

CPSC 3640/5400: Decentralized Payments, Contracts, and Finance for Humans and AI

Lecture 9b: DeFi advantage: Privacy-preserving asset transfers

Fan Zhang
Yale University
2026 Fall

Administrivia

- Lab 1 grade released
- PS1 released, due in a week
 - Covers lectures on Bitcoin & Ethereum

Section II: DeFi as the new fintech frontier

- Organized into a few categories
- **DeFi Primitives:** What they are, how they are implemented
 - Apps: [Tokens](#), [Tokenization](#), Decentralized Exchanges, Oracles, (re-)staking
 - Infra for fast transactions
- **DeFi advantages:** things not possible in TradFi
 -  [Privacy \[today\]](#), Interoperability, censorship resistance,
- **DeFi risks:** new problems unique to DeFi & mitigation
 - order manipulation, human factors, bugs

The gold standard of privacy-preserving payment system



- This lecture can end here.
- Except that we need **digital money**.

Physical money v.s. digital money



 PayPal

 venmo

 WeChat Pay

 Alipay™

- No third party needed for cash transactions.
- In digital money like PayPal, the company is *always* involved.
- **What about decentralized payments like Bitcoin?**

Bitcoin's privacy

How private is Bitcoin?

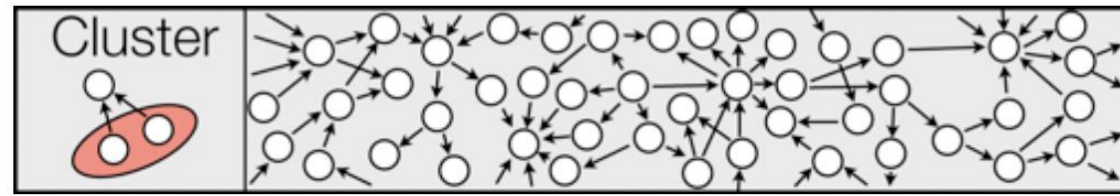


Supposed privacy stems from

- Users never reveal real names
- A user can have *many* PKs and UTXOs
- PKs and UTXOs are “unlinkable”

But we can still link them to some extent

Clustering by inputs



8b6008b2e369499c5c51058f5f09e549c160a84692c00cb97dfa2b4881e9cc27

142Z7VauMVdSV5DADb62DsJ7vvW9ccq18t (30.28851 BTC - Output)
16H9oN1JFXSHEv16X8PLeS77MMF3EKqEiH (30.28851 BTC - Output)
17RHwSeN5Ky8gGwTHCH8j4mZH3eqQNbrav (30.58936 BTC - Output)
13b78oU4oCid4gQw87bvUMUZ1XpnZqwNQ1 (12.9148 BTC - Output)
16BzfEpwF9P6ULmmMcdbdag3m2ZaETGgwYN (29.55 BTC - Output)
1CusinkMvW53WtupuspCMDyi8gZ2sb13zv (30.28851 BTC - Output)
1PXA5YNC2MWYtssfsBTBPMWXW8cDkPuMTB (30.28851 BTC - Output)
1Gg2D33ySPndnSELBnmze1QsmyscSdeGVkX (30.28851 BTC - Output)
1FdPwjg7XJfrEqdQnduusg2K51UwJDACci (30.28851 BTC - Output)
178AKou6Q2741uPqt9FQfB26ZUK16f3yDt (29.36578 BTC - Output)
1LZe2eSEKr8ik6ja8k8YNSh1amR2czmwwe (30 BTC - Output)
16ah8vzFqtnyCPtp57Y55bkXwSot7Bd3ic (29.84 BTC - Output)
1Dn92DXHrPNVH7EMrD5oawDedWdk43Jjkk (29.83951 BTC - Output)
1NvC4vQVbwJUjXWHBHKiGKAmKRMkRKe7gv (30.13 BTC - Output)
14kSwoX2cPkwRtKW5KTWbFGtraYpXrYckW (99.45 BTC - Output)
1AkMojTEiUXUa3f9SNjLZcvLtxY54wyC6n (141.9995 BTC - Output)
1KRMiP4uLy2hm86MRRQv4ghkQthVK6BH6 (29.6 BTC - Output)



1Z9ADFwVMZvgjN3HoNf91XoT2Lpth559F - (Spent)
17iCsx5w55KcNdCRRp9xXFDcMU7btNhqpm - (Unspent)

0.01001 BTC
675 BTC

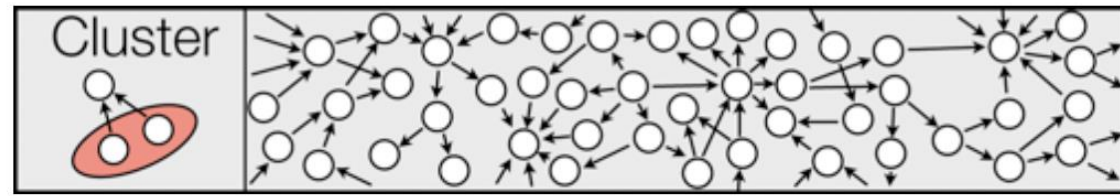
6 Confirmations

675.01001 BTC

Heuristic #1: the same user controls these addresses

Next several slides are taken from Sarah Meiklejohn's talk at IMC'13 (public).

Clustering by change



8b6008b2e369499c5c51058f5f09e549c160a84692c00cb97dfa2b4881e9cc27

142Z7VauMVdSV5DADb62DsJ7wvW9ccq18t (30.28851 BTC - Output)
16H9oN1JFXSHEv16X8PLeS77MMF3EKqEiH (30.28851 BTC - Output)
17RHwSeN5Ky8gGwTHCH8j4mZH3eqQNbrav (30.58936 BTC - Output)
13b78oU4oCid4gQw87bvUMUZ1XpnZqwNQ1 (12.9148 BTC - Output)
16BzfEpwF9P6ULmmMcdbdag3m2ZaETGgwYN (29.55 BTC - Output)
1CusinkMvW53WtupuspCMDyi8gZ2sb13zv (30.28851 BTC - Output)
1PXA5YNC2MWYtssfsBTBPMWXW8cDkPuMTB (30.28851 BTC - Output)
1Gg2D33ySPndnSELBnmze1QsmyscSdeGVkX (30.28851 BTC - Output)
1FdPwjg7XJfrEqdQnduusg2K51UwJDACci (30.28851 BTC - Output)
178AKou6Q2741uPqt9FQfB26ZUK16f3yDt (29.36578 BTC - Output)
1LZe2eSEKr8ik6ja8k8YNSh1amR2czmwwe (30 BTC - Output)
16ah8vzFqtnyCPtp57Y55bkXwSot7Bd3ic (29.84 BTC - Output)
1Dn92DXHrPNVH7EMrD5oawDedWdk43Jjkk (29.83951 BTC - Output)
1NvC4vQVbwJUjXWHBHKiGKAmKRMkRKe7gv (30.13 BTC - Output)
14kSwoX2cPkwRtKW5KTWbFGtraYpXrYckW (99.45 BTC - Output)
1AkMojTEiUXUa3f9SNjLZcvLtxY54wyC6n (141.9995 BTC - Output)
1KRMiP4uLyy2hm86MRRQv4ghkQthVK6BH6 (29.6 BTC - Output)



1Z9ADFwVMZvgjN3HoNf91XoT2Lpth559F - (Spent) 0.01001 BTC
17iCsx5w55KcNdCRRp9xXFDcMU7btNhqpm - (Unspent) 675 BTC

6 Confirmations 675.01001 BTC

Heuristic #2: the same user also controls this address

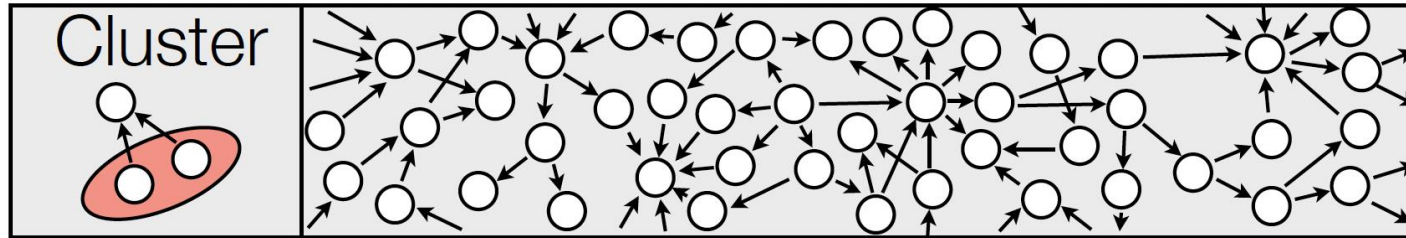
How to deanonymize users?

- Step 1: clustering PKs -> users

How to identify users?

Users can use **arbitrarily many public keys** (pseudonyms); as a result the Bitcoin graph is complicated and has **12 million public keys**

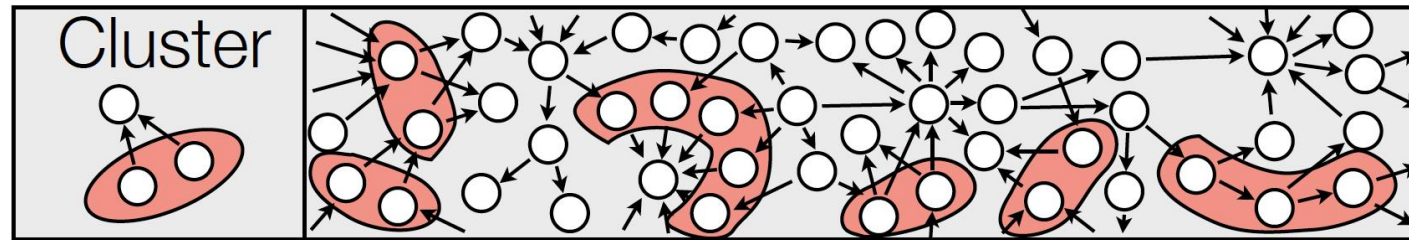
(number in 2013)



How to identify users?

Users can use **arbitrarily many public keys** (pseudonyms); as a result the Bitcoin graph is complicated and has **12 million public keys**

(number in 2013)



Collapse into a more manageable graph of **clusters of public keys** representing distinct entities

How to deanonymize users?

- Step 1: clustering PKs -> users
- Step 2: profile users (e.g., are they in US or China, do they use Coinbase or Kraken?)
- Step 3: wait until user accidentally reveal real-world identity behind **any** public key, e.g.,
 - When user uses an exchange under KYC laws (Know-your-customer),
 - When user posts one of its address online

Commercial blockchain analytics



Elliptic

Founded in London in 2013. Its tools trace funds across blockchains and screen wallets and transactions for links to illicit activity.



Chainalysis

Founded in 2014. Its Reactor software maps transaction flows and links addresses to identified services and entities to support investigations.



TRM Labs

Founded in 2018. Its tools trace funds across blockchains and combine address labels with threat intelligence to investigate financial crime.

Privacy-preserving payment systems

Privacy-preserving payment systems



e-cash

David Chaum introduced e-cash in 1982, later commercialized by DigiCash. Blind signatures unlink bank-issued digital coins from their withdrawal.



Tornado Cash

An Ethereum mixing protocol launched in 2019. Zero-knowledge proofs let users withdraw from a shared pool without revealing their deposit.



Monero

Born from CryptoNote in 2014. Ring signatures, stealth addresses and RingCT provide privacy by default for senders, recipients and amounts.



Zcash

Grew out of Zerocash research and launched in 2016. Shielded transfers use zk-SNARKs to hide payment details while proving validity.

Untraceable Electronic Cash †

(Extended Abstract)

*David Chaum*¹ *Amos Fiat*² *Moni Naor*³



Pioneering cryptographer known for

- e-cash
- Mixnet and DC net (anonymous communication system)
- Verifiable electronic voting

- e-cash design is classic and visionary (in 1988(!))
- Elegantly simple
- Not terribly successful commercially

Coming back to this



- No third party needed for cash transactions.
- In digital money, the company is *always* involved.
- **E-cash asked: How to make digital money that can be verified without a third party?**

Setup of e-Cash

- **Bank** trusted for everything but privacy (i.e., honest but interested in tracking users)
- Bank's "API" includes mint, spend, and redeem.
- **Mint:** User deposits \$ and gets "digital cash"
 - Possibly non-anonymously
- **Spend:** Alice sends "digital cash" to Bob
- **Redeem:** Bob submits "digital cash" to Bank and gets \$

Goals



- Prevent counterfeiting
- Prevent double spending
- Cash-like privacy: prevent bank from tracking

Warmup: e-cash with no privacy

- Suppose Bank has a key pair $(sk_{\text{Bank}}, pk_{\text{Bank}})$
- Assuming we only have \$1 bills (can be extended)
- No privacy yet.

Mint:

- Alice chooses a random serial number **sn**
- deposit \$1 at bank
- **Bank:** produces digital cash (sn, σ) where $\sigma = \text{Sig}(sk_{\text{Bank}}, sn)$

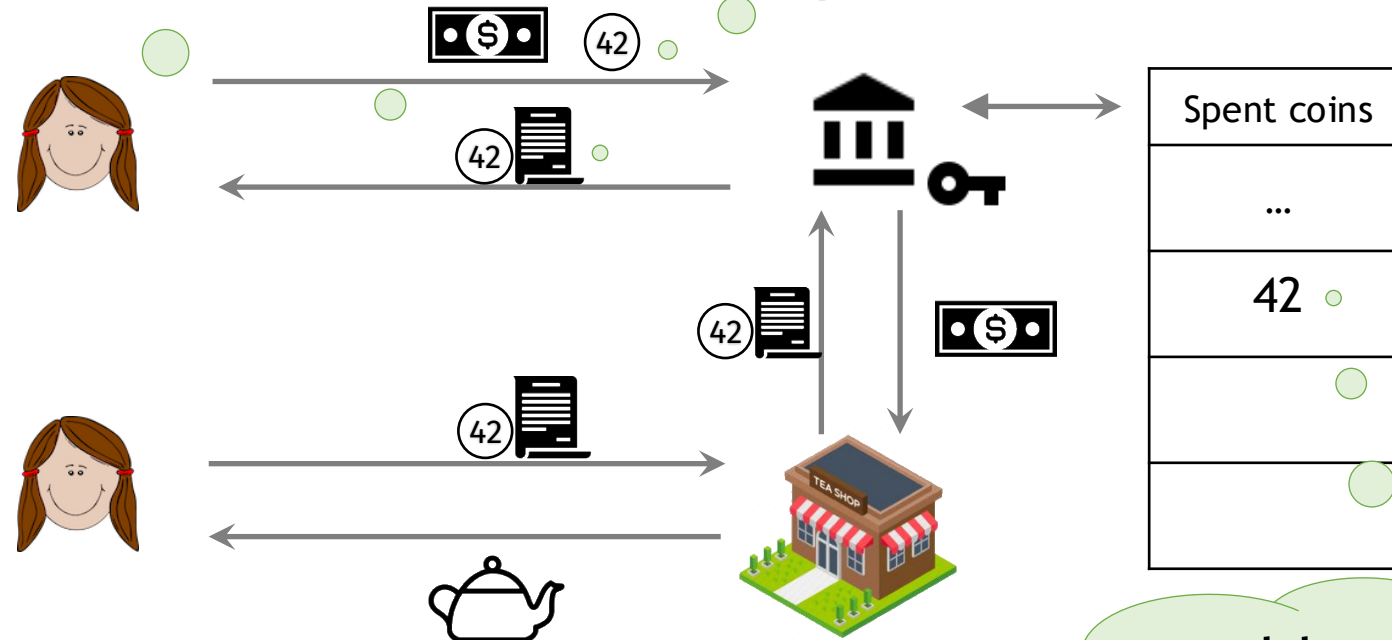
Pay:

- Alice sends (sn, σ) to Bob
- Bob sends the coin to bank
- **Bank:** If sn is not in the list of spent coins, credit Bob \$1
- Now Bob accepts the payment

Warmup: e-cash with no privacy

Counterfeiting
is prevented by
digital sigs

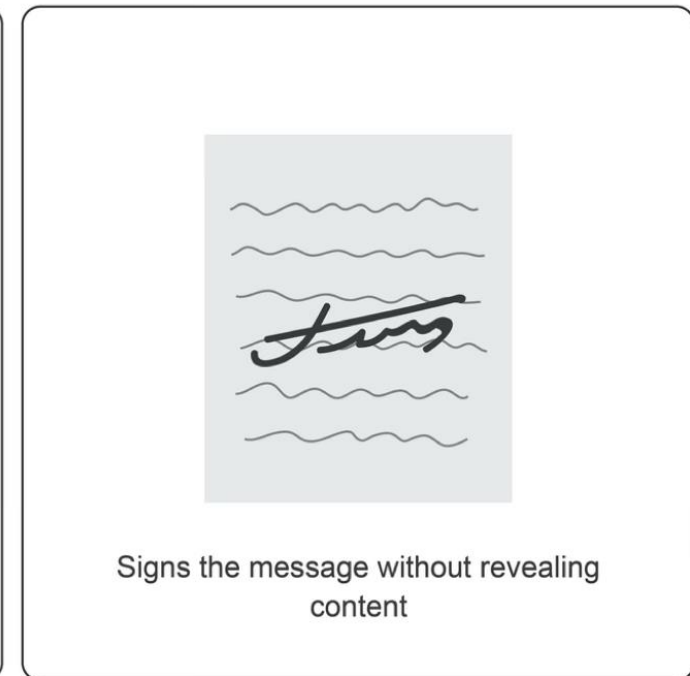
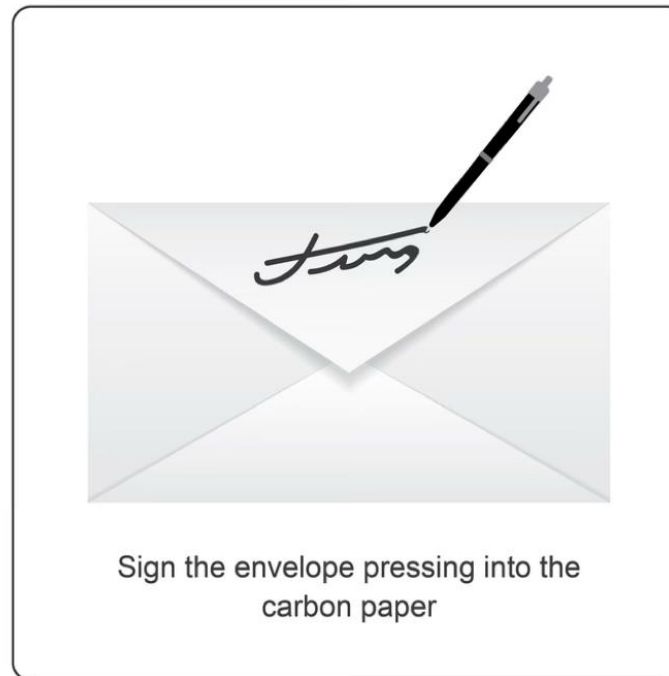
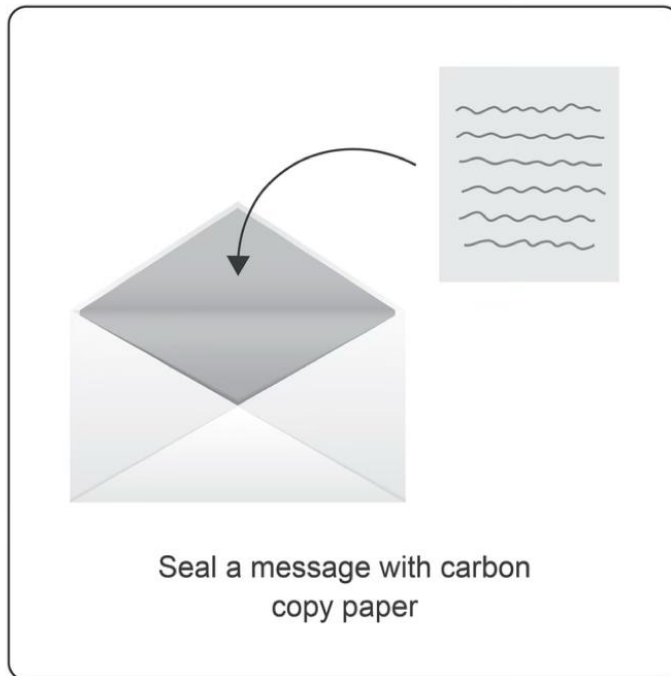
User chosen
serial number



Double spending
is prevented

What about privacy?

- Bank still learns that Alice paid Bob \$10 dollar.
- Idea: Blind Signature (can be realized in several ways...)



E.g., RSA signature ('77)

RSA assumption:

Given y , it is hard to take cube root modulo N , i.e., hard to find x s.t.

$$x^3 = y \pmod{N}$$

Yale'69 (BA)

Rivest, Shamir, and
Adleman Receive
2002 Turing Award



Ronald L. Rivest



Adi Shamir



Leonard M. Adleman

E.g., RSA signature

- KeyGen: sample p, q two random primes
 - PK: $N = pq$
 - SK: (p, q)
- Sign(SK, m)
 - returns $H(m)^{\frac{1}{3}} \bmod N$
- Verify(m, σ , PK)
 - returns $\sigma^3 \stackrel{?}{=} H(m) \bmod N$

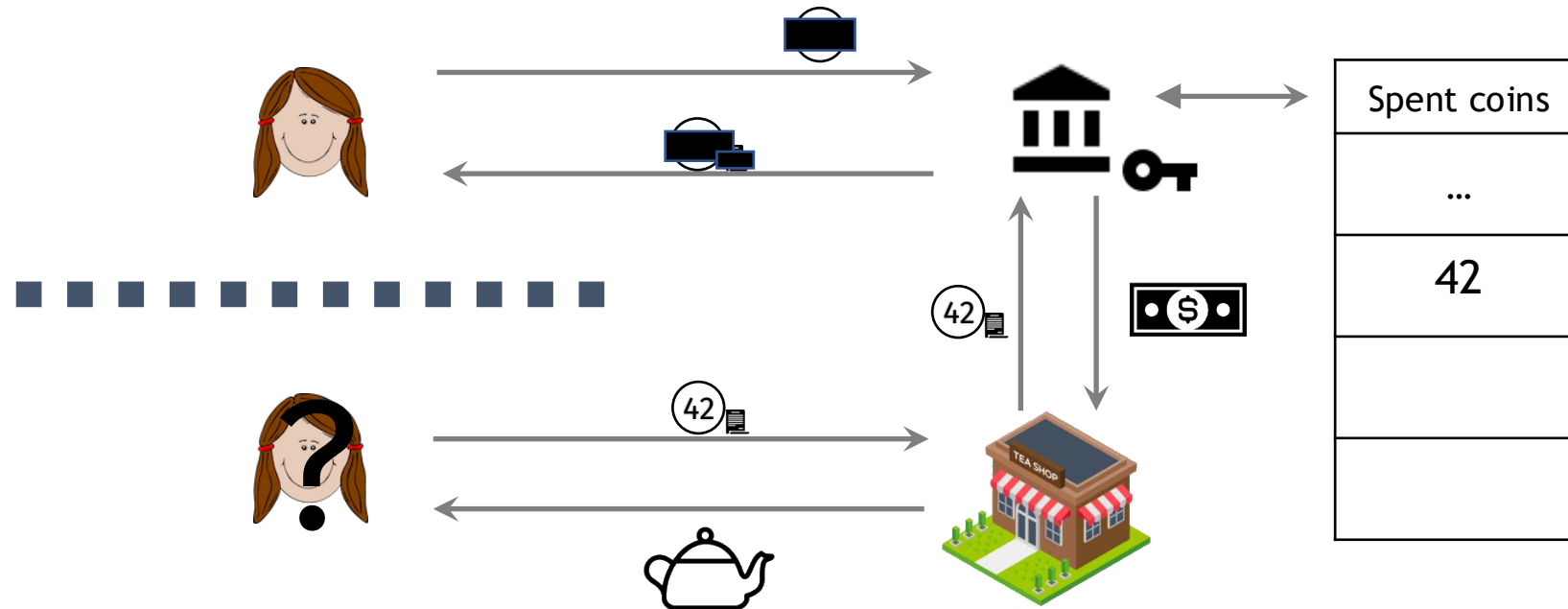
Why H?
O.w. Easy to forge
valid (message,
signature) pair
 $(m = \sigma^3, \sigma)$

Blind Signature based on RSA

- An interactive protocol between **sender** & **signer**.
- **Sender** with message m picks a *random blinding factor* r
 - **Sender** sends $H(m) \times r^3$ to the signer
 - **Signer** sends back $H(m)^{\frac{1}{3}} \times r$
 - **Sender** computes $H(m)^{\frac{1}{3}}$ -- a signature on m
- Dozens of new protocols have been invented since.

e-cash via blind signatures

- Now Bank doesn't learn which coins are spent or who are involved
- Just like cash transactions



Limitation: Double spending

- We saw the **online protocol** for double spending prevention: Merchant calls the bank to make sure sn hasn't been spent for every txn.
- This is undesirable in practice: The merchant may go offline (e.g., at a farmer's market with poor cellular coverage)
- Can this be done **offline**?
 - i.e., without bank in the loop

The 2nd nice ideas in e-cash

- Encode user identifier (e.g., account number) in coins
- In a way that the identifier is **hidden** *unless* a user spend a coin more than once
- $\text{coin} = \prod_{i=1}^{k/2} \sigma(x_i, y_i)$
- $\sigma(m)$: the bank's signature on m
- k : security parameters affects the chance of a cheater succeeding
- Basically, product of $k/2$ signatures

Double spend prevention in e-cash

- $\text{coin} = \prod_{i=1}^{k/2} \sigma(x_i, y_i)$
- $x_i = H(a_i)$ where a_i is a **random serial number** chosen by Alice
- $y_i = H(a_i + ID)$ where ID is Alice's **identifier**
- Goal: when Alice double-spends, both a_i and $a_i + ID$ will be revealed to bank, then the bank can compute ID and catch Alice cheating.

- $x_i = H(a_i)$ where a_i is a random serial number chosen by Alice
- $y_i = H(a_i \oplus ID)$ where ID is Alice's identification info

Bob (merchant)'s protocol

1. Bob receives coin = $\prod_{i=1}^{k/2} \sigma(x_i, y_i)$
2. Bob flips $k/2$ coins and asks Alice to reveal pre-image of x_i or y_i

0	1	2	...	$k/2$
				
1	1	0		1
(a_0, y_0)	(a_1, y_1)	$(x_2, a_2 \oplus ID)$		$(a_{k/2}, y_{k/2})$

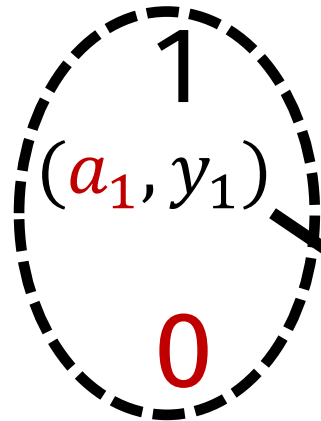
3. Verify coin is consistent with revealed information.
4. If so, Bob accepts the payment (without talking to Bank)!

Double spend prevention in e-cash

0	1	2	...	$k/2$
				

Bob

1
 (a_0, y_0)



1
 (a_1, y_1)

0
 $(x_2, a_2 \oplus ID)$

1
 $(a_{K/2}, y_{K/2})$

Charlie

1

0
 $(x_1, a_1 \oplus ID)$

0

1

- When Bob and Charlie deposit the same coins, Bank catches Alice for double spending.

Not done yet

- $x_i = H(a_i)$ where a_i is a random serial number chosen by Alice
- $y_i = H(a_i \oplus ID)$ where ID is Alice's identification info

- With blind signature, how can bank make sure Alice has embedded the correct ID?
- Bank asks Alice to open **a random half**.
 - Verify if any numbers are ill-formed.
 - Sign the rest.
- Bank catches Alice with overwhelming probability. Yet Alice will can use the rest w/ privacy.



Cut and choose

The commercialization of e-cash

- DigiCash Inc. was founded in 1989 on the idea of e-cash
- In USA: only one bank (the Mark Twain bank in Saint Louis, MO) implemented e-cash, and attracted 5,000 users in a three-year trial.
- Went bankrupt in 1998, despite flourishing electronic commerce, but with credit cards as the “currency of choice”
- David Chaum opined then *“As the Web grew, the average level of sophistication of users dropped. It was hard to explain the importance of privacy to them”*

Problem for innovation

- We trust banks (maybe because we have to)
- But unlikely to trust a *new* bank
- Even less likely to trust a new bank with "private" "digital" "money"
- **Finding trusted/centralized parties is a major obstacle**

- E-cash requires trusted bank, which is perhaps hard to find.
- Bitcoin and cryptocurrencies) doesn't require trusted bank, but also isn't private.
- **How to make Bitcoin more private?**
 - CoinJoin, Monero, Zerocash
- **How to make Ethereum (or ERC-20) more private?**
 - It's in fact harder (we will see Tornado Cash)