Admin:

Today:

MIT Bitcoin Expo March 7-8
Bitcoin Club every Wed 8pm in 8-205

E-cash & Bitcoin

- Money - basics
- Electronic checks
- Signed coin ID
- Identities
- (Public) Ledgers
- Bitcoin

Project ideas:

- How to make bitcoin work in solar system (or galaxy)?
  [Distributed consensus with communication delays?]
- What if 71 party is using Eyal/Siver strategy?
  (Ref "Majority is not Enough: Bitcoin Mining is Vulnerable")
- Study bitcoin alternatives ("alt-coins") & improvements
- Study bitcoin scalability
Admin:
- Talks start Monday
- Zeldovich talks Wed 5/11 on Stuxnet

Today: "Electronic Cash"
- Basics: Atoms/Bit, Tokens/Accumb
- Electronic checks
- Properties
- Double-spending
- Coins & Peppervin
- Anonymity
"Electronic Money"

- What properties should it have?
- " " can " "?

Atoms vs Bits

- What can "possessing value" (money) mean?
- How can we transfer value?

Easy to answer if we use (gold) atoms to represent value:
- gold atoms are hard to make
- only one person at a time can "own" an atom

Things get complicated if we want to use bits:
- easy to generate bits
- bits can be copied \Rightarrow \text{double-spending becomes a problem!}

(Token-based)

Possession-based vs Account-based methods

- In a possession-based method, owning the representation \equiv owning the value

- In an account-based method, there is usually some TTP who "maintains accounts" (e.g. a "bank"); transactions cause value to be shifted from one account to another.
- Most "bit-based" methods are account-based.
Simple example: Electronic checks

- Account-based: Bank has PK_B, SK_B
- User has PK_u, SK_u, \text{cert on (}u, PK_u\text{)} by bank
- Check = \begin{align*}
&\text{cert (on PK_u, signed by SK_B)} \\
&\text{sign (SK_u, "Pay Bob $100, date, serial #")}
\end{align*}
- Bank deposits check just once (using serial #)
- Usual problem of overdrawn acct (bad check)
- Bank knows exact details: payer, payee, amt, date
- Merchant " " "

This works.

What else is possible?

Can we make payments more like cash?
Desirable (?) Properties

- Non-forgeable (prevent fraud, inflation)
- Not double-spendable
- Reliability: can "back up" your $
- Exclusive ownership
- Transferability: A can pay B
- Transitivity: B can use A's payment to pay C
- Variable denominations
- Divisibility & combiniability
- Efficiency (esp. for small costs)
- On-line vs. off-line transactions
- Scalability
- Anonymity
- Security
- Conversion to "ordinary" money

traditional:
- medium of exchange
- store of value
- unit of account
  - "unit of measure"
Double-spending
- essentially a "replay attack"
- if you can backup your $, then "restore" gives you your spent money back!?
- prevention seems really tough (unless you use atoms)
- detection requires convergence of spending records
  (e.g. at bank) and large databases (?) ledger
- even if you can detect double-spending - what do you do?
  - roll back/deny transaction
    (2nd merchant to get same electronic coin can't deposit it)
- punishing perpetrator may be impossible if we have (true) anonymity: payer is not identifiable
- furthermore: is payer or payee the culprit?
  (can merchant "frame" consumer?)
- deterrence may be hard... how to punish
  (pay fine from account?)
Some approaches:

**Signed coin ID**

Bank (TTP)
Alice (payer)
Bob (payee)

3 protocols to support:
1. withdrawal/authorization
   - Alice becomes "able to pay"
   - (e.g. coin issuance in check scheme)
2. payment
3. deposit

1. withdrawal:
   - Bank gives Alice \( R, \text{sign}(SK_B, R) \triangleq \text{unforgeable object!} \)
   - \( R \) is coin ID
   - Bank keeps \( R \) in database of unspent coins
   - Bank debits Alice's account for withdrawal

2. payment:
   - Alice gives coin to Bob; Bob checks Bank sig

3. deposit:
   - Bob gives coin to Bank; Bank checks sig & \( R \) in DB
   - Flags \( R \) as "spent"
Peppercoin (Mizuki & Rivest)

- "probabilistic payments":
  - paying 10¢ = paying $10 with probability 1/100
    \[ \text{(micropayment)} \]
  - paying 10¢ = paying $10 with probability 1/100
    \[ \text{(macropayment, sometimes)} \]
- based on electronic checks method

- Alice pays Bob 10¢ as follows:
  - she gives Bob electronic check for $10 that contains condition:
    - "This check valid if and only if \( E \) is true" (where \( E \) holds with probability 1/100)
  - Bob must be able to test if \( E \) is true
    - if so, he can deposit check
    - if not, he throws check away (but gives Alice her purchase)
  - he gets paid correctly on the average (law of large numbers)
  - Alice should not be able to tell if \( E \) is true when she writes check (else she can filter checks...)
- Bank should be able to tell if \( E \) is true (so "bad checks" when \( E \) is false, don't get deposited.
Bitcoin:

- ID's are PK's (used for signing transactions) ECDSA
- Public ledger records all transactions
  (account-based) PK=acct name
- money created by/h for those maintaining ledger ("miners")
  no other way to create money
  (digress: discuss need for loans!)
- transaction detail:
  \[
  \text{from acct: } \rightarrow \rightarrow \quad \text{time} \quad \text{to acct: } \rightarrow \rightarrow \quad \begin{cases} \text{valid if from acct have enough value} \\
  \text{(S, ams)} \end{cases}
  \]
- ledger detail: "block chain"

\[
\begin{array}{c}
\text{Block} \\
\text{Prev hash} \\
T_i, T_{i+1}, \ldots \\
\end{array} \rightarrow \begin{array}{c}
\text{Block} \\
\text{Prev hash} \\
\text{None} \\
\end{array}
\]

Block valid if hash (block) begins with enough zeros &Tx's valid

- need to search nonce to find one that works (hish hash col)
- solved blocks are "published"
- longest chain wins
- difficulty level adapts to
difficulty level adapts to
- yield 2 or 3 solution every 10 minutes
- wait for enough (6) confirmations

no TTP!
view blockchain info

ledger is transaction log

implies account balances

Verifying a transaction involves checking account balances

Issues:

- Scalability?
  - Can it do e.g. 4000 tps (like Visa?)
  - Blocks may reach 1/4 GB, block now is only 300 KB or so
  - ECDSA signature verification main computational need

- Phasing out of miner's fee? (in 2140 or so ... 21 M BTC) (reward halving every 4 years) as BTC/block now

- Act-based but no loans!
  (Can't create money by giving a loan, as you can with current monetary system)

  Like "gold standard" in terms of "in creation"

- Protocol vulnerabilities:
  - large pool of miners (>50%)
  - "majority is not enough" (Eyal/Sirer)