- Welcome (to 6.892 :) Shrobe Pub Leads
- SIGN UP Sheet
- What is a Pub bgan. ?
- Why good?
  1. Comm K Neha
  2. Comm T Silvio
- Expectations low / high Silvio
- Grades: Impress us! Occasional readings Neha
- ~~Project~~ New Application * FORMALIZING
- Mini Syllabus — Neha (Bitcoin ~~Lattices~~ Analysis) / Silvio Silvio
- Introduction Secret Silvio
- Hand to Neha

Lecture 1. Outline

Goal: digital money.

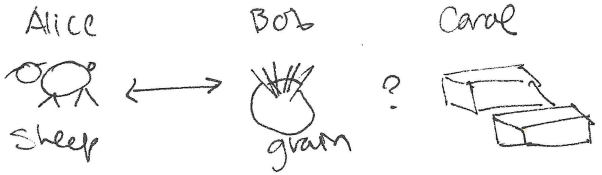
Primitives: Hash functions
Commitments
Hash chain
~~Merkle trees~~
Public key Crypto (ex: RSA)
Digital Signatures

Simple e-cash
Blinded Signatures
Chaumian e-cash
Bitcoin

① Money.

~~Features of cash~~

First barter.



Money made things more efficient.
- Credit - Notes - Shells
- Receipts - Coins - Beads

~~Solves~~ problem of coordination
Moves

Features of real cash → can't double spend
- Integrity - "Private"
- ~~Security~~
- P2P

How to design digital money.

② Digital Money.

Bank transfer
Credit Cards
Paypal / Venmo
} Not really money mediated

ASK CLASS for properties

What is "real" money?
Issues fiat currency

Digital currency?

③

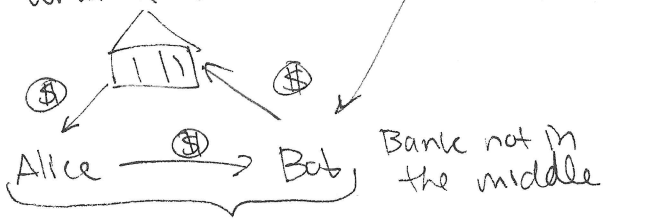
Alice	~~$5~~	$3
Bob	~~$10~~	$12
Carol	$7	$7

A → B $2

Draw like this

Pros: real money!
Cons: - CB has to be online - bank can deny ~~transfer~~ transfer
- No privacy - not P2P

④ What if...



How to construct such a thing....
2 critical tools of cryptography

⑤ Hash Functions

Goal: Take input of arbitrary length $H(x) = h$
produce fixed size output. "Fingerprint"
Collision resistance: very hard to find $x \neq y$ where $H(x) = H(y)$.

But note there are many

SHA256: $\{0,1\}^{<2^{64}} \rightarrow \{0,1\}^{256}$

Model as a Random Oracle. What does that mean?

Table: 2^{64} rows

000....0	0110..01
000....1	1001..10
⋮	
0110...0	inevitable
⋮	

Inevitable because $2^{64} >> 256$.

Take all of Wikipedia, produce hash h which (practically) uniquely identifies it!

Tweak x to produce same h VERY HARD.

A hash is a commitment to the original data.

Also provides Privacy!

Imagine proof for P=NP: [Proof]

But... I'm about to go on vacation. Want to be first without releasing it.

$H($ [Proof] $) = h$.

Publish h → can come back later + reveal

[Proof].

How to timestamp h?

② Publish on twitter → Blackmail Jack Dorsey

① NYTimes (slow!)

⑥ Hash chains

Would be really great if we had a tamper-proof way of timestamping data!

⑦ Public-key Crypto

public key: pk ← ~~encrypt~~

secret key: sk ← decrypt

1 is the "remainder"

$\exists k$ s.t. $ed = k \cdot (p-1)(q-1) + 1$

Ex: RSA.

Choose 2 large primes p, q. $N = p \cdot q$.

choose e, d s.t. $ed = 1 \bmod (p-1)(q-1)$

pk: (e, N) Encrypt $(m) = m^e \bmod N$

sk: (d, N) Decrypt $(c) = c^d \bmod N$

$(m^e)^d \bmod N = m^{ed} \bmod N = m^{k(p-1)(q-1)+1} \bmod N$

$= (m^{(p-1)(q-1)})^k \cdot m \bmod N = 1^k \cdot m \bmod N = m$ (Euler)

** This exact scheme not used in practice!

⑧ Digital Signatures.

Alice → Bob. How does Bob know it's REALLY Alice?

Can use this to "sign" data! (not just about encryption).

Idea here: digitally sign (like handwritten sig).

1) key-gen$(u) \to (pk, sk)$ Note: m is public.

2) sign$(m, sk) \to sig(m)$

3) verify$(c, \overset{H(m)}{m}, pk) \to$ yes or no.

Properties:

- Legitimate sigs are always verified

  verify$(sig(m)^{sk}, m, pk) \to$ yes

- Hard to forge. VERY HARD to compute ~~sig(bs)~~ ~~without~~ s s.t. verify$(s, m, pk) \to$ yes w/out knowing sk.

Ex: RSA

Alice → Bank "send Bob $2".

Alice: (pk, sk)

pk: (e, N)

sk: (d, N)

~~Encrypt~~ Sign $(m) = m^d \bmod N = c$

Bank: verify$(c, m, (e, N)) = c^e \bmod N \to$ ~~m~~ de → m

$m \overset{?}{=} m$

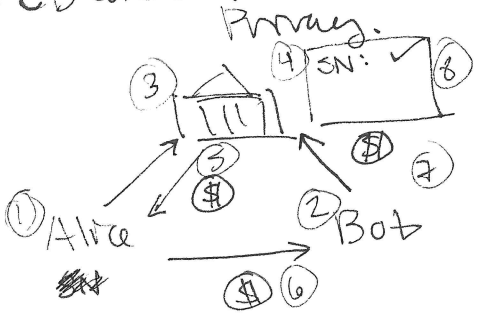
Q: ok that Alice used d?

⑨ Simple e-cash

CB cons: Bank online, Privacy.

In the middle of payment Bank generates ④ SN: uniq random #

$: $sig_{bank}(SN)$



Bob can verify ⑦

verify($, SN, PK_{bank}) → yes

copies for free

Pros: real money! payment P2P

Cons: privacy

Bank can censor

bank online to verify (double spends)

(10) Blind Signatures

Alice can "blind" the message to the bank.

Bank: pk: (e, N), sk: (d, N)

$m \to b(m)$.

Ex: Alice chooses k^e. Sends $k^e m \to$ Bank.

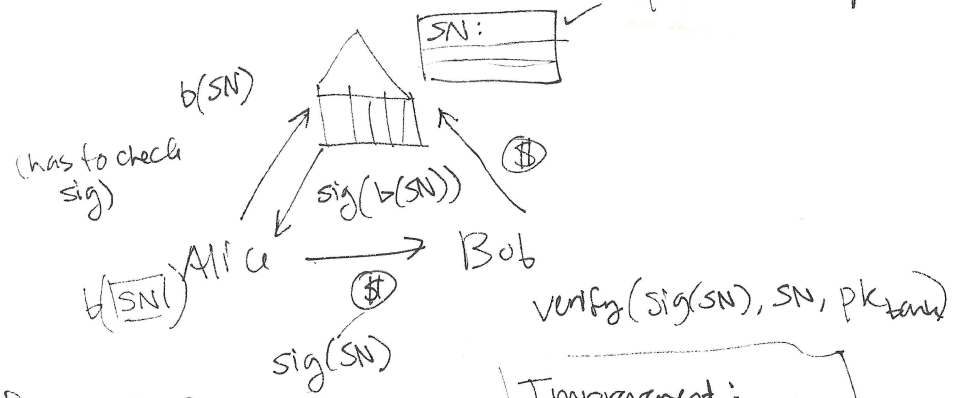
Bank signs $(k^e m)^d \bmod N \to k^{ed} m^d = k m^d$.

Alice "unblinds" signature w/ $\frac{1}{k}$

$\frac{1}{k} \cdot k \cdot m^d = m^d \leftarrow$ valid Bank signature.

(11) Add Privacy

Chaumian e-cash: Same picture, except.



verify(sig(SN), SN, pk_{bank})

Pros: Real Money!
P2P
Privacy (sort of)

Cons: ~~Bank online to verify~~
Bank can censor.

Improvement:
Detect Double Spends!
(after the fact).

TABLE OF DIGITAL MONEY

	No Double Spend	Real	P2P	Tolerate Offline	Privacy	Central ~~Control~~
Central Bank		✓	×	×	×	✓
Simple e-cash		✓	✓	?	×	×
Chaumian e-cash		✓	✓	w/ double spend detection	✓	×
(Credit cards?)		×	~~✓~~ ×	8.	×	×

ENTER BITCOIN.

Recap → Silvio.