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 and accepted
- Hard to fake
- Hold its value
- Convenient to use
- Anonymous/not anonymous

Convenience and wide acceptability are the most important points

Reg.E/Reg.Z

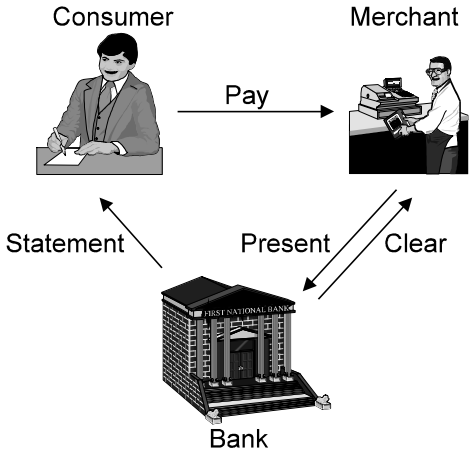
US consumer protection laws for credit cards and ATM cards

- Congress passed laws guaranteeing repudiation to force banks to provide appropriate consumer protection
- Similar protection exists in most other parts of the world

Report loss within 2 days: No liability

Report loss within 2-60 days (time to get a bank statement): Liability of $50 (value of one average transaction at the time the law was passed)

Cheques



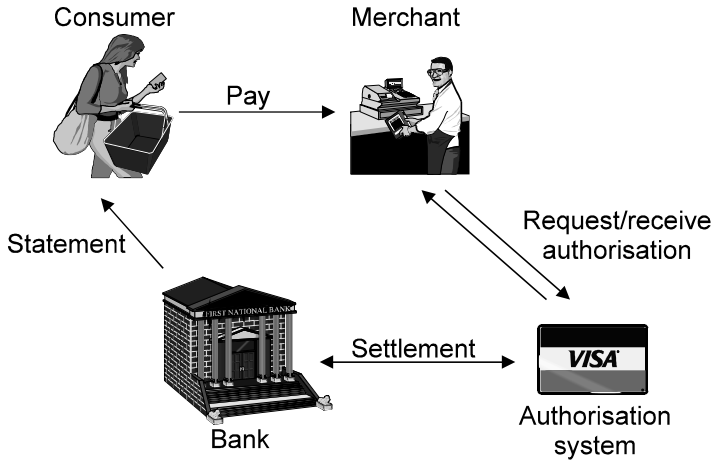
Merchant doesn't know whether the cheque is valid until it's cleared

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 the end of the day)

Credit Cards (ctd)

Consumer can't detect fraud until the statement arrives

- Internet access has reduced the window somewhat

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 (Reg.E/Reg.Z)

Originally far more merchant fraud than consumer fraud

- Internet has shifted the balance towards the consumer side

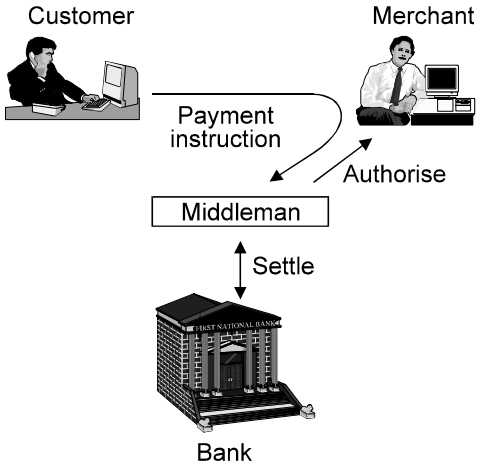
Credit card companies assume liability for their merchants; banks with cheques don't

Transactions on the Internet

Transactions are a fairly conventional card not present transaction and follow the precedent set by phone orders (MOTO)

- 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 pay to reduce current costs

Payment Systems

Book entry systems

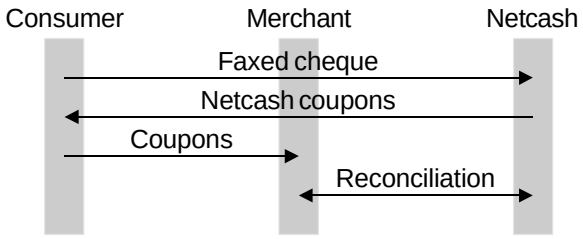
- Credit cards over SSL
- e-cheques (Netcash)
- Virtual credit cards (First Virtual)
- Encrypted credit cards (Cybercash)
- Mondex/SET
- Many, many others

Bearer certificate systems

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

Netcash, 1993 onwards

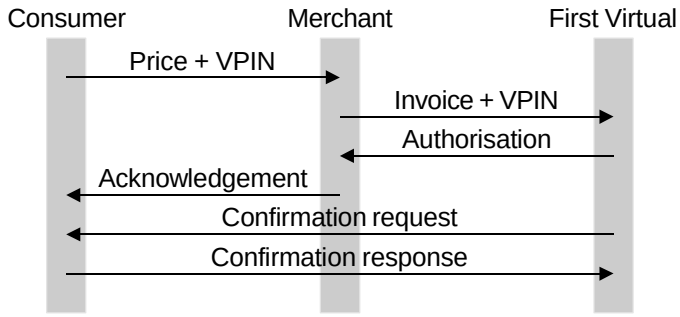
Coupons bought via faxed cheques



- USC/MIT research collaboration, continuously enhanced over time
- Later versions became quite complex

First Virtual, 1994

Virtual PINs (VPINs) over email



- First Virtual ↔ consumer confirmation verifies the consumer ↔ merchant communication

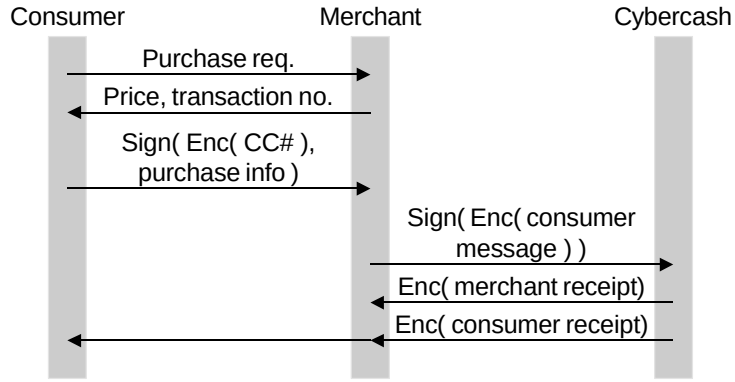
First Virtual, 1994 (ctd)

Inadvertently acted as a test load for the Internet infrastructure

- Broken mail clients
- Broken mail servers
- Broken DNS
- Broken routers
- Broken …

Cybercash, 1994

Encrypted credit cards



- Consumer signs the purchase info so the merchant can verify it, but encrypts the CC# so only Cybercash can process it
- Merchant countersigns the consumer message, forwards it to Cybercash

Cybercash, 1994 (ctd)

Main goals

- Prevent exposure of credit card details
- Prevent merchant fraud

Book Entry System Variations

Some systems (e.g. 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

Paypal

Had ebay as its killer app

Credit cards over SSL again

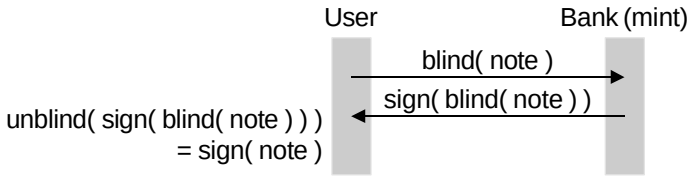
- Paypal can act as an escrow agent
- Starts to look a lot like a bank

Used mostly in the US

- Non-US countries allow user-initiated bank transfers

Digicash, 1994

Digicash issuing protocol



User ends up with a note signed by the bank

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

Digicash (ctd)

Using e-cash

- Send the note to the merchant
- Merchant redeems the note at the 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 (paranoia about anonymous cash)
- 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 probably to earn seignorage by selling it

- Bank earns interest on real cash corresponding to digital bits held by the 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) that 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
  - User-initiated funds transfers to other users are easy to do
  - Paypal is largely redundant outside the US
- 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 payer's bank writes a cheque and sends it to the payee
- Payments are batched and held until a sufficient number have been accumulated
  - The fact that funds leave the payer's account on a given day doesn't guarantee timely arrival in the payee's 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 be able to handle all of these

e-cheque Design

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

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

Signatures are accompanied by bank-issued certificates

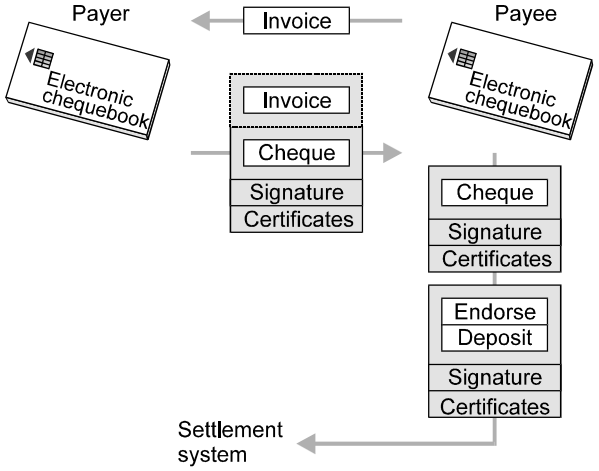
- Tie the signer's key to a bank account
- Different account is used for e-cheques to protect the standard cheque account against fraud

e-cheque Design (ctd)

Private key is held in a 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 was rejected because of security concerns
  - "If hackers acquire signing keys and perpetuate fraud, payee confidence in the system would be destroyed"
- Use of PDAs 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 in an invoice to avoid an attacker substituting a different invoice

Mechanisms can be extended to provide certified cheques

- Payer's bank
  - Verifies the details of the cheque
  - Places a hold on the payer's funds
  - Countersigns the 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>`	Cheque amount (+ 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=x>`	Hash of next block
`<nonce>`	Random value to make blocks unpredictable
`<certissuer>`	Optional identity of issuing certificate
`<algorithm>`	Hash and signature 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

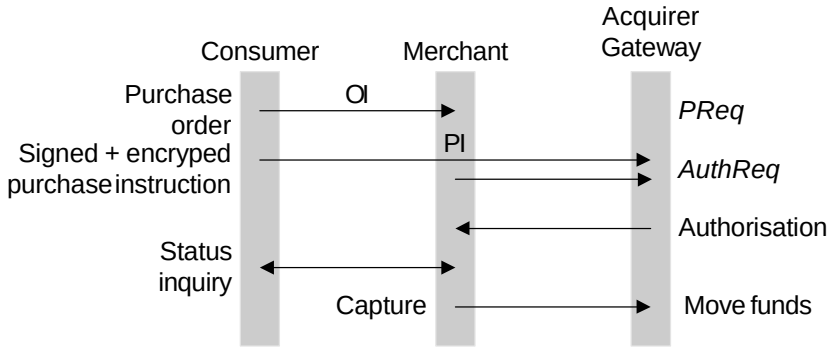
- One component of a larger architecture
- Provision for strong encryption
- Completely new system
- More carefully thought out from a security standpoint

SET (ctd)

SEPP

- General architectural design rather than a precise specification
- Lowest-common-denominator crypto (export controls again)
- Fits in with the 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 the 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
- Client's 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

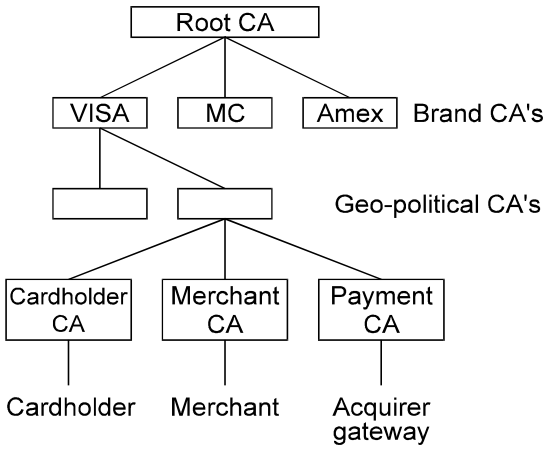
SET Features (ctd)

Capture can take place later (e.g. 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 CAs are rarely utilised and have very high security

- Unlike PEM, this is a case where a single hierarchy does work

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

- Make use of just enough of X.509 to be workable

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

Until IPsec, SET was 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
- SETco charged a huge amount of money for compliance testing of implementations
  - Hard on small companies, who were doing the implementation work

SET in Practice: Problems (ctd)

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 messages are huge, ~100× larger than a standard ISO 8583 card payment message

- Some gateways would get a SET message, check the signature, then throw everything away and set a flag in the 8583 message saying "Verified OK"

Although SET was specifically designed for exportability, you couldn't export the reference implementation until export controls were finally abolished

SET in Practice: Problems (ctd)

Huge numbers of merchants use the credit card number as the primary key for their customer databases

- "Solved" by making the card number visible to merchants
- Defeats the major purpose of SET (protecting the CC number)

SET requires

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

SET in Practice: Problems (ctd)

All the liability was carried by the issuing bank

- All the benefit was obtained by the acquiring bank
- Some attempts were made to mitigate this by splitting the costs

VISA/MC didn't care if SET succeeded or not

- SET was a counter to Cybercash, Mondex, etc
- When those didn't go anywhere, SET was superfluous
- Credits cards over SSL are far more profitable, since they're charged as card-not-present transactions

SET is still seeing some use in countries where Reg.E/Reg.Z don't apply

SET Successors

Verified by VISA™ — roll-your-own SET

- Everyone gets to independently reinvent the wheel…
  … badly

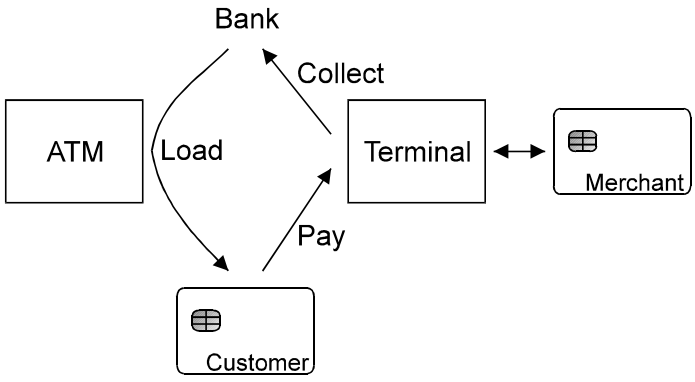
Design target seems more to impress VISA's auditors than to provide real security

C-SET (Chip-SET)

- Vaguely SET-like protocol for use with smart cards
- Eventually moved over to EMV

prEN 1546

Inter-sector electronic purse (IEP) standard, 1995



Both customers and merchants use smart-card based electronic purses to handle payment transactions

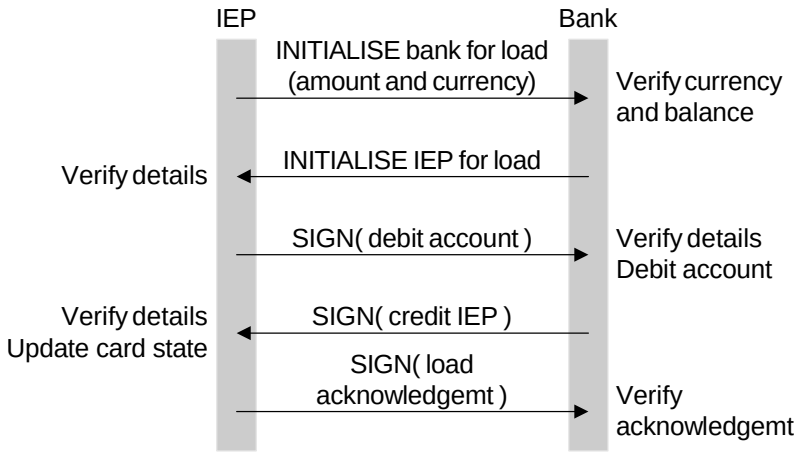
prEN 1546 (ctd)

Defines the overall framework in some detail, but leaves algorithms, payment types and parameters, and other details to implementers

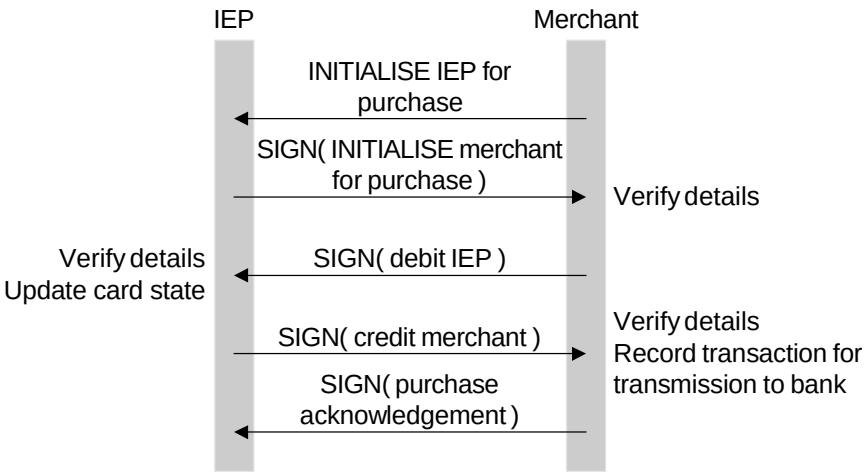
- Specifies the file layout and data elements for the IEP
- Defines commands INITIALISE IEP, CREDIT IEP, DEBIT IEP, CONVERT IEP CURRENCY, and UPDATE IEP PARAMETER
- Specifies exact payment routines in a BASIC-like pseudo-language
- All messages are "signed" (typically with a 4-byte DES MAC)
- Handles everything but purse-to-purse transactions

Includes many variants including a cut-down version for phonecards and extra acknowledgements for transactions

Credit IEP Transaction



Credit Merchant Transaction



TeleQuick

Austrian CEN 1546 Quick electronic purse adapted for online use

- Merchant ↔ customer = Internet
- Merchant ↔ bank = traditional X.25

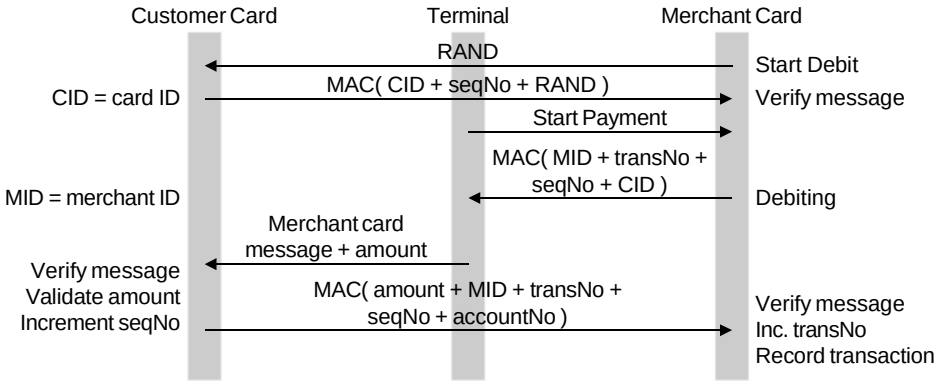
All communications are protected using TLS

Conceived as a standard Quick transaction with terminals a long way apart

- Transaction rollback in case of communications faults
- Virtual ATM must handle multiple simultaneous transactions
  – Handled via host security modules (HSM's)
- Windows PC is an insecure platform
  – Move functionality into reader (LCD,keypad,crypt module)

Geldkarte

Electronic purse developed by the ZKA (Zentraler Kreditausschuß, association of all German banks)



Geldkarte (ctd)

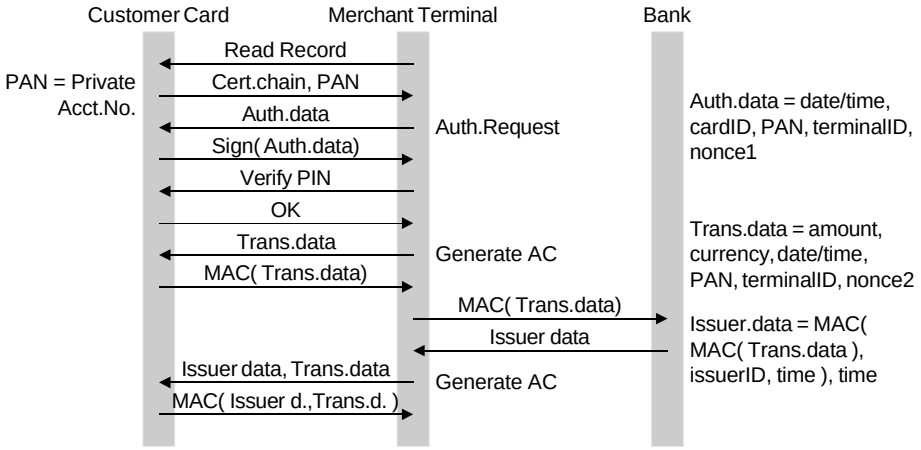
Nonces and sequence counters prevent replays

Merchant handles settlement offline

- transNo prevents replays

EMV

Europay, Mastercard, Visa, 1994 onwards



- Later updated to make use over the Internet more secure
  - Original EMV wasn't designed to run 'naked' over the Inet

EMV (ctd)

Popular in Asia due to huge reductions in fraud

- 17% → 8% fraud and still dropping as they move away from mag.stripe cards

Less popular elsewhere

- US has low fraud anyway (around 1%)
- Australia/New Zealand have all-online PIN checks
- UK Chip & PIN is vaguely EMV-ish, but seems mostly designed to push liability back onto customers

Micropayments

Allow payment of small quantities (cents or fractions of cents)

- Not handled by conventional payment mechanisms

Handle by having something act as an accumulator

- Accumulate micropayments until the amount is sufficient to be handled via a macropayment
- Issue jetons/scrip from a macropayment debit
- Betting protocols, e.g. 10 cents = 10% chance of paying $1.00

Many variants exist, following the general model for macropayments

Micropayments (ctd)

Netbill: Public-key Kerberos

- Netbill acts as an accumulator ("till") until sufficient balance accumulates to debit a bank account (credit card model)

Cybercoin

- Cybercash extended to micropayments
- Cybercash debits a bank account for a fixed amount, then pays out micropayments from that (debit card model)

Millicent, MicroMint, …

- Scrip-based systems
- Debit bank account, issue scrip in equal value (debit card model)