Electronic Payment Systems

- Overview
- eCash
- Bitcoin
- Micropayments

Development of Money

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

Monetary History

ABSTRACTION

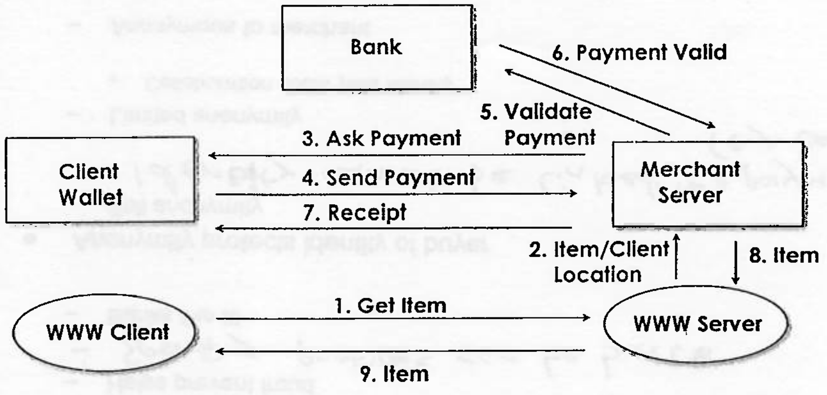
- Barter (direct exchange of goods)
- Medium of exchange (arrowheads, salt)
- Coins (gold, silver)
- Tokens (paper)
- Notational money (bank accounts)
- Dematerialized schemes (pure information)

NEED BANKS

Types of Money

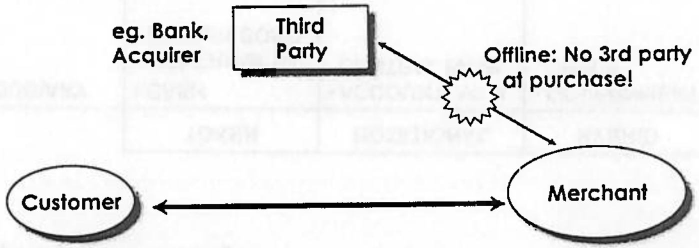
	TOKEN	NOTATIONAL	HYBRID
FIDUCIARY	• CASH • GOVERNMENT BEARER BOND	• ACCOUNT WITH CENTRAL BANK	• GOVERNMENT CHECK
SCRIPTURAL	• CERTIFIED CHECK • TRAVELER'S CHECK	• BANK ACCOUNT • FREQUENT FLYER MILES	• PERSONAL CHECK • GIFT CERTIFICATE

Generic Web Payment Example



On-line/Off-line

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



eg. Bank, Acquirer

Third Party

Offline: No 3rd party at purchase!

Customer

Merchant

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 (e.g. cash)
  - Limited anonymity
    - Collaboration could yield identity
  - Anonymous to merchant
  - Privacy
    - Payment details hidden from outsiders

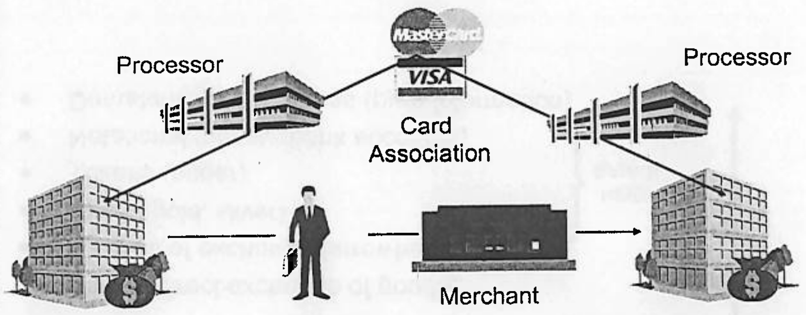
Payment Methods

Macropayments
- > $1
- Strong Crypto

Micropayments
- < $1
- Lightweight 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
- **Cheque/EFT**
  - Potentially large amounts
  - Person-to-person
  - Vital for B2B transactions
- **Micro**
  - Very small payments
  - Possibly 10c or less
  - Information goods
    - e.g. Payment for stock quote $0.001

Card Payment - Participants



Processor

Card Association

Processor

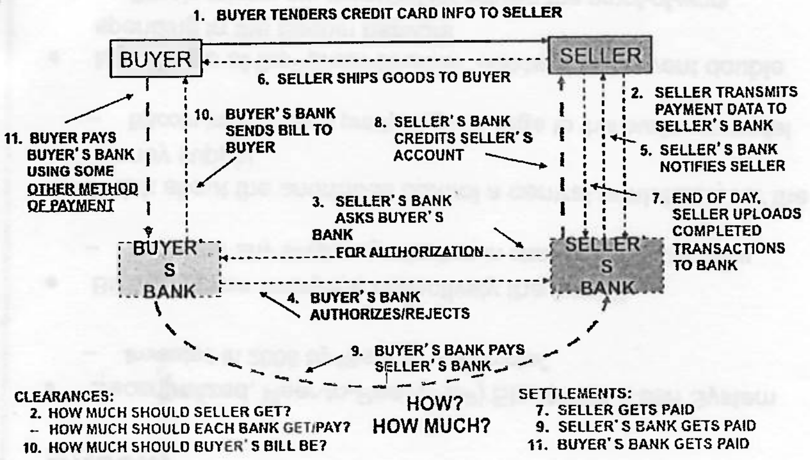
Consumer

Merchant

- Issuing Bank
- Issues card
- Extends credit
- Assumes risk of card
- Cardholder reporting

- Merchant Bank (Acquirer)
- Sets up merchant
- Extends credit
- Assumes risk of merchant
- Funds merchant

Card Payment - Transaction



1. BUYER TENDERS CREDIT CARD INFO TO SELLER

BUYER

SELLER

6. SELLER SHIPS GOODS TO BUYER

2. SELLER TRANSMITS PAYMENT DATA TO SELLER'S BANK

10. BUYER'S BANK SENDS BILL TO BUYER

8. SELLER'S BANK CREDITS SELLER'S ACCOUNT

5. SELLER'S BANK NOTIFIES SELLER

11. BUYER PAYS BUYER'S BANK USING SOME <u>OTHER METHOD OF PAYMENT</u>

3. SELLER'S BANK ASKS BUYER'S BANK FOR AUTHORIZATION

7. END OF DAY, SELLER UPLOADS COMPLETED TRANSACTIONS TO BANK

BUYER'S BANK

SELLER'S BANK

4. BUYER'S BANK AUTHORIZES/REJECTS

9. BUYER'S BANK PAYS SELLER'S BANK

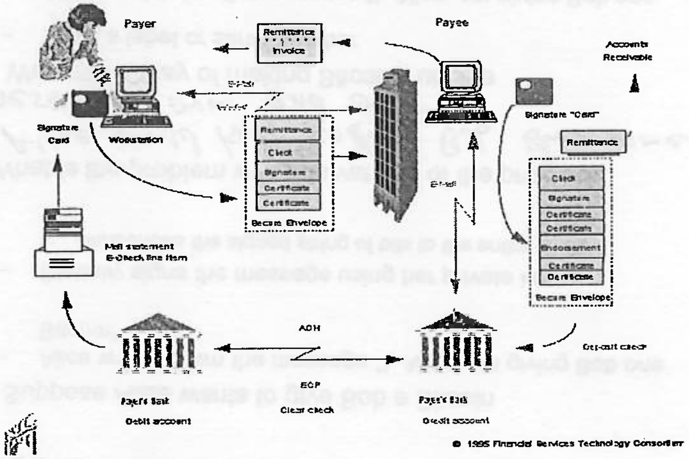
CLEARANCES:
2. HOW MUCH SHOULD SELLER GET?
– HOW MUCH SHOULD EACH BANK GET/PAY?
10. HOW MUCH SHOULD BUYER'S BILL BE?

HOW?
HOW MUCH?

SETTLEMENTS:
7. SELLER GETS PAID
9. SELLER'S BANK GETS PAID
11. BUYER'S BANK GETS PAID

Electronic Cheques

Electronic Check Concept

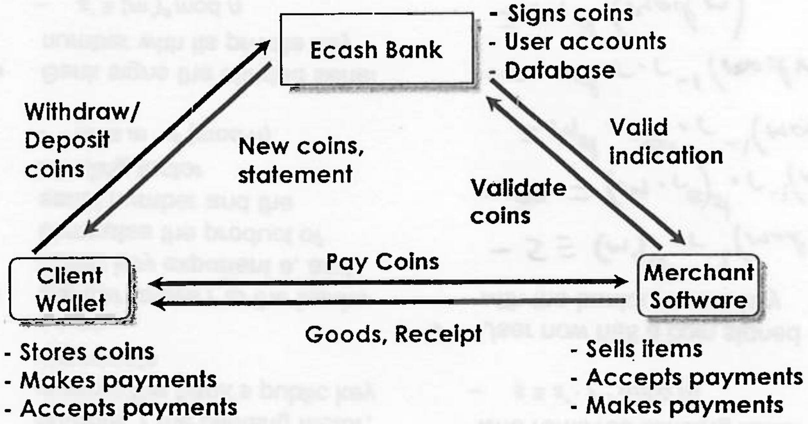


© 1995 Financial Services Technology Consortium

eCash

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

eCash Payment Model



Ecash Bank
- Signs coins
- User accounts
- Database

Withdraw/ Deposit coins

New coins, statement

Valid indication

Validate coins

Client Wallet

Pay Coins

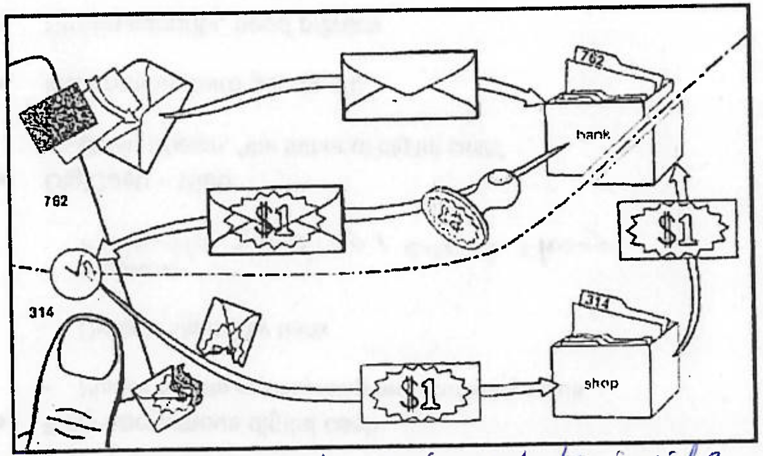
Goods, Receipt

Merchant Software

- Stores coins
- Makes payments
- Accepts payments

- Sells items
- Accepts payments
- Makes payments

Anonymous Digital Cash



Stamp envelope w/o- seeing whats inside.
Blank paper is now stamped inside envelope

Blind Signature Protocol

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

$- s \equiv (m')^d \cdot r^{-1} (\bmod n)$

$- \ = (m \cdot r^e)^d \cdot r^{-1} (\bmod n)$

$= m^d \cdot r^{ed} \cdot r^{-1} (\bmod n)$ – d is inver of e

$= m^d \cdot r \cdot r^{-1} (\bmod n)$

$= m^d (\bmod n)$

Bitcoin

- Decentralized, Peer-to-Peer (P2P) Electronic Cash System
  - Invented in 2008 by "Satoshi Nakamoto"
- Bitcoin makes everyone collectively the bank!!
  - No longer any single organization 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

Bitcoin – Version 1

- Suppose Alice wants to give Bob a Bitcoin
  - Alice writes down the message "I, Alice, am giving Bob one Bitcoin"
  - Digitally signs 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?

Alice could keep sending Bob the same message over and over

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

• Requires on-line verification

Bitcoin – Version 2

- Make *everyone* collectively the bank
  - Everyone keeps a complete record of which Bitcoins belong to which person
    - i.e. a shared public ledger showing all Bitcoin transactions

-Known as "Blockchain"

- 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
    - Broadcasts both Alice's msg and his acceptance of the transaction to the entire network
      - Everyone updates their copy of the Blockchain

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

~~Bob should~~.

Bitcoin – 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
- Q: Can Alice double spend in this version of network-based protocol?

-Yes she could do this by taking over the bitcoin network by creating a billion separate identities ~~with~~ which are under her control

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 *l* and hash the combination, the output hash begins with a long run of 0s

-"k-bit partial collision"

- 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

PoW Example

- For example, if we use *l* = "Hello, world!" and the nonce *x* = 0
  - h("Hello, world!0") = 1312af178c253f84028d480a6adc1e25e81caa44c749ec819761 92e2ec934c64
    - *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") = 0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464 e12dcd4e9
- 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

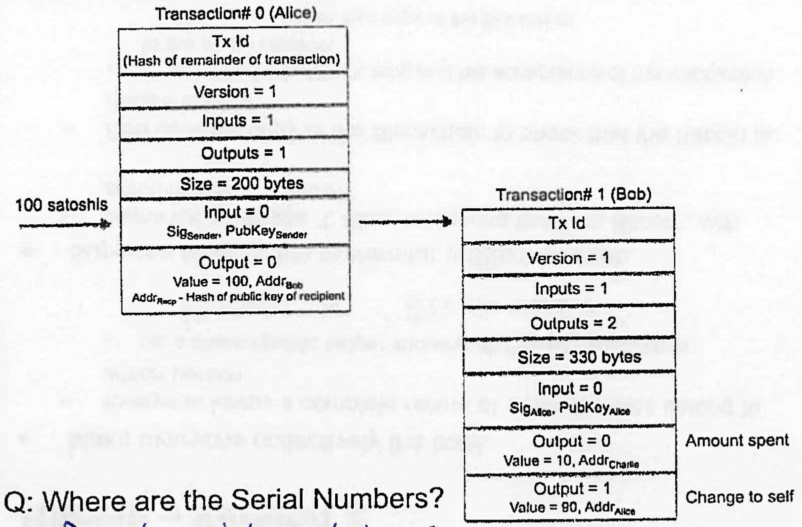
• A challenging task, requires lots of computing power

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 for *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

• Currently at 12.5 bitcoins

Bitcoin Transactions


Transaction# 0 (Alice)

Tx Id
(Hash of remainder of transaction)

Version = 1

Inputs = 1

Outputs = 1

Size = 200 bytes

100 satoshis

Input = 0
Sig_{Sender}, $PubKey_{Sender}$

Output = 0
Value = 100, $Addr_{Bob}$
$Addr_{Recp}$ - Hash of public key of recipient

Transaction# 1 (Bob)

Tx Id

Version = 1

Inputs = 1

Outputs = 2

Size = 330 bytes

Input = 0
Sig_{Alice}, $PubKey_{Alice}$

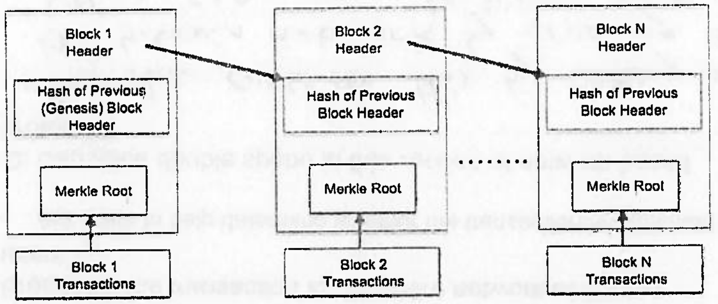
Output = 0
Value = 10, $Addr_{Charlie}$ — Amount spent

Output = 1
Value = 90, $Addr_{Alice}$ — Change to self

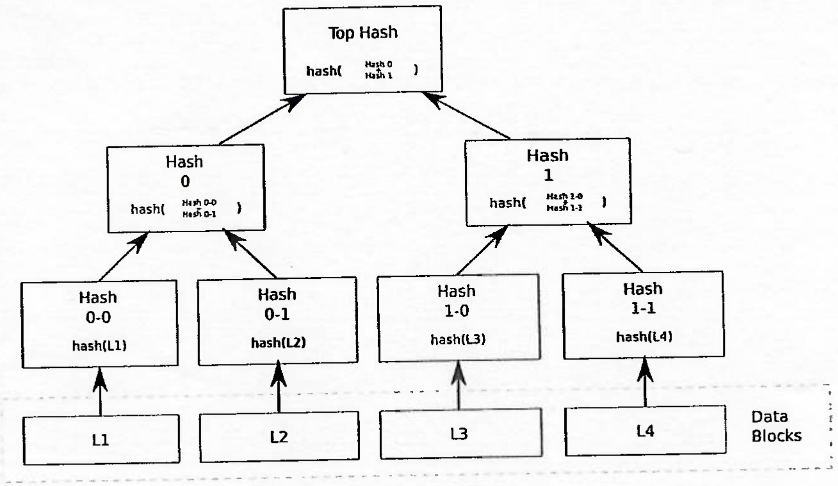
Q: Where are the Serial Numbers?

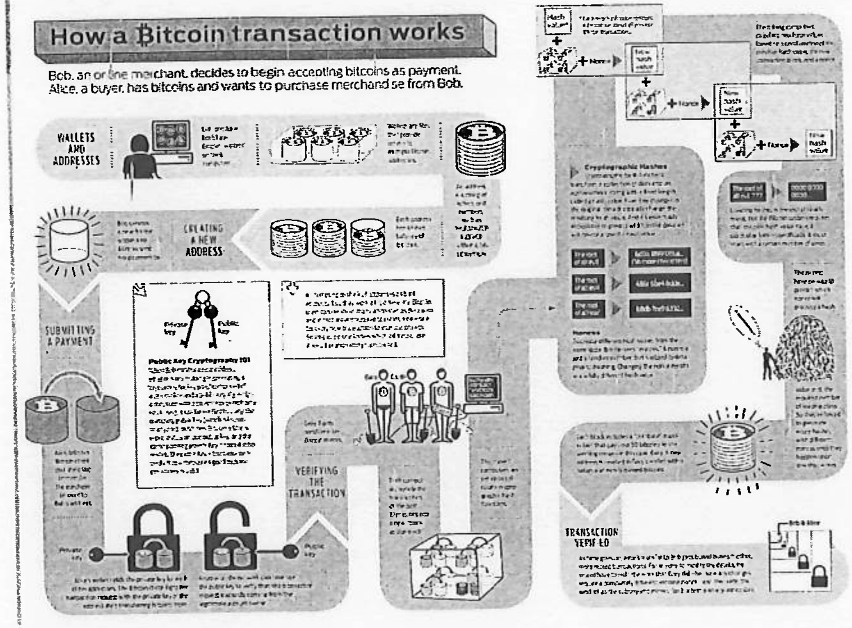
Don't exist in bitcoin

Blockchain



Merkle Tree





How a ₿itcoin transaction works

Bob, an online merchant, decides to begin accepting bitcoins as payment. Alice, a buyer, has bitcoins and wants to purchase merchandise from Bob.

Micropayments

- Repeated small payments for low value information
- Macropayment Problems
  - Minimum price set by transaction processing costs

    ~ e.g. credit card ~3% transaction value
  - Maximum number of transactions/second
    - 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
  - Millicent, PayWord, MicroMint, Subscrip

Micropayments Enable

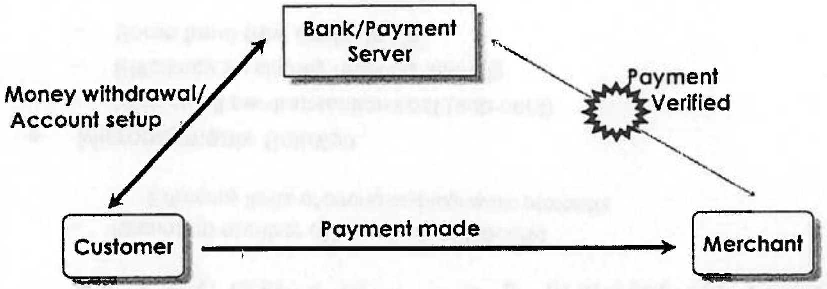
- No minimum price for information and services
  - New Internet opportunities
- Quality information due to financial reward

To buy information
- Articles and Web pages
- Stock quotes and DB queries
- Cartoons and clip art
- Music and videos

To buy software
- Java applets
- Apps
- Software add-ons
- Games

To bill access
- To applications
- For services
- Education
- To shared resources

Micropayment Purchase



On-line verification with 3rd party <u>removed</u>