Electronic Commerce

SET is the answer, but you have to phrase the question very carefully

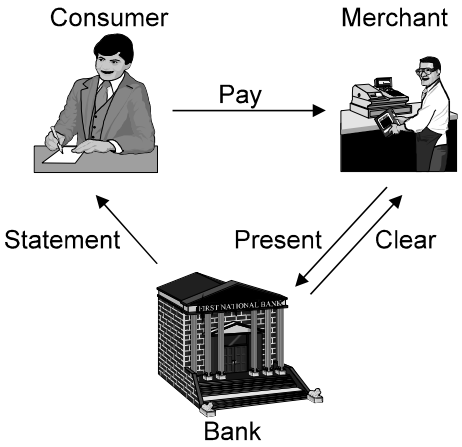
Electronic Payments

An electronic payment system needs to be

- Widely recognised
- Hard to fake
- Hold its value
- Convenient to use
- Anonymous/not anonymous

Convenience is the most important point

Cheques



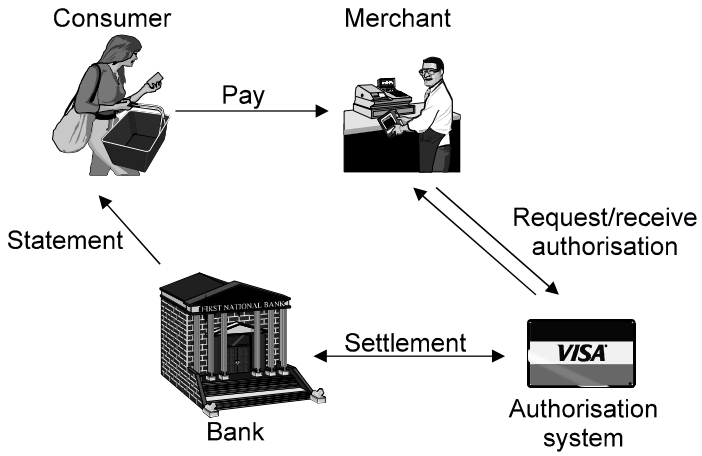
Merchant doesn't know whether the cheque is valid until it's cleared

Cheques (ctd)

Consumer can't detect fraud until the statement arrives

Cost of processing errors vastly outweighs the cost of normal actions

Credit Cards



Authentication is online

Settlement is usually offline (batch processed at end of day)

Credit Cards (ctd)

Consumer can't detect fraud until the statement arrives

Cost of processing errors vastly outweighs the cost of normal actions

Merchant carries the risk of fraud in card not present transactions

Consumer liability is limited to $50

Far more merchant fraud than consumer fraud

Credit card companies assume liability for their merchants; banks with cheques don't

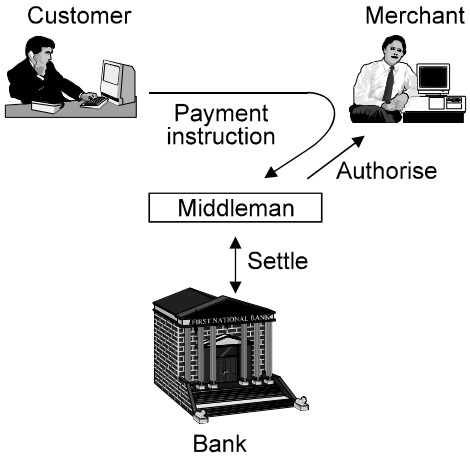
Transactions on the Internet

Transactions are fairly conventional card not present transactions and follow the precedent set by phone orders

Online nature provides instant verification

Biggest problems are authentication and confidentiality

General Model of Internet Transactions



Virtually all net payment systems consist of some variant of this

Everyone wants to be the middleman

Retail vs Business-to-business Commerce

Retail commerce

- Small dollar amounts
- Stranger-to-stranger transactions

Business-to-business commerce

- Large dollar amounts
- Based on trust relationships
- Banks play a direct role — they guarantee the transaction
  - – You can't disintermediate the banks

Business-to-business commerce is where the money is

- For retail transactions, you can't beat a credit card over SSL

Business customers will buy to reduce current costs

Payment Systems

Book entry systems

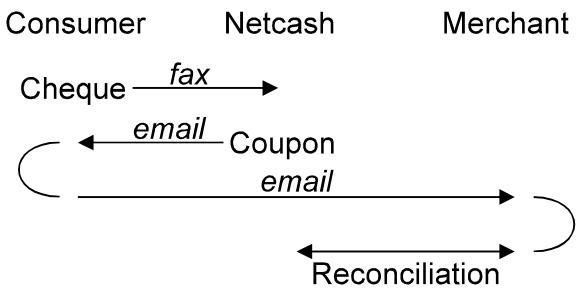
- Credit cards over SSL
- Encrypted credit cards (Cybercash)
- Virtual credit cards (First Virtual)
- e-cheques (Netcash)
- Mondex/SET
- Many, many others

Bearer certificate systems

- Scrip (Millicent)
- True digital cash (Digicash)

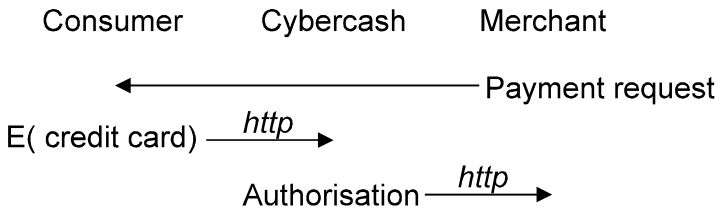
Netcash

e-cheques, `http://www.teleport.com/~netcash`



Cybercash

Encrypted credit cards, `http://www.cybercash.com`



Book Entry System Variations

Some systems (eg GlobeID) have the consumer (instead of the merchant) do the messaging

Credit cards don't handle small transactions very well. Some options are

- Don't handle micropayments at all
- Middleman has to act as a bank
- Use a betting protocol: 10 cent transaction = 1% chance of a $10 transaction

Digicash

Digicash issuing protocol

User		Bank (mint)
blind(note)	→	
	←	sign(blind(note))
unblind(sign(blind(note))) = sign(note)		

User ends up with a note signed by the bank

- Note is not tied to user
- Implemented as an electronic purse which holds arbitrary denominations

Digicash (ctd)

Using e-cash

- Send note to merchant
- Merchant redeems note at bank
- Double spending is avoided by having the user ID revealed if the note is banked twice (ZKP)
  - The fielded system just keeps a record of already spent notes, which is easier

Digicash (ctd)

Problems

- Banks don't like it (anyone can be a bank)
- Governments don't like it
- Not used much (awkward/fluctuating licensing requirements)
  - Licensed as if it were an RSA-style monopoly patent

By the time they figure it out, the patent will expire (2007)

- Digicash principals are great cryptographers, not so good business managers
- Patents are currently in limbo after Digicash Inc. collapsed

Making e-cash work

Best e-cash business model is to earn seignorage by selling it

- Bank earns interest on real cash corresponding to digital bits held by consumer
- US Federal Reserve earns $20B/year in interest on outstanding dollar bills
- Phone cards and gift vouchers are a small-scale example of this

Consumers may demand interest on e-cash

e-cash is useful for small transactions (micropayments) which other systems can't handle

- But what do you buy over the net for 10 cents?

echecks

Background for a US audience

- Non-US automated payment processing is relatively sophisticated
- Automatic payments (rent, utilities, wages) are handled via direct funds transfer
- Funds are moved electronically from one account to another on the same day
  - Checks are used rarely
  - Electronic check proposals are met with bafflement

echecks (ctd)

Background for a non-US audience

- US cheque and payment processing is very primitive
- "Automatic payment" frequently means the payers bank writes a cheque and sends it to the payee
- Payments are batched and held until a sufficient number have accumulated
  - The fact that funds leave the payers account on a given day doesn't guarantee timely arrival in the payees account
- Cheques are used extensively
- Electronic cheques would be a significant advance on the current situation

Electronic Cheque Design Requirements

Cheques can involve

- One or more signers
- One or more endorsers
- Invoice(s) to be paid
- Deposit to account or cash

Electronic version must be flexible enough to able to handle all of these

e-cheque Design

e-cheques are defined using FSML (Financial Services Markup Language)

- FSML allows addition and deletion of document blocks, signing, co-signing, endorsing, etc.

Signatures are accompanied by bank-issued certificates

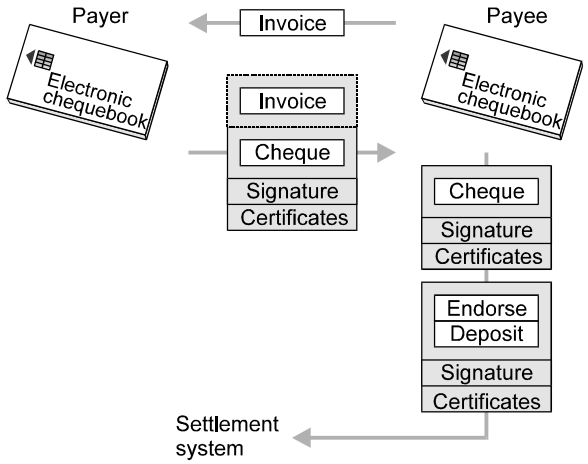
- Tie the signers key to a bank account
- Different account is used for e-cheques to protect standard cheque account against fraud

e-cheque Design (ctd)

Private key is held in smart card (electronic cheque book)

- Card numbers each signature/cheque
  - Attempts to re-use cheques will be detected
- Card keeps record of cheques signed
  - Provides some degree of protection against trojan horse software
- Card provides some degree of non-repudiation
- Use of software implementations rejected because of security concerns
  - "If hackers acquire signing keys and perpetuate fraud, payees confidence in the system would be destroyed"
- Use of PDA's as e-chequebooks was also considered

e-cheque Processing



Settlement is handled via existing standards

- ANSI X9.46 with FSML representation instead of cheque image
- ANSI X9.37 cash letter contained in X9.46 encapsulation

e-cheque Processing (ctd)

Cheque signature may also bind and invoice to avoid an attacker substituting a different invoice

Mechanisms can be extended to provide certified cheques

- Payers bank
  - Verifies details of cheque
  - Places hold on payers funds
  - Countersigns cheque

e-cheque design is a good example of carefully designing a protocol to meet certain security requirements

- Work around shortcomings in existing laws
- Work around shortcomings in existing security technology

e-cheque Format

Tag	Field
<check>	Start tag of cheque block
<checkdata>	Start tag of elements logged in electronic chequebook
<checknum>	Cheque number
<dateissued>	Date cheque was issued
<datevalid>	Date cheque is payable
<amount>	Amount of cheque (+optional currency)
<payto>	Payee (+optional bank, account, etc)
</checkdata>	End of elements logged
<checkbook>	ID of electronic chequebook
<restrictions>	Optional “duration”, “deposit only”, etc
<legalnotice>	“Subject to standard cheque law”
</check>	End of cheque block

e-cheque Format (ctd)

Tag	Field
<signature>	Start tag of signature block
<blkname>	Name of this block
<sigdata>	Start tag of signed data
<blockref>	Name of next block
<hash alg=*xxx*>	Hash of next block
<nonce>	Random string to make blocks unpredictable
<certissuer>	Optional identity of issuing certificate
<algorithm>	Hashing and signing algorithm used
</sigdata>	End of signed data
<sig>	Signature computed by electronic chequebook
</signature>	End of signature block

SET

Secure Electronic Transactions

Based on two earlier protocols, STT (VISA/Microsoft) and SEPP (MasterCard/IBM)

STT

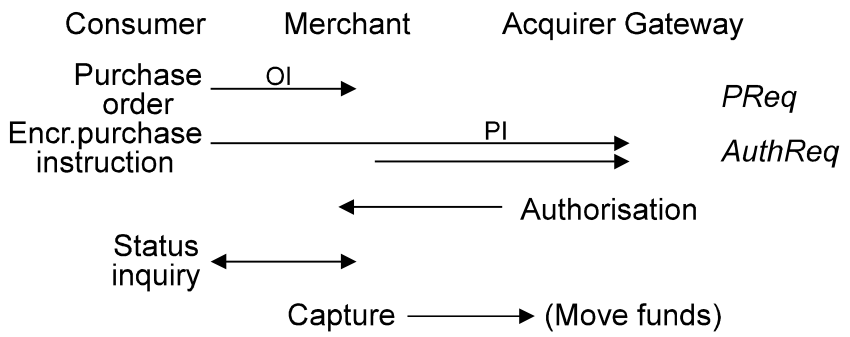
- One component of a larger architecture
- Provision for strong encryption
- Completely new system
- More carefully thought out from a security standpoint

SET (ctd)

SEPP

- General architectural design rather than a precise specification
- Lowest-common-denominator crypto
- Fits in with existing infrastructure
- More politically and commercially astute

SET (ctd)



Acquirer gateway is an Internet interface to the established credit card authorisation system and cardholder/merchant banks

SET Features

Card details are never disclosed to merchant

- Encrypted purchase instruction (PI) can only be decrypted by the acquirer
  - In practice the acquirer usually reveals the card details to the merchant after approval, for purchase tracking purposes
- PI is cryptographically tied to the order instruction (OI) processed by the merchant
- Clients digital signature protects the merchant from client repudiation

Authorisation request includes the consumer PI and merchant equivalent of the PI

- Acquirer can confirm that the cardholder and merchant agree on the purchase details

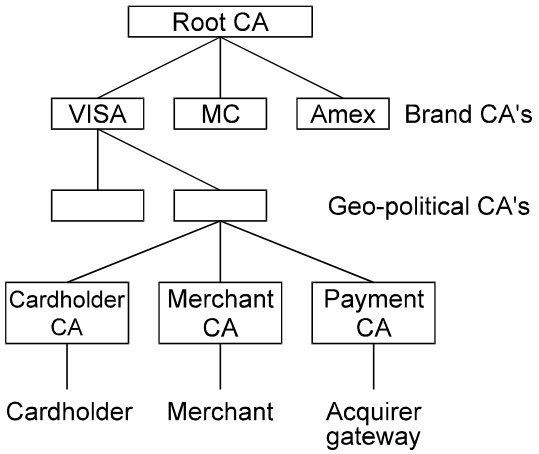
SET Features (ctd)

Capture can take place later (eg when the goods are shipped)

- User can perform an inquiry transaction to check the status

The whole SET protocol is vastly more complex than this

SET Certification



SET root CA and brand CA’s are rarely utilised and have very high security

SET Certification (ctd)

SET includes a complete PKI using customised X.509

- Online certificate requests
- Certificate distribution
- Certificate revocation

SET certificates are implemented as an X.509 profile with SET-specific extensions

SET Certification (ctd)

Card-based infrastructure makes certificate management (relatively) easy

- Users are identified by their cards
- Certificates are revoked by cancelling the card
- Because everything is done online, “certificate management” is easy
- Acquirer gateways have long-term signature keys and short-term encryption keys
  - Encryption keys can be revoked by letting them expire

SET in Practice: Advantages

SET will enable e-commerce, eliminate world hunger, and close the ozone hole

- SET prevents fraud in card not present transactions

SET eliminates the need for a middleman (the banks love this)

SET leverages the existing infrastructure

SET in Practice: Problems

SET is the most complex (published) crypto protocol ever designed

- > 3000 lines of ASN.1 specification
- 28-stage (!) transaction process
  - "The SET reference implementation will be available by mid 1996"
  - "SET 1.0 " " " mid 1997"
  - "SET 2.0 " " " mid 1998"
- Interoperability across different implementations is a problem

SET is awfully slow (6 RSA operations per transaction)

- Great for crypto hardware accelerator manufacturers
- For comparison, VISA interchange gateway currently has to handle 2000 pure DES-based transactions/second

SET in Practice: Problems (ctd)

Although SET was specifically designed for exportability, you still can't export the reference implementation

SET requires

- Custom wallet software on the cardholders PC
- Custom merchant software
- Special transaction processing software (and hardware) at the acquirer gateway.