A Social Problem

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?”
A Social Problem

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.

“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.”

The Bank’s Problem

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
The Distraction: Bitcoin

- Unregulated payment system and currency:
  ⇒ lack of regulation is a feature!
- Implemented in free software
- Decentralised peer-to-peer system
The Distraction: 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
The Distraction: 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
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 (ZCash) offer anonymity.
Do you want to have a libertarian economy?

Do you want to live under total surveillance?
GNU Taler

Digital cash, made socially responsible.

Privacy-Preserving, Practical, Taxable, Free Software, Efficient
What is Taler?

Taler is an electronic instant payment system.

- Uses electronic coins stored in wallets on customer’s device
- Like cash
- Pay in existing currencies (i.e. EUR, USD, BTC), or use it to create new regional currencies
Taler Overview

- Exchange
  - Customer: withdraw coins
  - Merchant: deposit coins
  - Merchant: spend coins
Architecture of Taler

⇒ Convenient, taxable, privacy-enhancing, & resource friendly!
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.
Social Impact of Taler

for the disadvantaged
anti-discrimination
financial education / self-responsibility
accessibility
regional markets
alternative economies
economic integration (migrants)
for new economies

for a better Internet
privacy
anti-DDoS
Internet security
anti-spam

for ordinary citizens
libre
anti-corruption
green / efficient
comfort

for a better market economy
economic independence
improves competition
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
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
Use Cases: Refugee Camps

Today:

- Non-bankable
- Direct distribution of goods to population
- Limited economic activity in camps
- High level of economic dependence

With GNU Taler:

- Local currency issued as basic income backed by aid
- Taxation possible based on economic status
- Local governance enabled by local taxes
- Increased economic independence and political participation
Use Cases: Refugee Camps

Today:

- Non-bankable
- Direct distribution of goods to population
- Limited economic activity in camps
- High level of economic dependence

With GNU Taler:

- Local currency issued as basic income backed by aid
- Taxation possible based on economic status
- Local governance enabled by local taxes
- Increased economic independence and political participation
Use Case: Anti-Spam

Today, p≡p provides authenticated encryption for e-mail:

- Free software
- Easy to use opportunistic encryption
- Available for Outlook, Android, Enigmail
- Spies & spam filters can no longer inspect content
Use Case: Anti-Spam

Today, p≡p provides authenticated encryption for e-mail:

▶ Free software
▶ Easy to use opportunistic encryption
▶ Available for Outlook, Android, Enigmail
▶ Spies & spam filters can no longer inspect content

With GNU Taler:

▶ Peer-to-peer payments via e-mail
▶ If unsolicited sender, hide messages from user & automatically request payment from sender
▶ Sender can attach payment to be moved to inbox
▶ Receiver may grant refund to sender
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.
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
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

Capability: $m \Rightarrow M$
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
Customer: Unblind coin (RSA)

1. Receive $s'$.
2. Compute $s := s'b^{-1} \mod n$
Withdrawing coins on the Web

**Taler (Withdraw coins)**

1. User authentication
2. Send account portal
3. Initiate withdrawal (specify amount and exchange)
4. Request coin denomination keys and wire transfer data
5. Send coin denomination keys and wire transfer data
6. Execute withdrawal

**Opt**
7. Request transaction authorization
8. Transaction authorization
9. Withdrawal confirmation

10. Execute wire transfer
11. Withdraw request
12. Signed blinded coins
13. Unblind coins

Customer Browser

Bank Site

Taler Exchange
Customer: Build shopping cart
Merchant Integration: Wallet Detection

```html
<script src="taler-wallet-lib.js"></script>
<script>
  taler.onPresent(() => {
    alert("Taler wallet is installed");
  });
  taler.onAbsent(() => {
    alert("Taler wallet is not installed");
  });
</script>
```
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": 0, "value": 1},
    "max_fee": {"currency": "EUR", "fraction": 100000, "value": 0},
    "auditors": [{"auditor_pub": "42V6TH91Q83FB846DK1GW3JQ5E8DS273W4..."}],
    "exchanges": [{"master_pub": "1T5FA8VQHMMKBHDMYPRZA2ZFK2S63AKF0Y..."},
                  "url": "https://exchange/"],
    "fulfillment_url": "https://shop/article/42?tid=249&time=14714744",
    "merchant": {"address": "Mailbox 4242", "jurisdiction": "Jersey",
                  "name": "Shop Inc."},
    "merchant_pub": "Y1ZAR5346J3ZTEXJCHQY9NZN78EZ2H8KZK8M0MYTNRJG5N...",
    "products": [{
        "description": "Essay: The GNU Project",
        "price": {"currency": "EUR", "fraction": 0, "value": 1},
        "product_id": 42, "quantity": 1}],
    "pay_deadline": "/Date(1480119270)/",
    "refund_deadline": "/Date(1471522470)/",
    "timestamp": "/Date(1471479270)/",
    "transaction_id": 249960194066269
}
```
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)

\[ s^e \equiv m \mod n \]
Payment processing with Taler

1. Choose goods by navigating to offer URL
2. Send signed digital contract proposal
3. Select Taler payment method (skippable with auto-detection)
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
11. Reload fulfillment URL for delivery
12. Provide product resource
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.
Diffie-Hellman (ECDH)

1. Create private keys $c, t \mod o$
2. Define $C = cG$
3. Define $T = tG$
4. Compute DH
   $$cT = c(tG) = t(cG) = tC$$
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 $f_{new} := FDH(C_{new}), m < n.$
5. Transmit $f'_{new} := f_{new} b_{new}^e$ mod $n$

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

Problem: Owner of $c_{new}$ may differ from owner of $c_{old}$!
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 \( f_{new} := FDH(C_{new}), m < n \).
5. Transmit \( f'_{new} := f_{new} b_{new}^e \mod n \)
... and sign request for change with \( c_{old} \).

Problem: Owner of \( c_{new} \) may differ from owner of \( c_{old} \)!
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 $f_{new} := FDH(C_{new})$
8. Transmit $f_{new}' := f_{new}b_{new}^e$
Cut-and-Choose

$\text{Cut-and-Choose}$

1. $c_{old}\rightarrow t_1\rightarrow \text{Exchange}$
2. $c_{old}\rightarrow t_2\rightarrow \text{Exchange}$
3. $c_{old}\rightarrow t_3\rightarrow \text{Exchange}$

$b_{new,1}, c_{new,1}$

$b_{new,2}, c_{new,2}$

$b_{new,3}, c_{new,3}$
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 ($\gamma = 2$)
Exchange: Blind sign change (RSA)

1. Take $f'_{\text{new},\gamma}$.
2. Compute $s' := f'_{\text{new},\gamma}^{id} \mod n$.
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 $T_\gamma, s := s' b_{new, \gamma}^{-1} \mod n$. 
Customer: Link (threat!)

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

Diagram:

- **Exchange**
  - Link: $T_\gamma$
  - $b_{new,\gamma}$
  - $c_{new,\gamma}$
  - $s$
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.
- To handle protocol aborts.

Transactions via refresh are equivalent to sharing a wallet.
## Competitor comparison

<table>
<thead>
<tr>
<th></th>
<th>Cash</th>
<th>Bitcoin</th>
<th>Zerocoin</th>
<th>Creditcard</th>
<th>GNU Taler</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Online</strong></td>
<td>−−−−</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+++</td>
</tr>
<tr>
<td><strong>Offline</strong></td>
<td>++++</td>
<td>−−</td>
<td>−−−</td>
<td>+</td>
<td>−−</td>
</tr>
<tr>
<td><strong>Trans. cost</strong></td>
<td>+</td>
<td>−−−−</td>
<td>−−−</td>
<td>−</td>
<td>++</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>+</td>
<td>−−−−</td>
<td>−−−</td>
<td>o</td>
<td>++</td>
</tr>
<tr>
<td><strong>Taxation</strong></td>
<td>−</td>
<td>−−</td>
<td>−−−</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td><strong>Payer-anon</strong></td>
<td>++</td>
<td>o</td>
<td>++</td>
<td>−−−</td>
<td>+++</td>
</tr>
<tr>
<td><strong>Payee-anon</strong></td>
<td>++</td>
<td>o</td>
<td>++</td>
<td>−−−</td>
<td>−−−</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>−</td>
<td>o</td>
<td>o</td>
<td>−</td>
<td>++</td>
</tr>
<tr>
<td><strong>Conversion</strong></td>
<td>++++</td>
<td>−−−−</td>
<td>−−−</td>
<td>+++</td>
<td>+++</td>
</tr>
<tr>
<td><strong>Libre</strong></td>
<td>−</td>
<td>++++</td>
<td>+++</td>
<td>−−−</td>
<td>+++</td>
</tr>
</tbody>
</table>
How to support?

- Join: taler@gnu.org, #taler
- Coding & design: https://gnunet.org/bugs/
- Translation: https://git.taler.net/www.git/tree/locale/fr/LC_MESSAGES/messages.po
- Integration: https://docs.taler.net/
- Donations: https://gnunet.org/ev
- Funding: https://taler.net/en/investors.html
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:


Let money facilitate trade; but ensure capital serves society.