What domain of digital communication should we be most concerned about?
Surveillance concerns

- Everybody knows about Internet surveillance.
- But is it that bad?
Surveillance concerns

▶ Everybody knows about Internet surveillance.
▶ But is it **that** bad?
  ▶ You can choose when and where to use the Internet
  ▶ You can anonymously access the Web using Tor
  ▶ You can find open access points that do not require authentication
  ▶ IP packets do not include your precise location or name
  ▶ ISPs typically store this meta data for days, weeks or months
Where is it worse?

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?”
Where is it worse?

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?”
What is worse:

- When you pay by CC, the information includes your name
- When you pay in person with CC, your location is also known
- You often have no alternative payment methods available
- You hardly ever can use someone else’s CC
- Anonymous prepaid cards are difficult to get and expensive
- Payment information is typically stored for at least 6 years
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 sovereingity 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
Do you want to live under total surveillance?
Banking, Surveillance and Physical Security

https://www.youtube.com/watch?v=GyJZViNF2VkJ (4'2019)
Break
GNU Taler

Digital cash, made socially responsible.

 ⟨ Taler ⟩

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**
Design goals for the GNU Taler Payment System

GNU Taler must ...

1. ... be implemented as free software.
2. ... protect the privacy of buyers.
3. ... must 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

- **Customer**
  - Action: withdraw coins
  - Direction: to Exchange

- **Exchange**
  - Action: verify
  - Direction: to Auditor
  - Action: deposit coins
  - Direction: to Merchant

- **Auditor**
  - Action: verify
  - Direction: from Exchange

- **Merchant**
  - Action: spend coins
  - Direction: from Exchange
Architecture of Taler

1. Pay exchange
2. Wire transfer
3. Withdraw coins
4. Spend coins
5. Deposit coins
6. Wire transfer
7. View balance

Exchange

Customer
- Browser/Mobile Wallet

Merchant
- Webshop

Customer’s Bank

Merchant’s Bank

Exchange’s Bank

Database

Fees
Architecture of Taler

Customer's Bank

Customer Wallet extension

Browser

1. pay exchange

2. wire transfer

3. withdraw coins

4. spend coins

5. deposit coins

6. wire transfer

7. view balance

Merchant's Bank

Exchange's Bank

Database

Exchange

⇒ Convenient, taxable, privacy-enhancing, & resource friendly!
Usability of Taler

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

https://demo.taler.net/
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 $(e, n)$.  

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

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

User authentication
Customer Browser
Taler Exchange
Bank Site
Taler (Withdraw coins)
Customer Browser
Customer: Build shopping cart
Merchant: Propose contract (EdDSA)

1. Complete proposal $D$.
2. Send $D$, $EdDSA_m(D)$
1. Receive proposal $D$, $EdDSA_m(D)$.
2. Send $s$, $C$, $EdDSA_c(D)$
Merchant and Exchange: Verify coin (RSA)

\[ s^e \equiv FDH(C) \mod n \]
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

Payer (Shopper) Browser → Payee (Merchant) Site → Payee (Merchant) Site → Taler Exchange
Warranting deposit safety

Exchange has another online signing key \( W = wG \):

Sends \( \text{EdDSA}_w(M, H(D), FDH(C)) \) 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.
Break
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 $c_{old}$.
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 $f'_{new} := f_{new} b_{new}$
Cut-and-Choose

\[ c_{old} \rightarrow t_1 \rightarrow c_{new,1} \]

\[ b_{new,1} \rightarrow c_{new,2} \rightarrow b_{new,2} \rightarrow c_{new,3} \rightarrow b_{new,3} \]

\[ \text{Exchange} \]

\[ \text{transmit} \]
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 $s' := f'^d_{\text{new},\gamma} \mod n$.
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 $T_\gamma$ and

$s := s' b_{new,\gamma}^{-1} \mod n$. 
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 $s := s' b_{new,\gamma}^{-1} \mod n$
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.
Example: The Taler Snack Machine

Integration of a MDB/ICP to Taler gateway.
Implementation of a NFC or QR-Code to Taler wallet interface.

---

1By M. Boss and D. Hofer
Software architecture for the Taler Snack Machine

- Application
  - gnu:net
  - libnfc
  - libqrencode
  - Taler

- Raspbian

- Raspberry Pi

- TCP/IP
  - MDB
  - NFC
  - TFT
User story: Install App on Android

User story: Withdraw e-cash
User story: Use machine!
Summary

- 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:
  - Gives change, can provide refunds
  - Integrates nicely with HTTP, handles network failures
  - High performance
  - Free Software
  - Formal security proofs
Break
Anastasis\textsuperscript{3}

\textsuperscript{3}Based on a BFH Bachelor’s thesis by D. Neufeld and D. Meister
THE PROBLEM

Confidentiality requires only consumer is in control of key material.

Consumers are unable to simultaneously ensure confidentiality and availability of keys.

Cryptographic key-splitting solutions so far are not usable.

European e-money issuers using electronic wallets must:\(^1\)
- Enable consumers to always recover their electronic funds (i.e. if devices are lost)
- Not assume consumers are able to remember or securely preserve key material

\(^1\) According to communication from ECB to Taler Systems SA.
THE PROBLEM

Confidentiality requires only consumer is in control of key material

Consumers are unable to simultaneously ensure confidentiality and availability of keys

Cryptographic key-splitting solutions so far are not usable

European e-money issuers using electronic wallets must:¹
- Enable consumers to always recover their electronic funds (i.e. if devices are lost)
- Not assume consumers are able to remember or securely preserve key material

¹ According to communication from ECB to Taler Systems SA.
WHAT IS ANASTASIS?

ANASTASIS IS A KEY RECOVERY SERVICE.

Users split their secret keys across multiple service providers.

Service providers learn nothing about the user, except possibly some details about how to authenticate the user.

Only the authorized user can recover the key by following standard authentication procedures (SMS TAN, Video-Ident, Security Question, eMail, etc.)
OVERVIEW

End User Application/Wallet Secret

Backup
Upload secret shares

Anastasis Protocol

Anastasis Escrow Providers

Secret Share 1 + postident auth

Secret Share 2+ email auth

Secret Share 3 + SMS auth

Restore
Return secret shares after successful authentication
SIMPLIFIED PROCESS FLOW
STEP 1: RECOVERY INFORMATION
SIMPLIFIED PROCESS FLOW
STEP 2: SPLIT RECOVERY INFORMATION
SIMPLIFIED PROCESS FLOW
STEP 3: USER IDENTIFICATION

IDENTITY
- First name
- Last name
- Social security number

Argon2

User ID
SIMPLIFIED PROCESS FLOW

STEP 4: KEY DERIVATION

User ID

HKDF 1

HKDF 2

K1

K2
SIMPLIFIED PROCESS FLOW
STEP 5: ENCRYPT PARTS
SIMPLIFIED PROCESS FLOW

STEP 6: ADD TRUTH

- H (answer to security question)
- Picture
- Phone number
- E-mail address
SIMPLIFIED PROCESS FLOW
STEP 7: ENCRYPT TRUTH

- K2
- H (answer to security question)
- Picture
- Phone number
- E-mail address
SIMPLIFIED PROCESS FLOW
STEP 8: STORE DATA

- H (answer to security question)
- Picture
- Phone number
- E-mail address
SIMPLIFIED PROCESS FLOW
STEP 9: USER IDENTIFICATION

IDENTITY

- First name
- Last name
- Social security number

Argon2

User ID
SIMPLIFIED PROCESS FLOW
STEP 10: KEY DERIVATION

User ID

HKDF 1

HKDF 2

K1

K2
SIMPLIFIED PROCESS FLOW

STEP 11: PROVIDE KEY

- H (answer to security question)
- Picture
- Phone number
- E-mail address
SIMPLIFIED PROCESS FLOW
STEP 12: DECRYPT TRUTH

PROVIDER A
+ H (answer to security question)

PROVIDER B
+ Picture

PROVIDER C
+ Phone number

PROVIDER D
+ E-mail address
SIMPLIFIED PROCESS FLOW

STEP 13: AUTHENTICATION

- Provide H (answer to security question)
- Provide TAN received by SMS
- Pass video identification
- Provide TAN received by mail
- Provide E-mail address
SIMPLIFIED PROCESS FLOW
STEP 14: RECEIVE PARTS

+ H (answer to security question)

+ Picture

+ Phone number

+ E-mail address
SIMPLIFIED PROCESS FLOW
STEP 15: DECRYPT PARTS
SIMPLIFIED PROCESS FLOW
STEP 16: REASSEMBLY
SIMPLIFICATIONS
THE PREVIOUS ILLUSTRATION MAKES VARIOUS SIMPLIFICATIONS

- Policies to allow more flexible splitting than 4/4
- Recovery document to remember policies and providers
- Distinction between core secret and master secret
- Payment processing
- Provider salts
- Anti-DoS provisions in protocol / request limits
- Versioning
- Liability limitations
UNIQUE SALES PROPOSITIONS (USPS)

1. Distributed trust instead of single point of failure
2. Maximum privacy with respect to authentication data
3. Post-quantum security
4. Ease of use
5. Low cost, scalable cloud-based solution
6. Transparent, Free Software solution
7. Generic API suitable for a range of applications
8. Customers can remain anonymous:
   - Minimizes risk to Anastasis service provider in case database is exposed
   - Makes it more difficult for attackers to fool authentication procedure
9. E-money issuer does not have to protect consumer data against its own staff and can respect consumer privacy
SOCIAL IMPACT OF ANASTASIS

Low-cost solution with minimal environmental impact
 Increases informational self-determination by keeping consumers in control of their data
 Free Software contributes to the global Commons
**OPERATING MODEL**

**REVENUE**
- E-money issuers pay Anastasis UG to offer service to consumers with wallets to satisfy their regulatory requirements (service must exist)
- Wallet operators pay Anastasis UG to assist with technical integration
- Consumers pay Anastasis UG for safekeeping and/or recovery (subscription)

**EXPENSES**
- Development and operations (staff costs)
- Server infrastructure

Presentation | Anastasis UG | 25
Electronic wallets for blockchain wallets and/or fiat currencies

Key store for communication keys, such as OpenPGP or X.509

Identity management solutions

Password managers and disk encryption key material (*)

(*) This is the only entry not yet validated by letters of interest or hard commitments.
MAIN RISKS AND MITIGATIONS

1. IMPLEMENTATION RISK
   Straightforward design simplifies work

2. INFORMATION SECURITY RISK
   Privacy-by-design minimizes loss

3. DISTRIBUTION ON CUSTOMER SIDE
   Strong partners with implementation need

4. CASH FLOW
   Cloud-based deployment with outsourcing of procedures that amortize only at scale

5. USABILITY
   Will work with UX expert
GNU Taler: Next Steps

- Implementation still needs:
  - Escrowed wallet backup and synchronization solution
    ⇒ Anastasis!
  - Finish integration with existing banking system (EBICS, FinTS) ⇒ LibFinTS
  - Code security audit ⇒ Exchange report now public!
  - Improved design and usability
    ⇒ Discussions with UX experts on public list!
  - Internationalization ⇒ https://weblate.taler.net/
  - Porting to more platforms (Web shops, Android, POS, iOS)

- Regulatory approval (withdraw and deposit limits, independent auditor, KYC/AML process validation)
- In discussions with various (commercial and central) banks
Be paid to read advertising, starting with spam
Give welfare without intermediaries taking huge cuts
Forster regional trade via regional currencies
Eliminate corruption by making all income visible
Stop the mining by making crypto-currencies useless for anything but crime