户向这些读卡器出示 EMV 信用卡或借记卡时需要进行认证，如“在商店中使用信用卡和借记卡付款”中所述。

在 iOS 12.3 或更高版本中，Apple 钱包中现有的部分 EMV 信用卡/借记卡可启用快捷交通模式。“快捷交通”可让用户在搭乘支持的公交运营商的交通工具时，无需使用面容 ID、触控 ID 或密码即可支付交通费用。当用户预置 EMV 信用卡或借记卡时，第一张预置到 Apple 钱包中的卡片会启用快捷交通模式。若要停用该卡的快捷交通模式，用户可在 Apple 钱包中轻点卡片正面的“更多”按钮，然后将“快捷交通设置”设为“无”。用户还可以使用 Apple 钱包将另一张信用卡或借记卡选为快捷交通卡。重新启用或选择另一张卡启用快捷交通模式需要使用面容 ID、触控 ID 或密码。

Apple Card 和 Apple Cash 也符合快捷交通模式的使用条件。

Apple 钱包中的证件

在运行 iOS 15.4 或更高版本的 iPhone 8 或后续机型以及运行 watchOS 8.4 或更高版本的 Apple Watch Series 4 或后续机型中，用户可将其州证件或驾照添加到 Apple 钱包，然后一刷 iPhone 或 Apple Watch 即可向支持的地点无缝、安全地出示证件。

【注】此功能仅在美国支持的州中可用。

Apple 钱包中的证件使用了内建于用户设备软硬件中的安全性功能，有助于保护用户身份及其个人信息安全。

将驾照或州证件添加到 Apple 钱包

在 iPhone 上，用户只需在 Apple 钱包中轻点屏幕顶部的“添加”(+) 按钮，即可开始添加其驾照或证件。如果用户在设置时配对了 Apple Watch，还会收到将驾照或证件添加到 Apple Watch 上 Apple 钱包的提示。

首先系统会要求用户使用 iPhone 扫描实体驾照或州证件的正面和背面。iPhone 会评估图像的质量和类型，以帮助确保所提供的图像可被州发证机构接受。此类身份证件图像会在设备端加密为州发证机构的密钥，然后发送给州发证机构。

接着，系统会要求用户完成一系列面部及头部动作。用户设备和 Apple 会评估此类动作，以帮助减少他人通过照片、视频或面罩来试图将非本人证件添加到 Apple 钱包的风险。然后，针对此类动作的分析结果会发送至州发证机构，而动作本身的视频不会发送。

为了帮助确保 Apple 钱包中身份证件的添加者与身份证件持有人为同一人，系统会要求用户自拍一张照片。在用户照片提交至州发证机构前，Apple 服务器和用户设备会将照片与面部及头部系列动作执行者的肖像进行对比，并帮助确保已提交照片上的人是真人，且与证件上的人肖像一致。该对比完成后，照片立即在设备端进行加密，并随后发送至州发证机构以根据证件文件上的图像进行对比。

最后，系统会要求用户执行面容 ID 或触控 ID 认证。用户设备将此唯一匹配的面容 ID 或触控 ID 生物识别信息与州证件相绑定，以帮助确保只有将该证件添加到此 iPhone 的人才能出示它；其他注册的生物识别信息则无法用于授权出示该证件。此过程会严格在设备端上执行，且不会发送至州发证机构。

州发证机构将收到用于设置数字身份的必要信息。其中包括用户证件的正反两面图像，从 PDF417 条形码中读取的数据，以及用户在身份验证流程中自拍的照片。发证州还将收到用于防止欺诈行为的一位数值，其基于用户的设备使用模式、设置数据以及个人 Apple ID 的相关信息。最后，证件是否批准添加到 Apple 钱包将取决于发证州的决定。

在州发证机构授权将州证件或驾照添加到 Apple 钱包后，iPhone 会在安全元件中生成密钥对，以将用户证件锚定到该特定设备。如果是添加到 Apple Watch，则 Apple Watch 会在安全元件中生成密钥对。

iPhone 上有证件后，Apple 钱包中的用户证件上所反映的信息会以由安全隔区保护的加密格式进行储存。

使用 Apple 钱包中的驾照或州证件

若要使用 Apple 钱包中的证件，用户需要先使用与 Apple 钱包中该证件关联的面容 ID 或触控 ID 设备进行认证，然后 iPhone 才会将信息出示给证件读卡器。

若要在 Apple Watch 上使用 Apple 钱包中的证件，用户需要在每次佩戴 Apple Watch 时都使用关联的面容 ID 外貌或触控 ID 指纹解锁 iPhone。然后，用户无需认证即可使用 Apple 钱包中的证件，直到其再次摘下 Apple Watch。此功能利用了 watchOS 系统安全性中所详述的“自动解锁”基础功能。

用户在将 iPhone 或 Apple Watch 靠近证件读卡器时，会在设备上看到提示，显示谁正在请求哪些特定信息及其是否打算进行储存。使用关联的面容 ID 或触控 ID 进行授权后，设备会发放所请求的身份信息。

【重要事项】用户无需解锁、展示或传递设备即可出示其证件。

如果用户未启用面容 ID 或触控 ID，而是启用了诸如“语音控制”、“切换控制”或“辅助触控”的辅助功能，则可以使用密码访问并出示其信息。

身份数据到证件读卡器的传输遵循 ISO/IEC 18013-5 标准，其中提供了多种可用的安全性机制，可检测、阻止和减少安全性风险。此类机制涵盖了身份数据完整性和防伪性、设备绑定、知情同意，以及通过无线电线路传输的用户数据保密性。

身份数据完整性和防伪性

Apple 钱包中的证件使用由发证机构提供的签名，使任何符合 ISO/IEC 18013-5 要求的读卡器均可验证 Apple 钱包中的用户证件。此外，“钱包”中证件的所有数据元素均单独受防伪保护。这使证件读卡器可请求 Apple 钱包中证件上存在的一部分特定数据元素，并使 Apple 钱包中的证件可用对应相同部分作出响应，因而只会共享所请求的数据并最大限度地保护用户隐私。

设备绑定

Apple 钱包中的证件认证使用设备签名来防止证件克隆和证件事务重放。通过将证件认证密钥储存在 iPhone 设备的安全元件中，证件会绑定到州发证机构创建证件时所针对的同一设备。

知情同意

针对 Apple 钱包中证件的读卡器认证会使用 ISO/IEC 18013-5 标准中所定义的协议来认证证件读卡器。出示期间，由读卡器证书派生的图标会显示出来，以保证用户正与预期的一方进行交互。

通过无线电线路传输的用户数据保密性

会话加密有助于确保在 Apple 钱包中的证件和证件读卡器之间交换的所有个人可识别信息 (PII) 都已加密。加密过程由应用层执行。因此会话加密的安全性不依赖于传输层（例如 NFC、蓝牙和无线局域网）提供的安全性。

Apple 钱包中的证件有助于保护用户信息隐私

Apple 钱包中的证件遵守 ISO/IEC 18013-5 中所概述的“设备检索”流程。设备检索消除了在出示期间调用服务器的需要，因而保护用户不受 Apple 和发证机构的跟踪。

iMessage 信息

iMessage 信息安全性概览

Apple 推出的 iMessage 信息是一项适用于 iOS 和 iPadOS 设备、Apple Watch 和 Mac 电脑的信息收发服务。iMessage 信息支持文本和附件，例如照片、联系人、位置和链接，以及在信息中直接发送附件，如竖起的大拇指图标。信息会显示在用户所有注册的设备上，这样用户就可以在其他设备上继续对话。iMessage 信息充分利用了 Apple 推送通知服务 (APNs)。Apple 不会记录信息或附件的内容，这些内容受端对端的加密服务保护，因此只有发送者和接收者可以访问它们。Apple 不能解密这些数据。

当用户在设备上打开 iMessage 信息后，设备会生成加密和签名密钥对供这一服务使用。加密使用了用于加密的 RSA 1280 位密钥和 NIST P-256 曲线上用于加密的 EC 256 位密钥。签名使用了椭圆曲线数字签名算法 (ECDSA) 256 位签名密钥。私钥存储在设备的钥匙串中，并且只有在首次解锁后才可用。公钥则与设备的 APNs 地址一起发送至 Apple 身份识别服务 (IDS)，在身份识别服务中，公钥会与用户的电话号码或电子邮件地址关联在一起。

在用户启用其他设备来使用 iMessage 信息时，它们的加密和签名公钥、APNs 地址以及所关联的电话号码都会添加至目录服务中。用户还可以添加更多电子邮件地址，这些电子邮件地址会通过发送确认链接进行验证。电话号码通过运营商网络和 SIM 卡进行验证。对于部分网络，需要使用短信验证（如果短信需要收费，还会向用户显示确认对话框）。除了 iMessage 信息外，一些系统服务（例如 FaceTime 通话和 iCloud）可能需要进行电话号码验证。当有新设备、电话号码或电子邮件地址添加进来时，用户所有已注册的设备都会显示一条警告信息。

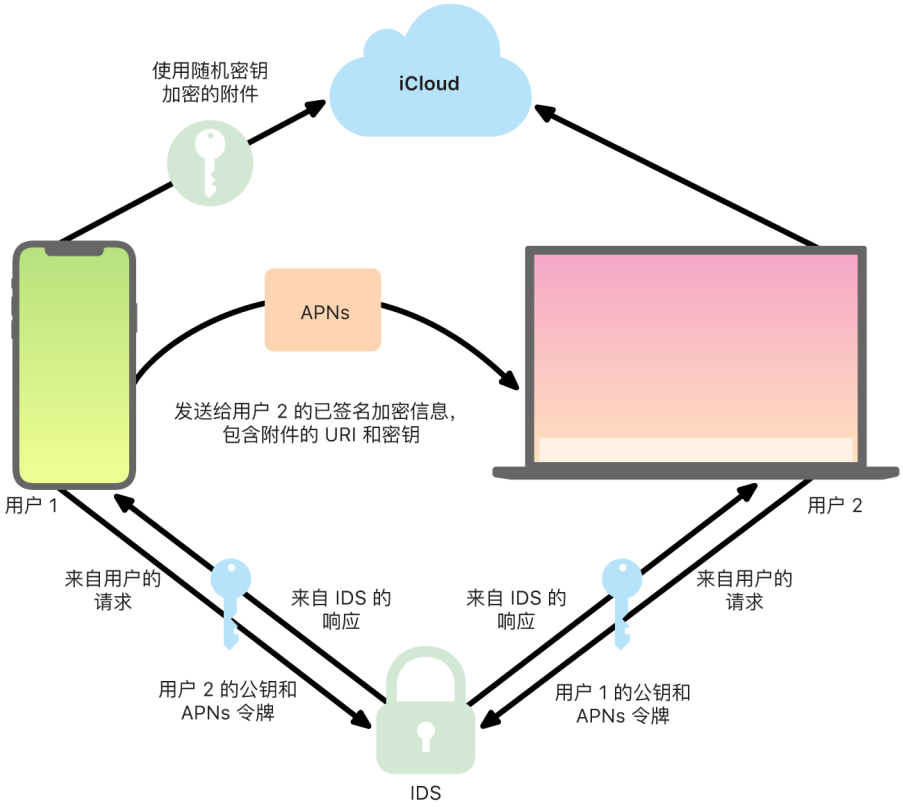
iMessage 信息如何安全发送和接收信息

用户通过输入一个地址或姓名来开始一次新的 iMessage 对话。如果他们输入一个电话号码或电子邮件地址，设备就会与 Apple 身份识别服务 (IDS) 进行联系，来提取与该联系人相关联的所有设备的公钥和 APNs 地址。如果用户输入的是一个名字，设备首先会使用用户的“通讯录” App 来收集与该名字相关联的电话号码和电子邮件地址，然后再从 IDS 中获取公钥和 APNs 地址。

对于每个接收者的设备，用户发出的信息都会单独进行加密。接收设备的公共加密密钥和签名密钥取自 IDS。发送设备针对每个接收设备生成一个 88 位随机值，并将其用作 HMAC-SHA256 密钥，以构成一个派生于发送者和接收者公钥以及明文的 40 位值。88 位值和 40 位值串联构成 128 位密钥，并在计数器 (CTR) 模式下使用 AES 加密连带的信息。接收方使用 40 位值验证解密后明文的完整性。此信息独有的 AES 密钥采用接收设备上用于加密公钥的 RSA-OAEP（算法）进行加密。之后使用 SHA-1 对加密的信息文本和加密的信息密钥进行混编，该哈希值会使用发送设备的专用签名密钥通过椭圆曲线数字签名算法 (ECDSA) 签名。在 iOS 13 或更高版本和 iPadOS 13.1 或更高版本中，设备可能使用椭圆曲线集成加密方案 (ECIES) 加密，而不是 RSA 加密。

针对每部接收设备所生成的每条信息包含加密的信息文本、加密的信息密钥和发送者的数字签名。信息然后会分派至 APNs 以进行发送。时间戳和 APNs 路由信息等元数据则不加密。与 APNs 的通信使用前向保密 TLS 频道加密。

APNs 最多只能转发大小为 4KB 或 16 KB 的信息，具体取决于 iOS 或 iPadOS 的版本。如果信息文本过长，或者附件中有照片等文件，那么附件会使用 AES 在 CTR 模式下通过随机生成的 256 位密钥进行加密并上传至 iCloud。附件的 AES 密钥、其统一资源标识符 (URI) 以及加密形式的 SHA-1 哈希值会作为 iMessage 信息的内容发送给收件人。常规的 iMessage 信息加密会保护以上内容的机密性和完整性，具体如下图所示。



对于群组对话，每一位接收者及其设备之间都会重复此过程。

对于接收方，每台设备接收到的是 APNs 发来的信息的副本，而且如有需要，设备会从 iCloud 提取附件。如果发送人的电话号码或电子邮件地址与接收者的通讯录相匹配，则会在可能的情况下显示一个名字。

与所有推送通知一样，信息在发出之后就会从 APNs 中删除。然而与其他 APNs 通知不同的是，如果设备不在线，iMessage 信息会列入队列等待发送。信息在 Apple 服务器上最长储存 30 天。

iMessage 信息姓名和照片安全共享

iMessage 信息的“姓名和照片共享”允许用户使用 iMessage 信息共享姓名和照片。用户可选择使用其“我的名片”信息，或自定姓名并包括所选的任何图像。iMessage 信息的“姓名和照片共享”使用二级系统来分发姓名和照片。

数据按栏位细分，每一栏按照以下流程分别加密和认证并且统一认证。具体有以下三栏：

- 姓名
- 照片
- 照片文件名

创建数据的先期步骤之一就是在设备上随机生成一个 128 位的记录密钥。此记录密钥随后通过 HKDF-HMAC-SHA256 派生创建三个子密钥，即Key 1:Key 2:Key 3 = HKDF(record key, “nicknames”)。每一栏都会随机生成一个 96 位初始化向量 (IV)，并且数据使用 AES-CTR 和 Key 1 加密。随后会使用 Key 2 并包括栏位名、栏 IV 以及栏位加密文本通过 HMAC-SHA256 来计算信息认证代码 (MAC)。最后，一组单独的栏位 MAC 值会被连在一起，并且其 MAC 使用 Key 3 通过 HMAC-SHA256 计算。256 位 MAC 与加密的数据一起储存。此 MAC 的前 128 位被用作 RecordID。

这个已加密的记录之后会储存在 CloudKit 公用数据库的 RecordID 下方。此记录永不会改变，并且每次当用户选择更改其姓名和照片时，便会生成新的加密记录。当用户 1 选择与用户 2 共享其姓名和照片时，记录密钥以及 recordID 会在加密的 iMessage 信息有效负载中一起发送。

当用户 2 的设备收到此 iMessage 信息有效负载时，它会注意到有效负载中包含“昵称和照片” recordID 及密钥。用户 2 的设备随后前往 CloudKit 公用数据库取回记录 ID 处已加密的姓名和照片并使用 iMessage 信息发送。

取回信息后，用户 2 的设备解密有效负载并使用 recordID 自身验证签名。如果验证通过，则会向用户 2 显示姓名和照片，用户 2 可以选择将此添加到其通讯录，或在“信息”中使用。

安全的 Apple Messages for Business

Apple Messages for Business 是一项信息收发服务，可让用户使用“信息” App 与商家交流。通过 Apple Messages for Business，用户始终可以掌控对话。还可以删除对话并阻止商家在今后向其发送信息。为了保护隐私，商家不会收到用户的电话号码、电子邮件地址或 iCloud 帐户信息。相反，Apple 身份识别服务 (IDS) 会生成一个称为**“无意义 ID”**的自定唯一标识符并与商家共享。“无意义 ID”对于用户的 Apple ID 和商家的商户 ID 之间的关系而言是唯一的。用户在通过 Apple Messages for Business 联系每个商家时会使用不同的“无意义 ID”。是否与商家共享个人可识别信息以及何时共享由用户决定，且 Apple Messages for Business 服务永不会储存对话历史记录。

Apple Messages for Business 支持“Apple 商务管理”中的“管理式 Apple ID”，并在“Apple 商务管理”中确定是否将其用于 iMessage 信息和 FaceTime 通话。

发送给商家的信息会在用户设备和 Apple 信息服务器之间进行加密，使用与 iMessage 信息相同的安全性和 Apple 信息服务器。Apple 信息服务器会在内存中解密这些信息并使用 TLS 1.2 通过加密的链接将信息中继给商家。信息在通过 Apple Messages for Business 服务传输时绝不会以未加密形式储存。商家的回复也使用 TLS 1.2 发送给 Apple 信息服务器，在服务器中它们使用每个接收设备的唯一公钥进行加密。

如果用户设备在线，信息会立即发送到设备而不会缓存到 Apple 信息服务器上。如果用户的设备不在线，加密信息会缓存最多 30 天以便用户在设备重新在线时接收信息。设备重新在线后，信息会立即发送至设备并从缓存中删除。未发送的缓存信息会在 30 天后过期并永久删除。

FaceTime 通话安全性

FaceTime 通话是 Apple 的视频和音频通话服务。与 iMessage 信息类似，FaceTime 通话使用 Apple 推送通知服务 (APNs) 与用户已注册过的设备建立初始连接。FaceTime 通话的音频/视频内容由端对端的加密进行保护，因此只有发送者和接收者可以访问它们。Apple 不能解密这些数据。

初始 FaceTime 通话通过 Apple 服务器基础架构建立连接，服务器在用户所注册的设备间中继数据包。设备在中继连接期间，通过 APNs 通知和 NAT 会话穿越实用程序 (STUN) 信息，验证其身份证书并为每个会话建立共享密钥。共享密钥用于为通过“安全实时传输协议”(SRTP) 进行流化的媒体通道派生会话密钥。SRTP 数据包使用“计数器模式”中的 AES256 加密并使用 HMAC–SHA1 认证。在完成初始连接和安全性设置之后，FaceTime 通话会在可能的情况下，使用 STUN 和互联网连接建立 (ICE) 在设备间建立点对点连接。

FaceTime 群聊可支持多达 33 位参与者同时进行 FaceTime 通话。与传统的一对一 FaceTime 通话一样，通话在受邀参与者的设备间使用端对端加密。FaceTime 群聊通话沿用了一对一 FaceTime 通话的绝大部分基础架构和设计，还加入了一套以 Apple 身份识别服务 (IDS) 提供的真实性为基础的密钥建立机制。此协议提供了正向保密，这意味着用户的设备即便遭到入侵也不会泄露之前的通话内容。会话密钥使用 AES-SIV 封装，然后使用 ECIES 结构与临时的 P-256 ECDH 密钥在参与者之间分发。

新的电话号码或电子邮件地址加入到进行中的 FaceTime 群聊通话后，活跃的设备会建立新的媒介密钥，并且绝对不会与新的受邀设备共享之前使用过的密钥。

查找

“查找”安全性

Apple 设备的“查找” App 植根于高级公钥加密系统。

概览

iOS、iPadOS 和 macOS 中的“查找” App 将“查找我的 iPhone”和“查找我的朋友”整合成单个 App。“查找”可帮助用户定位丢失的设备，即使 Mac 已离线。在线设备只需将其位置通过 iCloud 报告给用户。“查找”离线工作的原理是由丢失的设备发出可由附近使用中的其他 Apple 设备检测到的短距离蓝牙信号。这些附近的设备随后将检测到的丢失设备的位置中继到 iCloud，以便用户可以在“查找” App 中定位它；与此同时保护了所涉及到所有用户的隐私和安全。即使 Mac 已离线且处于睡眠状态，“查找”仍可对其进行定位。

通过蓝牙和世界各地亿万使用中的 iOS、iPadOS 和 macOS 设备，用户可定位其丢失的设备，即使它无法接入无线局域网或蜂窝网络。在“查找”设置中启用了“离线查找”的任何 iOS、iPadOS 或 macOS 设备都可成为“查找设备”。这意味着该设备可通过蓝牙检测到另一台丢失的离线设备的存在，然后使用其网络连接向所有者报告大致的位置。当设备启用了离线查找时，意味着它可由其他参与者以同样的方式进行定位。整个交互过程采用了端对端加密且以匿名方式进行，旨在实现对电池和数据的有效利用。电池续航和蜂窝数据套餐用量受其影响极小，且用户隐私得到更佳保护。

【注】“查找”可能并非在所有国家或地区都可用。

端对端加密

“查找”植根于高级公钥加密系统。在“查找”设置中启用了离线查找后，设备上会直接生成记录为 {d,P} 的椭圆曲线 (EC) P-224 加密私钥对，其中 **d** 为私钥，**P** 为公钥。另外，还有一个 256 位的密钥 SK_0，并且计数器 $_i$ 初始化为零。此私钥对和密钥永不会发送给 Apple，并只会通过 iCloud 钥匙串采用端对端加密的方式在用户的其他设备间同步。密钥和计数器用于通过以下递归构造派生当前的对称密钥 SK_i：$SK_i = KDF(SK_{i-1}, \text{“update”})$。

基于密钥 SK_i，使用 $(u_i,v_i) = KDF(SK_i, \text{“diversify”})$ 计算两个较大的整数 u_i 和 v_i。记作 **d** 的 P-224 私钥和称为 **P** 的对应公钥使用包含两个整数的仿射关系进行派生，以计算出短期有效的密钥对：派生的私钥为 d_i，其中 $d_i = u_i * d + v_i$（以 P-224 曲线的阶为模），对应的公钥为 P_i 并验证 $P_i = u_i*P + v_i*G$。

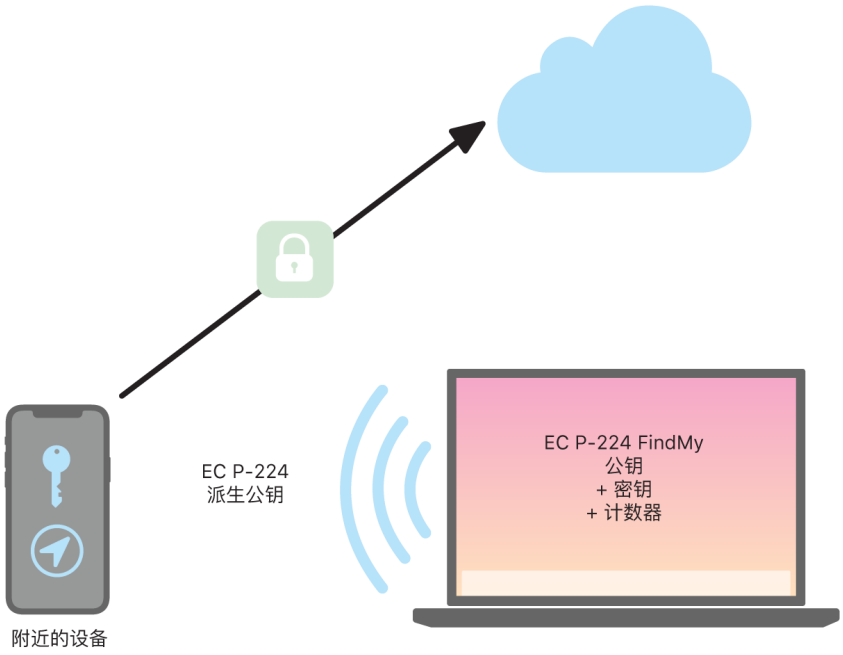
当设备丢失且无法接入无线局域网或蜂窝网络时（例如，MacBook Pro 被忘在公园长椅上），设备会开始在短时间内定期广播蓝牙有效负载中派生的公钥 P_i。通过使用 P-224，公钥代表可匹配单个蓝牙有效负载。然后周围的设备就可在公钥中加密自己的位置来帮助查找离线的设备。大约每隔 15 分钟，就会使用计数器的增量值并按照上述流程生成的新公钥进行替换，这样其他人就无法通过永久标识符跟踪用户。派生机制旨在使各种公钥 P_i 无法链接到同一台设备。

保持用户和设备匿名

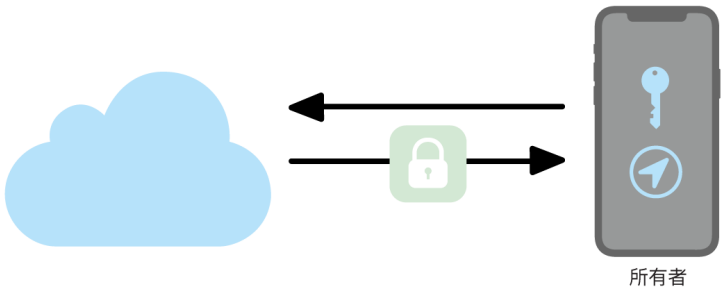
除了确保位置信息和其他数据完全加密外，参与者的身份也对其他人和 Apple 保密。由查找设备发送给 Apple 的流量在内容或标题中都不包含认证信息。因此 Apple 不知道查找设备或寻回设备的所有者身份。而且，Apple 不会记录可泄露查找设备所有者身份的信息，也不会保留允许其他人关联查找设备和设备所有者的信息。设备所有者只会收到加密的位置信息，该信息在“查找” App 中解密和显示且不会指示谁寻回了该设备。

使用“查找”定位丢失的 Apple 设备

任何位于蓝牙覆盖范围内且打开了离线查找的 Apple 设备都可检测来自其他配置为允许“查找”的 Apple 设备的信号并读取当前广播密钥 P_i。查找设备使用 ECIES 结构和广播中的公钥 P_i 加密其当前位置信息并中继给 Apple。加密的位置与服务器索引关联，该索引使用获取自蓝牙有效负载的 P-224 公钥 P_i 计算为 SHA256 哈希值。Apple 从不拥有解密密钥，因此 Apple 无法读取由查找设备加密的位置。丢失设备的所有者可重构索引并解密已加密的位置。



尝试定位丢失的设备时，会为位置搜索期间预估一个计数器值的预期范围。设备所有者知道搜索期间计数器值范围中的原始 P-224 私钥 **d** 和密钥值 SK_i 后，便可重构整个搜索期间的值集 $\{d_i, SHA256(P_i)\}$。然后，用于定位丢失设备的设备所有者可使用索引值集 $SHA256\ (P_i)$ 来查询服务器并从服务器下载已加密的位置。“查找” App 随后使用匹配的私钥 d_i 在本地解密已加密的位置，并在 App 中显示丢失设备的大致位置。多台查找设备报告的位置会由所有者的 App 整合以生成更精确的位置。



定位离线的设备

如果用户在其设备上启用了“查找我的 iPhone”，设备升级到 iOS 13 或更高版本、iPadOS 13.1 或更高版本和 macOS 10.15 或更高版本后，离线查找会默认启用。此设计旨在确保每位用户在设备丢失时都有尽可能高的几率定位到其设备。但是，如果用户不想参与，可随时在其设备上的“查找”设置中停用离线查找。离线查找停用时，设备不可再用作查找设备，也不可被其他查找设备检测到。不过只要设备接入了无线局域网或蜂窝网络，用户仍可进行定位。

定位到丢失的离线设备时，用户会收到告知其设备已找到的通知和电子邮件。若要查看丢失设备的位置，用户可打开“查找” App 并选择“设备”标签页。“查找”会显示带有大致地址的地图位置以及设备多久前被检测到的信息，而不是将设备显示在设备被定位前的空白地图上。如果收到更多位置报告，当前位置和时间戳均会自动更新。用户无法在离线设备上播放声音或远程抹掉，但他们可使用位置信息来追溯走过的路线或采取其他措施来帮助寻回。

连续互通

连续互通安全性概览

连续互通充分利用了诸如 iCloud、蓝牙和无线局域网等技术，让用户在另一台设备上继续进行在前一台设备上进行的活动、拨打和接听电话、发送和接收文本信息以及共享蜂窝互联网连接。

接力安全性

无论是从一台设备传送到另一台设备，还是在原生 App 与网站之间传送，甚至是传送较大数据，Apple 都可以安全地处理“接力”。

“接力”如何安全工作

当用户的 iOS、iPadOS 和 macOS 设备彼此接近时，用户可以使用“接力”功能，自动将正在处理的内容从一台设备传送到另一台设备。用户可以使用“接力”功能来切换设备并立即继续工作。

当用户在第二台支持“接力”功能的设备上登录 iCloud 时，两台设备使用 APNs 建立频段外的低功耗蓝牙 (BLE) 4.2 配对。单条信息采用与 iMessage 信息中的信息类似的方式加密。设备配对后，每台设备都会生成对称的 256 位 AES 密钥，并储存在设备的钥匙串中。此密钥可加密和认证 BLE 广播。BLE 广播会在 GCM 模式下使用 AES256 并采用重放保护措施，将设备的当前活动传递给其他已配对的 iCloud 设备。

设备首次接收到来自新密钥的广播时，它会建立与发起设备之间的 BLE 连接并交换广播加密密钥。该连接使用标准的 BLE 4.2 加密进行保护，而且单个信息也会被加密（与 iMessage 信息的加密方式类似）。在某些情况下，这些信息会使用 APNs 发送，而不是 BLE。活动负载采用与 iMessage 信息相同的方式进行保护和传输。

在原生 App 和网站之间使用“接力”功能

“接力”功能可允许原生 iOS、iPadOS 或 macOS App 恢复由 App 开发者合法控制域中的网页上的用户活动。它还允许原生 App 的用户活动在网页浏览器中继续进行。

为帮助阻止原生 App 要求继续访问不是由其开发者控制的网站，App 必须证明对其要继续访问的网站域具有合法控制权。对网站域的控制是使用共享的网站凭证所使用的机制来建立。有关详细信息，请参阅 App 访问已存储密码的权限。在允许 App 接受使用“接力”功能的用户活动前，系统必须验证 App 的域名控制。

使用“接力”功能传送的网页来源可以是任何采用了“接力” API 的浏览器。当用户浏览网页时，系统会使用加密的“接力”广播字节来广播网页的域名。只有用户的其他设备能够解密该广播字节。

在接收设备上，系统会检测到安装的原生 App 接受了来自已经广播域名的“接力”，并将该原生 App 图标显示为“接力”选项。启动后，原生 App 会接收完整的 URL 以及网页标题。浏览器中的其他信息不会被传送到原生 App。

相反，如果“接力”接收设备未安装相同的原生 App，原生 App 可能会指定回退 URL。如果出现这种情况，系统会将用户的默认浏览器显示为“接力” App 选项（如果该浏览器采用了“接力”API）。请求使用“接力”时，系统会启动浏览器并使用来源 App 提供的回退 URL。回退 URL 并不一定要限制为由原生 App 开发者控制的域名。

使用“接力”传送较大的数据

除了使用“接力”的基本功能外，一些 App 可能会选择使用支持发送大量数据（通过 Apple 开创的点对点无线局域网技术，与“隔空投送”类似）的 API。例如，“邮件” App 使用这些 API 来支持通过“接力”传送可能包含较大附件的邮件草稿。

App 使用这些 API 时，两台设备间开始交换，如同使用“接力”传送一样。但在使用低功耗蓝牙 (BLE) 收到初始负载后，接收设备会通过无线局域网发起新的连接。此连接会使用 TLS 加密，并通过由 iCloud 钥匙串共享的身份派生信任。身份证书中的身份标识会针对每位用户的身份进行验证。其他负载数据会通过此加密的连接进行发送，直到传输完成。

通用剪贴板

“通用剪贴板”利用“接力”在设备间安全传输剪贴板的内容，使用户可以在一台设备上拷贝，然后粘贴到另一台设备上。内容与其他“接力”数据采用同样的方式进行保护，并默认使用“通用剪贴板”共享，除非 App 开发者选择不允许共享。

不论用户是否已将剪贴板粘贴到 App 中，App 都可以访问剪贴板数据。通过“通用剪贴板”，此类数据访问会扩展到用户其他设备上的 App（在 iCloud 登录时建立）。

iPhone 蜂窝网络通话中继安全性

当用户的 Mac、iPad、iPod touch 或 HomePod 与其 iPhone 接入相同的无线局域网时，便可以使用 iPhone 的蜂窝网络连接来拨打和接听电话。这样的配置要求使用相同的 Apple ID 帐户，同时登录 iCloud 和 FaceTime 通话。

来电时，会通过 Apple 推送通知服务 (APNs) 来通知所有已配置的设备，每个通知都会使用与 iMessage 信息相同的端对端加密技术。连接到相同网络的设备上会显示来电通知用户界面。用户接通电话时，会使用安全的点对点连接技术在两台设备间无缝传输用户 iPhone 的音频。

当在一台设备上接通电话时，该设备使用低功耗蓝牙 (BLE) 进行短暂广播，使附近的 iCloud 配对设备停止响铃。广播的字节使用与“接力”广播相同的方法来加密。

去电也会使用 APNs 中继到 iPhone，并通过安全的点对点链接在设备间采用类似的方式传输音频。用户可以在 FaceTime 通话设置中关闭“iPhone 蜂窝网络通话”来停用设备的电话中继功能。

iPhone 短信转发安全性

“短信转发”会自动将 iPhone 上接收的短信发送到用户注册的 iPad、iPod touch 或 Mac 上。每台设备均须使用相同的 Apple ID 帐户登录 iMessage 信息服务。“短信转发”启用后，如果启用了双重认证，用户信任圈内的设备会自动注册。否则，会要求在每台设备上输入由 iPhone 随机生成的 6 位数字代码来验证注册。

设备链接后，iPhone 会使用 iMessage 信息全性概览中所述的方法，将发来的短信进行加密并转发给每台设备。回复也使用相同的方法发送回 iPhone，iPhone 随后使用运营商的短信传输机制将回复以短信形式发送。用户可以在“信息”设置中打开或关闭“短信转发”。

智能热点安全性

智能热点将其他 Apple 设备连接到 iOS 或 iPadOS 个人热点。支持智能热点的 iOS 和 iPadOS 设备使用低功耗蓝牙 (BLE) 来发现设备并与所有设备通信，前提是这些设备分别使用与“家人共享”（在 iOS 13 和 iPadOS 中）中所使用的同一个 iCloud 帐户登录。兼容智能热点且运行 OS X 10.10 或更高版本的 Mac 电脑，使用相同的技术来发现支持智能热点的 iOS 和 iPadOS 设备，并与之通信。

最初当用户进入设备上的无线局域网设置时，设备会发出 BLE 广播，其中包含可被所有登录到相同 iCloud 帐户设备接受的标识符。该标识符由绑定到 iCloud 帐户的 DSID（目的地发讯识别器）生成，并会定期更新。当其他登录到相同 iCloud 帐户的设备彼此接近且支持个人热点时，它们会检测到信号并作出响应，以表示可以使用智能热点。

不属于“家人共享”成员的用户选择 iPhone 或 iPad 使用“个人热点”时，该设备会收到打开“个人热点”的请求。而该请求会通过加密的链接（使用 BLE 加密方法）进行发送；请求的加密方式与 iMessage 信息的加密方式类似。之后，设备会使用包含个人热点连接信息的相同信息独有加密方式，通过同一 BLE 链接作出响应。

对于参与了“家人共享”的用户，个人热点连接信息会使用与 HomeKit 设备用来同步信息的机制类似的机制安全共享。具体来说，在用户间共享热点信息的连接会使用 ECDH (Curve25519) 临时密钥加密，该密钥通过用户各自设备特定的 Ed25519 公钥进行认证。所使用的这些公钥是之前建立“家人共享”时，使用 IDS 在“家人共享”成员间同步的公钥。

网络安全性

网络安全性概览

除了 Apple 用于保护 Apple 设备上所储存数据的内置安全保护，还有许多措施可供组织采用，以借此确保信息在来往于设备时安全无虞。所有这些安全保护和措施皆属于网络安全性范畴。

因为用户必须能够在全球任何地方访问公司网络，所以帮助确保用户得到授权且其数据在传输期间受到保护十分重要。为了实现这些安全目标，iOS、iPadOS 和 macOS 集成了经证实的技术和最新标准来进行无线局域网和蜂窝数据网络连接。这就是为何我们的操作系统使用标准联网协议并使开发者能够访问这些协议，以实现经认证和授权的加密通信。

TLS 安全性

iOS、iPadOS 和 macOS 支持传输层安全协议（TLS 1.0、TLS 1.1、TLS 1.2、TLS 1.3）和数据包传输层安全协议（DTLS）。TLS 协议同时支持 AES-128 和 AES-256 并首选提供正向保密的密码套件。Safari 浏览器、“日历”和“邮件”等互联网 App 会自动使用此协议在设备与网络服务之间建立一条加密的通信通道。上层 API（如 CFNetwork）使开发者可以轻松在其 App 中采用 TLS，而底层 API（如 Network.framework）则提供精细控制。CFNetwork 不接受 SSL 3，且使用 WebKit 的 App（如 Safari 浏览器）禁止建立 SSL 3 连接。

在 iOS 11 或更高版本和 macOS 10.13 或更高版本中，除非用户信任，否则不允许再将 SHA-1 证书用于 TLS 连接。也不允许使用 RSA 密钥短于 2048 位的证书。RC4 对称密码套件在 iOS 10 和 macOS 10.12 中未予使用。默认情况下，通过 SecureTransport API 实现的 TLS 客户端或服务器未启用 RC4 密码套件，并且当 RC4 是唯一可用的密码套件时无法连接。为了更加安全，需要 RC4 的服务或 App 在升级后才能使用更安全的密码套件。在 iOS 12.1 中，2018 年 10 月 15 日之后由系统信任根证书签发的证书必须在可信任证书透明度日志中记录才可进行 TLS 连接。在 iOS 12.2 中，针对 Network.framework 和 NSURLSession API 默认启用 TLS 1.3。使用 SecureTransport API 的 TLS 客户端不能使用 TLS 1.3。

App 传输安全性

App 传输安全性提供默认的连接要求，这样 App 在使用 NSURLConnection、CFURL 或 NSURLSession API 时会遵守安全连接的最佳实践。默认情况下，App 传输安全性将密码选择范围限制为仅包括提供正向保密的密码套件，具体而言：

- 伽罗瓦/计数器模式 (GCM) 下的 ECDHE_ECDSA_AES 和 ECDHE_RSA_AES
- 密码块链接 (CBC) 模式

App 可以按域停用正向保密要求，这种情况下，RSA_AES 会添加到可用密码集中。

服务器必须支持 TLS 1.2 和正向保密，且证书必须有效并使用 SHA256 或更强的加密算法签名，以及包含至少 2048 位 RSA 密钥或 256 位椭圆曲线密钥。

不满足这些要求的网络连接会失败，除非 App 重写了 App 传输安全性。无效的证书始终导致硬故障和无连接。App 传输安全性会自动应用到针对 iOS 9 或更高版本和 macOS 10.11 或更高版本编译的 App。

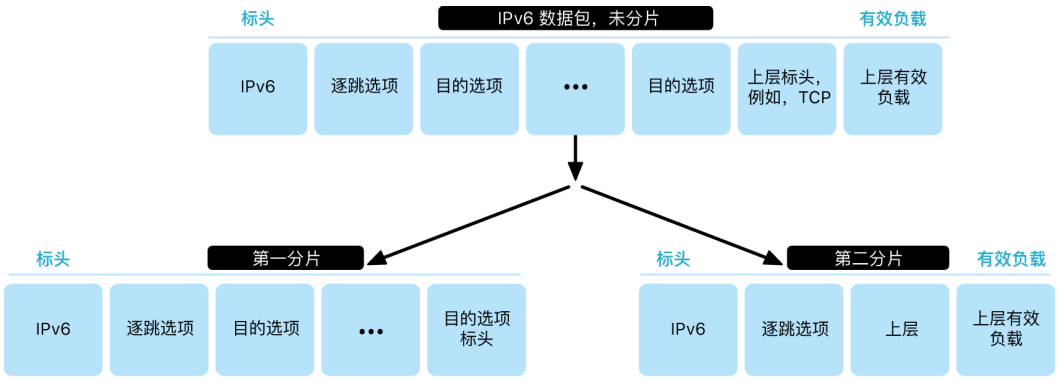
证书有效性验证

对 TLS 证书信任状态的评估依照既定的行业标准（如 RFC 5280 所列）执行，且整合了新出现的标准，如 RFC 6962（证书透明度）。在 iOS 11 或更高版本以及 macOS 10.13 或更高版本中，Apple 设备会随撤销及受限证书的当前列表来进行定期更新。该列表由 Apple 信任的每个内建根证书颁发机构及其从属的 CA 签发者所发布的证书撤销清单 (CRL) 聚合而成，列表还可能根据 Apple 的要求酌情包括其他限制。每当使用网络 API 函数进行安全连接时都会使用此信息。如果需要单独列出的 CA 撤销证书太多，信任评估可能需要在线证书状态响应 (OCSP)，否则将无法通过信任评估。

IPv6 安全性

所有 Apple 操作系统都支持 IPv6，为保护用户的隐私和网络堆栈的稳定性实施了多种机制。使用无状态地址自动配置 (SLAAC) 时，所有接口的 IPv6 地址的生成方式既可以帮助防止跨网络跟踪设备，同时可在网络未发生改变时确保地址的稳定性，从而提供良好的用户体验。自 RFC 3972 起，地址生成算法基于加密生成地址，并通过特定于接口的修饰符进行增强，以保证同一网络上甚至不同的接口最终都具有不同的地址。此外，创建的临时地址的首选生命周期为 24 小时，默认用于任何新连接。为了与 iOS 14、iPadOS 14 和 watchOS 7 中引入的“私有无线局域网地址”功能保持一致，将为设备加入的每个无线局域网生成唯一的本地链接地址。网络的 SSID 将合并为地址生成的附加元素，类似于自 RFC 7217 之后的 Network_ID 参数。iOS 14、iPadOS 14 和 watchOS 7 使用此方法。

为了防止基于 IPv6 扩展标头和分段的攻击，Apple 设备实施了 RFC 6980、RFC 7112 和 RFC 8021 中指定的保护措施。与其他措施相比，这些措施可以阻止仅可在第二个分段中发现上层标头（如下所示）时发生的攻击，而这种攻击可能导致无状态数据包过滤等安全性控制执行混乱。



此外，为帮助确保 Apple 操作系统中 IPv6 堆栈的可靠性，Apple 设备对与 IPv6 相关的数据结构强制实施各种限制，如每个接口的前缀数。

虚拟专用网络 (VPN) 安全性

像虚拟专用网络这样的安全网络服务通常只需简单的设置和配置，便可配合 iOS、iPadOS 和 macOS 设备使用。

支持的协议

若 VPN 服务器支持以下协议和认证方式，便可在这些设备上使用：

- IKEv2/IPsec，此协议通过共享密钥、RSA 证书、椭圆曲线数字签名算法 (ECDSA) 证书、EAP-MSCHAPv2 或 EAP-TLS 认证
- SSL-VPN，此协议使用来自 App Store 的合适的客户端 App
- L2TP/IPsec，此协议通过 MS-CHAPV2 密码进行用户认证，并通过共享密钥（iOS、iPadOS 和 macOS）和 RSA SecurID 或 CRYPTOCard（仅限 macOS）进行机器认证
- Cisco IPsec，此协议通过密码、RSA SecurID 或 CRYPTOCard 进行用户认证，并通过共享密钥和证书（仅限 macOS）进行机器认证

支持的 VPN 部署

iOS、iPadOS 和 macOS 支持以下功能：

- **请求 VPN 域：**针对使用基于证书认证的网络。IT 策略通过使用 VPN 配置描述文件来指定哪些域需要 VPN 连接。
- **为 App 单独设置 VPN：**用于帮助更精确地建立 VPN 连接。移动设备管理 (MDM) 解决方案可为每个被管理的 App 和 Safari 浏览器中特定的域指定连接。这有助于确保进出企业网络的数据始终是安全的，而用户的个人数据不会进出企业网络。

iOS 和 iPadOS 支持以下功能：

- **始终打开 VPN：**适用于由 MDM 解决方案管理、以及通过 Mac 版 Apple Configurator、Apple 校园教务管理或 Apple 商务管理监督的设备。“始终打开 VPN”使得用户在接入蜂窝网络和无线局域网时无需手动打开 VPN 以启用保护。它还通过将所有 IP 流量回传至组织，使组织拥有设备流量的完整控制权。IKEv2 是后续加密交换参数和密钥使用的默认协议，可通过数据加密来保护流量传输的安全。组织可以监控并过滤设备上传入或传出的流量、保护组织网络内的数据安全并限制设备访问互联网。

无线局域网安全性

安全访问无线网络

所有 Apple 平台均支持行业标准的无线局域网认证和加密协议，以在接入以下安全无线网络时提供经认证的访问和保密性：

- WPA2 个人级
- WPA2 企业级
- WPA2/WPA3 过渡模式
- WPA3 个人级
- WPA3 企业级
- WPA3 企业级 192 位安全模式

WPA2 和 WPA3 会认证每个连接，并提供 128 位 AES 加密以帮助确保无线发送的数据的机密性。这为用户提供了最高级别的安全保障，在通过无线局域网连接发送和接收通信时确保用户数据始终受到保护。

WPA3 支持

以下 Apple 设备支持 WPA3：

- iPhone 7 或后续机型
- 第 5 代 iPad 或后续机型
- Apple TV 4K 或后续机型
- Apple Watch Series 3 或后续机型
- Mac 电脑（2013 年末或后续机型，支持 802.11ac 或更高版本）

较新的设备支持 WPA3 企业级 192 位安全模式认证，其中包括在接入兼容的无线接入点 (AP) 时支持 256 位加密。这甚至为无线传输的流量提供了更强大的机密性保护。iPhone 11、iPhone 11 Pro、iPhone 11 Pro Max 及后续 iOS 和 iPadOS 设备支持 WPA3 企业级 192 位安全模式。

PMF 支持

除了保护无线传输的数据，Apple 平台还通过 802.11w 中定义的保护管理框架 (PMF) 服务，将 WPA2 和 WPA3 级别的保护扩展到单播和多播管理框架。以下 Apple 设备支持 PMF：

- iPhone 6 或后续机型
- iPad Air 2 或后续机型
- Apple TV HD 或后续机型
- Apple Watch Series 3 或后续机型
- Mac 电脑（2013 年末或后续机型，支持 802.11ac 或更高版本）

由于支持 802.1X，Apple 设备可集成于各种 RADIUS 认证环境中。支持的 802.1X 无线认证方式包括 EAP-TLS、EAP-TTLS、EAP-FAST、EAP-SIM、PEAPv0 和 PEAPv1。

平台保护措施

Apple 操作系统保护设备免受网络处理器固件漏洞的威胁。这意味着具有无线局域网连接的网络控制器只能有限地访问应用程序处理器内存。

- 使用 USB 或 SDIO（安全数字输入输出）与网络处理器建立接口时，网络处理器无法发起与应用程序处理器间的直接内存访问 (DMA) 事务。
- 使用 PCIe 时，每个网络处理器均位于自己独立的 PCIe 总线上。每个 PCIe 总线上的输入/输出内存管理单元 (IOMMU) 会进一步限制网络处理器的 DMA 对仅包含其网络数据包和控制结构的内存和资源的访问。

已弃用的协议

Apple 产品支持以下已弃用的无线局域网认证和加密协议：

- 开放式 WEP，同时具有 40 位及 104 位密钥
- 共享式 WEP，同时具有 40 位及 104 位密钥
- 动态 WEP
- 暂时密钥集成协议 (TKIP)
- WPA
- WPA/WPA2 过渡模式

这些协议不再被认为是安全的，且这些协议的使用很大程度上受到兼容性、可靠性、性能和安全性因素的制约。对这些协议的支持仅作向后兼容性之用，且可能在未来软件版本中移除。

建议将所有无线局域网转为采用 WPA3 个人级或 WPA3 企业级模式，以尽可能地提供最强大、最安全和最兼容的无线局域网连接。

无线局域网隐私

MAC 地址随机化

Apple 平台会在未与无线局域网关联的情况下执行无线局域网扫描时使用随机介质访问控制地址（MAC 地址）。执行这些扫描可用于发现和连接已知的无线局域网，或为使用地理围栏的 App 提供“定位服务”协助，如基于位置的提醒事项或在 Apple 的“地图”中固定一个位置。请注意，尝试接入首选无线局域网时发生的无线局域网扫描不是随机的。iPhone 5 或后续机型均支持无线局域网 MAC 地址随机化。

在设备未与无线局域网关联或设备处理器处于睡眠状态的情况下进行增强型首选网络卸载 (ePNO) 扫描时，Apple 平台也会使用随机 MAC 地址。设备为使用地理围栏的 App 使用“定位服务”（如基于位置的提醒事项会确定设备是否接近某个特定位置）时，会运行 ePNO 扫描。

由于设备从无线局域网断开连接时其 MAC 地址会更改，因此即使设备接入蜂窝网络，无线局域网流量的被动观察程序也不能使用该地址持续跟踪设备。Apple 已告知无线局域网生产企业 iOS 和 iPadOS 无线局域网扫描会使用随机的 MAC 地址，且 Apple 和生产企业都无法预测这些随机的 MAC 地址。

在 iOS 14 或更高版本、iPadOS 14 或更高版本以及 watchOS 7 或更高版本中，当 iPhone、iPad、iPod touch 或 Apple Watch 接入无线局域网时，其会在每个网络中使用唯一的（随机）MAC 地址标识自己。此功能可由用户或使用无线局域网有效负载中的新选项停用。在某些情况下，设备将回退到实际的 MAC 地址。

有关更多信息，请参阅 Apple 支持文章：在 iPhone、iPad、iPod touch 和 Apple Watch 上使用私有无线局域网地址。

无线局域网帧序号随机化

无线局域网帧包含序号，它被底层 802.11 协议用于启用高效可靠的无线局域网通信。由于这些序号在每个传输帧上递增，因此它们及同一设备上传输的其他帧可用于将无线局域网扫描期间所传输的信息关联起来。

为了保护安全不受此特性影响，当 MAC 地址更改为新的随机地址时，Apple 设备会将序号随机化，其中包括为每次设备未关联时所发起的新扫描请求进行序号随机化。以下设备支持此随机化：

- iPhone 7 或后续机型
- 第 5 代 iPad 或后续机型
- Apple TV 4K 或后续机型
- Apple Watch Series 3 或后续机型
- iMac Pro（视网膜 5K 显示屏，27 英寸，2017 年）或后续机型
- MacBook Pro（13 英寸，2018 年）或后续机型
- MacBook Pro（15 英寸，2018 年）或后续机型
- MacBook Air（视网膜显示屏，13 英寸，2018 年）或后续机型
- Mac mini（2018 年）或后续机型
- iMac（视网膜 4K 显示屏，21.5 英寸，2019 年）或后续机型
- iMac（视网膜 5K 显示屏，27 英寸，2019 年）或后续机型
- Mac Pro（2019 年）或后续机型

无线局域网连接

Apple 为用于“隔空投送”和“隔空播放”的点对点无线局域网连接生成随机 MAC 地址。随机地址还用于 iOS 和 iPadOS（带 SIM 卡）中的个人热点，以及 macOS 中的互联网共享。

每次建立这些网络接口均会生成新的随机地址，且会根据需要为每个接口独立生成唯一的地址。

隐藏网络

无线局域网由其网络名称（即**服务集标识符 (SSID)**）标识。部分无线局域网被配置为隐藏其 SSID，导致无线接入点不广播网络的名称。此类网络被称为**隐藏网络**。iPhone 6s 及后续设备会自动检测隐藏的网络。如果网络已隐藏，则 iOS 或 iPadOS 设备会发送包括探查 SSID 的请求，否则不会发送。这可帮助防止设备广播用户之前接入过的隐藏网络的名称，从而进一步保证了隐私。

蓝牙安全性

Apple 设备中有两种蓝牙，即经典蓝牙和低功耗蓝牙 (BLE)。两种版本的蓝牙安全模型包含以下不同的安全性功能：

- **配对：**创建一个或多个共享密钥的过程
- **绑定：**储存在配对期间创建的密钥以在后续连接中用于形成可信设备对的行为
- **认证：**验证两个设备具有相同的密钥
- **加密：**消息机密性
- **消息完整性：**防止消息伪造的保护措施
- **安全简易配对：**防止被动窃听和中间人攻击的保护措施

蓝牙 4.1 版已将安全连接功能添加到经典蓝牙 (BR/EDR) 物理传输。

各类蓝牙的安全性功能如下。

支持	经典蓝牙	低功耗蓝牙
配对	P-256 椭圆曲线	FIPS 批准的算法（AES-CMAC 和 P-256 椭圆曲线）
绑定	配对信息储存在 iOS、iPadOS、macOS、Apple tvOS 和 watchOS 设备上的安全位置中	配对信息储存在 iOS、iPadOS、macOS、Apple tvOS 和 watchOS 设备上的安全位置中
认证	FIPS 批准的算法（HMAC-SHA256 和 AES-CTR）	FIPS 批准的算法
加密	在“控制器”中执行 AES-CCM 加密	在“控制器”中执行 AES-CCM 加密
消息完整性	针对消息完整性使用 AES-CCM	针对消息完整性使用 AES-CCM
安全简易配对：防止被动窃听的保护措施	临时椭圆曲线迪菲-赫尔曼交换 (ECDHE)	椭圆曲线迪菲-赫尔曼交换 (ECDHE)
安全简易配对：防止中间人 (MITM) 攻击的保护措施	两种用户辅助型数字方法：数字比较 (Numerical Comparison) 和密钥接入 (Passkey Entry)	两种用户辅助型数字方法：数字比较 (Numerical Comparison) 和密钥接入 (Passkey Entry) 配对需要用户的响应，包括所有非 MITM 配对模式
蓝牙 4.1 或更高版本	iMac 2015 年末或后续机型 MacBook Pro 2015 年初或后续机型	iOS 9 或更高版本 iPadOS 13.1 或更高版本 macOS 10.12 或更高版本 Apple tvOS 9 或更高版本 watchOS 2.0 或更高版本
蓝牙 4.2 或更高版本	iPhone 6 或后续机型	iOS 9 或更高版本 iPadOS 13.1 或更高版本 macOS 10.12 或更高版本 Apple tvOS 9 或更高版本 watchOS 2.0 或更高版本

低功耗蓝牙隐私

为帮助保护用户隐私，BLE 包括了两种功能：地址随机化和交叉传输密钥派生。

地址随机化是一种功能，可以通过频繁更改蓝牙设备地址来提高长时间跟踪 BLE 设备的难度。为了让使用隐私功能的设备可与已知设备重新连接，设备地址（也被称为**私有地址**）必须能被其他设备解析。私有地址使用配对过程中所交换的设备身份解析密钥而生成。

iOS 13 或更高版本和 iPadOS 13.1 或更高版本可以跨传输派生链接密钥，这一功能称为**交叉传输密钥派生**。例如，使用 BLE 生成的链接密钥可用于派生经典蓝牙的链接密钥。此外，针对支持《蓝牙核心规范 4.1》（请参阅《蓝牙核心规范 5.1》）所推出的安全连接功能的设备，Apple 在 BLE 中加入了对经典蓝牙的支持。

iOS 中的超宽带安全性

新的 U1 芯片由 Apple 设计，使用提供空间感知的超宽带技术，使 iPhone 11、iPhone 11 Pro 和 iPhone 11 Pro Max 或后续 iPhone 机型可精确定位其他搭载 U1 芯片的 Apple 设备。超宽带技术使用与其他支持的 Apple 设备相同的技术来随机化数据：

- MAC 地址随机化
- 无线局域网帧序号随机化

单点登录

单点登录安全性

单点登录

iOS 和 iPadOS 支持通过单点登录 (SSO) 对企业网络进行认证。SSO 与基于 Kerberos 的网络配合使用，针对用户有权访问的服务对用户进行认证。SSO 可用于各种网络活动，从安全的 Safari 浏览器会话到第三方 App。同时还支持基于证书的认证，如 PKINIT。

macOS 支持使用 Kerberos 对企业网络进行认证。App 可以使用 Kerberos 针对用户有权访问的服务对用户进行认证。Kerberos 还可用于各种网络活动，从安全的 Safari 浏览器会话和网络文件系统认证到第三方 App。还支持基于证书的认证，但需要 App 采用开发者 API。

iOS、iPadOS 和 macOS 的 SSO 使用 SPNEGO 令牌和 HTTP Negotiate 协议，与基于 Kerberos 的认证网关和支持 Kerberos 票证的“Windows 集成身份验证”系统配合使用。SSO 支持基于开源 Heimdal 项目。

iOS、iPadOS 和 macOS 中支持以下加密类型：

- AES-128-CTS-HMAC-SHA1-96
- AES-256-CTS-HMAC-SHA1-96
- DES3-CBC-SHA1
- ARCFOUR-HMAC-MD5

Safari 浏览器支持 SSO，且使用标准 iOS 和 iPadOS 联网 API 的第三方 App 也可进行配置来使用它。为了配置 SSO，iOS 和 iPadOS 支持配置描述文件有效负载，允许移动设备管理 (MDM) 解决方案向下推送必要的设置。其中包括：设置用户主体名称（即 Active Directory 用户帐户）和 Kerberos 领域设置，以及配置应允许哪些 App 和 Safari 浏览器网站 URL 使用 SSO。

若要在 macOS 中配置 Kerberos，请使用“票证显示程序”获取票证，登录 Windows Active Directory 域，或者使用 `kinit` 命令行工具。

可扩展单点登录

App 开发者可使用 SSO 扩展来执行自己的单点登录。当原生 App 或网页 App 需要将某些身份提供者用于用户认证时，会调用 SSO 扩展。开发者可提供两种扩展：重定向至 HTTPS 的扩展和使用质询/响应机制（如 Kerberos）的扩展。这使可扩展单点登录支持 OpenID、OAuth、SAML2 和 Kerberos 认证方案。

若要使用单点登录扩展，App 可以使用 AuthenticationServices API，或者依靠操作系统所提供的 URL 拦截机制。WebKit 和 CFNetwork 为任何原生 App 或 WebKit App 提供一个可无缝支持单点登录的拦截层。若要调用单点登录扩展，则必须通过移动设备管理 (MDM) 描述文件来安装管理员所提供的配置文件。此外，重定向类型的扩展必须使用“关联域”有效负载以证明其所支持的身份识别服务器知道其存在。

Kerberos SSO 扩展是操作系统所提供的唯一扩展。

“隔空投送”安全性

支持“隔空投送”的 Apple 设备使用低功耗蓝牙 (BLE) 和 Apple 创建的点对点无线局域网技术向附近的设备发送文件和信息，包括具有“隔空投送”功能并运行 iOS 7 或更高版本的 iOS 设备和 iPad 设备，以及运行 OS X 10.11 或更高版本的 Mac 电脑。无线局域网信号用于在设备间进行直接通信，无需使用任何互联网连接或无线接入点 (AP)。此连接使用 TLS 加密。

默认情况下，“隔空投送”的共享对象设置为“仅限联系人”。用户还可以选择使用“隔空投送”与所有人进行共享，或者完全关闭这一功能。针对通过移动设备管理 (MDM) 解决方案管理的设备或 App，组织可以限制“隔空投送”的使用。

“隔空投送”操作

“隔空投送”使用 iCloud 服务帮助用户完成认证。当用户登录 iCloud 时，设备上会储存一个 2048 位 RSA 身份标识，且当用户打开“隔空投送”时，设备还会根据与用户 Apple ID 相关联的电子邮件地址和电话号码，创建一个“隔空投送”短身份标识哈希值。

当用户选择使用“隔空投送”共享项目时，发送设备会通过 BLE 发出包括用户的“隔空投送”短身份标识哈希值的“隔空投送”信号。附近处于唤醒状态且打开了“隔空投送”的其他 Apple 设备检测到这一信号，并使用点对点无线局域网进行响应，这样发送设备就可以发现任何响应设备的身份标识。

在“仅限联系人”模式下，接收到的“隔空投送”短身份标识哈希值会与接收设备的“通讯录” App 中联系人的哈希值进行对比。如果找到匹配，接收设备将使用其身份标识信息通过点对点无线局域网进行响应。如果没有匹配，设备将不作响应。

在“所有人”模式中也会采用大致相同的过程。但是，即使设备的“通讯录” App 中没有匹配，接收设备也会作出响应。

发送设备随即使用点对点无线局域网发起“隔空投送”连接，并使用此连接将长身份标识哈希值发送给接收设备。如果长身份标识哈希值与接收方“通讯录”中某个已知联系人的哈希值匹配，则接收方会使用其长身份标识哈希值作出响应。

如果哈希值通过验证，接收方的名字和照片（如果“通讯录”中有）会显示在发送方的“隔空投送”共享表单中。在 iOS 和 iPadOS 中，这些信息均显示在“联系人”或“设备”部分中。未验证或未认证的设备会显示在发送方的“隔空投送”共享表单中，并带有一个剪影图标及设备名称，该名称可在“设置”>“通用”>“关于本机”>“名称”中找到。在 iOS 和 iPadOS 中，设备会放置在“隔空投送”共享表单的“其他人”部分中。

然后发送方用户可选择要与之共享的人。用户选择后，发送设备会与接收设备建立一个加密的 (TLS) 连接，此连接会交换其 iCloud 身份证书。身份证书中的身份标识会针对每位用户的“通讯录” App 进行验证。

如果证书通过验证，则会请求接收方用户接收来自经过验证的用户或设备即将传输的内容。如果选择了多个接收者，将针对每个目标重复此过程。

iPhone 和 iPad 上的无线局域网密码共享安全性

支持无线局域网密码共享的 iOS 和 iPadOS 设备使用与“隔空投送”类似的机制将无线局域网密码从一台设备发送到另一台设备。

当用户选择一个无线局域网（请求方）且系统提示输入无线局域网密码时，Apple 设备发起低功耗蓝牙 (BLE) 广播，表示其想要使用无线局域网密码。附近处于唤醒状态且拥有所选无线局域网密码的其他 Apple 设备，会使用 BLE 连接正在请求的设备。

拥有无线局域网密码的设备（授予方）要求请求方的联系信息，而请求方必须使用与“隔空投送”类似的机制来证明自己的身份。一旦身份获得证明，授予方会将用于加入网络的密码发送给请求方。

针对通过移动设备管理 (MDM) 解决方案管理的设备或 App，组织可以限制无线局域网密码共享的使用。

macOS 中的防火墙安全性

macOS 包括内建防火墙，以保护 Mac 免受网络访问攻击和拒绝服务攻击。防火墙可以在“系统偏好设置”的“安全性与隐私”面板中配置，并支持以下配置：

- 无视 App 而阻止所有传入连接。
- 自动允许内建软件接收传入连接。
- 自动允许已下载的签名软件接收传入连接。
- 基于用户指定的 App 添加或拒绝访问。
- 阻止 Mac 对 ICMP（互联网控制消息协议）探查和端口扫描请求作出响应。

开发者套件安全性

开发者套件安全性概览

Apple 提供了许多“套件”框架，便于第三方开发者扩展 Apple 服务。这些框架以用户隐私和安全为核心而构建：

- HomeKit
- CloudKit
- SiriKit
- DriverKit
- ReplayKit
- ARKit

HomeKit 安全性

HomeKit 通信安全性

HomeKit 奠定了家庭自动化的基础，通过使用 iCloud 以及 iOS、iPadOS 和 macOS 安全性来保护和同步专用数据，而无需将这些数据透露给 Apple。

HomeKit 身份标识和安全性基于 Ed25519 公私密钥对。iOS、iPadOS 和 macOS 设备上会为每位用户针对 HomeKit 生成 Ed25519 密钥对，即用户的 HomeKit 身份标识。该密钥对被用来认证 iOS、iPadOS 和 macOS 设备之间以及 iOS、iPadOS 和 macOS 设备与配件之间的通信。

储存在钥匙串中且仅包括在加密的钥匙串备份中的密钥使用“iCloud 钥匙串”在设备间保持最新（可用时）。HomePod 和 Apple TV 通过轻点设置或如下所述的设置模式来接收密钥。iPhone 通过 Apple 身份识别服务 (IDS) 将密钥共享到配对的 Apple Watch。

HomeKit 配件间的通信

HomeKit 配件会生成自己的 Ed25519 密钥对，用来与 iOS、iPadOS 和 macOS 设备进行通信。如果将配件恢复为出厂设置，则会生成新的密钥对。

为了在 iOS、iPadOS 和 macOS 设备与 HomeKit 配件之间建立联系，会使用安全远程密码（3072 位）协议来交换密钥：用户在 iOS、iPadOS 设备上输入由配件生产企业提供的八位数代码，然后使用 HKDF-SHA512 派生密钥按照 ChaCha20-Poly1305 AEAD 进行加密。在设置过程中，还会对配件的 MFi 认证进行验证。不搭载 MFi 芯片的配件可内建对 iOS 11.3 或更高版本中软件认证的支持。

当 iOS、iPadOS 和 macOS 设备以及 HomeKit 配件在使用过程中进行通信时，它们使用上述过程中交换的密钥相互进行认证。每个会话均使用端对端协议建立，并基于逐个会话 Curve25519 密钥通过 HKDF-SHA512 派生密钥进行加密。这同时适用于基于 IP 的配件和低功耗蓝牙 (BLE) 配件。

对于支持广播通知的 BLE 设备，已配对的 iOS、iPadOS 和 macOS 设备会通过安全会话在配件中预置广播加密密钥。此密钥用于加密有关配件状态改变的数据，通过 BLE 广播通知状态改变。广播加密密钥是 HKDF-SHA512 派生密钥，数据使用 ChaCha20-Poly1305 AEAD 算法来加密。广播加密密钥由 iOS、iPadOS 和 macOS 设备定期更改，并使用 iCloud 更新到其他设备（请参阅 HomeKit 数据安全性）。

HomeKit 与 Siri

Siri 可用来询问和控制配件以及激活各种场景。Siri 只会匿名获得有关家庭配置的极少量信息。所提供有关房间名称、配件和场景的信息为命令识别所需。发送给 Siri 的音频可能会指示具体的配件或命令，但此类 Siri 数据并不会关联到 HomeKit 等 Apple 其他功能。

支持 Siri 的 HomeKit 配件

用户可以使用“家庭” App 在支持 Siri 的配件上启用 Siri 等新功能以及其他 HomePod 功能，如计时器、闹钟、广播和门铃。这些功能启用后，配件会与本地网络上具有这些 Apple 功能的已配对 HomePod 协作。音频使用 HomeKit 和“隔空播放”协议通过加密通道在设备之间交换。

“用‘嘿 Siri’唤醒”打开时，配件使用本地运行的触发短语检测引擎来听取“嘿 Siri”短语。如果此引擎检测到该短语，它会使用 HomeKit 将音频帧直接发送到配对的 HomePod。HomePod 会对音频执行第二次检查，如果短语似乎不包含触发短语，则可能取消音频会话。

“触碰使用 Siri”打开时，用户可以按下配件上的专用按钮以开始与 Siri 对话。音频帧会直接发送到配对的 HomePod。

在检测到成功激活 Siri 后，HomePod 会将音频发送到 Siri 服务器，并使用安全、隐私和加密保护措施（与 HomePod 应用于用户激活 HomePod 本身时一样）来满足用户的意图。如果 Siri 有音频回复，则 Siri 的回复会通过“隔空播放”音频通道发送到配件。部分 Siri 请求需要用户提供更多信息（例如，询问用户是否想要听取更多选项）。在这种情况下，配件会收到应该提示用户的指示，然后更多音频会流播放到 HomePod。

配件需要有视觉指示器，以便在其听取活跃时向用户发出信号（例如，LED 指示灯）。除了访问音频流之外，配件不知道 Siri 请求的意图，也不会储存任何用户数据。

HomeKit 数据安全性

HomeKit 数据可在使用 iCloud 和 iCloud 钥匙串的用户 iOS、iPadOS 和 macOS 设备间安全更新。更新期间，HomeKit 数据使用派生自用户 HomeKit 身份标识的密钥和随机数进行加密，并会作为不透明二进制大对象（即 **blob**）处理。最近处理的二进制大对象会储存在 iCloud 中，但不会用作其他目的。因为 HomeKit 数据加密所使用的密钥仅在用户的 iOS、iPadOS 和 macOS 设备上可用，因此在传输和 iCloud 储存过程中无法对其内容进行访问。

HomeKit 数据还可在同一个家庭的多个用户间同步。这一过程所采用的认证和加密方法与 iOS、iPadOS 和 macOS 设备及 HomeKit 配件之间所使用的相同。当用户加入家庭时，设备间会交换 Ed25519 公钥进行认证。新用户加入家庭后，会使用端对端协议和逐个会话密钥来认证和加密所有进一步的通信。

最初在 HomeKit 中创建家庭的用户或具有编辑权限的其他用户能添加新用户。家庭所有者的设备会使用新用户的公钥来配置配件，这样配件就能认证并接受新用户的命令。当具有编辑权限的用户添加一名新用户时，此过程委托给家居中枢来完成操作。

HomeKit 和 Apple TV

预置 Apple TV 以与 HomeKit 配合使用的过程在用户登录 iCloud 时自动执行。iCloud 帐户需要启用双重认证。Apple TV 与家庭所有者的设备通过 iCloud 交换临时 Ed25519 公钥。当家庭所有者的设备与 Apple TV 处于同一个本地网络时，临时密钥用于通过端对端协议和逐个会话密钥来维护本地网络上连接的安全性。这一过程所采用的认证和加密方法与 iOS、iPadOS 和 macOS 设备及 HomeKit 配件之间所使用的相同。通过这一安全的本地连接，家庭所有者的设备将用户的 Ed25519 公私密钥对传输到 Apple TV。然后，这些密钥会被用来维护 Apple TV 与 HomeKit 配件之间以及 Apple TV 与属于 HomeKit 家庭配置中的其他 iOS、iPadOS 和 macOS 设备之间通信的安全性。

如果用户没有多台设备且未批准其他用户访问其家庭，则 HomeKit 数据不会传输到 iCloud。

家庭数据和 App

用户可通过“隐私”设置来控制 App 对家庭数据的访问。当 App 请求访问家庭数据（与请求访问“通讯录”、“照片”及其他 iOS、iPadOS 和 macOS 数据源类似）时，会要求用户授予访问权限。如果用户批准，App 可以访问房间名称、配件名称、每个配件所处的房间以及在 HomeKit 开发者文稿中详述的其他信息，文稿网址为 https://developer.apple.com/homekit/

本地数据储存

HomeKit 将有关家庭、配件、场景和用户的数据储存在用户的 iOS、iPadOS 和 macOS 设备上。储存的数据会使用派生自用户 HomeKit 身份标识密钥的密钥和随机数进行加密。此外，HomeKit 数据还会使用“首次用户认证前保护”数据保护类进行储存。HomeKit 数据仅备份在加密的备份中，因此诸如通过 USB 备份到“访达”（macOS 10.15 或更高版本）或 iTunes（macOS 10.14 或更低版本）的未加密备份就不包含 HomeKit 数据。

使用 HomeKit 保护路由器

支持 HomeKit 的路由器可让用户管理 HomeKit 配件通过无线局域网访问本地网络和互联网的权限，从而提高其家庭网络的安全性。这些路由器还支持私有 PSK (PPSK) 认证，因此可以使用特定于配件并可根据需要撤销的密钥将配件添加到无线局域网。PPSK 认证不会将主无线局域网密码透露给配件，同时又允许路由器在配件更改其 MAC 地址的情况下安全地识别配件，从而提高了安全性。

用户可以通过“家庭” App 按照以下方法配置配件组的访问限制：

- **无限制：**允许无限制访问互联网和本地网络。
- **自动：**此为默认设置。允许基于配件生产企业提供给 Apple 的互联网站点和本地端口列表访问互联网和本地网络。此列表包括配件所需的所有站点和端口以确保配件正常运行。（在此类列表可用之前，会实施“无限制”。）
- **限制在家中：**不能访问互联网或本地网络，除非 HomeKit 要求进行连接以发现和控制本地网络中的配件（包括家居中枢要求进行的连接以支持远程控制）。

PPSK 是由 HomeKit 自动生成且特定于配件的 WPA2 个人级强密码短语，会在配件之后从“家庭”中移除时被撤销。在通过 HomeKit 路由器配置的家庭中，当 HomeKit 将配件添加到无线局域网时会使用 PPSK；此添加操作在“家庭” App 中的配件设置屏幕上反映为**“无线局域网凭证：HomeKit 管理”**。在添加路由器之前添加到无线局域网的配件将重新配置为使用 PPSK（如果配件支持）；否则将保留其现有的凭证。

为了进一步提高安全性，用户必须使用路由器生产企业开发的 App 来配置 HomeKit 路由器，这样 App 便能验证用户可以访问该路由器且可以将路由器添加到“家庭” App。

HomeKit 摄像头安全性

HomeKit 中具有互联网协议地址 (IP 地址) 的摄像头将视频和音频流直接发送到本地网络中访问该流的 iOS、iPadOS、Apple tvOS 和 macOS 设备。该流使用设备和互联网协议摄像头 (或 IP 摄像头) 上随机生成的密钥进行加密，密钥通过与摄像头之间的安全 HomeKit 会话进行交换。当设备未接入本地网络时，加密的流通过家居中枢中继到设备。家居中枢不会解密该流；它仅充当设备和 IP 摄像头间的中继站。当 App 向用户显示 HomeKit IP 摄像头视频视图时，HomeKit 会通过单独的系统进程安全地渲染视频帧。因此，App 无法访问或储存视频流。此外，App 也不允许在此流中进行截屏。

HomeKit 安防视频

HomeKit 提供了一种安全的端对端保密机制，可录制、分析并查看来自 HomeKit IP 摄像头的片段，而不会将视频内容透露给 Apple 或任何第三方。当 IP 摄像头检测到动作后，会通过家居中枢和 IP 摄像头之间专用的本地网络连接直接将视频片段发送到用作家居中枢的 Apple 设备。本地网络连接由会话独有的 HKDF-SHA512 派生密钥对加密，该密钥对由家居中枢和 IP 摄像头在 HomeKit 会话中进行协商。HomeKit 解密家居中枢上的音频和视频流并在本地分析视频帧以检测任何重要的事件。如果检测到重要的事件，HomeKit 会使用 AES-256-GCM 并通过随机生成的 AES256 密钥来加密该视频片段。HomeKit 还会为每个片段生成海报帧，这些海报帧使用相同的 AES256 密钥加密。加密的海报帧、音频和视频数据会上传到 iCloud 服务器。每个片段相关的元数据 (包括加密密钥) 会使用 iCloud 端对端加密上传到 CloudKit。

对于人脸分类，HomeKit 使用 iCloud 端对端加密储存用于归类 CloudKit 中特定人脸的所有数据。储存的数据包括每个人的信息，如名称和相应人脸的代表性图像。这些人脸图像可源自用户的“照片” (如果用户选择)，或可从之前分析的 IP 摄像头视频中收集而来。HomeKit 安防视频分析会话将此分类数据用于识别安防视频流中直接从 IP 摄像头接收的人脸，并且包括上文提及的片段元数据中的身份信息。

用户使用“家庭” App 查看摄像头中的片段时，这些数据会从 iCloud 下载并使用 iCloud 端对端解密在本地解封用于解密流媒体的密钥。加密的视频内容会从服务器流传输并在 iOS 设备上本地解密，之后才会显示在查看设备上。每个视频片段会话可能拆分为多个子部分，每个子部分使用其自己独有的密钥来加密内容流。

Apple TV 的 HomeKit 安全性

HomeKit 通过安全的方式将部分第三方遥控器配件连接到 Apple TV，还支持将用户描述文件添加到家庭所有者的 Apple TV。

配合 Apple TV 使用第三方遥控器配件

部分第三方遥控器配件会向关联的 Apple TV (通过“家庭” App 添加) 提供人机界面设计 (HID) 事件和 Siri 音频。遥控器通过安全会话将 HID 事件发送到 Apple TV。当用户使用专用的 Siri 按钮明确激活了具有 Siri 功能的电视遥控器上的麦克风时，遥控器会将音频数据发送到 Apple TV。遥控器使用专用的本地网络连接将音频帧直接发送到 Apple TV。会话独有的 HKDF-SHA512 派生密钥对由 Apple TV 和电视遥控器在 HomeKit 会话中进行协商，且电视遥控器用于加密本地网络连接。HomeKit 会解密 Apple TV 上的音频帧，并将它们转发到 Siri App，在此使用与所有 Siri 音频输入相同的隐私保护策略来处理音频帧。

HomeKit 家庭的 Apple TV 描述文件

HomeKit 家庭的某个用户将其描述文件添加到家庭所有者的 Apple TV 时，该用户即可访问其电视节目、音乐和播客。每个用户在 Apple TV 上使用其描述文件的设置会通过 iCloud 端对端加密共享到所有者的 iCloud 帐户。数据为每个用户所有，以只读方式共享给所有者。家庭的每个用户可以在“家庭” App 中更改这些值，所有者的 Apple TV 会使用这些设置。

设置打开后，用户的 iTunes 帐户即在 Apple TV 上可用。设置关闭后，Apple TV 上与该用户关联的所有帐户和数据将被删除。初始 CloudKit 共享由用户的设备发起，建立安全 CloudKit 共享的令牌将通过同样用于在家庭用户之间同步数据的安全通道发送。

iOS、iPadOS 和 watchOS 的 SiriKit 安全性

Siri 使用 App 扩展项系统与第三方 App 通信。设备端的 Siri 可以访问用户的联系信息和设备的当前位置。但是在向 App 提供受保护的数据之前，Siri 会检查该 App 的用户控制的访问权限。根据这些权限，Siri 仅将原始用户话语的相关部分传递给 App 扩展项。例如，假设 App 没有权限访问联系信息，则 Siri 不会在诸如“使用‘<付款 App>’给妈妈付一百块钱”这样的用户请求中解析人物关系。在这种情况下，该 App 仅会看到“妈妈”这个称呼。

但是，如果用户授权 App 访问联系信息，App 会收到用户母亲的相关解析信息。如果信息正文中提到了某个关系，例如“用信息App给妈妈发消息说我哥哥棒极了”，则不论 App 有无权限，Siri 都不会解析“我哥哥”。

启用 SiriKit 的 App 可以将特定于 App 或用户的词条发给 Siri，如用户联系人的名字。此信息与随机标识符相关联，可让 Siri 的语音识别和自然语言理解识别该 App 的词条。只要标识符仍在使用中、在用户于“设置”中停用 App 的 Siri 集成之前或者在启用 SiriKit 的 App 卸载之前，自定信息将保持可用状态。

诸如“使用拼车App叫一辆去妈妈家的车”这种请求，需要获取用户的联系人的位置数据。Siri 会仅针对该请求将需要的信息提供给 App 的扩展项，而不管用户会如何设置该 App 对位置或联系信息的访问权限。

macOS 的 DriverKit 安全性

DriverKit 是一个允许开发者创建用户在其 Mac 上安装的设备驱动程序的框架。使用 DriverKit 构建的驱动程序在用户空间中运行，而不是作为内核扩展运行，以提高系统的安全性和稳定性。这使得安装更加简单，并提高了 macOS 的稳定性和安全性。

用户只需下载 App（使用系统扩展或 DriverKit 时无需安装器），且扩展仅在需要时会被启用。在许多用例中，这些方式取代了 Kext，它需要管理员权限才能安装在“/系统/资源库”或“/资源库”中。

建议使用需要内核扩展的设备驱动程序、云端储存解决方案、网络和安全性 App 的 IT 管理员升级到基于系统扩展而构建的更新版本。这些更新版本极大地降低了 Mac 内核崩溃的可能性，同时也减少了攻击面。这些新扩展在用户空间中运行，安装时无需特殊的权限，并在捆绑的 App 被移到废纸篓时自动移除。

DriverKit 框架为 I/O 服务、设备匹配、内存描述符和调度队列提供 C++ 类。还为数字、集合、字符串和其他常见类型定义适合 I/O 的类型。用户将其与家庭特定的驱动程序框架（如 USBDriverKit 和 HIDDriverKit）配合使用。使用“系统扩展”框架来安装和升级驱动程序。

iOS 和 iPadOS 中的 ReplayKit 安全性

ReplayKit 是一个允许开发者给其 App 添加录音和实时广播功能的框架。另外，它还允许用户给使用设备前置摄像头和麦克风进行录制和广播的内容添加注解。

影片录制

影片录制中构建了多层安全性：

- **许可对话框：**开始录制前，ReplayKit 会显示一则征求用户同意的提醒，请求用户确认其录制屏幕、麦克风和前置摄像头的意图。这则提醒在每个 App 进程中显示一次，并且当 App 处于后台 8 分钟后会再次出现。
- **屏幕和音频捕捉：**屏幕和音频捕捉发生在 ReplayKit 监控程序 replayd 中的 App 进程之外。这旨在确保录制的内容绝对不会被 App 进程访问。
- **App 内屏幕和音频捕捉：**这允许 App 获取视频和样本缓冲，由权限对话框保护。
- **影片创建和储存：**影片文件被写入的目录只能由 ReplayKit 子系统访问，绝对不会被任何 App 访问。这有助于防止第三方未经用户同意而使用录制内容。
- **终端用户预览和共享：**用户可以通过 ReplayKit 提供的用户界面预览和共享影片。此用户界面通过 iOS 扩展项基础架构在进程外呈现，并且能够访问生成的影片文件。

ReplayKit 广播

影片广播中构建了多层安全性：

- **屏幕和音频捕捉：**广播中的屏幕和音频捕捉机制发生在 `replayd` 中，与影片录制完全一样。
- **广播扩展项：**第三方服务若要参与到 ReplayKit 广播中，需要创建两个配置了 com.apple.broadcast-services endpoint 终端的新扩展项：
  - 允许用户设置其广播的用户界面扩展项
  - 用于将视频和音频数据上传到服务的后端服务器的上传扩展项

这个架构帮助确保主 App 对广播的视频和音频内容没有权限。只有 ReplayKit 和第三方广播扩展项能够访问。

- **广播挑选器：**通过广播挑选器，用户可以使用通过“控制中心”即可访问且由系统定义的相同用户界面直接从 App 中启动系统广播。此用户界面使用私有 API 进行实施，是包含在 ReplayKit 框架内部的一个扩展项。它处于主 App 的进程外。
- **上传扩展项：**第三方广播服务用来处理广播期间视频和音频内容的扩展项使用原始未编码样本缓冲。在此处理模式下，视频和音频数据将序列化并通过直接的 XPC 连接实时传递到第三方上传扩展项。视频数据编码方式如下：从视频样本缓冲中提取 IOSurface 对象，将其作为 XPC 对象安全地进行编码，通过 XPC 发送到第三方扩展项，再安全地解码回 IOSurface 对象中。

iOS 和 iPadOS 中的 ARKit 安全性

ARKit 是一个允许开发者在其 App 或游戏中提供增强现实体验的框架。开发者可以使用 iOS 或 iPadOS 设备的前置或后置摄像头添加 2D 或 3D 元素。

Apple 设计相机时考虑了隐私问题，所以第三方 App 必须获得用户的同意才能访问相机。在 iOS 和 iPadOS 中，在用户授权 App 访问其相机后，该 App 可以访问前置和后置摄像头的实时图像。若无明确告知正在使用相机，则 App 无法使用相机。

使用相机拍摄的照片和视频可能包含其他信息，如拍摄地点和时间、景深以及全景拍摄。如果用户不想使用“相机” App 拍摄的照片和视频包含位置信息，随时可以前往“设置”>“隐私”>“定位服务”>“相机”进行控制。如果用户不想在共享照片和视频时包含位置信息，可以在共享表单的“选项”菜单中关闭位置。

使用 ARKit 的 App 可以使用其他相机的世界或面孔追踪信息来更好地定位用户的 AR 体验。世界追踪使用用户设备上的算法处理来自这些传感器的信息，以确定它们相对于物理空间的位置。世界追踪启用了“地图”中诸如“光学朝向”等功能。

安全设备管理

安全设备管理概览

iOS、iPadOS、macOS 和 Apple tvOS 支持一系列灵活的安全性策略和配置，易于强制执行和管理。组织可以通过它们来保护企业信息并帮助确保员工符合企业要求，即便员工使用的是自带设备也无妨，例如，在参与“自带设备办公”(BYOD) 计划的过程中。

组织可以使用密码保护、配置描述文件、远程擦除和第三方移动设备管理 (MDM) 解决方案等资源来管理设备群，并确保公司数据的安全，甚至员工在自己的个人设备上访问这些数据时也能确保安全。

在 iOS 13 或更高版本、iPadOS 13.1 或更高版本和 macOS 10.15 或更高版本中，Apple 设备新增专为 BYOD 计划而设计的用户注册选项。用户注册为用户在其个人设备上提供了更多自主控制权，与此同时，通过在单独且加密保护的 APFS（Apple 文件系统）宗卷上储存企业数据来增强企业数据的安全性。这使得 BYOD 计划在安全性、隐私和用户体验之间取得了更好的平衡。

iPhone 和 iPad 配对模型安全性

iOS 和 iPadOS 使用配对模型从主机电脑控制对设备的访问。配对会在设备及其连接的主机之间通过公钥交换来建立信任关系。iOS 和 iPadOS 还会使用这种信任关系来启用与连接的主机之间的附加功能，例如数据同步。在 iOS 9 或更高版本中：

- 要求配对的服务只有在用户解锁设备后才能开始
- 如果设备最近未解锁，则服务不会开始
- 服务（如照片同步等）可能需要解锁设备才能开始

配对过程要求用户解锁设备并接受来自主机的配对请求。在 iOS 9 或更高版本中，用户还需要输入其密码，然后主机和设备会交换并存储 2048 位 RSA 公钥。随后主机会获得一个 256 位密钥，可解锁储存在设备上的托管密钥包。设备在将受保护的数据发送给主机或启动服务（例如 iTunes 或“访达”同步、文件传输、Xcode 开发等）前，会要求使用交换的密钥来启动加密的 SSL 会话。若要为所有通信使用此加密的会话，设备需要主机通过无线局域网进行连接，因此之前必须通过 USB 进行配对。配对还会启用多项诊断功能。在 iOS 9 中，超过 6 个月未使用的配对记录将会过期。在 iOS 11 或更高版本中，此时间期限缩短为 30 天。

某些诊断服务（包括 com.apple.mobile.pcapd）限制为仅通过 USB 工作。此外，com.apple.file_relay 服务要求安装 Apple 签名的配置描述文件。在 iOS 11 或更高版本中，Apple TV 可以使用“安全远程密码”协议以无线方式建立配对关系。

用户可以使用“还原网络设置”或者“还原位置与隐私”选项清除受信任的主机列表。

移动设备管理

移动设备管理安全性概览

Apple 操作系统支持移动设备管理 (MDM)，可让组织安全地配置和管理规模化的 Apple 设备部署。

MDM 如何安全工作

MDM 功能以现有的操作系统技术为基础，如配置描述文件、无线注册和 Apple 推送通知服务 (APNs)。例如，APNs 用于唤醒设备，使设备可以通过安全的连接与其 MDM 解决方案直接通信。使用 APNs，不会传输任何机密或专有信息。

通过 MDM，IT 部门可在企业环境中注册 Apple 设备、无线配置和更新设置、监控公司政策的遵循情况、管理软件更新策略，甚至可以远程擦除或锁定被管理的设备。

除了 iOS、iPadOS、macOS 和 Apple tvOS 支持的传统设备注册之外，iOS 13 或更高版本、iPadOS 13.1 或更高版本和 macOS 10.15 或更高版本还增加了一个注册类型：用户注册。用户注册是特别针对“自带设备办公” (BYOD) 部署的 MDM 注册，此类设备归个人所有，但在受管理的环境中使用。与未受监督的设备注册相比，用户注册授予 MDM 解决方案更多有限的权限并提供用户和企业数据的独立加密。

注册类型

- **自动设备注册：**自动设备注册允许组织从设备开箱时起就可配置和管理设备（该过程称为**自动前进部署**）。这些设备称为**被监督的设备**，用户可以选择阻止 MDM 描述文件被用户移除。自动设备注册针对组织所有的设备而设计。
- **设备注册：**设备注册允许组织让用户手动注册设备并管理设备使用的各个不同方面，包括抹掉设备的功能。设备注册还具有大量有效负载和访问限制，可应用到设备。用户移除注册描述文件后，基于该注册描述文件的所有配置描述文件、用户设置和被管理的 App 都将随之移除。
- **用户注册：**用户注册专为用户所有的设备而设计且与“管理式 Apple ID”集成，用于在设备上建立用户身份。“管理式 Apple ID”是用户注册描述文件中的一部分，用户必须成功认证才能完成注册。“管理式 Apple ID”可与用户已登录的个人 Apple ID 共同使用。受管理的 App 和帐户使用“管理式 Apple ID”，而个人 App 和帐户使用个人 Apple ID。

设备访问限制

访问限制可由管理员启用，在部分情况下也可停用。它的作用是帮助阻止用户访问 MDM 解决方案中已注册 iPhone、iPad、Mac 或 Apple TV 的特定 App、服务或功能。作为配置描述文件的一部分，访问限制有效负载中的访问限制会被发送到设备。iPhone 上的某些访问限制可以镜像到配对的 Apple Watch 上。

密码设置管理

默认情况下，用户的密码可以设置为一个数字 PIN 码。在配备面容 ID 或触控 ID 的 iOS 和 iPadOS 设备中，密码长度最短为 4 位数。建议使用较长、较复杂的密码，因为这样的密码很难被猜中或遭到攻击。

管理员可以使用 MDM 或 Microsoft Exchange ActiveSync，或者要求用户手动安装配置描述文件，以强制实施复杂密码的要求及其他策略。安装 macOS 密码策略有效负载需要管理员密码。部分密码策略可以要求特定的密码长度、密码组合或其他属性。

配置描述文件执行

配置描述文件是 MDM 解决方案在受管理的设备上提供和管理策略和访问限制的主要方法。如果组织需要配置大量设备或者向大量设备提供许多自定电子邮件设置、网络设置或证书，那么可以通过配置描述文件安全可靠地执行此类操作。

配置描述文件

配置描述文件是一个由有效负载组成的 XML 文件（以 .mobileconfig 结尾），这些有效负载可将设置和授权信息载入到 Apple 设备上。配置描述文件将自动化设置、帐户、访问限制和凭证的配置。此类文件可由 MDM 解决方案或 Mac 版 Apple Configurator 创建，也可通过手动创建。在向 Apple 设备发送配置描述文件前，组织必须使用注册描述文件在 MDM 解决方案中注册该设备。

注册描述文件

注册描述文件是包含 MDM 有效负载的配置描述文件，在特定于设备的 MDM 解决方案中注册该设备。这允许 MDM 解决方案将命令和配置描述文件发送到设备以及查询设备的特定方面。用户移除注册描述文件后，基于该注册描述文件的所有配置描述文件、用户设置和被管理的 App 都将随之移除。一台设备上一次只能有一个注册描述文件。

配置描述文件设置

配置描述文件在特定的有效负载中包含多项可以指定的设置，包括（但不限于）：

- 密码策略
- 针对设备功能的访问限制（例如停用相机）
- 网络和 VPN 设置
- Microsoft Exchange 设置
- 邮件设置
- 帐户设置
- LDAP 目录服务设置
- CalDAV 日历服务设置
- 凭证和密钥
- 软件更新

描述文件签名和加密

可以对配置描述文件进行签名来验证其来源，以及对其进行加密来帮助确保其完整性并保护其内容。iOS 和 iPadOS 的配置描述文件使用 RFC 5652 中规定的“密码讯息语法”(CMS) 进行加密，CMS 支持 3DES 和 AES128。

描述文件安装

用户可以使用 Mac 版 Apple Configurator 直接在其设备上安装配置描述文件，或者可以使用 Safari 浏览器下载配置描述文件、以邮件附件发送、在 iOS 和 iPadOS 中使用隔空投送或“文件” App 传输或使用移动设备管理 (MDM) 解决方案以无线方式发送。用户在“Apple 校园教务管理”或“Apple 商务管理”中设置设备时，设备会下载并安装用于 MDM 注册的描述文件。有关如何移除描述文件的信息，请参阅《Apple 平台部署》中的移动设备管理介绍。

【注】在受监督的设备上，还可以将配置描述文件锁定到设备。这旨在防止其移除，或者可以只允许使用密码将其移除。由于许多组织使用的是自己的 iOS 和 iPadOS 设备，因此将设备绑定到 MDM 解决方案的配置描述文件可以移除，但这样做也会移除所有被管理的配置信息、数据和 App。

自动设备注册

在用户获得设备前，组织无需实际操作或者准备设备，即可在移动设备管理 (MDM) 中自动注册 iOS、iPadOS、macOS 和 Apple tvOS 设备。注册其中一个服务后，管理员会登录到该服务网站，并将该计划关联到 MDM 解决方案。然后他们购买的设备就可以通过 MDM 分配给用户了。设备配置过程中，确保实施适当的安全措施可提高敏感数据的安全性。例如：

- 激活期间，在 Apple 设备“设置助理”的初始化设置流程中加入用户认证环节。
- 提供包含有限访问权限的初始配置，以及需要更多设备配置才能访问敏感数据。

分配用户后，所有 MDM 指定的配置、访问限制或控制都会自动安装。设备与 Apple 服务器之间的所有通信在通过 HTTPS (TLS) 传输时均经过加密。

通过在设备的“设置助理”中移除特定的步骤，可进一步简化用户的设置过程，方便用户快速使用。管理员还可以控制用户能否将 MDM 描述文件从设备移除，并帮助确保设备访问限制在设备的整个生命周期里准备到位。设备开箱并激活后，会在组织的 MDM 解决方案中注册，且所有管理设置、App 和图书均会按 MDM 管理员指定的方式安装。

Apple 校园教务管理、Apple 商务管理和 Apple 商务必备

“Apple 校园教务管理”、“Apple 商务管理”和“Apple 商务必备”服务可让 IT 管理员部署由组织直接从 Apple 或合作的 Apple 授权经销商和运营商处购买的 Apple 设备。

结合 MDM 解决方案使用时，管理员可以简化用户的设置过程，配置设备设置以及通过这三种服务分发购买的 App 和图书。“Apple 校园教务管理”还直接或通过 SFTP 集成了学生信息系统 (SIS)，并且这三种服务均能够将跨域身份管理系统 (SCIM) 或联合认证与 Microsoft Azure Active Directory (Azure AD) 配合使用，以便管理员可以快速创建帐户。

Apple 维护符合 ISO/IEC 27001 和 27018 标准的认证，以协助 Apple 客户履行其监管和合同义务。这些认证针对特定范围内的系统向客户提供了 Apple 信息隐私和安全实践的独立保障。有关更多信息，请参阅《Apple 平台认证》中的 Apple 互联网服务安全认证。

【注】若要了解 Apple 计划在某个国家或地区是否可用，请参阅 Apple 支持文章：面向教育机构和企业的 Apple 计划和付款方式的可用情况。

设备监督

监督通常意味着设备归组织所有，使组织拥有对设备配置和使用限制的额外控制。有关更多信息，请参阅《Apple 平台部署》中的关于 Apple 设备监督。

激活锁安全性

Apple 实施激活锁的方式各有不同，具体取决于设备是 iPhone 或 iPad、搭载 Apple 芯片的 Mac 还是基于 Intel 且搭载 Apple T2 安全芯片的 Mac。

iPhone 和 iPad 上的表现方式

在 iPhone 和 iPad 设备上，激活锁通过 iOS 和 iPadOS “设置助理”中无线局域网选择屏幕之后的激活过程来实施。当设备表示正在进行激活时，设备会向 Apple 服务器发送请求来获取激活证书。启用了激活锁的设备会提示用户输入启用激活锁时所使用的 iCloud 凭证。iOS 和 iPadOS “设置助理”只有在获得有效证书后才会继续。

搭载 Apple 芯片的 Mac 上的表现方式

在搭载 Apple 芯片的 Mac 中，LLB 会验证设备是否存在有效的 LocalPolicy 且 LocalPolicy 策略随机数值是否与安全储存组件中所储存的值匹配。如果出现以下情况，LLB 会启动进入 recoveryOS：

- 当前 macOS 不存在任何 LocalPolicy
- LocalPolicy 对该 macOS 无效
- LocalPolicy 随机哈希值与储存在安全储存组件中的哈希值不匹配

recoveryOS 检测到 Mac 电脑未激活后会联系激活服务器以获取激活证书。如果设备启用了激活锁，recoveryOS 会提示用户输入启用激活锁时所使用的 iCloud 凭证。在获得有效的激活证书后，该激活证书密钥会用于获得 RemotePolicy 证书。Mac 电脑会使用 LocalPolicy 密钥和 RemotePolicy 证书来生成有效的 LocalPolicy。LLB 只有在存在有效 LocalPolicy 的情况下，才会允许启动 macOS。

基于 Intel 的 Mac 电脑上的表现方式

在基于 Intel 且搭载 T2 芯片的 Mac 中，T2 芯片固件会先确认存在有效的激活证书，然后才会允许电脑启动进入 macOS。T2 芯片所载入的 UEFI 固件负责从 T2 芯片查询设备的激活状态，且如果不存在有效的激活证书，则会启动进入 recoveryOS 而非 macOS。recoveryOS 检测到 Mac 未激活后会联系激活服务器以获取激活证书。如果设备启用了激活锁，recoveryOS 会提示用户输入启用激活锁时所使用的 iCloud 凭证。UEFI 固件只有在有效激活证书存在的情况下，才会允许启动 macOS。

受管理的丢失模式和远程擦除

“受管理的丢失模式”用于定位被盗的受监督设备。定位后，可远程锁定或擦除这些设备。

受管理的丢失模式

如果运行 iOS 9 或更高版本的受监督 iOS 或 iPadOS 设备丢失或被盗，移动设备管理 (MDM) 管理员可以在该设备上远程启用“丢失模式”（名为“受管理的丢失模式”）。启用“受管理的丢失模式”后，当前用户将退出登录，且设备不能被解锁。屏幕上显示一则可由管理员自定的信息，例如显示一个电话号码（以便在他人发现设备时拨打）。管理员还可以请求设备发送其当前位置（即使“定位服务”已关闭），也可以选择播放声音。当管理员关闭“受管理的丢失模式”（这是退出该模式的唯一方式）时，用户将会收到此操作通知，通知方式是在锁定屏幕上显示信息或在主屏幕上显示提醒。

远程擦除

管理员或用户可以远程抹掉 iOS、iPadOS 和 macOS 设备（即时远程擦除仅当 Mac 已启用“文件保险箱”时可用）。通过安全地丢弃可擦除存储器中的媒介密钥，使所有数据均不可读，以实现即时远程擦除。如果通过 Microsoft Exchange ActiveSync 进行远程擦除，在执行擦除前，设备会签入 Microsoft Exchange 服务器。

通过 MDM 或 iCloud 触发远程擦除命令后，iPhone、iPad、iPod touch 或 Mac 设备会向 MDM 解决方案发送确认并执行擦除操作。

远程擦除在以下情况中不可用：

- 使用“用户注册”
- 帐户通过“用户注册”安装时，使用 Microsoft Exchange ActiveSync
- 设备受监督时使用 Microsoft Exchange ActiveSync

用户还可以使用“设置” App 来擦除自己的 iOS 和 iPadOS 设备。如上所述，可以将 iOS 和 iPadOS 设备设为在连续多次输入密码失败后自动擦除。

iPadOS 中的“共享 iPad”安全性

“共享 iPad”是一种用于 iPad 部署的多用户模式。该模式允许用户共用一部 iPad，同时单独存储每位用户的文稿和数据。每位用户都会获得自己专有的保留储存位置，该位置以 APFS（Apple 文件系统）宗卷的形式实施并受到用户凭证的保护。“共享 iPad”要求使用组织签发和拥有的“管理式 Apple ID”。

通过“共享 iPad”，用户可以登录任何为组织所有且配置为供多用户使用的设备。用户数据被划分到独立的目录中，每个目录都位于自己的数据保护域中且都受到 UNIX 权限和沙盒保护。在 iPadOS 13.4 或更高版本中，用户还可以登录临时会话。用户退出登录临时会话后，系统会删除其 APFS 宗卷并将保留的空间返还给系统。

登录“共享 iPad”

登录“共享 iPad”时同时支持使用原生及联合“管理式 Apple ID”。首次使用联合帐户时，用户会被重定向至身份提供商 (IdP) 的登录门户。认证后，支持的“管理式 Apple ID”会获得一个短时访问令牌，其登录流程与原生“管理式 Apple ID”的登录流程相似。登录后，“共享 iPad”上的“设置助理”会提示用户建立密码（凭证），用于保护设备本地数据的安全以及后续登录屏幕的认证。手持单用户设备的用户可以使用其联合帐户登录一次“管理式 Apple ID”，然后通过密码解锁其设备。与上述流程类似，用户在“共享 iPad”上使用其联合帐户登录一次，此后便可使用其建立的密码。

用户不通过联合认证进行登录时，Apple 身份识别服务 (IDS) 会使用 SRP 协议对“管理式 Apple ID”进行认证。如果认证成功，则会授予一个特定于该设备的短时访问令牌。如果用户此前已经使用过该设备，则他们已有一个使用相同凭证解锁的本地用户帐户。

如果用户此前没有使用过该设备或使用的是临时会话功能，则“共享 iPad”会预置新的 UNIX 用户 ID、用于储存用户个人数据的 APFS 宗卷和本地钥匙串。因为在创建 APFS 宗卷时已经为用户分配（保留）了储存空间，可能没有足够的空间用于创建新的宗卷。在这种情况下，系统会识别数据已经完成同步到云端的现有用户并将该用户从设备中登出，从而允许新用户登录。如果所有现有用户的云端数据上传都未完成，则新用户无法登录，这种情况极少发生。若要登录，新用户需要等待一位用户完成其数据同步，或者让管理员强制删除一个现有的用户帐户，但会存在数据丢失的风险。

如果设备未接入互联网（例如，如果用户没有无线局域网接入点），则仅在有限的天数内可以认证本地帐户。在那种情况下，只有之前拥有现有本地帐户或使用临时会话功能的用户才能登录。时限过期后，即使本地帐户已经存在，用户仍需要在线进行认证。

解锁或创建用户的本地帐户后，若以远程方式认证，由 Apple 服务器签发的短时令牌便会转换为允许登录 iCloud 的 iCloud 令牌。接着就会恢复用户的设置，并从 iCloud 同步其文稿和数据。

当用户会话处于活跃状态且设备保持在线时，文稿和数据将在创建或修改时储存到 iCloud。另外，后台同步机制帮助确保在用户退出登录后将更改推送到 iCloud 或其他使用 NSURLSession 后台会话的网络服务。该用户的后台同步完成后就会卸载用户的 APFS 宗卷，如果用户不重新进行登录便无法再次装载。

临时会话不会与 iCloud 同步数据，并且虽然临时会话可以登录第三方同步服务，如 Box 或 Google Drive，但临时会话结束后便无法继续同步数据。

退出登录“共享 iPad”

用户退出登录“共享 iPad”时，该用户的密钥包会立即锁定，所有 App 都会关闭。若要加速新用户登录过程，iPadOS 会暂时推迟某些普通的退出登录操作，并向新用户显示登录窗口。如果用户在此期间（约 30 秒）登录，“共享 iPad”会在新用户帐户登录的过程中执行推迟的清除操作。但是如果“共享 iPad”处于空闲状态，会触发推迟的清除操作。清除期间登录窗口会重新启动，就像发生了另一次退出登录操作一样。

临时会话结束后，“共享 iPad”会执行完整的退出登录序列，并立即删除临时会话的 APFS 宗卷。

Apple Configurator 安全性

Mac 版 Apple Configurator 采用灵活、安全且以设备为中心的设计，可让管理员快速轻松地配置一部或多部通过 USB 连接到 Mac 的 iOS、iPadOS 和 Apple tvOS 设备（或经 Bonjour 配对的 Apple tvOS 设备），然后再将其分发给用户。通过 Mac 版 Apple Configurator，管理员可以更新软件、安装 App 和配置描述文件、重新命名和更改设备上的墙纸、导出设备信息和文稿以及执行其他操作。

Mac 版 Apple Configurator 还可以修复或恢复搭载 Apple 芯片和 Apple T2 安全芯片的 Mac 电脑。Mac 以这种方式进行修复或恢复时，包含操作系统（macOS、Apple 芯片机型的 recoveryOS 或 T2 安全芯片机型的 sepOS）最新次要更新的文件会从 Apple 服务器安全下载，并直接在 Mac 上安装。修复或恢复成功后，文件会从运行 Apple Configurator 的 Mac 上删除。用户在任何时候都不能在 Apple Configurator 之外检查或使用此文件。

即使设备不是直接从 Apple、Apple 授权经销商或 Apple 授权运营商处购买，管理员也可以选择使用 Mac 版 Apple Configurator 或 iPhone 版 Apple Configurator 将设备添加到“Apple 校园教务管理”、“Apple 商务管理”或“Apple 商务必备”。在管理员设置已手动注册的设备时，其表现与这些服务之一中的任何其他通过强制监督和移动设备管理 (MDM) 注册的设备相同。对于非直接购买的设备，用户拥有 30 天的临时期限来从这些服务之一、监督和 MDM 中解除设备。

组织还可以使用 Mac 版 Apple Configurator 激活完全没有互联网连接的 iOS、iPadOS 和 Apple tvOS 设备，方法是在设备设置期间把它们连接到已接入互联网的主机 Mac。管理员可使用其必要配置（包括 App、描述文件和文稿）恢复、激活和准备设备，不再需要接入无线局域网或蜂窝网络。此功能不允许管理员在非网络共享激活期间绕过任何正常所需的现有激活锁要求。

屏幕使用时间安全性

“屏幕使用时间”是用于查看和管理成人及其子女在 App、网站等内容上所花费时间的内建功能。包含两类用户：成人和（被管理的）儿童。

“屏幕使用时间”并非新的系统安全性功能，但很有必要了解它如何保护设备间所收集和共享的数据的隐私及安全性。“屏幕使用时间”在 iOS 12 或更高版本、iPadOS 13.1 或更高版本、macOS 10.15 或更高版本中可用，只有部分功能在 watchOS 6 或更高版本中可用。

以下表格描述了“屏幕使用时间”的主要功能。

功能	支持的操作系统
查看使用数据	iOS iPadOS macOS
执行其他访问限制	iOS iPadOS macOS watchOS
设定网页使用限额	iOS iPadOS macOS
设定 App 限额	iOS iPadOS macOS watchOS
配置停用时间	iOS iPadOS macOS watchOS

对于管理自己的设备使用情况的用户，“屏幕使用时间”控制项目和使用数据可通过 CloudKit 端对端加密在关联了相同 iCloud 帐户的设备间同步。此功能需要用户帐户启用双重认证（同步默认打开）。“屏幕使用时间”替代了 iOS 和 iPadOS 早期版本中的“访问限制”功能以及 macOS 早期版本中的“家长控制”功能。

在 iOS 13 或更高版本、iPadOS 13.1 或更高版本和 macOS 10.15 或更高版本中，如果“屏幕使用时间”用户和受管理孩子的 iCloud 帐户启用了双重认证，则他们的使用情况会自动在设备间共享。用户清除 Safari 浏览器历史记录或删除 App 时，对应的使用数据也会从该设备和所有同步的设备中移除。

家长和“屏幕使用时间”

家长还可以使用 iOS、iPadOS 和 macOS 设备上的“屏幕使用时间”来了解和控制孩子的设备使用情况。如果家长是 iCloud“家人共享”中的家人共享组织者，他们可以查看孩子的使用数据，以及管理孩子的“屏幕使用时间”设置。家长打开“屏幕使用时间”时孩子会收到通知，孩子也可以监控自己的使用情况。为孩子打开“屏幕使用时间”时，家长需要设定密码以让孩子无法进行更改。孩子可在成年（成年年龄取决于国家或地区）后关闭此监控。

使用数据和配置设置使用端对端加密的 Apple 身份识别服务 (IDS) 协议在家长和孩子的设备间传输。加密数据可能会短暂储存在 IDS 服务器上，直到接收设备读取了此数据（例如，如果 iPhone、iPad 或 iPod touch 处于关机状态，则只要开机，就视为已读取数据）。Apple 无法读取此数据。

“屏幕使用时间”分析

如果用户打开了“共享 iPhone 与手表分析”，Apple 只会收集以下匿名数据以更充分了解“屏幕使用时间”的使用方式：

- “屏幕使用时间”是在“设置助理”过程中打开还是稍后在“设置”中打开
- 创建限额后“类别”使用情况的变化（90 天内）
- “屏幕使用时间”是否打开
- “停用时间”是否启用
- “请求更多使用时间”请求的使用次数
- App 限额的数量
- 用户在“屏幕使用时间”设置中查看使用情况的次数，按用户类型或查看类型（本地、远程、小组件）统计
- 用户忽略限额的次数，按用户类型统计
- 用户删除限额的次数，按用户类型统计

Apple 不会收集特定 App 或网页的使用数据。用户在“屏幕使用时间”使用信息中看到 App 列表时，App 图标是直接从 App Store 拉取的，不会保留这些请求中的任何数据。

术语表

安全储存组件 芯片设计为使用不可更改的 RO 代码、硬件随机数生成器、加密引擎和物理篡改检测。在支持的设备上，安全隔区与用于储存反重放随机数的安全储存组件配对。为了读取和更新随机数，安全隔区和储存芯片采用安全协议来帮助确保对随机数的排他访问。此技术已更迭多代，提供了不同的安全性保证。

地址空间布局随机化 (ASLR) 操作系统所采用的一项技术，旨在让恶意利用软件错误的成功几率极大地降低。通过确保内存地址和偏移量不可预测，使攻击代码无法对这些值进行硬编码。

底层引导载入程序 (LLB) 在具有两步启动架构的 Mac 电脑上，LLB 包含由 Boot ROM 调用的代码，该代码随后会载入 iBoot，成为安全启动链的一环。

恢复模式 该模式用于在无法识别用户设备的情况下恢复许多 Apple 设备，使用户可以重新安装操作系统。

集成电路 (IC) 也称为**微芯片**。

可擦除存储器 NAND 存储器中一个用于储存加密密钥的专用区域，可被直接寻址和安全擦除。尽管当攻击者实际占有设备时，可擦除存储器无法提供保护，但其中存储的密钥可用作密钥层级的一部分，用于实现快速擦除和前向安全性。

联合测试行动小组 (JTAG) 一个由程序员和电路开发者采用的标准硬件调试工具。

临时椭圆曲线迪菲-赫尔曼交换 (ECDHE) 一种基于椭圆曲线的密钥交换机制。ECDHE 允许双方就私密密钥达成一致，其使用的方式可防止窃听双方信息的窃听者发现密钥。

媒介密钥 加密密钥层次的一部分，可帮助实现安全的立即擦除。在 iOS、iPadOS、Apple tvOS 和 watchOS 中，媒介密钥会封装数据宗卷上的元数据（因此，没有媒介密钥便无法访问所有文件独有密钥，也就无法访问受数据保护加密方法所保护的文件）。在 macOS 中，媒介密钥会封装文件保险箱所保护宗卷上的密钥材料、所有元数据和数据。在上述任何一种情况下，擦除媒介密钥会让加密的数据变得不可访问。

门禁 macOS 中的一项技术，其设计旨在帮助确保仅受信任的软件可在用户的 Mac 上运行。

密封密钥保护 (SKP) 数据保护中的一种技术，其使用系统软件的测量值和仅在硬件中可用的密钥来保护（或**密封**）加密密钥。

密码派生密钥 (PDK) 用户密码与长期 SKP 密钥和安全隔区的 UID 配合使用，由此派生加密密钥。

密钥包 一种用于储存一组类密钥的数据结构。每种类型（用户、设备、系统、备份、托管或 iCloud 云备份）的格式都相同。

包含以下内容的标头：版本（在 iOS 12 或更高版本中设为四）；类型（系统、备份、托管或 iCloud 云备份）；密钥包 UUID；HMAC（若密钥包已签名）；用于封装类密钥的方法：配合盐和迭代计数使用 Tangling 及 UID 或 PBKDF2。

类密钥列表：密钥 UUID；类（哪个文件或钥匙串数据保护类）；封装类型（仅 UID 派生密钥；UID 派生密钥和密码派生密钥）；封装的类密钥；非对称类的公钥。

密钥封装 使用一个密钥来加密另一个密钥。iOS 和 iPadOS 根据 RFC 3394 使用 NIST AES 密钥封装。

内存控制器 片上系统中的子系统，控制片上系统与其主内存之间的接口。

片上系统 (SoC) 一种将多种组件整合到单个芯片上的集成电路 (IC)。应用程序处理器、安全隔区和其他协处理器都是 SoC 的组件。

启动进程寄存器 (BPR) 一组片上系统 (SoC) 硬件旗标，软件可使用其跟踪设备进入的启动模式，如设备固件更新 (DFU) 模式和恢复模式。启动进程寄存器旗标经设定后就不可清除。这样就允许后续软件获得系统状态的可信指示。

启动转换 支持在受支持的 Mac 电脑上安装 Microsoft Windows 的 Mac 实用工具。

软件种子位 安全隔区 AES 引擎中的专用位，从 UID 生成密钥时追加到 UID 末尾。每个软件种子位都有对应的锁定位。只要对应的锁定位还未设定，安全隔区 Boot ROM 和操作系统就可独立更改每个软件种子位的值。设定锁定位后，其本身或软件种子位都不能修改。安全隔区重启时，软件种子位和其锁定位会还原。

设备固件升级 (DFU) 模式 设备的 Boot ROM 代码在等待通过 USB 进行恢复时所处的模式。在 DFU 模式下，设备为黑屏，但在连接到运行 iTunes 或“访达”的电脑时，会出现以下提示：“iTunes（或‘访达’）检测到一个处于恢复模式的（iPad、iPhone 或 iPod touch）。用户必须先恢复此（iPad、iPhone 或 iPod touch），然后才能将它与 iTunes（或‘访达’）配合使用。”

输入/输出内存管理单元 (IOMMU) 输入/输出内存管理单元。集成芯片中的子系统，用于控制从其他输入/输出设备和外围设备访问地址空间的权限。

数据保护 支持的 Apple 设备的文件和钥匙串保护机制。它也可以指 App 用来保护文件和钥匙串项的 API。

数据保险箱 由内核强制实施的一种机制，用于防止未经授权访问数据，不管请求的 App 本身是否经过沙盒化。

随机数 用于各种安全协议的唯一一次性数字。

统一可扩展固件接口 (UEFi) 固件 一种 BIOS 的替代技术，用于将固件连接到电脑的操作系统。

统一资源标识符 (URI) 可识别基于网络的资源的字符串。

椭圆曲线数字签名算法 (ECDSA) 一种基于椭圆曲线加密的数字签名算法。

唯一 ID (UID) 一个 256 位的 AES 密钥，在设备制造过程中刻录在每个处理器上。这种密钥无法由固件或软件读取，只能由处理器的硬件 AES 引擎使用。若要获取实际密钥，攻击者必须对处理器的芯片发起极为复杂且代价高昂的物理攻击。UID 与设备上的任何其他标识符均无关，包括但不限于 UDID。

文件独有密钥 数据保护用于在文件系统上加密文件的密钥。文件独有密钥使用类密钥封装，储存在文件的元数据中。

文件系统密钥 用于加密每个文件的元数据的密钥，包括其类密钥。存储在可擦除存储器中，用于实现快速擦除，并非用于保密目的。

纹路走向角度映射 一种从部分指纹中提取以用于描述纹路走向和宽度的数学呈现方式。

系统软件授权 将植入硬件的加密密钥与在线服务相结合的过程，以检查在升级时是否仅提供和安装由 Apple 发布且适用于受支持设备的正版软件。

系统协处理器完整性保护 (SCIP) Apple 采用的一种机制，其设计旨在阻止修改协处理器固件。

钥匙串 一种基础架构和一组 API、Apple 操作系统和第三方 App 用来储存和检索密码、密钥及其他敏感凭证。

移动设备管理 (MDM) 一种可让管理员远程管理已注册设备的服务。设备注册后，管理员可通过网络使用 MDM 服务来在设备上配置设置以及执行其他任务，无需进行用户交互。

硬件安全模块 (HSM) 专门防篡改的电脑，可保障数字密钥的安全并对其进行管理。

预置描述文件 Apple 签名的属性列表（.plist 文件），其中列明允许在 iOS 或 iPadOS 设备上安装和测试 App 的实体和授权。开发预置描述文件列出开发人员选择要进行 Ad Hoc 分发的设备，分发预置描述文件中包含企业开发的 App 的 App ID。

增强的串行外设接口 (eSPI) 设计用于同步串行通信的一体化总线。

直接内存访问 (DMA) 一项允许硬件子系统绕过 CPU 直接访问主内存的功能。

专有芯片 ID (ECID) 每台 iOS 和 iPadOS 设备上的处理器所独有的一个 64 位标识符。当在一台设备上接通电话时，该设备通过低功耗蓝牙 (BLE) 4.0 进行短暂广播，使附近的 iCloud 配对设备停止响铃。广播的字节使用与“接力”广播相同的方法来加密。作为个性化流程的一部分，此标识符不被视为机密。

组 ID (GID) 类似于 UID，但同一类中的每个处理器的 GID 都相同。

AES 加密引擎 执行 AES 的专用硬件组件。

AES-XTS IEEE 1619-2007 中所规定的一种 AES 模式，用于加密存储介质。

AES（高级加密标准） 一种流行的全局加密标准，用于加密数据以保护其隐私。

APFS（Apple 文件系统） 这是 iOS、iPadOS、Apple tvOS、watchOS 和使用 macOS 10.13 或更高版本的 Mac 电脑的默认文件系统。APFS 具有强加密、空间共享、快照、快速目录大小统计功能以及改进的文件系统基础。

Apple 安全性奖金 Apple 向研究人员发放的奖金，奖励其报告影响最新发布的操作系统及相关最新硬件的漏洞。

Apple 商务管理 一个针对 IT 管理员且基于网站的简单门户，为组织提供了一种快速简单的方法来部署 Apple 设备，无论设备是组织直接从 Apple 购买的，还是购买自 Apple 合作授权经销商或运营商处。在用户获得设备前，管理员无需实际操作或者准备设备，即可在移动设备管理 (MDM) 解决方案中自动注册设备。

Apple 身份识别服务 (IDS) Apple 的 iMessage 信息公钥、APNs 地址和电话号码及电子邮件地址目录，用于查找密钥和设备地址。

Apple 推送通知服务 (APNs) 一项由 Apple 提供的全球服务，用于向 Apple 设备传送推送通知。

Apple 校园教务管理 一个针对 IT 管理员且基于网站的简单门户，为组织提供了一种快速简单的方法来部署 Apple 设备，无论设备是组织直接从 Apple 购买的，还是购买自 Apple 合作授权经销商或运营商处。在用户获得设备前，管理员无需实际操作或者准备设备，即可在移动设备管理 (MDM) 解决方案中自动注册设备。

Boot ROM 设备的处理器在首次启动时所执行的第一个代码。作为处理器不可分割的一部分，Apple 或攻击者均无法修改。

CKRecord 键值对词典，键值对包含通过 CloudKit 存储或获取的数据。

HMAC 一种基于加密哈希函数的信息认证码。

iBoot 适用于所有 Apple 设备的 2 级启动载入程序。载入 XNU 以作为安全启动链一部分的代码。iBoot 可由底层引导载入程序载入或直接由 Boot ROM 载入，具体取决于片上系统 (SoC) 的版本。

NAND 非易失性闪存。

sepOS 安全隔区固件，基于 Apple 定制版本的 L4 微内核。

SSD 控制器 管理存储介质（固态硬盘）的硬件子系统。

Tangling 用户密码转换为密钥并使用设备的 UID 加强的过程。此过程帮助确保暴力攻击只能在特定设备上执行，因此可限制攻击的频度且避免多部设备同时遭到攻击。Tangling 算法是 PBKDF2。这种算法为每次迭代使用加入设备 UID 的 AES 密钥作为伪随机函数 (PRF)。

xART eXtended 反重放技术的缩写。一组为具有反重放功能（基于物理储存架构）的安全隔区提供加密且经认证的永久储存区的服务。请参阅“安全储存组件”。

XNU Apple 操作系统中央的内核。默认为受信任状态，并强制执行代码签名、沙盒化、授权核对和地址空间布局随机化 (ASLR) 等安全措施。

XProtect macOS 中的一项防病毒技术，可基于签名检测和移除恶意软件。

文稿修订记录文稿修订记录

文稿修订记录文稿修订记录

日期	摘要
2022 年 12 月	添加的主题： • iCloud 高级数据保护 更新的主题： • iCloud 安全性概览 • iCloud 加密 • iCloud 云备份安全性 • 帐户恢复联系人安全性 • 遗产联系人安全性

日期	摘要
2022 年 5 月	针对以下系统更新： • iOS 15.4 • iPadOS 15.4 • macOS 12.3 • Apple tvOS 15.4 • watchOS 8.5 添加的主题： • 配对 recoveryOS 的访问限制 • 本地操作系统版本 (love) • 健康共享 • 帐户恢复联系人安全性 • 遗产联系人安全性 • Tap to Pay on iPhone 安全性 • 使用 Apple 钱包访问 • 访问凭证类型 • Apple 钱包中的证件 • 支持 Siri 的 HomeKit 配件 更新的主题： • 配备触控 ID 的妙控键盘 • 面容 ID、触控 ID 和密码 • 脸部匹配安全性 • 通过备用电量使用快捷卡 • 搭载 Apple 芯片的 Mac 的启动模式 • 搭载 Apple 芯片的 Mac 的 LocalPolicy 文件内容 • iOS、iPadOS 和 macOS 中的签名系统宗卷安全性 • watchOS 系统安全性 • Apple 安全性研究设备 • Apple 文件系统作用 • 保护 App 对用户数据的访问 • macOS 中的 App 安全性介绍 • 在 macOS 中防范恶意软件 • iCloud 安全性概览 • 钥匙串安全同步 • iCloud 钥匙串安全恢复 • 通过 Apple Pay 使用付款卡支付 • Apple Pay 中的免接触式凭证 • 停用 Apple Pay 付款卡 • Apple Card 申请 • Apple Cash 安全性 • 将交通卡和电子货币卡添加到 Apple 钱包 • 安全的 Apple Messages for Business • FaceTime 通话安全性 • iOS 中的车钥匙安全性 • Apple Configurator 安全性 移除的主题： • HomeKit 配件和 iCloud

日期	摘要
2021 年 5 月	针对以下系统更新： • iOS 14.5 • iPadOS 14.5 • macOS 11.3 • Apple tvOS 14.5 • watchOS 7.4 添加的主题： • 配备触控 ID 的妙控键盘。 • 安全意图和与安全隔区的连接。 • 自动解锁与 Apple Watch。 • CustomOS Image4 清单哈希值 (coih)。 编辑的主题： • 在通过备用电量使用快捷卡中添加了两种新的“快捷模式”交易。 • 编辑了安全隔区功能摘要。 • 在安全 Multi-Boot (smb3) 中添加了软件更新内容。 • 在密封密钥保护 (SKP) 中添加了更多内容。

日期	摘要
2021 年 2 月	针对以下系统更新： • iOS 14.3 • iPadOS 14.3 • macOS 11.1 • Apple tvOS 14.3 • watchOS 7.2 添加的主题： • 内存安全 iBoot 实施 • 搭载 Apple 芯片的 Mac 的启动过程 • 搭载 Apple 芯片的 Mac 的启动模式 • 搭载 Apple 芯片的 Mac 的启动磁盘安全性策略控制 • LocalPolicy 签名密钥创建和管理 • 搭载 Apple 芯片的 Mac 的 LocalPolicy 文件内容 • iOS、iPadOS 和 macOS 中的签名系统宗卷安全性 • Apple 安全性研究设备 • 密码监视 • IPv6 安全性 • iOS 中的车钥匙安全性 更新的主题： • 安全隔区 • 硬件麦克风断联 • 基于 Intel 的 Mac 的 recoveryOS 和诊断环境 • Mac 电脑的直接内存访问保护 • macOS 中的内核扩展 • 系统完整性保护 • watchOS 系统安全性 • 在 macOS 中管理文件保险箱 • App 访问已存储密码的权限 • 密码安全建议 • Apple Cash 安全性 • 安全的 Apple Messages for Business • 无线局域网隐私 • 激活锁安全性 • Apple Configurator 安全性
2020 年 4 月	针对以下系统更新： • iOS 13.4 • iPadOS 13.4 • macOS 10.15.4 • Apple tvOS 13.4 • watchOS 6.2 更新内容： • 硬件麦克风断联中新增了 iPad 麦克风断联。 • 保护 App 访问用户数据中新增了数据保险箱。 • 更新了在 macOS 中管理文件保险箱以及命令行工具。 • 在 macOS 中防范恶意软件中新增了恶意软件移除工具。 • 更新了 iPadOS 中的“共享 iPad”安全性。

日期	摘要
2019 年 12 月	整合《iOS 安全保护手册》、《macOS 安全保护概览》和《Apple T2 安全芯片概览》 针对以下系统更新： • iOS 13.3 • iPadOS 13.3 • macOS 10.15.2 • Apple tvOS 13.3 • watchOS 6.1.1 移除了隐私控制、Siri 和 Siri 建议，以及 Safari 浏览器智能防跟踪。若要了解这些功能的最新信息，请访问 https://www.apple.com/cn/privacy/。
2019 年 5 月	针对 iOS 12.3 的更新 • 支持 TLS 1.3 • 修订“隔空投送”安全性描述 • DFU 模式和恢复模式 • 配件连接的密码要求
2018 年 11 月	针对 iOS 12.1 的更新 • FaceTime 群聊
2018 年 9 月	针对 iOS 12 的更新 • 安全隔区 • 操作系统完整性保护 • 通过备用电量使用快捷卡 • DFU 模式和恢复模式 • HomeKit 电视遥控器配件 • 免接触式凭证 • 学生证 • Siri 建议 • Siri 中的快捷指令 • “快捷指令” App • 用户密码管理 • 屏幕使用时间 • 安全性认证和计划
2018 年 7 月	针对 iOS 11.4 的更新 • 生物认证策略 • HomeKit • Apple Pay • 商务聊天 • iCloud 云端“信息” • Apple 商务管理
2017 年 12 月	针对 iOS 11.2 的更新 • Apple Pay Cash

日期	摘要
2017 年 10 月	针对 iOS 11.1 的更新 • 安全性认证和计划 • 触控 ID/面容 ID • 共享备忘录 • CloudKit 端对端加密 • TLS 更新 • Apple Pay、使用 Apple Pay 在网上支付 • Siri 建议 • 共享 iPad
2017 年 7 月	针对 iOS 10.3 的更新 • 安全隔区 • 文件数据保护 • 密钥包 • 安全性认证和计划 • SiriKit • HealthKit • 网络安全性 • 蓝牙 • 共享 iPad • 丢失模式 • 激活锁 • 隐私控制
2017 年 3 月	针对 iOS 10 的更新 • 系统安全性 • 数据保护类 • 安全性认证和计划 • HomeKit、ReplayKit、SiriKit • Apple Watch • 无线局域网、VPN • 单点登录 • Apple Pay、使用 Apple Pay 在网上支付 • 信用卡、借记卡和储值卡预置 • Safari 建议
2016 年 5 月	针对 iOS 9.3 的更新 • 管理式 Apple ID • Apple ID 双重认证 • 密钥包 • 安全性认证 • 丢失模式、激活锁 • 安全备忘录 • Apple 校园教务管理 • 共享 iPad

日期	摘要
2015 年 9 月	针对 iOS 9 的更新 • Apple Watch 激活锁 • 密码策略 • 触控 ID API 支持 • A8 上数据保护使用 AES-XTS • 适用于无人值守式软件更新的密钥包 • 认证更新 • 企业级 App 信任模型 • 对于 Safari 浏览器书签的数据保护 • App 传输安全性 • VPN 规格 • 针对 HomeKit 的 iCloud 远程访问 • Apple Pay 回馈卡、Apple Pay 发卡机构的 App • “聚焦”设备上索引 • iOS 配对模型 • Apple Configurator 2 • 访问限制

Apple
One Apple Park Way
Cupertino, CA 95014
apple.com

CN028-00625