On Digital Monies

July 2, 2011

Digital Payment vs. Digital Money

- Payment System $<$ Money

Digital Payment vs. Digital Money

- Payment System < Money
- Functions of money (texbook)
  1. Payment
  2. Measure of value
  3. Saving / investment (store of value)

Digital Payment vs. Digital Money

- Payment System < Money
- Functions of money (texbook)
  1. Payment
  2. Measure of value
  3. Saving / investment (store of value)
- Why bother with the “digital” part?

Digital Payment vs. Digital Money

- Payment System $<$ Money
- Functions of money (texbook)
  1. Payment
  2. Measure of value
  3. Saving / investment (store of value)
- Why bother with the “digital” part?
- **Merry Crisis!**

Digital Monies: Past, Present and Future

1. DigiCash
    - David Chaum, 1990
    - Emphasis on untraceability
2. WebMoney
    - WM Transfer Ltd., 1997
    - Emphasis on finality of transactions
3. BitCoin
    - Satoshi Nakamoto, 2009 (2007)
    - Emphasis on guaranteed scarcity
4. ePoint
    - D. N. & friends, 2007 (2005)
    - Emphasis on issuer transparency

Technical Challenge #1: Double Spending

- **DigiCash**
  Reactive security measures
- **WebMoney**
  Proactive: centralized account-keeping
- **BitCoin**
  Long-term proactive: approx. 1h confirmation time
- **ePoint** (future)
  All of the above. :-)

Economic Challenge #1: Acceptance

- **DigiCash**
  Backing by banking system.
- **WebMoney**
  Backing by escrow services and contractual acceptance.
- **BitCoin**
  Purely speculative.
- **ePoint** (future)
  Backing by securitized debt.

Legal Challenge #1: State Monopoly

- **DigiCash**
  Banking license
- **WebMoney**
  Ownership & purchase certificate
- **BitCoin**
  Outside of state jurisdiction
- **ePoint** (future)
  Purchase certificate

Architectural considerations

- Open source infrastructure; the only secrets are keys
- Most of the work is done by paranoid clients
  Paranoid users only need to trust their client sw/hw
- Weakly coupled server nodes provide a *sufficiently consistent* database of transactions and balances
- Server nodes are not trusted, but rewarded
- There is *one* transaction type: transfer of a given amount of funds from one account to another.
- Issuing is simply incurring a negative balance.

Implementation details

- Transactions are split into two: *give* transactions signed by the payer and *take* transactions signed by the recipient.
- Partial balances are calculated by clients and checked by both clients and server nodes.
- Transactions refer to earlier transactions by hash values, checked by all parties
- References are included to
  - related transactions
  - very recent transactions
  - random transactions in the past
- Volutary transaction fees refer to the corresponidng transactions

User experience

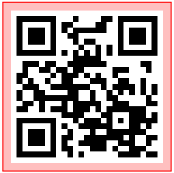
- Naïve transactions are possible
- Peer-to-peer payment over any channel
  - by cellphone
  - by email
  - over the web
  - in online chat
  - by handing over pieces of paper
  - ... even verbally (over the phone or in person)

User experience

- Naïve transactions are possible
- Peer-to-peer payment over any channel
  - by cellphone
  - by email
  - over the web
  - in online chat
  - by handing over pieces of paper
  - ... even verbally (over the phone or in person)
- Cash-like behavior
  - locally stored tokens vs. centrally kept accounts
  - no identification (hence no risk of identity theft)
  - some measure of privacy

Payment tokens: rands

- Each payment token is a short *rand*om code called “**rand**”.
- Rands have many faces:
  - **textual** representation
    `vTOe2RutvrF8`
  - **QR code**

    
  - **paper** rands
  - **electronic** representation

Thank you for your attention!

`www.epointsystem.org`