Northeastern

Electronic Cash

Creence Lin
December 10, 2009

Outline

- Motivation for electronic cash payment system
  - Pros and cons of traditional cash
  - Pros and cons of payments by instruction
  - Properties of an ideal payment system
- Model of Electronic Cash
  - How participants authenticate each other
    - About Authentication Methods Not Studied in Class
  - Representing electronic cash itself
- Fraud Detection and Prevention
- Privacy Concerns
  - Blinding

Pros and cons of traditional cash

Pros

- Privacy
- High acceptability
- Person to person payments without bank involvement
- Instantaneous use

Cons

- May not always have exact change
- Costly to produce and handle
- No loss theft protection
- Criminal activity
- Requirement for physical proximity of payer and payee

Pros and cons of payments by instruction (credit, debit cards, etc.)

Pros

- Transportable and storable with convenience and low cost
- Less risks of theft and loss since actual value resides at banks
- No requirement for physical proximity of payer and payee

Cons

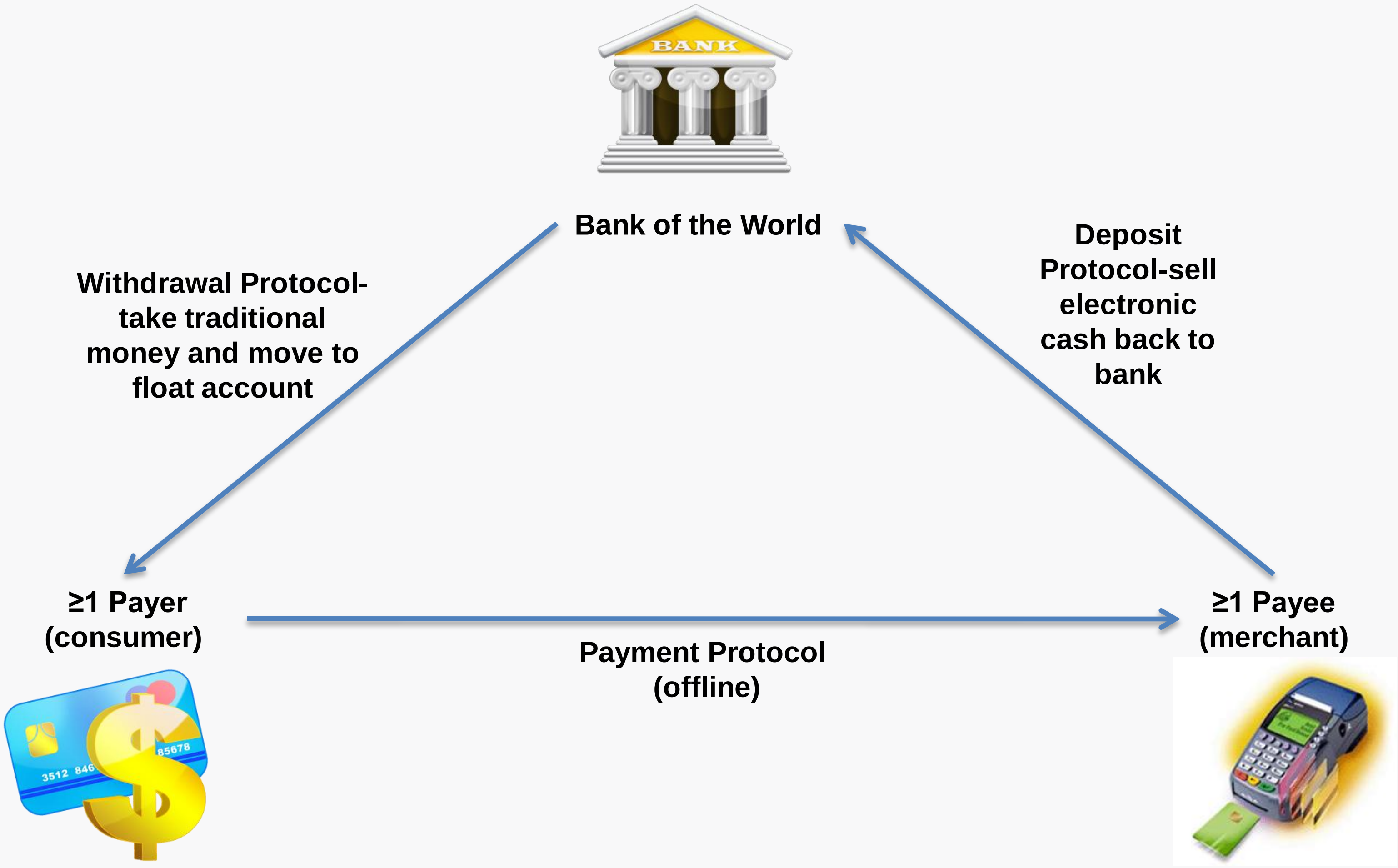
- Difficult to ensure authenticity without chip cards and cryptographic authentication
- Chip cards do not address online authentication techniques
- Online verification and processing is expensive for payees
- Lack of privacy

What would we want in a payment system?

- spontaneous payments including offline payment verification
- no bank involvement –payment from person to person
- privacy
- hard to forge
- transportable and storable with convenience and low cost
- cheap to replace when worn out
- payable without many bank notes
- able to trace criminal activity
- Loss theft protection
- No requirement for physical proximity of payer and payee
- Low processing and handling costs
- High acceptability
- Cost effective for low value purchases

The best of both traditional cash and payments by instruction.

Electronic Cash Model



Need at least 1 computing device per participant (payer, payee, etc.)

Authentication

- Paying device computes a response to receiving device based on its secret key and a challenge
  - Design challenge response protocols can be conventional (MACs) or based on public key cryptography, zero knowledge authentication, DES, RSA, digital signatures, Lamport signatures, matrix based signatures, tree authentication etc.
- Dynamic authentication prevents replay attacks
- Diversified keys with digital signatures on the paying device's ID number provide additional security over a system wide secret key
- Can also use session key.

Matrix Based Signatures

- Matrix based signature signs messages of int($\log_2 r^c$) bits expanded into an r-ary representation.

Uses matrices generated from c random numbers and a hash function applied to elements of the matrices with r rows.

$$\begin{pmatrix} f^{r-1}(s_{1j}) & f^{r-1}(s_{2j}) & \cdots & f^{r-1}(s_{cj}) \\ \vdots & \vdots & \vdots & \vdots \\ f(s_{1j}) & f(s_{2j}) & \cdots & f(s_{cj}) \\ s_{1j} & s_{2j} & \cdots & s_{cj} \end{pmatrix}$$

Hash functions

Random numbers

- A better signature storage that verifies by computing top rows of message and control matrix
- Example: Sign the 6-bit message 010011, with 4-ary expansion 103

Message matrix ($j=1$) is based on rows

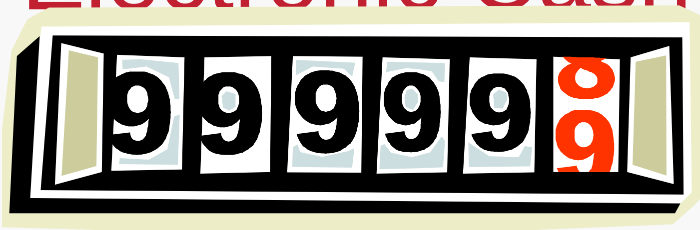
Control matrix ($j=2$) based on hash exponent?

$$\begin{matrix} \text{row } 0 \\ \text{row } 1 \\ \text{row } 2 \\ \text{row } 3 \end{matrix} \begin{pmatrix} f^3(s_{11}) & f^3(s_{21}) & f^3(s_{31}) \\ f^2(s_{11}) & f^2(s_{21}) & f(s_{31}) \\ f(s_{11}) & f(s_{21}) & f(s_{31}) \\ s_{11} & s_{21} & s_{31} \end{pmatrix} \quad \begin{pmatrix} f^3(s_{12}) & f^3(s_{22}) & f^3(s_{32}) \\ f^2(s_{12}) & f^2(s_{22}) & f(s_{32}) \\ f^1(s_{12}) & f^1(s_{22}) & f^1(s_{32}) \\ s_{12} & s_{22} & s_{32} \end{pmatrix}$$

Tree Authentication

- Tree Authentication
  - One time signature scheme using single public key (Root node)
  - To compute a digital signature on a message, the paying device uses a leaf of the tree that has not been used before

Electronic Cash Representations



Register Based Cash

- Amount of electronic cash maintained in a chip register by means of the value of a counter
- Minimal storage space
- If tampered, counters can be bypassed or updated without bank authorization
- Paying device must authenticate amount transferred
- Depending on authentication type, receiving device can store as register based cash, electronic check for deposit, or electronic coins for spending

Electronic Coins

- Cryptographic tokens assigned a fixed denomination and currency (message) that are digitally signed by the bank
- Verifiable solely by using the signature public key of the bank
- Storage space must be allocated
- Two part form has (message, signature) pair, but subject to coin theft or copying.
- Three part form has secret key used to compute the digital signature from challenge message of receiving device, public key, certificate on public key.
  - Receiving device cannot reuse coins -must deposit coin since it does not know the secret key, but can verify payment using public key.

Fraud Detection and Prevention

- Certain electronic cash models are better for fraud detection:
  - The digitally signed message of the paying device ties the received value for electronic checks and coins transactions. (Coins more difficult to overspend)
  - System wide secret keys make it difficult to trace compromised devices
  - Three part coins have secret keys that are stored solely in the paying device making fraud easier to trace
  - Including intended payee in signature prevents man in middle attack
- Banks should take common sense precautions:
  - Tracing to a device does not mean owner is guilty, but compromised devices, checks, and coins should still be blacklisted
  - Master and certification keys should be refreshed on a regular basis indicated by expiration dates.
  - Devices should be able to resend last message in event of interruption
  - Control account access, software should be secure, etc.

Privacy

- Goal: Balance untraceability without encouraging criminal activity

Bad idea: Relax standards

- Collect only aggregate data, have anonymous devices or accounts
  - cannot trace criminal activity
- In the case of anonymous accounts:
  1. is not truly private since transactions can be linked to a single account/device
  2. sometimes illegal

Better idea: Blinding

- A receiver can obtain digitally signed information that remains hidden from the issuer.
- Obtains the same signature as if you had not hidden the information

Blinding

Simplified blinding example: signer doesn't know message

Regular RSA signatures

- A receiver gives a message *m* to the signer who signs with the private decryption key $d=1/e$.
  - $m^{1/e}$ mod n

Blind RSA signatures

- The message *m* is multiplied by r^e where *r* is a random nonzero integer and *e* is the public encryption key
- A receiver gives a message $r^e m$ to the signer who signs with the private decryption key $d=1/e$.
  - $(r^e m)^{1/e} \bmod n = rm^{1/e} \bmod n$
- The receiver multiplies result by r^{-1}
  - $r^{-1} rm^{1/e} \bmod n = m^{1/e} \bmod n$

- This example shows how the receiver can get the same signature from a signer without revealing the message.
- The receiver (payer) could be asking the signer (bank) to sign a fraudulent message and the bank will not be able to trace and blacklist the device.
- One show blinding, a variation of the concept, traces double spent coins without sacrificing untraceability

Cut and Choose Blinding

Type of one show blinding - variant of blind RSA signature

Blind RSA signatures

- The message *m* is multiplied by r^e where *r* is a random nonzero integer and *e* is the public encryption key
- A receiver gives a message $r^e m$ to the signer
- who signs with the private decryption key $d=1/e$.
  - $(r^e m)^{1/e}$ mod $n = rm^{1/e}$ mod n
- The receiver multiplies result by r^{-1}
  - $r^{-1} rm^{1/e}$ mod $n = m^{1/e}$ mod n

Cut and choose

- For each of *k* *i*'s the one way function *f* (instead of message m) with inputs a_i, c_i, d_i is multiplied by r_i^e where a_i, c_i, d_i, r_i are random nonzero integers and *e* is the public encryption key
- A receiver gives all the messages $r_i^e f(a_i, c_i, d_i)$ to the signer.
- The bank chooses a subset of the *i*'s to be revealed and the signer will reveal a_i, c_i, d_i, r_i for those *i*'s.
- If the revealed candidates have been received properly, the signer signs the remaining the product of the remaining *i*'s with the private decryption key $d=1/e$.
- One of the cons of this method is too much data exchange.