NEXT GENERATION INTERNET

The GNU Taler Payment System

Christian Grothoff

December 2025

Agenda

Motivation & Background

GNU Taler: Introduction

Protocol Basics

Offline payments

Future Work & Conclusion

Component Zoo

A Social Problem

This was a question posed to RAND researchers in 1971:

> *"Suppose you were an advisor to the head of the KGB. Suppose you are given the assignment of designing a system for the surveillance of all citizens and visitors within the boundaries of the USSR. The system is not to be too obtrusive or obvious. What would be your decision?"*

A Social Problem

This was a question posed to RAND researchers in 1971:

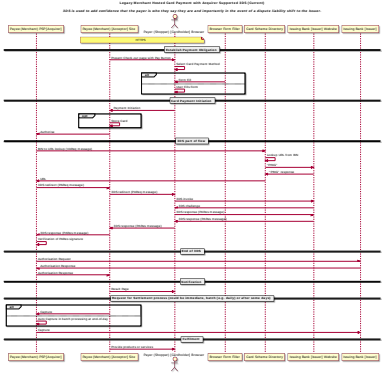
> *"Suppose you were an advisor to the head of the KGB. Suppose you are given the assignment of designing a system for the surveillance of all citizens and visitors within the boundaries of the USSR. The system is not to be too obtrusive or obvious. What would be your decision?"*

"I think one of the big things that we need to do, is we need to get away from true-name payments on the Internet. The credit card payment system is one of the worst things that happened for the user, in terms of being able to divorce their access from their identity." –Edward Snowden, IETF 93 (2015)

Banks have Problems, too!

3D secure (“verified by visa”) is a nightmare:

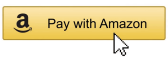
- Complicated process
- Shifts liability to consumer
- Significant latency
- Can refuse valid requests
- Legal vendors excluded
- No privacy for buyers



Online credit card payments will be replaced, but with what?

The Bank's Problem

- Global tech companies push oligopolies
- Privacy and federated finance are at risk
- Economic sovereignty is in danger



Predicting the Future

- Google and Apple will be your bank and run your payment system
- They can target advertising based on your purchase history, location and your ability to pay
- They will provide more usable, faster and broadly available payment solutions; our federated banking system will be history
- After they dominate the payment sector, they will start to charge fees befitting their oligopoly size
- Competitors and vendors not aligning with their corporate “values” will be excluded by policy and go bankrupt
- The imperium will have another major tool for its financial warfare

Central Bank Digital Currency?

Speech by Augustin Carstens, Bank of International Settlements (October 2020) on the difference between Central Bank Digital Currencies and cash.

Central Bank Digital Currency vs. Cash

https://www.youtube.com/watch?v=R_E4Uu7ycqE (10'2020)

GNU Taler: Introduction

GNU Taler [1, 3, 2]

Digital cash, made **socially responsible**.

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

What is Taler?

https://taler.net/en/features.html

Taler is

- a Free/Libre software *payment system* infrastructure project
- ... with a surrounding software ecosystem
- ... and a company (Taler Systems S.A.) and community that wants to deploy it as widely as possible.

However, Taler is

- *not* a currency or speculative asset
- *not* a long-term store of value
- *not* a network or instance of a system
- *not* based on proof-of-work or proof-of-stake

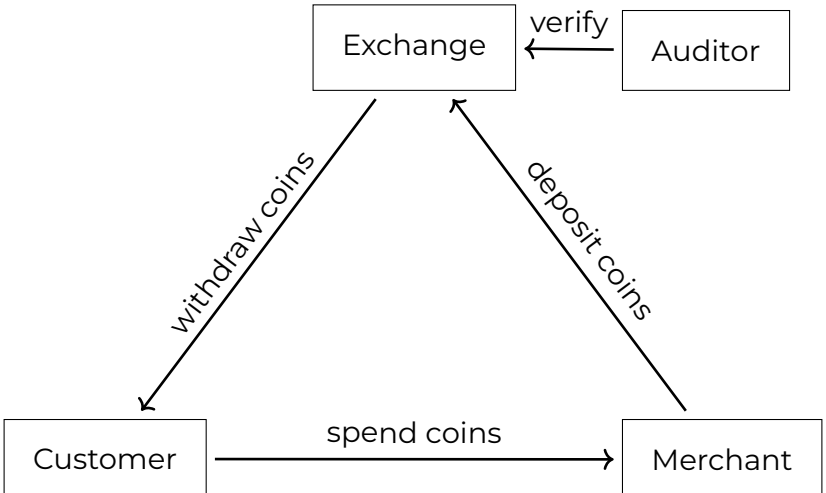
Design principles

https://taler.net/en/principles.html

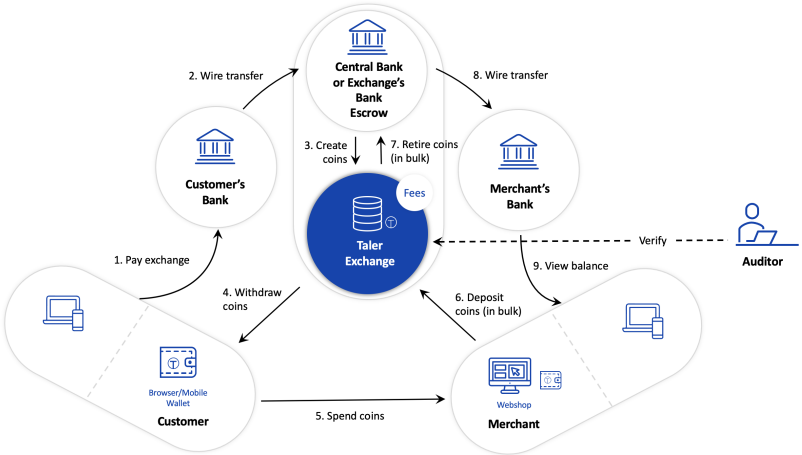
GNU Taler must ...

1. ... be implemented as **free software**.
2. ... protect the **privacy of buyers**.
3. ... enable the state to **tax income** and crack down on illegal business activities.
4. ... prevent payment fraud.
5. ... only **disclose the minimal amount of information necessary**.
6. ... be usable.
7. ... be efficient.
8. ... avoid single points of failure.
9. ... foster **competition**.

Taler Overview



Architecture of Taler



Consumer Impact of Taler

- **Convenient:** pay with one click instantly — in Euro, Dollar, Yen or Bitcoin
- **Friction-free security:** Payments do not require sign-up, login or multi-factor authentication
- **Privacy-preserving:** payment requires/shares no personal information
- **Bank account:** not required

Merchant Impact of Taler

- **Instant clearance:** one-click transactions and instant clearance at par
- **Easy & compliant:** GDPR & PCI-DSS compliance-free and without any effort
- **Major profit increase:** efficient protocol + no fraud = extremely low costs
- **1-click checkout:** without Amazon and without false positives in fraud detection

Usability of Taler

`https://demo.taler.net/`

1. Install browser extension.
2. Visit the `bank.demo.taler.net` to withdraw coins.
3. Visit the `shop.demo.taler.net` to spend coins.

Protocol Basics

BFH Bachelor’s thesis video

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) or Unique signatures (1977) or VRF (1999)
- Cut-and-choose zero-knowledge proof (1985)

But of course we use modern instantiations.

Definition: Taxability

We say Taler is taxable because:

- Merchant's income is visible from deposits.
- Hash of contract is part of deposit data.
- State can trace income and enforce taxation.

Definition: Taxability

We say Taler is taxable because:

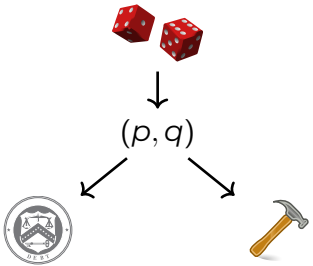
- Merchant's income is visible from deposits.
- Hash of contract is part of deposit data.
- State can trace income and enforce taxation.

Limitations:

- withdraw loophole
- *sharing* coins among family and friends

Exchange setup: Create a denomination key (RSA)

1. Generate 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)

- Generate random number $m \mod o$ as private key
- Compute public key $M := mG$

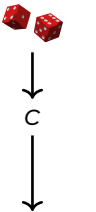
m

M

Capability:
$m \Rightarrow$

Customer: Create a planchet (EdDSA)

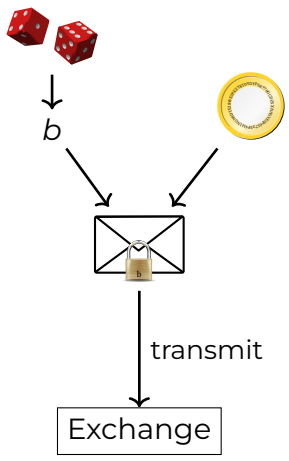
- Generate random number $c \mod o$ as private key
- Compute public key $C := cG$



Capability: $c \Rightarrow$

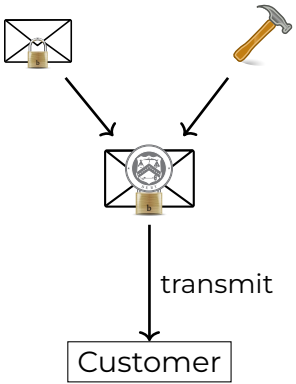
Customer: Blind planchet (RSA)

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



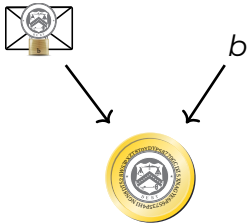
Exchange: Blind sign (RSA)

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

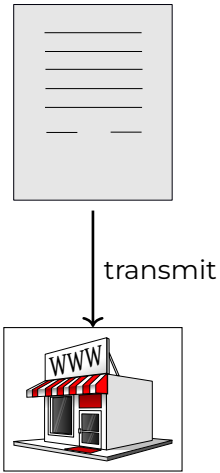


Customer: Unblind coin (RSA)

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

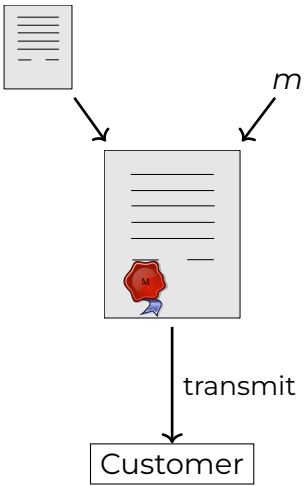


Customer: Build shopping cart



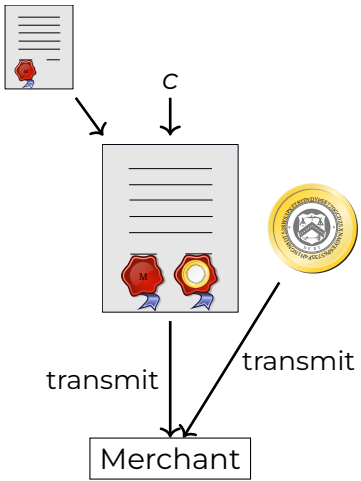
Merchant: Propose contract (EdDSA)

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)

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

The exchange does not only verify the signature, but also checks that the coin was not double-spent.

Merchant and Exchange: Verify coin (RSA)

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

The exchange does not only verify the signature, but also checks that the coin was not double-spent.

Taler is an online payment system.

Giving change

It would be inefficient to pay EUR 100 with 1 cent coins!

- Denomination key represents value of a coin.
- Exchange may offer various denominations.
- Wallet may not have exact change!
- Must be able to pay given sufficient total funds.

Giving change

It would be inefficient to pay EUR 100 with 1 cent coins!

- Denomination key represents value of a coin.
- Exchange may offer various denominations.
- Wallet may not have exact change!
- Must be able to pay given sufficient total funds.

Key goals:

- maintain unlinkability
- maintain taxability of transactions

Giving change

It would be inefficient to pay EUR 100 with 1 cent coins!

- Denomination key represents value of a coin.
- Exchange may offer various denominations.
- Wallet may not have exact change!
- Must be able to pay given sufficient total funds.

Key goals:

- maintain unlinkability
- maintain taxability of transactions

Method:

- Contract can specify to pay *partial value* of a coin.
- Allow wallet to obtain *unlinkable change*.

Unique Signatures

- Some public key operations depend on a nonce or “random” value
  - Ex.: DSA/ECDSA (signing)
  - \+ same plaintext, different ciphertext
  - \- security may break on nonce-reuse
- Generating the nonce deterministically by hashing all inputs (see also: Fiat-Shamir transformation) can make these algorithms **deterministic**
  - Ex.: EdDSA
- If only one form of a valid signature exists and the verifier can check this, a signature is **unique**.
  - Ex.: RSA, Verifiable Random Func.

Unique signatures:

Verifiable Random Functions

Micali, Rabin, & Vadhan (1999) proposed verifiable random functions.

Let M be some input.

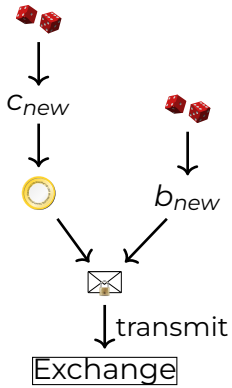
- $(sk, pk) := VRF_{keygen}()$
- *Verifier* picks M
- $(v, p) := VRF_{sign}(M, sk)$
- v is deterministic, unpredictable and high-entropy for any M and sk, and (v, p) can only be computed with sk
- $VRF_{verify}(M, pk, v, p)$ returns true only if v was computed correctly
- sk cannot be derived from M, pk, v and p

Straw-man solution

Given partially spent private coin key c_{old}:

1. Pick random $c_{new} \mod o$ private key
2. Compute $C_{new} := c_{new}G$ public key
3. Pick random b_{new}
4. Compute $f_{new} := FDH(C_{new})$, $m < n$.
5. Transmit $f'_{new} := f_{new}b^e_{new} \mod n$

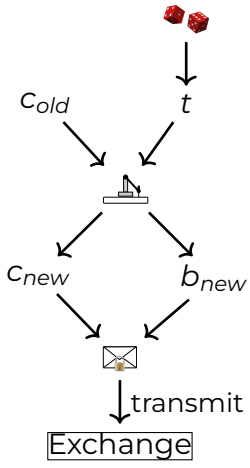
... and sign request for change with c_{old}.



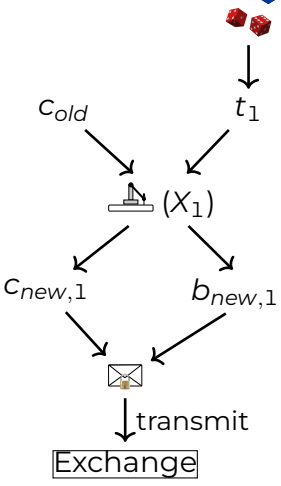
Customer: Transfer setup (UNISIG)

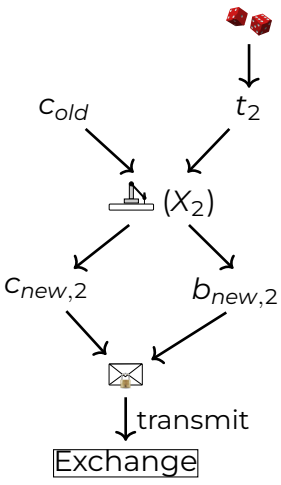
Given partially spent private coin key c_{old}:

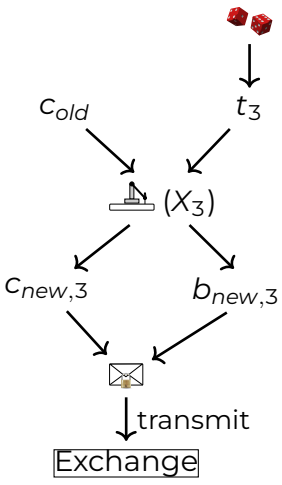
1. Let $C_{old} := c_{old}G$ (as before)
2. Create random nonce t
3. Compute unique signature $X := UNISIG_{c_{old}}(t)$
4. Derive c_{new} and b_{new} from X using HKDF
5. Compute $C_{new} := c_{new}G$
6. Compute $f_{new} := FDH(C_{new})$
7. Transmit $f'_{new} := f_{new}b^e_{new}$



Cut-and-Choose







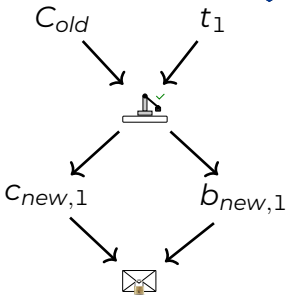
Exchange: Choose!

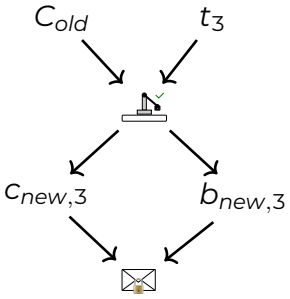
Exchange sends back random $\gamma \in \{1, 2, 3\}$ to the customer.

Customer: Reveal

1. If $\gamma = 1$, send $\langle t_2, X_2 \rangle$, $\langle t_3, X_3 \rangle$ to exchange
2. If $\gamma = 2$, send $\langle t_1, X_1 \rangle$, $\langle t_3, X_3 \rangle$ to exchange
3. If $\gamma = 3$, send $\langle t_1, X_1 \rangle$, $\langle t_2, X_2 \rangle$ to exchange

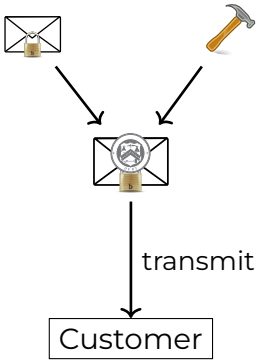
Exchange: Verify ($\gamma = 2$)





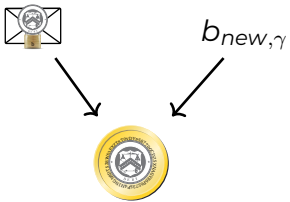
Exchange: Blind sign change (RSA)

1. Take $f'_{new,\gamma}$.
2. Compute $s' := f'^{d}_{new,\gamma} \mod n$.
3. Return signature s'.



Customer: Unblind change (RSA)

1. Receive s'.
2. Compute $s := s' b_{new,\gamma}^{-1} \mod n$.

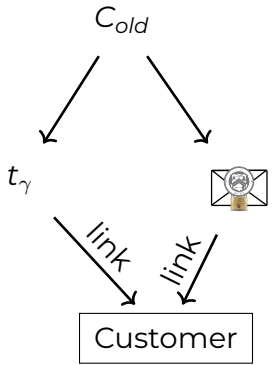


Exchange: Allow linking change

Given C_{old}

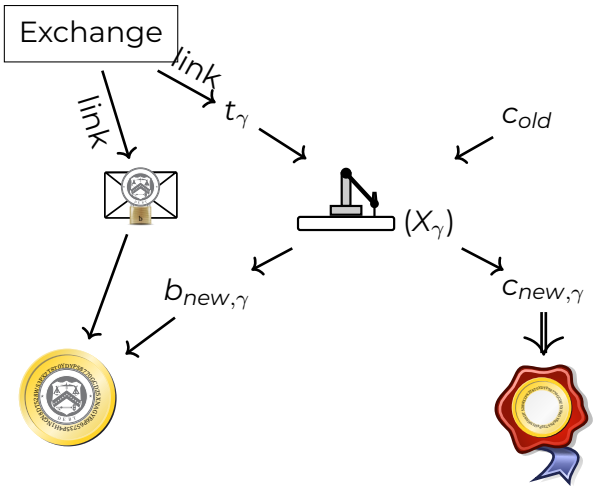
return t_γ and

$s := s' b_{new,\gamma}^{-1} \mod n.$



Customer: Link (threat!)

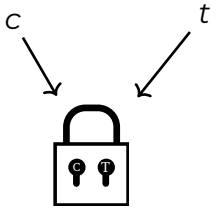
1. Have c_{old}.
2. Obtain T_γ, s from exchange
3. Compute $X_\gamma = UNISIG_{c_{old}}(t_\gamma)$
4. Derive $c_{new,\gamma}$ and $b_{new,\gamma}$ from X_γ
5. Unblind $s := s' b_{new,\gamma}^{-1} \mod n$



VRF vs. Dold'19 with Diffie-Hellman (ECDH)

VRF/unique signatures are *slightly* stronger than required!

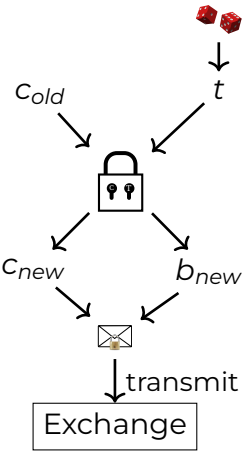
1. Create private keys $c, t \mod o$
2. Define $C = cG$
3. Define $T = tG$
4. Compute DH
   $cT = c(tG) = t(cG) = tC$
5. Sign T with EdDSA: DH is unique, with EdDSA we have a signature, t allows verifier to check!



Transfer setup with ECDH-based Refresh

Given partially spent private coin key c_{old}:

1. Let $C_{old} := c_{old}G$ (as before)
2. Create random private transfer key $t \mod o$
3. Compute $T := tG$
4. Compute $X := c_{old}(tG) = t(c_{old}G) = tC_{old}$
5. Derive c_{new} and b_{new} from X
6. Compute $C_{new} := c_{new}G$
7. Compute $f_{new} := FDH(C_{new})$
8. Transmit $f'_{new} := f_{new}b^e_{new}$



Refresh protocol summary

- Customer asks exchange to convert old coin to new coin
- Protocol ensures new coins can be recovered from old coin

$\Rightarrow$ New coins are owned by the same entity!

Thus, the refresh protocol allows:

- To give unlinkable change.
- To give refunds to an anonymous customer.
- To expire old keys and migrate coins to new ones.
- To handle protocol aborts.

Transactions via refresh are equivalent to *sharing* a wallet.

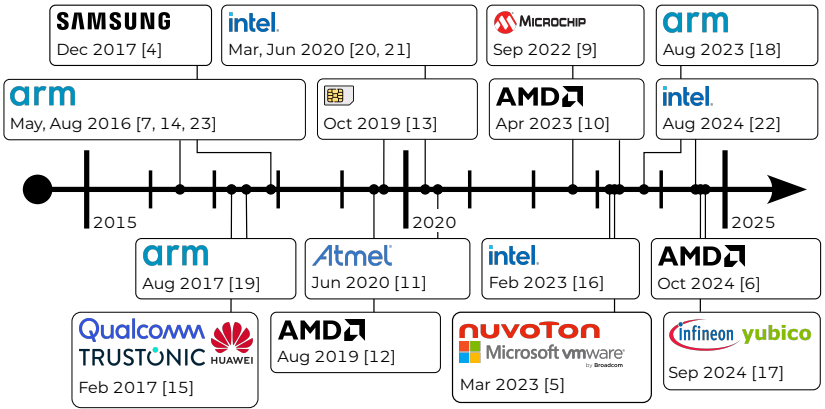
Offline payments

Digitaler Euro — Offline?

Many central banks today demand offline capabilities for CBDCs.

Digitaler Euro — Offline?

Many central banks today demand offline capabilities for CBDCs.



A Scenario

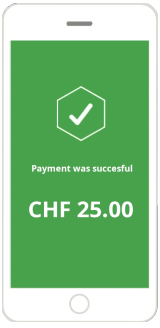
God is offline, but customer pays online

Typical Payment Process

All equivalent: Twint, PayPal, AliPay, PayTM

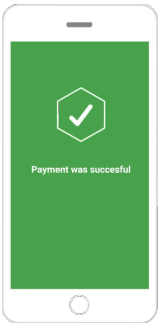
Secure Payment ...

Everything green?



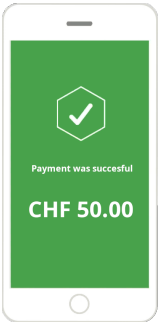
Exploit “Code”

Programming optional

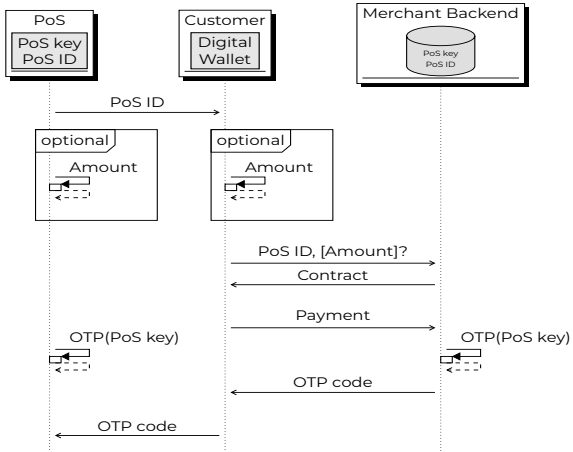


“Customers” *love* Twint …

Daily non-business for shops



Partially Offline Payments with GNU Taler [8]



The Emergency Act of Canada

Speech by Premier Kenney, Alberta, February 2022.

The Emergency Act of Canada

https://www.youtube.com/watch?v=NehMAj492SA (2'2022)

Future Work & Conclusion

Use Case: Journalism

Today:

- Corporate structure
- Advertising primary revenue
- Tracking readers critical for business success
- Journalism and marketing hard to distinguish

Use Case: Journalism

Today:

- Corporate structure
- Advertising primary revenue
- Tracking readers critical for business success
- Journalism and marketing hard to distinguish

With GNU Taler:

- One-click micropayments per article
- Hosting requires no expertise
- Reader-funded reporting separated from marketing
- Readers can remain anonymous

Taler: Project Status

https://docs.taler.net/

- Cryptographic protocols and core exchange component are stable
- Pilot project at Bern University of Applied Sciences cafeteria
- Netzbon (regional currency) in Basel launched
- Taler Operations AG live Swiss-wide
- Internal alpha deployment with GLS Bank (Germany)
- Internal alpha deployment with Magnet Bank (Hungary)

Competitor comparison

	Cash	Bitcoin	Zerocoin	Creditcard	GNU Taler
Online	−−−	++	++	+	+++
Offline	+++	−−	−−	+	++
Trans. cost	+	−−−	−−−	−	++
Speed	+	−−−	−−−	o	++
Taxation	−	−−	−−−	+++	+++
Payer-anon	++	o	++	−−−	+++
Payee-anon	++	o	++	−−−	−−−
Security	−	o	o	−−	++
Conversion	+++	−−−	−−−	+++	+++
Libre	−	+++	+++	− − −	+++

Other ongoing developments

- Privacy-preserving auctions (trading, currency exchange) (`oezguer@taler.net`)
- Hardware and software support for embedded systems (`mikolai@taler.net`)
- Tax-deductable receipts for donations to charities (donau.git)
- Unlinkable anonymous subscriptions and discount tokens (`ivan@taler.net`)
- Support for illiterate and innumerate users[1] (`marc@taler.net`)

[1] Background: `https://myoralvillage.org/`

Open Challanges

- Try to explain this to lawyers and AML staff of banks
- What are convincing arguments for citizens to switch?
- How to address anti-competitive cash-back from card payments?
- . . .

How to support?

Join: `https://lists.gnu.org/mailman/listinfo/taler`

Discuss: `https://ich.taler.net/`

Develop: `https://bugs.taler.net/`, `https://git.taler.net/`

Apply: `https://nlnet.nl/propose`, `https://nlnet.nl/taler`

Translate: `https://weblate.taler.net/`, `translation-volunteer@taler.net`

Integrate: `https://docs.taler.net/`

Donate: `https://gnunet.org/ev`

Partner: `https://taler-systems.com/`

Conclusion

What can we do?

- Suffer mass-surveillance enabled by credit card oligopolies with high fees, and
- Engage in arms race with deliberately unregulatable blockchains

OR

- Establish free software alternative balancing social goals!

References I

Jeffrey Burdges, Florian Dold, Christian Grothoff, and Marcello Stanisci.
Enabling secure web payments with GNU Taler.
In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *6th International Conference on Security, Privacy and Applied Cryptographic Engineering*, number 10076 in LNCS, pages 251–270. Springer, Dec 2016.

David Chaum, Christian Grothoff, and Thomas Moser.
How to issue a central bank digital currency.
In *SNB Working Papers*, number 2021-3. Swiss National Bank, February 2021.

References II

- Florian Dold.
  The GNU Taler system: practical and provably secure electronic payments. (Le système GNU Taler: Paiements électroniques pratiques et sécurisés).
  PhD thesis, University of Rennes 1, France, 2019.

- M. Dorjmyagmar, M. Kim, and H. Kim.
  Security analysis of samsung knox.
  In *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pages 550–553, 2017.

References III

Francisco Falcon.
Vulnerabilities in the tpm 2.0 reference implementation code.
https://blog.quarkslab.com/vulnerabilities-in-the-tpm-20-reference-implementation-code.html, March 2023.

References IV

Stefan Gast, Hannes Weissteiner, Robin Leander Schröder, and Daniel Gruss.
Counterseveillance: Performance-counter attacks on amd sev-snp.
In *Network and Distributed System Security (NDSS) Symposium 2025*, February 2025.
Network and Distributed System Security Symposium 2025 : NDSS 2025, NDSS 2025 ; Conference date: 23-02-2025 Through 28-02-2025.

References V

- R. Guanciale, H. Nemati, C. Baumann, and M. Dam.
  Cache storage channels: Alias-driven attacks and verified countermeasures.
  In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 38–55, May 2016.
- Priscilla Huang, Emmanuel Benoist, Christian Grothoff, and Sebastian Javier Marchano.
  Practical offline payments using one-time passcodes.
  SUERF Policy Briefs, (622), June 2023.

References VI

- Olivier Hériveaux.
  Triple exploit chain with laser fault injection on a secure element.
  In *2022 Workshop on Fault Detection and Tolerance in Cryptography (FDTC)*, pages 9–17, 2022.
- Hans Niklas Jacob, Christian Werling, Robert Buhren, and Jean-Pierre Seifert.
  faultpm: Exposing amd ftpms' deepest secrets.
  In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pages 1128–1142. IEEE, 2023.

References VII

- Jan Jancar, Vladimir Sedlacek, Petr Svenda, and Marek Sys. Minerva: The curse of ECDSA nonces (systematic analysis of lattice attacks on noisy leakage of bit-length of ECDSA nonces). *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(4):281–308, 2020.
- Mengyuan Li, Yinqian Zhang, Zhiqiang Lin, and Yan Solihin. Exploiting unprotected i/o operations in amd's secure encrypted virtualization. In *USENIX Security Symposium*, 2019.

References VIII

- Adaptive Mobile Security Limited.
  Simjacker technical report.
  https://www.enea.com/info/simjacker/, 2019.
- Moritz Lipp, Daniel Gruss, Raphael Spreitzer, Clémentine Maurice, and Stefan Mangard.
  Armageddon: Cache attacks on mobile devices.
  In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 549–564, USA, 2016. USENIX Association.

References IX

- Aravind Machiry, Eric Gustafson, Chad Spensky, Christopher Salls, Nick Stephens, Ruoyu Wang, Antonio Bianchi, Yung Ryn Choe, Christopher Kruegel, and Giovanni Vigna.
  Boomerang: Exploiting the semantic gap in trusted execution environments.
  In *NDSS*, 2017.
- Joseph Nuzman.
  Cve-2022-38090: Improper isolation of shared resources in some intel(r) processors when using intel(r) software guard extensions may allow a privileged user to potentially enable information disclosure via local access.
  `https://www.cve.org/CVERecord?id=CVE-2022-38090`, February 2023.

References X

- Thomas Roche.
  Eucleak: Side-channel attack on the yubikey 5 series—revealing and breaking infineon ecdsa implementation on the way.
  https://ninjalab.io/eucleak/, September 2024.
- Xhani Marvin Saß, Richard Mitev, and Ahmad-Reza Sadeghi.
  Oops..! i glitched it again! how to Multi-Glitch the Glitching-Protections on ARM TrustZone-M.
  In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 6239–6256, 2023.

References XI

Adrian Tang, Simha Sethumadhavan, and Salvatore Stolfo. Clkscrew: Exposing the perils of security-oblivious energy management.
In *Proceedings of the 26th USENIX Conference on Security Symposium*, SEC'17, page 1057–1074, USA, 2017. USENIX Association.

References XII

Jo Van Bulck, Daniel Moghimi, Michael Schwarz, Moritz Lipp, Marina Minkin, Daniel Genkin, Yarom Yuval, Berk Sunar, Daniel Gruss, and Frank Piessens.
LVI: Hijacking Transient Execution through Microarchitectural Load Value Injection.
In *41th IEEE Symposium on Security and Privacy (S&P'20)*, March 2020.

Stephan van Schaik, Andrew Kwong, Daniel Genkin, and Yuval Yarom.

SGAxe: How SGX fails in practice.
`https://sgaxeattack.com/`, June 2020.

References XIII

- Luca Wilke, Florian Sieck, and Thomas Eisenbarth.
  Tdxdown: Single-stepping and instruction counting attacks against intel tdx.
  In *ACM CCS 2024*, 2024.
- Ning Zhang, Kun Sun, Deborah Shands, Wenjing Lou, and Y Thomas Hou.
  Truspy: Cache side-channel information leakage from the secure world on arm devices.
  IACR Cryptol. ePrint Arch., 2016:980, 2016.

Component Zoo

The Taler Software Ecosystem: Overview

Taler is based on modular components that work together to provide a complete payment system:

- **Exchange:** Service provider for digital cash
  - Core exchange software (cryptography, database)
  - Air-gapped key management, real-time **auditing**
  - **libeufin**: Modular integration with banking systems
  - **challenger**: KYC service with OAuth 2.0 API
- **Merchant:** Integration service for existing businesses
  - Core merchant backend software (cryptography, database)
  - **Back-office interface** for staff
  - **Frontend integration** (E-commerce, Point-of-sale)
- **Wallet:** Consumer-controlled applications for e-cash
  - Multi-platform wallet software (for browsers & mobile phones)
  - Wallet backup storage providers (**sync** & **Anastasis**)

Taler Exchange

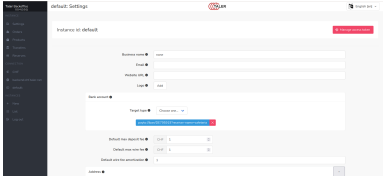
The **Exchange** is the core logic of the payment system.

- One exchange at minimum must be operated per currency
- Offers a REST API for merchants and customers
- Uses several helper processes for configuration and to interact with RTGS and cryptography
- KYC support via OAuth 2.0, KycAID or Persona APIs

Taler Merchant

The **Merchant** is the software run by merchants to accept GNU Taler payments.

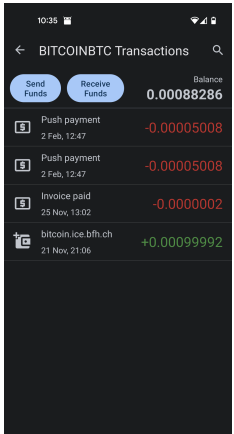
- REST API for integration with e-commerce
- SPA provides Web interface for administration
- Features include:
  - Multi-tenant support
  - Refunds
  - Templates
  - Webhooks
  - Inventory management (optional)



Taler Wallet

The **Wallet** is the software run by consumers to store their digital cash and authorize transactions.

- **wallet-core** is the logic shared by all interfaces
- Works on Android, F-Droid, iOS, Ubuntu Touch, WebExtension (Chrome, Chromium, Firefox, etc.)
- Features include:
  - Multi-currency support
  - Wallet-to-wallet payments (NFC or QR code)
  - CRDT-like data model



Taler Auditor

The **Auditor** is the software run by an independent auditor to validate the operation of an Exchange.

- REST API for additional report inputs by merchants (optional)
- Secure database replication logic

libeufin-nexus

libeufin-nexus allows Taler components to interact with a core banking system. It:

- provides an implementation of the Wire Gateway for the exchange
- supports EBICS 2.5 and 3.0
- other APIs such as FinTS or PSD2-style XS2A APIs can be added without requiring changes to the Exchange
- was tested with GLS Bank (DE) and Postfinance (CH) accounts and real EUR/CHF

libeufin-bank

libeufin-bank implements a standalone bank with a Web interface. It:

- provides the Taler Core Bank API for RESTful online banking using a Web interface (with multi-factor authentication)
- includes a Taler Wire Gateway for the exchange
- offers the Taler Bank Integration API to allow wallets to easily withdraw digital cash
- optionally provides the Taler Conversion Info API for currency conversion between fiat and regional currencies
- optionally integrates with libeufin-nexus to interact with a core banking system

Challenger

Challenger allows clients to obtain validated address (KYC) data about users:

- Customizable Web-based process for address validation
- Can validate phone numbers, e-mail addresses or physical mailing addresses
- Provides an exchange-compatible OAuth 2.0 API

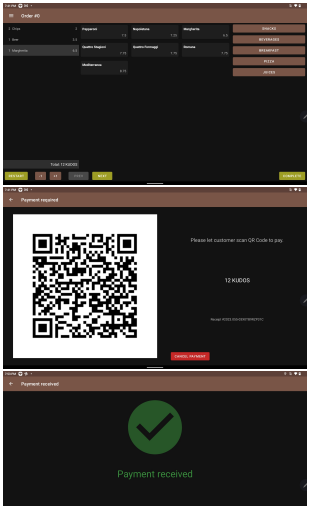
Depolymerization

Depolymerization is a bridge between GNU Taler and blockchains, making Taler a layer 2 system for crypto-currencies (like Lightning).

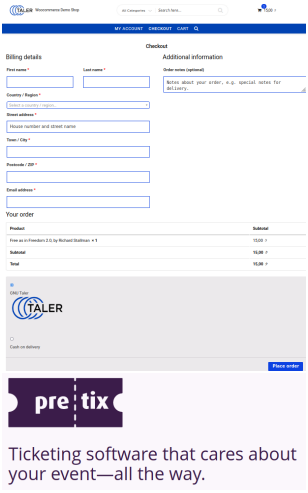
- provides an implementation of the Wire Gateway for the exchange
- Works on top of Bitcoin and Ethereum crypto-currencies, with the DLTs as the “RTGS”
- Provides same API to Exchange as libeufin-nexus

Point-of-Sale App for Android

- Allows merchant to generate orders against Taler backend and display QR code to enable customer to pay in person
- Patterned after ViewTouch restaurant UI



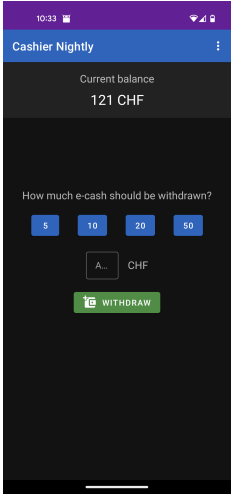
Payment plugins



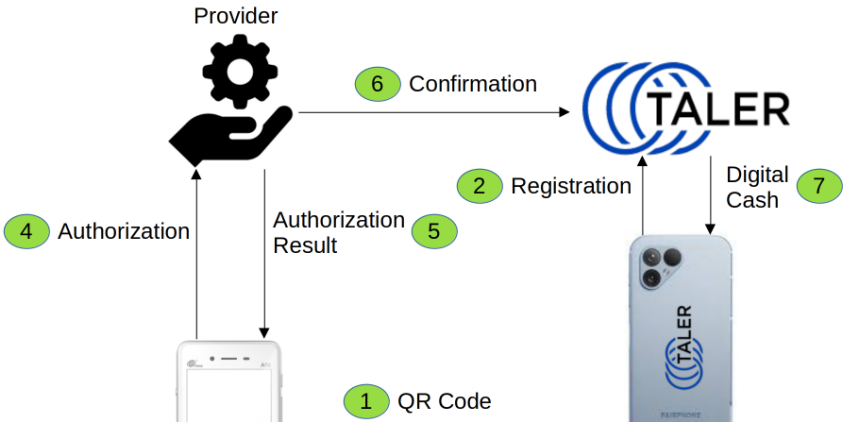
- Pretix, ticket sales system
- Joomla!, an e-commerce platform
- WooCommerce, an e-commerce solution on top of WordPress
- DrupalCommerce, an e-commerce solution on top of Drupal

Cashier App for Android

- Enables BFH staff to convert cash to e-cash
- Staff has special bank accounts with limited funds
- Students can pay staff in cash to receive e-cash
- The Cashier App is implemented in Kotlin



Cashless2ecash by Joel Haeberli



TalDir (WiP)

TalDir is an extension to the existing peer-to-peer payment functionality.

- Registry to associate wallets with network addresses
- Extensible to different types of network services:
  - E-mail
  - SMS
  - Twitter
  - ...
- Send payments or invoices to wallets associated with network address
- Will **not** require sending wallet to use same network service

Acknowledgments

Funded by the European Union (Project 101135475).

Co-funded by
the European Union

Funded by SERI (HEU-Projekt 101135475-TALER).

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs,
Education and Research EAER
State Secretariat for Education,
Research and Innovation SERI

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.