Contents

Development of Money

> Definition: "something generally accepted as a medium of exchange, a measure of value, or a means of payment."

- Barter (direct exchange of goods)
- Medium of exchange (arrowheads, salt)
- Coins (gold, silver)
- Tokens (paper)
- Notational money (bank accounts)
- Dematerialised schemes (pure information)

Types of Money

Types	Token	Notational	Hybrid
Fiduciary	Cash, Government bearer bond	Account with central bank	Government check
Scriptural	Certified check, Traveller's check	Bank account, Frequent flyer miles	Personal check, gift certificate

Generic Web Payment

1. www client: Request Item
2. www server: Find Item/Client Location
3. Merchant Server: Request Payment
4. Client Wallet: Send Payment
5. Merchant Server: Validate Payment
6. Bank: Payment is Valid
7. Merchant Server: Receipt of Payment
8. www server: Serve Item

On-line/Off-line

- In an on-line payment a third party is involved at the time of the purchase
  - Verification of payment to prevent fraud
- Transaction takes longer
- Third party can be a bottleneck

  - e.g. bank, acquirer

Anonymity vs Audit Trail

- Audit trail provides detailed log of all payments
  - Helps prevent fraud
  - Spending profiles can be built
  - Banks like it!
- Anonymity protects identity of buyer
  - Full anonymity
    - Identity cannot be linked to payment (eg. eCash)
  - Limited anonymity
    - Collaboration could yield identity
  - Anonymous to merchant
  - Privacy
    - Payment details hidden from outsiders

Payment Methods

Macropayments

- > $1
- Strong Crypto

Credit/Debit Cards

- International acceptability
- No fee for buyer

Cash

- Small amounts
- Person-to-person
- No (low) transaction fee
- No need for bank account

Check/EFT

- Potentially large amounts
- Person-to-person
- Vital for B2B transactions

Micropayments

- < $1
- Lightweight Crypto

Micro

- Very small payments
- Possibly 10c or less
- Information goods
    - e.g. Payments for stock quote $0.001

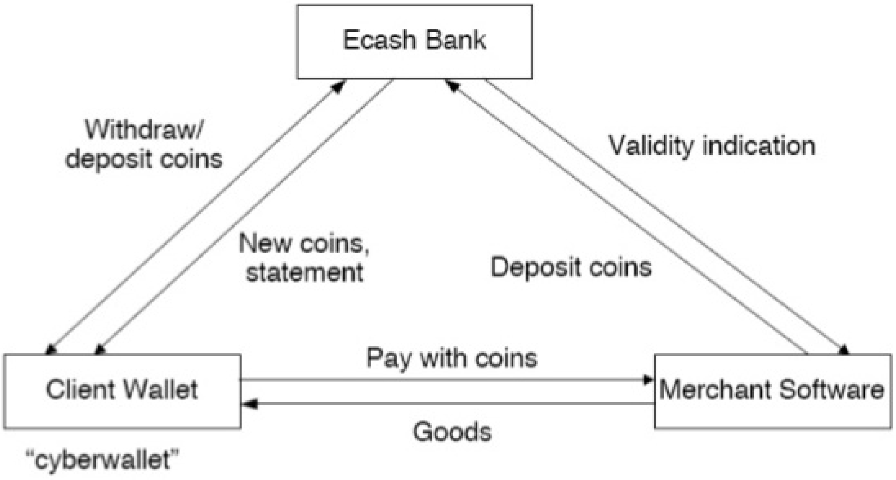
Card Payment

Participants

- Consumer
- Issuing bank
- Card association (e.g. Visa, Mastercard)
- Merchant
- Merchant bank
- Card association

eCash

- Fully anonymous digital cash
    - Pieces of data representing real monetary value
    - Digitally signed by bank
    - Problems
- DigiCash - 1990
    - David Chaum, “the father of digital cash”
- Information, hard goods, etc.
- Strong security, good privacy



Payment Model

Blind Signature Protocol

- Let m be the coin's real serial number, r the blinding factor, e and n the bank's public key exponents
- Sender raises r to the bank's public key exponent e and computes the product of the serial number and the blinding factor
  - $m' \equiv m \times r^e \mod n$
- Bank signs the blinded serial number with its private key
  - $s' \equiv (m')^d \mod n$
- Returns the coin to the user who removes blinding factor
  - $s \equiv s' \times r^{-1} \mod n$
- User now has a coin signed with the bank's private key
  - $s \equiv (m')^d \times r^{-1} \mod n$
  - $= (m \times r^e)^d \times r^{-1} \mod n$
  - $= m^d \times r^{ed} \times r^{-1} \mod n$
  - $= m^d \times r \times r^{-1} \mod n$
  - $= m^d \mod n$

Bitcoin

- Decentralised, Peer-to-Peer (P2P) Electronic Cash System

  - Invented in 2008 by “Satoshi Nakamoto”
- Bitcoin makes everyone collectively the bank!!
  - No longer any single organisation in charge of the currency
- Think about the enormous control a central bank has over the money supply
  - Bitcoin introduces a pretty huge change to this business model
- Makes use of the “proof-of-work” concept to prevent double spending in the Bitcoin network
  - Bitcoin miners are rewarded for solving the proof-of-work problem with newly minted bitcoins or transaction fees

Version 1

- Suppose Alice wants to give Bob a Bitcoin
  - Alice writes down the message “I, Alice, am giving Bob one Bitcoin”
  - Digitally sign the message using her private key
    * Announces the signed string of bits to the entire world

Q: What is the problem with this version of the protocol?

A: Alice could keep sending Bob the same signed message over and over.

- We need a way of making Bitcoins unique
  - Need a label or serial number
  - Alice would sign her message “I, Alice, am giving Bob one Bitcoin, with serial number 8740348”

Version 2

- Make *everyone* collectively the bank
  - Everyone keeps a complete record of which Bitcoin belong to which person
    * i.e. a shared public ledger showing all Bitcoin transactions
- Suppose Alice wants to transfer a Bitcoin to Bob
  - Signs the message “I, Alice, am giving Bob one Bitcoin, with serial number 1234567”
  - Bob uses his copy of the blockchain to check that the Bitcoin is Alice’s to give
  - Broadcast both Alice’s message and his acceptance of the transaction to the entire network
  - Everyone updates their copy of the blockchain

Version 3

- When Alice sends Bob a bitcoin
  - Bob should not try to verify the transaction alone
- Broadcast the transaction to the entire network of Bitcoin users
  - Ask them to help determine whether the transaction is legitimate

Proof-of-Work (PoW)

- Involves a combination of two ideas
  - Make it computationally costly for network users to validate transactions
  - Reward them for trying to help validate transactions
- As people on the network hear a message
  - Each adds it to a queue of pending transactions that they have been told about, but which have not yet been approved
  - A network user named David might have the following queue of pending transactions
    * "I, Tom, am giving Sue one Bitcoin, with serial number 1201174"
    * "I, Alice, am giving Bob one Bitcoin, with serial number 1234567"

Hash Collisions

- David checks his copy of the blockchain, and can see that each transaction is valid
  - Would like to help out by broadcasting news of that validity to the entire network
- As part of the validation protocol David is required to solve a hard computational puzzle - the "Proof-of-Work"
- David has to find a nonce x such that when we append x to the list of transactions I and hash the combination, the output hash begins with a long run of 0s
- The puzzle can be made more or less difficult by varying the number of zeroes
  - A simple puzzle might require four 0s at the start of the hash
  - A more difficult puzzle might require 15 consecutive zeros

Example

- If we use $I =$ "Hello, world!" and the none $x = 0$
  - h("Hello, world!") $= 957914327af08234$
    - $x = 0$ is a failure, since the output does not begin with any 0s
- We can keep trying different values for the nonce, $x = 1, 2, 3$...Finally, at $x = 4250$ we obtain
  - h("Hello, world!4250") $= 00004c2380f23408$
- If we want the output hash value to begin with 10 zeroes
  - Then on average, we need to try $16^{10} \approx 10^{12}$ different values for x before we find a suitable nonce

Bitcoin Miners

- Suppose David is lucky and finds a suitable nonce x
- Broadcasts the block of transactions he is approving to the network, together with the value of x
  - Other participants in the network can verify that x is a valid solution to the proof-of-work puzzle
    - Update their Blockchain to include the new block of transactions
- This validation process is called *mining*
  - For each block of transactions validated, the successful miner receives a bitcoin reward

Micropayments

- Repeated small payments for low value information
- Macropayment Problems
  - Minimum number of transactions/second
  - Maximum number of strong cryptographic protocols
    - Efficiency limits of strong cryptographic protocols
- Micropayments Solution
  - Very small per-transaction cost (sub-cent)
  - Efficiency by slightly relaxing security
  - Some fraud (few cents) is OK
- Systems
  - Millicient, PayWord, MicroMint, Subscrip

Enable

- No minimum price for information and services
  - New internet opportunities
- Quality information due to financial reward