Surviving Private Key Compromise in Electronic Payment Systems

GNU

⟨Taler⟩

taler.net

IRC#**taler**

(on freenode)

twitter@taler
mail@taler.net

Florian Dold &
Christian Grothoff
{dold,grothoff}@taler.net

Inria informatiques mathématiques

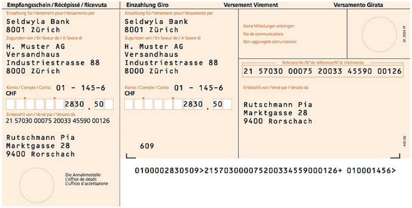
Berner Fachhochschule
Technik und Informatik

Prelude: `draft-dold-payto`

payto://

See also:

`https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml`



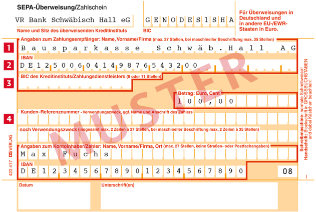
Empfangsschein / Récépissé / Ricevuta | Einzahlung Giro | Versement Virement | Versamento Girata

Einzahlung für / Versement pour / Versamento per
Seldwyla Bank
8001 Zürich
Zugunsten von / En faveur de / A favore di
H. Muster AG
Versandhaus
Industriestrasse 88
8000 Zürich
Konto / Compte / Conto 01 - 145-6
CHF 2830 . 50
Einbezahlt von / Versé par / Versato da
21 57030 00075 20033 45590 00126
Rutschmann Pia
Marktgasse 28
9400 Rorschach
Die Annahmestelle
L'office de dépôt
L'ufficio d'accettazione

Einzahlung für / Versement pour / Versamento per
Seldwyla Bank
8001 Zürich
Zugunsten von / En faveur de / A favore di
H. Muster AG
Versandhaus
Industriestrasse 88
8000 Zürich
Konto / Compte / Conto 01 - 145-6
CHF 2830 . 50
609

Keine Mitteilungen anbringen
Pas de communications
Non aggiungete comunicazioni
Referenz-Nr./N° de référence/N° di riferimento
21 57030 00075 20033 45590 00126
Einbezahlt von / Versé par / Versato da
Rutschmann Pia
Marktgasse 28
9400 Rorschach

0100002830509>215703000075200334559000126+ 010001456>



SEPA-Überweisung/Zahlschein
VR Bank Schwäbisch Hall eG
Name und Sitz des überweisenden Kreditinstituts
GENODES1SHA
BIC
Für Überweisungen in Deutschland und in andere EU-/EWR-Staaten in Euro.
1 Angaben zum Zahlungsempfänger: Name, Vorname/Firma (max. 27 Stellen, bei maschineller Beschriftung max. 35 Stellen)
Bausparkasse Schwäb.Hall AG
2 IBAN
DE12500604149876543200
3 BIC des Kreditinstituts/Zahlungsdienstleisters (8 oder 11 Stellen)
Betrag: Euro, Cent
100,00
4 Kunden-Referenznummer - Verwendungszweck, ggf. Name und Anschrift des Zahlers
noch Verwendungszweck (insgesamt max. 2 Zeilen à 27 Stellen, bei maschineller Beschriftung max. 2 Zeilen à 35 Stellen)
Angaben zum Kontoinhaber/Zahler: Name, Vorname/Firma, Ort (max. 27 Stellen, keine Straßen- oder Postfachangaben)
Max Fuchs
IBAN
DE12345678901234567890 08
Datum
Unterschrift(en)
MUSTER

payto: Uniform Identifiers for Payments and Accounts

Like mailto:, but for bank accounts instead of email accounts!

```
payto://<PAYMENT-METHOD>/<ACCOUNT-NR>
  ?subject=InvoiceNr42
  &amount=EUR:12.50
```

Default action: Open app to review and confirm payment.

Benefits of Payto

- Standardized way to represent financial resources (bank account, bitcoin wallet) and payments to them
- Useful on the client-side on the Web and for FinTech backend applications
- Payment methods (such as IBAN, ACH, Bitcoin) are registered with IANA and allow extra options

Digital cash, made **socially responsible**.

⟨**Taler**⟩

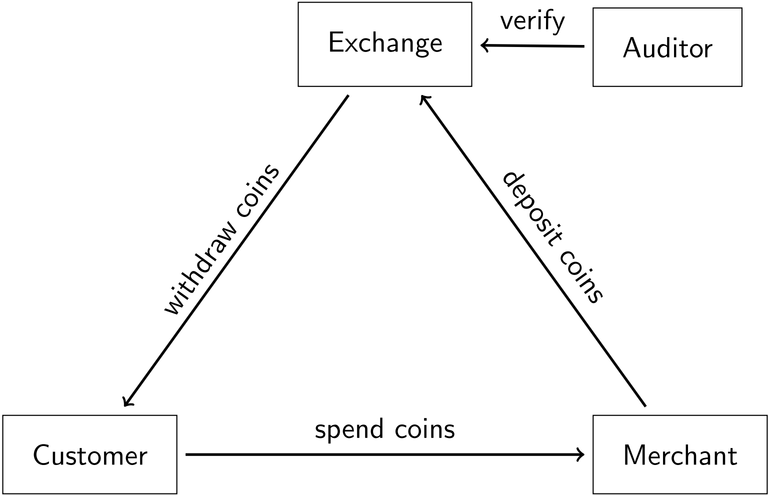
Privacy-Preserving, Practical, Taxable, Free Software, Efficient

What is Taler?

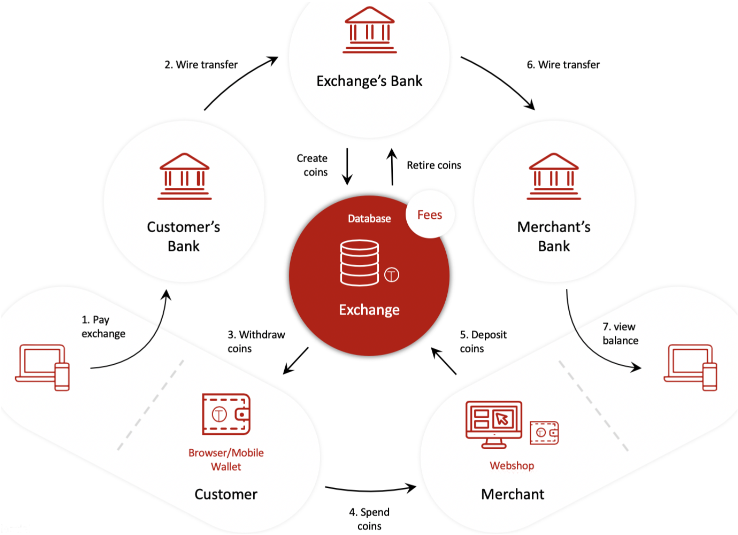
Taler is an electronic instant payment system suitable for a CBEC.

- Uses electronic coins stored in **wallets** on customer's device
- Like **cash**
- Pay in **existing currencies** (i.e. EUR, USD, BTC)

Taler Overview



Architecture of Taler



⇒ Convenient, taxable, privacy-enhancing, & resource friendly!

How does it work?

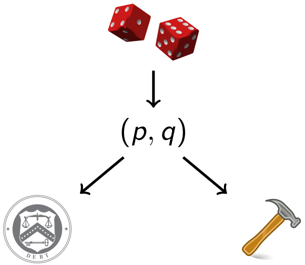
We use a few ancient constructions:

- Cryptographic hash function (1989)
- Blind signature (1983)
- Schnorr signature (1989)
- Diffie-Hellman key exchange (1976)
- Cut-and-choose zero-knowledge proof (1985)

But of course we use modern instantiations.

Exchange setup: Create a denomination key (RSA)

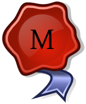
1. Pick random primes p, q.
2. Compute $n := pq$,
   $\phi(n) = (p-1)(q-1)$
3. Pick small $e < \phi(n)$ such that
   $d := e^{-1} \mod \phi(n)$ exists.
4. Publish public key (e, n).



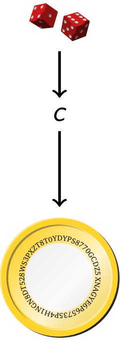
Merchant: Create a signing key (EdDSA)

- pick random $m \mod o$ as private key
- $M = mG$ public key

$\downarrow$
m
$\downarrow$
M

Capability: $m \Rightarrow$ 

Customer: Create a planchet (EdDSA)

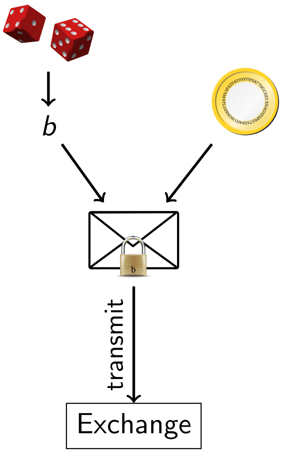


- Pick random $c \mod o$ private key
- $C = cG$ public key

Capability: $c \Rightarrow$ 

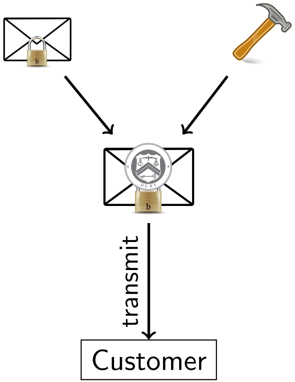
Customer: Blind planchet (RSA)

1. Obtain public key (e, n)
2. Compute $f := FDH(C)$, $f < n$.
3. Pick blinding factor $b \in \mathbb{Z}_n$
4. Transmit $f' := fb^e \mod n$



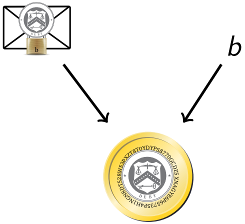
Exchange: Blind sign (RSA)

1. Receive f'.
2. Compute $s' := f'^{d} \mod n$.
3. Send signature s'.

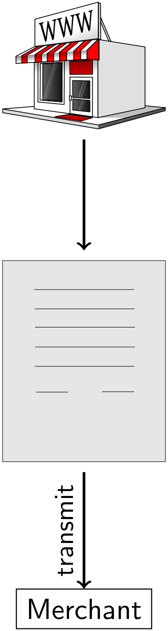


Customer: Unblind coin (RSA)

1. Receive s'.
2. Compute $s := s' b^{-1} \mod n$

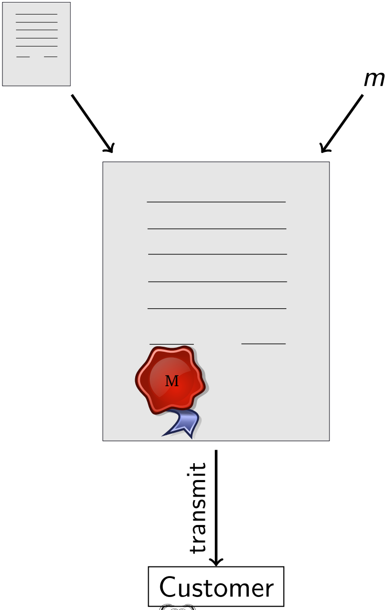


Customer: Build shopping cart



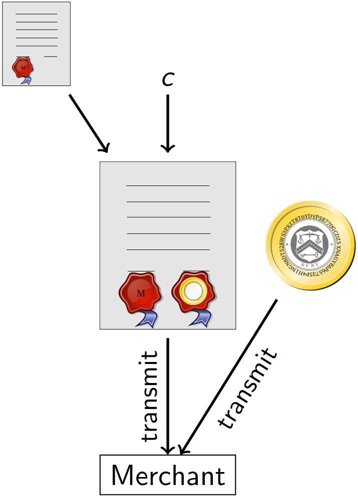
Merchant: Propose contract (EdDSA)

1. Complete proposal D.
2. Send D, $EdDSA_m(D)$



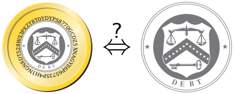
Customer: Spend coin (EdDSA)

1. Receive proposal D, $EdDSA_m(D)$.
2. Send s, C, $EdDSA_c(D)$



Merchant and Exchange: Verify coin (RSA)

$$s^e \stackrel{?}{\equiv} FDH(C) \mod n$$



Warranting deposit safety

Exchange has *another* online signing key $W = wG$:

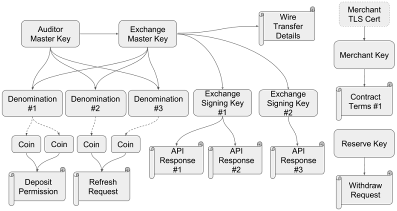
Sends E, $EdDSA_w(M, H(D), FDH(C))$ to the merchant.

This signature means that M was the *first* to deposit C and that the exchange thus must pay M.

Without this, an evil exchange could renege on the deposit confirmation and claim double-spending if a coin were deposited twice, and then not pay either merchant!

Online keys

- The exchange needs d and w to be available for online signing.
- The corresponding public keys W and (e, n) are certified using Taler's public key infrastructure (which uses offline-only keys).

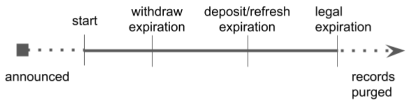


What happens if those private keys are compromised?

Denomination key (e, n) compromise

- An attacker who learns d can sign an arbitrary number of illicit coins into existence and deposit them.
- Auditor and exchange can detect this once the total number of deposits (illicit and legitimate) exceeds the number of legitimate coins the exchange created.
- At this point, (e, n) is *revoked*. Users of *unspent* legitimate coins reveal b from their withdrawal operation and obtain a *refund*.
- The financial loss of the exchange is *bounded* by the number of legitimate coins signed with d.

⇒ Taler frequently rotates denomination signing keys and deletes d after the signing period of the respective key expires.



Online signing key W compromise

- An attacker who learns w can sign deposit confirmations.
- Attacker sets up two (or more) merchants and customer(s) which double-spend legitimate coins at both merchants.
- The merchants only deposit each coin once at the exchange and get paid once.
- The attacker then uses w to fake deposit confirmations for the double-spent transactions.
- The attacker uses the faked deposit confirmations to complain to the auditor that the exchange did not honor the (faked) deposit confirmations.

The auditor can then detect the double-spending, but cannot tell who is to blame, and (likely) would presume an evil exchange, forcing it to pay both merchants.

Detecting online signing key W compromise

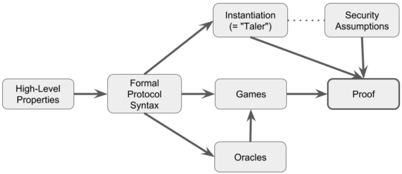
- Merchants are required to *probabilistically* report signed deposit confirmations to the auditor.
- Auditor can thus detect exchanges not reporting signed deposit confirmations.

⇒ Exchange can rekey if illicit key use is detected, then only has to honor deposit confirmations it already provided to the auditor *and* those without proof of double-spending *and* those merchants reported to the auditor.

⇒ Merchants that do not participate in reporting to the auditor risk their deposit permissions being voided in cases of an exchange's private key being compromised.

Summary and further reading

- We can design protocols that fail *soft*.
- GNU Taler's design limits financial damage even in the case private keys are compromised.
- GNU Taler does more:
  - Gives change, can provide refunds
  - Integrates nicely with HTTP, handles network failures
  - High performance
  - Free Software
  - Formal security proofs



- More information at `https://taler.net/`.

How to support?

- GNU, TUM, INRIA and BFH are *not* banks.
- We created Taler Systems SA for commercial support and development of GNU Taler.
- We are in discussions with central banks, commercial banks, suppliers, merchants and various Free Software projects to get GNU Taler into operation.
- More banking partners and venture capital would be welcome.

Talk to us!

Do you have any questions?

References:

1. Christian Grothoff, Bart Polot and Carlo von Loesch. *The Internet is broken: Idealistic Ideas for Building a GNU Network*. **W3C/IAB Workshop on Strengthening the Internet Against Pervasive Monitoring (STRINT)**, 2014.
2. Jeffrey Burdges, Florian Dold, Christian Grothoff and Marcello Stanisci. *Enabling Secure Web Payments with GNU Taler*. **SPACE 2016**.
3. Florian Dold, Sree Harsha Totakura, Benedikt Müller, Jeffrey Burdges and Christian Grothoff. *Taler: Taxable Anonymous Libre Electronic Reserves*. Available upon request. 2016.
4. Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer and Madars Virza. *Zerocash: Decentralized Anonymous Payments from Bitcoin*. **IEEE Symposium on Security & Privacy, 2016**.
5. David Chaum, Amos Fiat and Moni Naor. *Untraceable electronic cash*. **Proceedings on Advances in Cryptology, 1990**.
6. Phillip Rogaway. *The Moral Character of Cryptographic Work*. **Asiacrypt**, 2015.
7. Florian Dold. *The GNU Taler System: Practical and Provably Secure Electronic Payments*. **PhD thesis. University of Rennes 1**, 2019.