GNU Taler

A Technological Option to Save our Democracy and Economy from "Cashless" Totalitarianism

J. Burdges, F. Dold, **C. Grothoff**, M. Stanisci

Institut National de Recherche en Informatique et en Automatique (Inria)
The GNU Project
Ashoka Fellow

14.12.2016

"I think one of the big things that we need to do, is we need to get a way 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)

Motivation

Modern economies need currency ...

This was a question posed to RAND researchers in 1971:

> *"Suppose you were an advisor to the head of the KGB, the Soviet Secret Police. 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?"*

This was a question posed to RAND researchers in 1971:

> *"Suppose you were an advisor to the head of the KGB, the Soviet Secret Police. 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?"*

Mastercard/Visa are too transparent.

Bitcoin

- Unregulated payment system and currency:

  ⇒ lack of regulation is a feature!
- Implemented in free software
- Decentralised peer-to-peer system

Bitcoin

- Unregulated payment system and currency:
  $\Rightarrow$ lack of regulation is a feature!
- Implemented in free software
- Decentralised peer-to-peer system
- Decentralised banking requires solving Byzantine consensus
- Creative solution: tie initial accumulation to solving consensus

Bitcoin

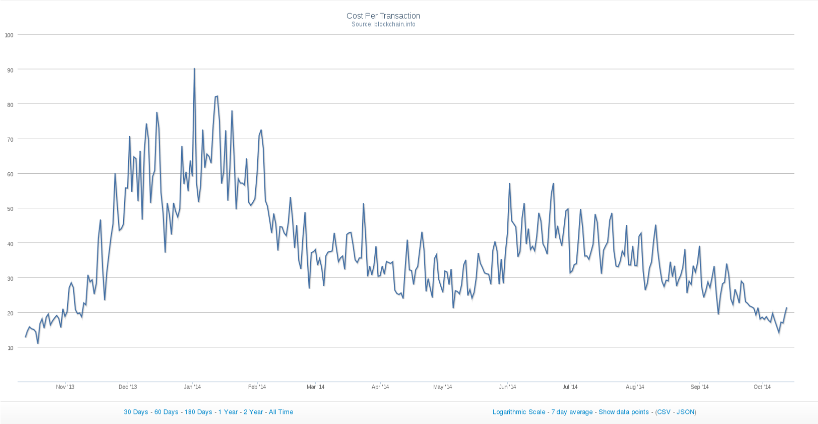
- Unregulated payment system and currency:

  ⇒ lack of regulation is a feature!
- Implemented in free software
- Decentralised peer-to-peer system
- Decentralised banking requires solving Byzantine consensus
- Creative solution: tie initial accumulation to solving consensus

  ⇒ Proof-of-work advances ledger

  ⇒ Very expensive banking

₿?


Cost Per Transaction
Source: blockchain.info

Current average transaction value: $\approx$ 1000 USD

₿?

Cryptography is rather primitive:

All Bitcoin transactions are public and linkable!

⇒ no privacy guarantees

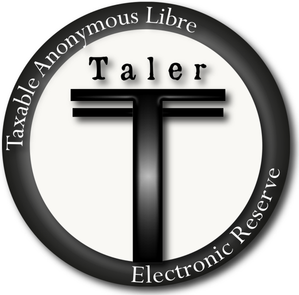
⇒ enhanced with “laundering” services

ZeroCoin, CryptoNote (Monero) and ZeroCash (ZCoin) offer anonymity.

Is society ready for an anarchistic economy?

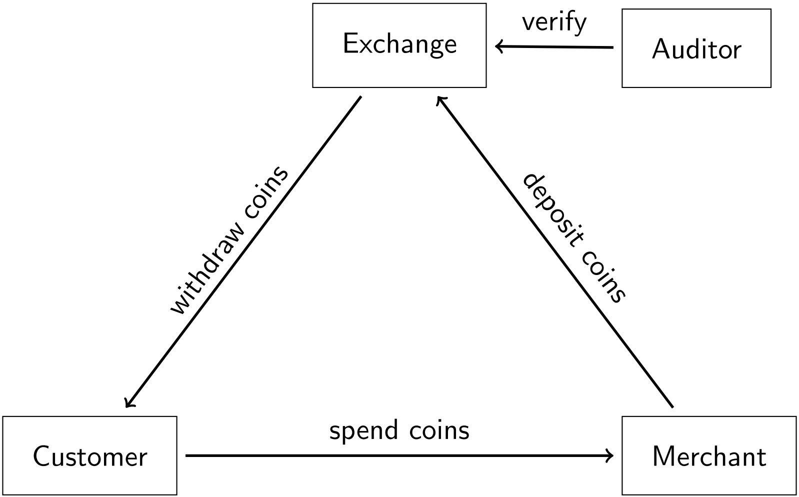
GNU Taler

Digital cash, made **socially responsible**.



Taxable, Anonymous, Libre, Practical, Resource Friendly

Architecture of GNU Taler



Usability of Taler

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

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

Value proposition: Customer

- Convenient: pay with one click
- Guaranteed: never fear being rejected by false-positives in the fraud detection
- Secure: like cash, except no worries about counterfeit
- Privacy-preserving: payment requires no personal information
- Stable: no currency fluctuations, pay in traditional currencies
- Free software: no hidden “gadgets”, third parties can verify

Value proposition: Merchant

- Fast: transactions at Web-speed
- Secure: signed contracts, no legitimate customer rejected by fraud decection
- Free software: competitive pricing and support
- Low fees: efficient protocol + no fraud = low costs
- Flexible: any currency, any amount
- Ethical: no fluctuation risk, no pyramid scheme, not suitable for illegal business
- Legal: complies with Regulation (EU) 2016/679 (GDPR)[1]

[1] Requires privacy by design and data minimization for all data processing in Europe after 25.5.2018.

Value proposition: Government

- Free software = commons: no monopoly, preserve independence
- Taxabiliy: reduces black markets
- Efficiency: high transaction costs hurt the economy
- Security: signed contracts, no counterfeit
- Audited: no bad banks
- Privacy: protection against foreign espionage

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.

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

Merchant Integration: Wallet Detection

```
<script src="taler-wallet-lib.js"></script>
<script>
  taler.onPresent(() => {
    alert("Taler␣wallet␣is␣installed");
  });
  taler.onAbsent(() => {
    alert("Taler␣wallet␣is␣not␣installed");
  });
</script>
```

Merchant Integration: Payment Request

```
HTTP/1.1 402 Payment Required
Content-Type: text/html; charset=UTF-8
X-Taler-Contract-Url: https://shop/generate-contract/42

<!DOCTYPE html>
<html>
  <!-- fallback for browsers without the Taler extension -->
  You do not seem to have Taler installed, here are other
  payment options ...
</html>
```

Merchant Integration: Contract

```
{
  "H_wire":"YTH0C4QBCQ10VDNTJN0DCTTV2Z6JHT5NF43F0RQHZ8JYB5NG4W4G...",
  "amount":{"currency":"EUR","fraction":1,"value":0},
  "auditors":[{"auditor_pub":"42V6TH91Q83FB846DK1GW3JQ5E8DS273W4..."}],
  "exchanges":[{"master_pub":"1T5FA8VQHMMKBHDMYPRZA2ZFK2S63AKF0Y...",
                "url":"https://exchange/"}],
  "expiry":"/Date(1480119270)/",
  "fulfillment_url": "https://shop/article/42?tid=249&time=14714744",
  "max_fee":{"currency":"EUR","fraction":01,"value":0},
    "merchant":{"address":"Mailbox␣4242","jurisdiction":"Jersey",
                "name":"Shop␣Inc."},
  "merchant_pub":"Y1ZAR5346J3ZTEXJCHQY9NJN78EZ2HSKZK8M0MYTNRJG5N...",
  "products":[{
    "description":"Essay:␣The␣GNU␣Project",
    "price":{"currency":"EUR","fraction":1,"value":0},
    "product_id":42,"quantity":1}],
  "refund_deadline":"/Date(1471522470)/",
  "timestamp":"/Date(1471479270)/",
  "transaction_id":249960194066269
}
```

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.

Global setup: Pick an Elliptic curve

Need:

G	generator in ECC curve, a point
o	size of ECC group, $o := \|G\|$, o prime

Now we can, for example, compute:

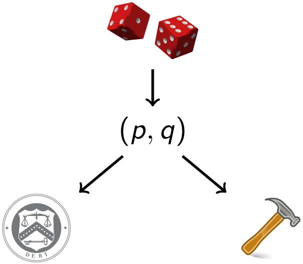
$$\begin{aligned} A &= G + G \\ &= 2G \\ B &= A + G \\ &= 3G \\ C &= cG \texttt{ for } c \in \mathbb{Z} \end{aligned}$$

Note:

$$G = (o + 1)G$$

Exchange setup: Create a denomination key (RSA)

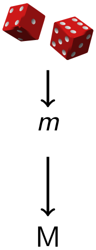
1. Pick random primes p, q.
2. Compute $n := pq$,
   $\phi(n) = (p-1)(q-1)$
3. Pick small $e < \phi(n)$ such that
   $d := e^{-1} \mod \phi(n)$ exists.
4. Publish public key (e, n).



(p, q)

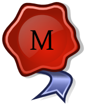
Merchant: Create a signing key (EdDSA)

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

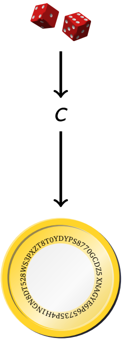


m

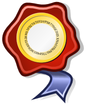
M

Capability: $m \Rightarrow$ 

Customer: Create a planchet (EdDSA)

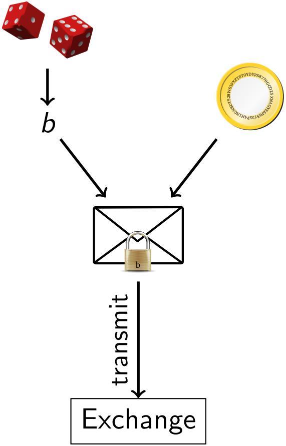


- Pick random $c \mod o$ private key
- $C = cG$ public key

Capability: $c \Rightarrow$ 

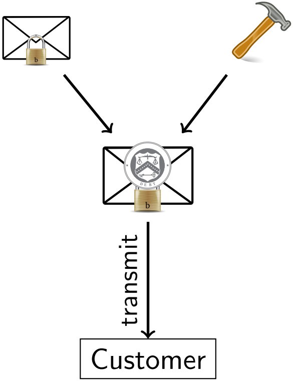
Customer: Blind planchet (RSA)

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



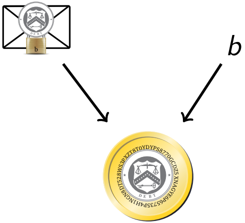
Exchange: Blind sign (RSA)

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

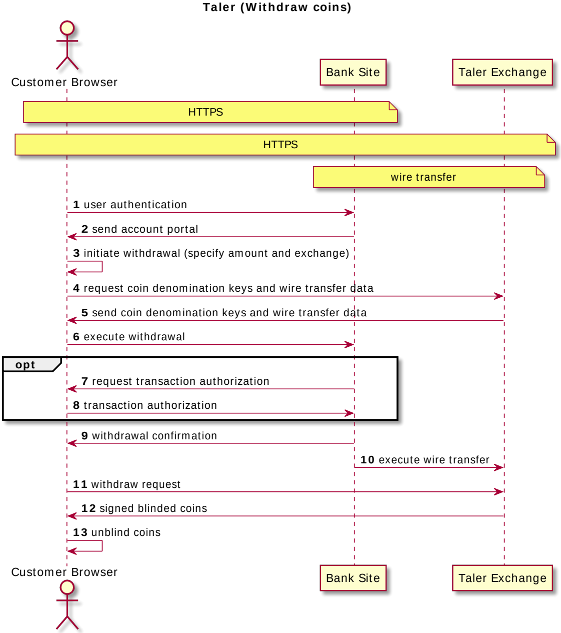


Customer: Unblind coin (RSA)

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

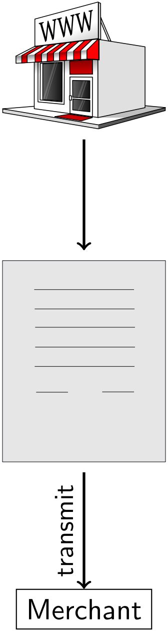


Withdrawing coins on the Web



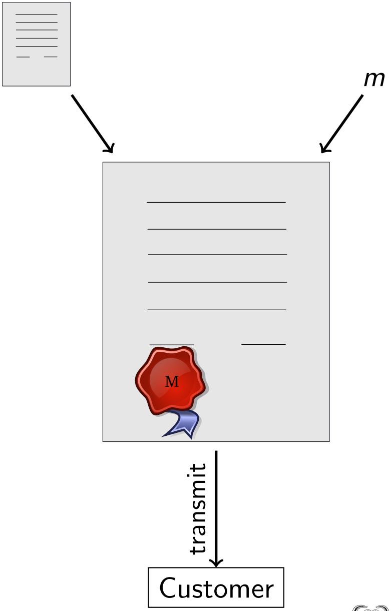
Taler (Withdraw coins)

Customer: Build shopping cart



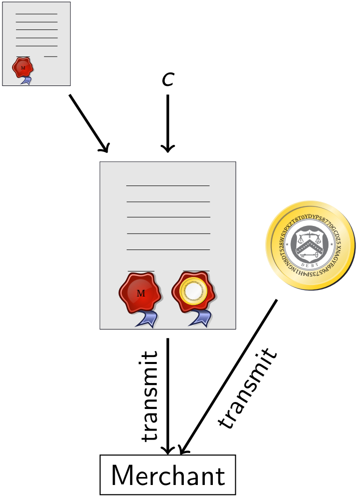
Merchant: Propose contract (EdDSA)

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



Customer: Spend coin (EdDSA)

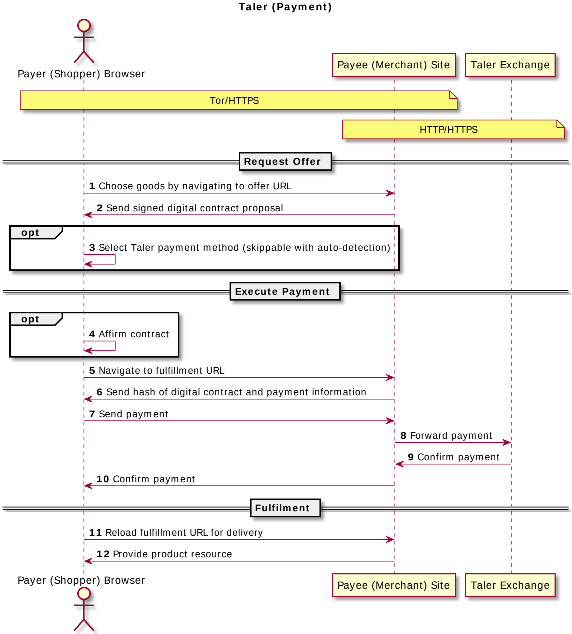
1. Receive proposal D, $EdDSA_m(D)$.
2. Send s, C, $EdDSA_c(D)$



Merchant and Exchange: Verify coin (RSA)

$$s^e \overset{?}{\equiv} m \mod n$$

Payment processing with Taler


Taler (Payment)
Payer (Shopper) Browser
Payee (Merchant) Site
Taler Exchange
Tor/HTTPS
HTTP/HTTPS
Request Offer
1 Choose goods by navigating to offer URL
2 Send signed digital contract proposal
opt
3 Select Taler payment method (skippable with auto-detection)
Execute Payment
opt
4 Affirm contract
5 Navigate to fulfillment URL
6 Send hash of digital contract and payment information
7 Send payment
8 Forward payment
9 Confirm payment
10 Confirm payment
Fulfilment
11 Reload fulfillment URL for delivery
12 Provide product resource
Payer (Shopper) Browser
Payee (Merchant) Site
Taler Exchange

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 for coins.
- Wallet may not have exact change!
- Usability requires ability 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 for coins.
- Wallet may not have exact change!
- Usability requires ability 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 for coins.
- Wallet may not have exact change!
- Usability requires ability to pay given sufficient total funds.

Key goals:

- maintain unlinkability
- maintain taxability of transactions

Method:

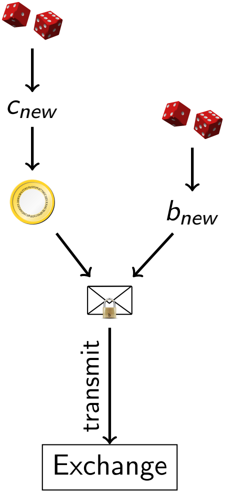
- Contract can specify to only pay *partial value* of a coin.
- Exchange allows wallet to obtain *unlinkable change* for remaining coin value.

Strawman solution

Given partially spent private coin key c_{old}:

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

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

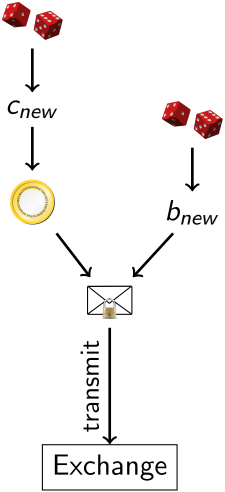


Strawman solution

Given partially spent private coin key c_{old}:

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

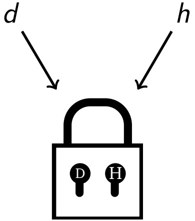
... and sign request for change with c_{old}.



Problem: Owner of c_{new} may differ from owner of c_{old}!

Diffie-Hellman (ECDH)

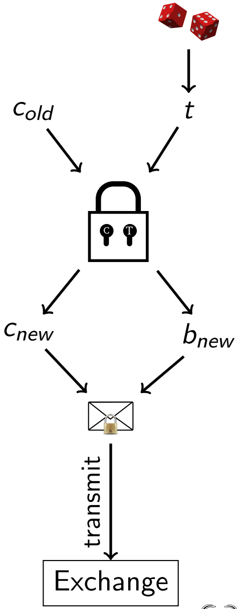
1. Create private keys $d, h \mod o$
2. Define $D = dG$
3. Define $H = hG$
4. Compute $DH := d(hD) = h(dH)$



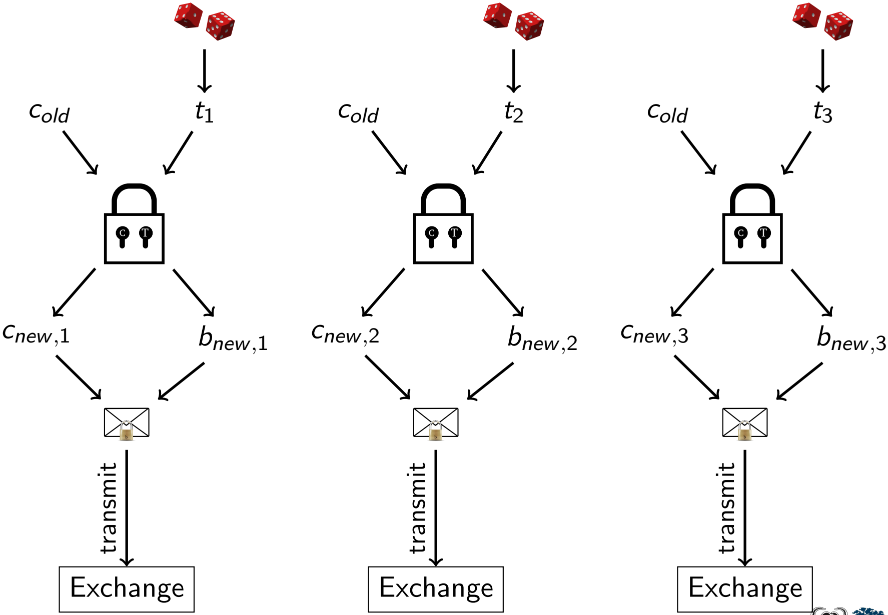
Customer: Transfer key setup (ECDH)

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 $m_{new} := FDH(C_{new})$
8. Transmit $m'_{new} := m_{new} b^{e}_{new}$



Cut-and-Choose



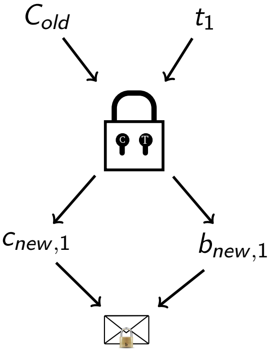
Exchange: Choose!

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

Customer: Reveal

1. If $\gamma = 1$, send t_2, t_3 to exchange
2. If $\gamma = 2$, send t_1, t_3 to exchange
3. If $\gamma = 3$, send t_1, t_2 to exchange

Exchange: Verify ($\gamma = 2$)



C_{old}

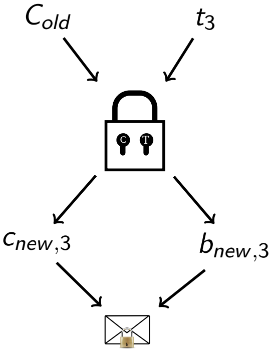
t_1

C T

$c_{new,1}$

$b_{new,1}$

b



C_{old}

t_3

C T

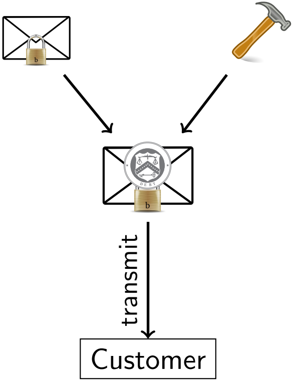
$c_{new,3}$

$b_{new,3}$

b

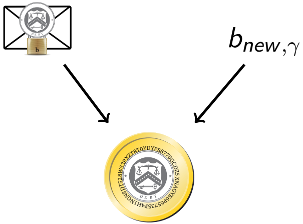
Exchange: Blind sign change (RSA)

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



Customer: Unblind change (RSA)

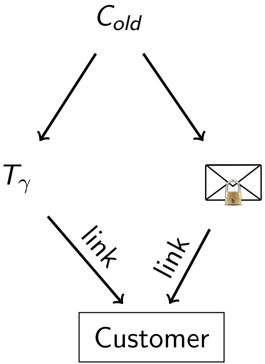
1. Receive s'.
2. Compute $s := s' b_{new,\gamma}^{-1} \mod n$.



Exchange: Allow linking change

Given C_{old}

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



C_{old}

T_γ

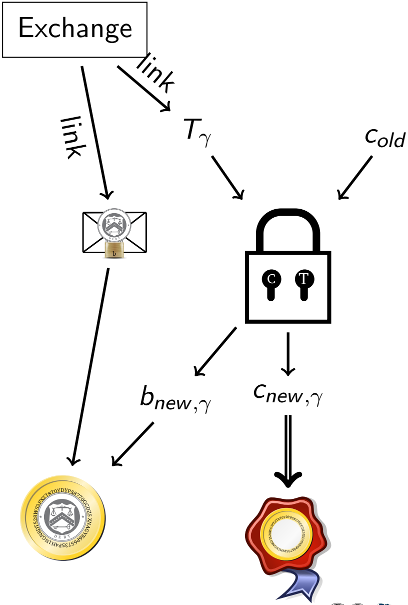
link

link

Customer

Customer: Link (threat!)

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



Refresh protocol summary

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

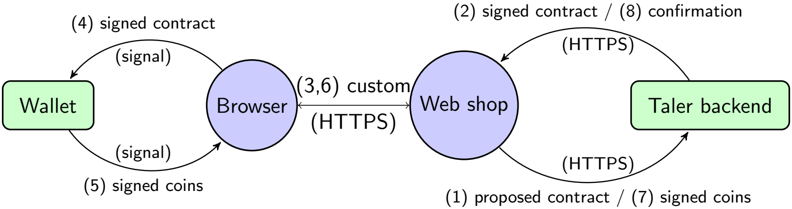
⇒ New coins are owned by the same entity!

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.

Transactions via refresh are equivalent to sharing a wallet.

Operational security



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	−	+++	+++	− − −	+++

Current technical developments

- Improving wallet (error handling, features, browser support)
- Ongoing work on exchange auditing
- Tutorial for merchants
- Tutorial for Web shop integration

`https://api.taler.net/`

How can you help?

- Current wallet only works for browsers and the Web.
- Protocol should work fine also over NFC.

⇒ **Write App for mobile phones and do POS integration!**

How can you help?

- Current wallet only works for browsers and the Web.
- Protocol should work fine also over NFC.

  ⇒ **Write App for mobile phones and do POS integration!**
- Existing Web shops support mostly only credit cards

  ⇒ **Write payment plugins for GNU Taler!**

How can you help?

- Current wallet only works for browsers and the Web.
- Protocol should work fine also over NFC.

  ⇒ **Write App for mobile phones and do POS integration!**
- Existing Web shops support mostly only credit cards

  ⇒ **Write payment plugins for GNU Taler!**
- Current documentation is mostly in English.

  ⇒ **Join our team and help with translations!**

How can you help?

- Current wallet only works for browsers and the Web.
- Protocol should work fine also over NFC.

  ⇒ **Write App for mobile phones and do POS integration!**
- Existing Web shops support mostly only credit cards

  ⇒ **Write payment plugins for GNU Taler!**
- Current documentation is mostly in English.

  ⇒ **Join our team and help with translations!**
- Exchange needs to be a legal (!) business to operate.
- Exchange operator income is from *transaction fees*.

  ⇒ **Find funding and create a startup!**

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, and
- Enjoy the "benefits" of cash

OR

- Establish free software alternative balancing social goals!

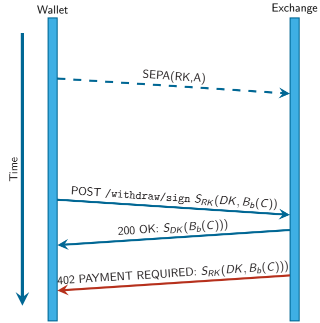
Do you have any questions?

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.

Let money facilitate trade; but ensure capital serves society.

Taler `/withdraw/sign`

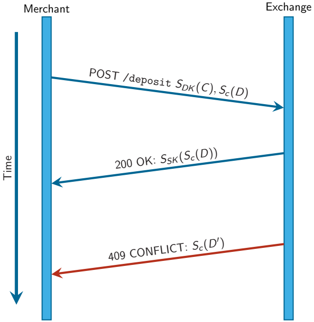

Wallet
Exchange
Time
SEPA(RK,A)
POST `/withdraw/sign` $S_{RK}(DK, B_b(C))$
200 OK: $S_{DK}(B_b(C)))$
402 PAYMENT REQUIRED: $S_{RK}(DK, B_b(C)))$

Result: $\langle c, S_{DK}(C) \rangle$.

A	Some amount, $A \geq A_{DK}$
RK	Reserve key
DK	Denomination key
b	Blinding factor
$B_b()$	RSA-FDH blinding
C	Coin public key $C := cG$
$S_{RK}()$	EdDSA signature
$S_{DK}()$	RSA-FDH signature

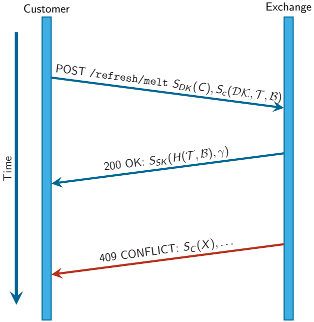
Taler `/deposit`

Merchant and exchange see only the public coin $\langle C, S_{DK}(C)\rangle$.



DK	Denomination key
$S_{DK}()$	RSA-FDH signature using DK
c	Private coin key, $C := cG$.
$S_C()$	EdDSA signature using c
D	Deposit details
SK	Exchange's signing key
$S_{SK}()$	EdDSA signature using SK
D'	Conficting deposit details $D' \neq D$

Taler `/refresh/melt`


Customer
Exchange
Time
POST /refresh/melt $S_{DK}(C), S_c(\mathcal{DK}, \mathcal{T}, \mathcal{B})$
200 OK: $S_{SK}(H(\mathcal{T}, \mathcal{B}), \gamma)$
409 CONFLICT: $S_C(X), \ldots$

κ System-wide security parameter, usually 3.

$\mathcal{DK}$ $:= [DK^{(i)}]_i$
List of denomination keys
$D + \sum_i A_{DK(i)} < A_{DK}$

t_j Random scalar for $j < \kappa$

$\mathcal{T}$ $:= [T_j]_\kappa$ where $T_j = t_j G$

k_j $:= cT_j = t_j C$ is an ECDHE

$b_j^{(i)}$ $:= KDF_b(k_j, i)$

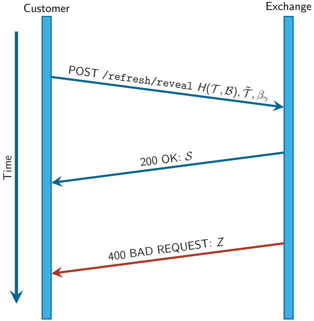
$c_j^{(i)}$ $:= KDF_c(k_j, i)$

$C_j^{(i)}$ $:= c_j^{(i)} G$

$\mathcal{B}$ $:= [H(\beta_j)]_\kappa$ where
$$\beta_j := \left[B_{b_j^{(i)}}(C_j^{(i)}) \right]_i$$

γ Random value in $[0, \kappa)$

Taler /refresh/reveal

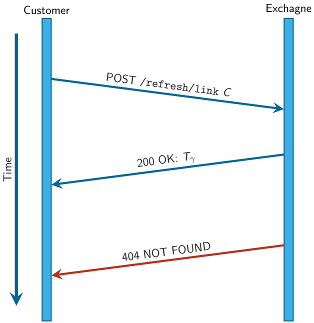


Customer
Exchange
Time
POST /refresh/reveal $H(\mathcal{T},\mathcal{B}), \tilde{\mathcal{T}}, \beta_\gamma$
200 OK: $\mathcal{S}$
400 BAD REQUEST: Z

$$\begin{aligned}
\mathcal{DK} &:= [DK^{(i)}]_i \\
t_j &\ .. \\
\tilde{\mathcal{T}} &:= [t_j | j \in \kappa, j \neq \gamma] \\
k_\gamma &:= cT_\gamma = t_\gamma C \\
b_\gamma^{(i)} &:= KDF_b(k_\gamma, i) \\
c_\gamma^{(i)} &:= KDF_c(k_\gamma, i) \\
C_\gamma^{(i)} &:= c_\gamma^{(i)} G \\
B_\gamma^{(i)} &:= B_{b_\gamma^{(i)}}(C_\gamma^{(i)}) \\
\beta_\gamma &:= \left[B_\gamma^{(i)}\right]_i \\
\mathcal{S} &:= \left[S_{DK^{(i)}}(B_\gamma^{(i)})\right]_i
\end{aligned}$$

Z Cut-and-choose missmatch information

Taler `/refresh/link`



C	Old coind public key
T_γ	Linkage data $\mathcal{L}$ at γ