EMV flaws and fixes: vulnerabilities in smart card payment systems

Steven J. Murdoch

www.cl.cam.ac.uk/users/sjm217

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

OpenNet Initiative

www.opennet.net

COSIC seminar, 11 June 2007, ESAT, K.U.Leuven, BE

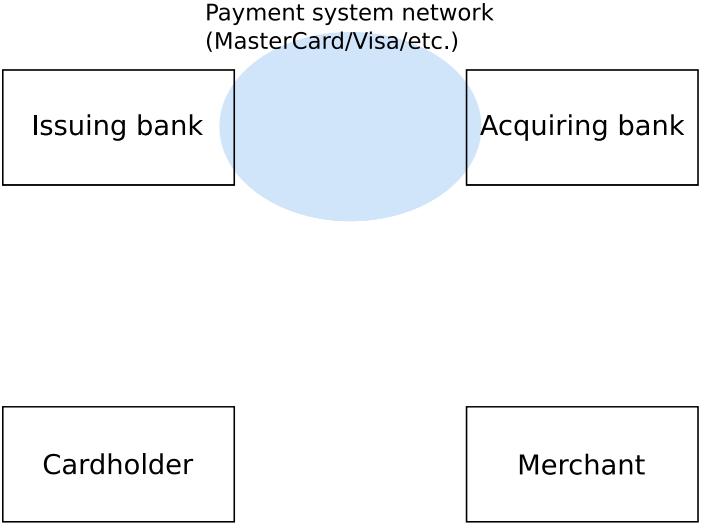
Structure of talk

- Summary of the EMV card payment system
  - ≈700 page specification, so I will simplify it somewhat, and omit secure messaging
- Some attacks on EMV, and corresponding defences
- How dispute resolution should and does work
- Generic weaknesses of the system, and how these can be resolved

EMV is a standard for smart card based payments

- Jointly developed by Europay, MasterCard and Visa
- Effort began in 1993 and current specification, v4.1 (the 6th revision), was released in 2004
- EMV is a self-contained standard, but the physical and electrical aspects are based on ISO/IEC 7816
- Freely available from `www.emvco.com`

Terminology



Payment system network
(MasterCard/Visa/etc.)

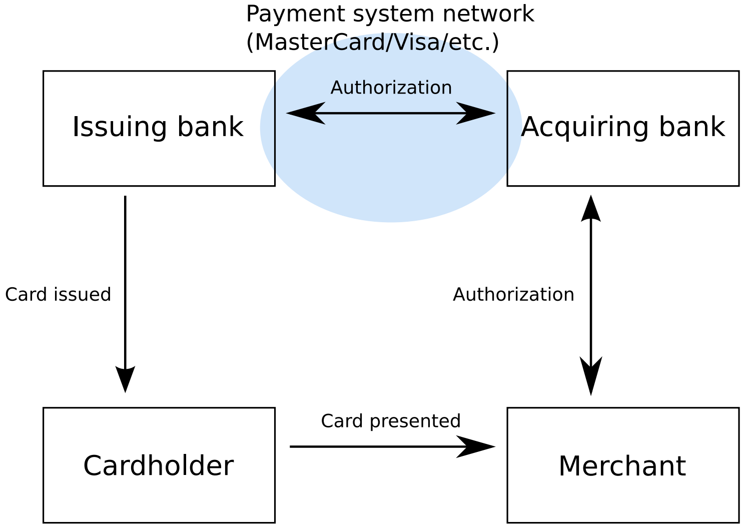
Issuing bank

Acquiring bank

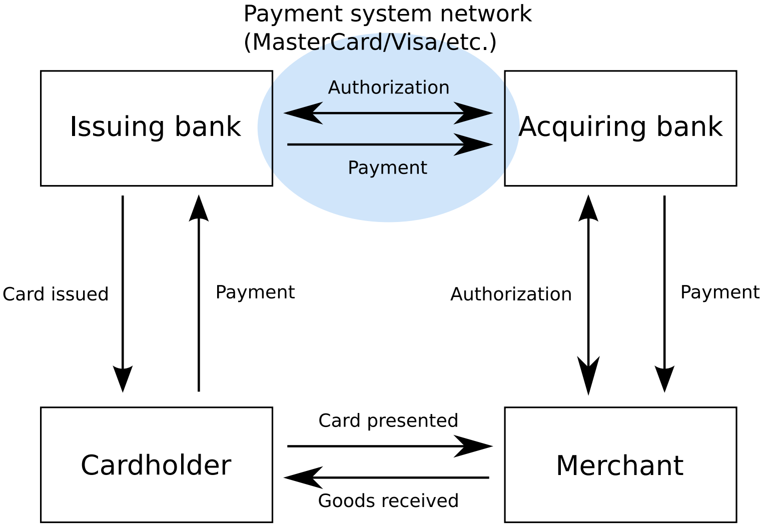
Cardholder

Merchant

Terminology



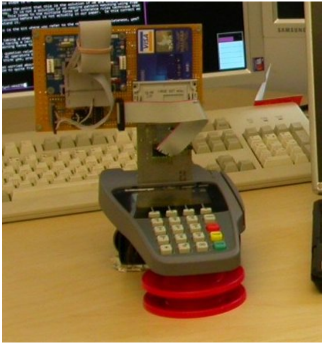
Terminology



EMV is primarily a compatibility standard

- Specifies (or at least intends to specify) enough for cards from one manufacturer/issuing bank to work with a terminal from another manufacturer/acquiring bank
- It is not a protocol, more a toolkit for building protocols
- It is not a security standard, although it specifies many security properties
  - Still possible to build a fully-EMV compliant, but broken, protocol
- Only (currently) defines that which is necessary for compatibility
  - Structure of fields exchanged between card and issuing bank is undefined or at most optional, but I will describe these too

To find out how EMV works, you need to look at real cards and transactions



This device records ISO 7816 transactions, and we monitored communications between real EMV terminals and cards. Also, we wrote a PC implementation of the core terminal software

There are 4 basic steps in a EMV transaction

1. *Read application data*: The terminal requests all information from the card that is necessary to process the transaction
2. *Card authentication*: The terminal confirms that the card is legitimate through Static Data Authentication (SDA), Dynamic Data Authentication (DDA) and/or Combined Data Authentication (CDA)
3. *Cardholder verification*: The terminal confirms that the person presenting the card is the legitimate cardholder (e.g. by PIN)
4. *Transaction authorization*: The terminal confirms that the card's account has adequate funds for the transaction

1: Read application data

Card → Terminal

- Account details (PAN, Cardholder name, Expiry date, etc.)
- Copy of the magnetic stripe details, for backwards compatibility
- Acceptable types of cardholder verification methods (CVM list)
- Issuer’s public key and certificate
- Signature of some data items under issuer’s key (SDA only)
- Card’s public key(s) and certificate(s) (DDA only)
- …

2: Card authentication (SDA)

Terminal

- Issuer's public key and certificate verified using CA (payment system network operator) public key
- Signature of static data items verified using issuer's public key

No nonce sent by terminal: signature of card data is static (SDA cards cannot perform RSA, and are hence cheaper)

Such signatures are thus copyable and replayable, once you have read the data or intercepted it in transit

Not all data items are signed; the issuer chooses which

2: Card authentication (DDA)

Terminal → Card

- Terminal nonce

Card → Terminal

- Signature using card's private key over terminal nonce and card nonce (and possibly other details)

Terminal

- Issuer's public key and certificate verified using CA public key
- Card's public key and certificate verified using issuer's public key
- Signature verified

3: Cardholder verification

CVM list specifies a set of rules for how terminals should select a cardholder verification method based on:

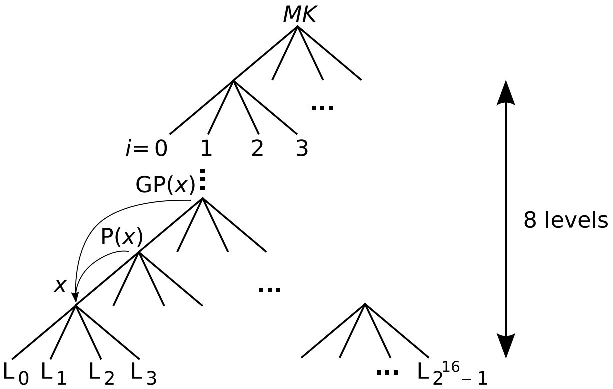
- Terminal capabilities (might not have a PIN pad)
- Attended or unattended terminal
- Type (cash, purchase, cashback) and value of transaction

Cardholder verification methods are:

- PIN
- Signature (generally if cardholder cannot remember/enter a PIN)
- Nothing (generally for unattended terminals)

For PIN verification, the entered PIN is sent to the card encrypted under its public key (DDA) or in the clear (SDA). The card reports success or failure (and decrements the PIN retry counter).

Session key derivation



x: Child; P(x): Parent; GP(x): Grandparent; P(MK) $= 0$

$x = \text{3DES}_{\text{P}(x)}(\text{GP}(x)_L \oplus i\ )\ ||\ \text{3DES}_{\text{P}(x)}(\text{GP}(x)_R \oplus i \oplus \texttt{<F0>})$

$\text{SK}_j = L_j \oplus \text{GP}(L_j)$ where L_j: jth leaf

4: Transaction authorization (step 1)

The terminal may choose to authorize the transaction offline or online

Terminal → Card

- Transaction currency, amount, country, date, type
- Result of cardholder verification
- Nonce

Card → Terminal

- DES/3DES CBC MAC, using derived session key (SDA/DDA) or asymmetric signature (CDA), over the above and additional data provided by the card
- Result of risk analysis: one of, *accept* (only if terminal specified offline authorization), *deny* or *go online*

4: Transaction authorization (step 2)

Terminal → Acquiring bank → Issuing bank

- Result for MAC/signature from card (ARQC)

Issuing bank → Acquiring bank → Terminal → Card

- Update to risk analysis parameters (CSU)
- CBC MAC over CSU and ARQC

Step 1 is then repeated

Result will be *accept* or *deny*

Part 2: Attacks on EMV

- SDA card cloning – *Yes cards*
- Fallback to magnetic stripe
- CVM manipulation
- Relay attacks

SDA card cloning and modification

- The lack of freshness in SDA means that data can be copied between cards
  - But SDA cards are cheaper, so used in the UK
- Cloned cards can be discovered by online transaction authorization, since the symmetric key is hard to clone
  - But offline verification is cheaper, so used in the UK for ≈20% of transactions
- PIN verification is performed by the card, so cloned ones can be programmed to accept any one – a *Yes card*

Result: If an SDA card is stolen, a fake can be created which is accepted for offline transactions, with any PIN

Fixes: Switch to DDA/CDA (expensive), cover any losses (probably cheaper since there are even easier attacks)

Magnetic stripe fallback

- EMV cards in the UK still have a magnetic stripe (for older terminals, chip failure, and use abroad)
- Data sent between card and terminal includes all information needed to make a fake magnetic stripe card
- Also commonly sent to the acquiring bank unencrypted
- With SDA cards, the PIN is sent in the clear from terminal to card
- Alternatively, fraudsters can capture it with a camera

Result: If a skimmer can be installed on a terminal, fake magnetic stripe cards can be created and used, with the correct PIN, in ATMs

Fixes: Turn off magnetic stripe fallback (disruptive), don't put magnetic stripe details on chip (happening, slowly)

Unsigned CVM manipulation

- The CVM list specifies acceptable types of cardholder verification.
- In some cards, this is not part of the signed data, so can be altered by a middleman
- So a fraudster, who has stolen a card but does not know the PIN, can make the terminal fall back to magnetic stripe

Result: Stolen cards may be usable online, even if the PIN is unknown

Fixes: Sign the CVM (expensive to roll out new cards), catch the problem at the back-end (look for unexpected signature transactions)

Middleman CVM attack

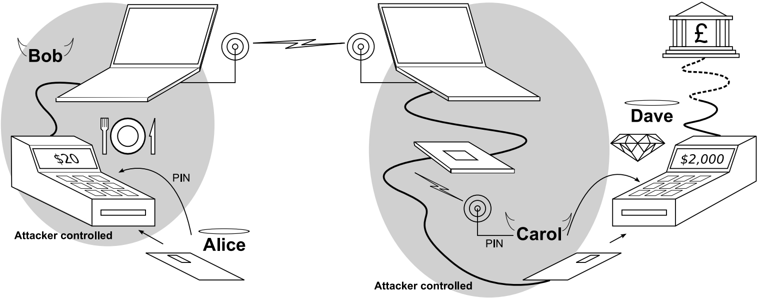
- Even for cards where the CVM is authenticated, a middleman could make the terminal think it is performing PIN verification, but tell the card it is doing signature
- Cards, according to the specification, report success or failure of the CVM, not what types were attempted

Result: Not much – cards we examined had extended the specification to also specify the CVM attempted (specifications are only part of the picture!)

Terminal tamper resistance

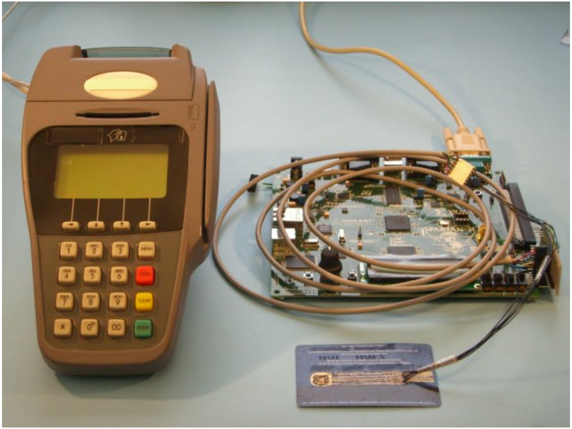
Cardholders have no way to verify the terminal they use is legitimate and untampered

Relay attacks in theory



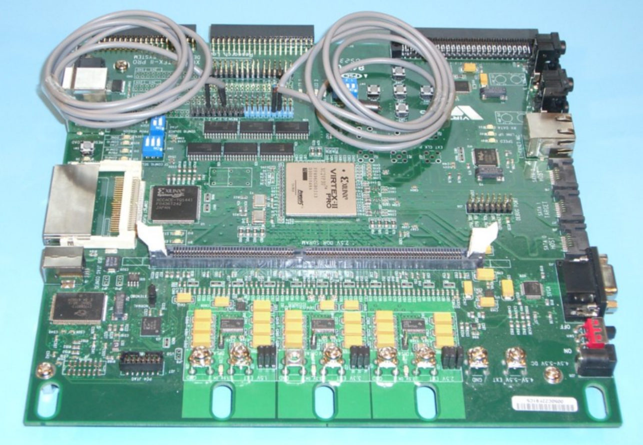
Alice thinks she is paying $20, but her card details are being relayed to Carol who is charging $2 000 to Alice’s card

Relay attacks in practice



We successfully tested our relay device in a real transaction, using 802.11b wireless (in front of TV cameras)

Distance bounding protocols



By measuring round-trip latency, the distance between the legitimate card and terminal can be securely established, defeating relay attacks

Dispute resolution (ideal)

Customer → Issuing bank

- Disputed transaction details

Issuing bank → Customer

- Card master key and transaction certificate (final MAC generated by card) of disputed transaction and all other information needed to repeat the calculation

or

- Refund

This is so the cardholder can tell whether the real card was used, or a SDA clone, made from recorded details

Dispute resolution (actual)

Customer → Issuing bank

- Disputed transaction details

Issuing bank → Customer

- Our systems are secure; your PIN was used. No, you can’t have your money back. Go away!

Dispute resolution (actual)

Customer → Financial ombudsman

- Disputed transaction details

Financial ombudsman → Customer

“The Firm has provided an ‘audit trail’ of the transactions disputed by you. This shows the location and times of the transactions and evidences that the card used was ‘CHIP’ read.”

Dispute resolution (actual)

Customer → Financial ombudsman

- Disputed transaction details

Financial ombudsman → Customer

"Although you have requested this information from the Firm yourself (and I consider that it is not obliged to provide it to you) I conclude that this will not make any difference, because this Service has already reviewed this information."

Dispute resolution (actual)

Customer → Financial ombudsman

- Disputed transaction details

Financial ombudsman → Customer

“Although you question The Firm’s security systems, I consider that the audit trail provided is in a format utilised by several major banks and therefore can be relied upon.”

Dispute resolution (actual)

Customer → Financial ombudsman

- Disputed transaction details

Financial ombudsman → Customer

> *"Although you question The Firm's security systems, I consider that the audit trail provided is in a format utilised by several major banks and therefore can be relied upon."*

Issuing bank → Financial ombudsman

£400

(issuing bank pays regardless of decision, but I wonder how repeated anti-bank decisions will affect the ombudsman's long-term funding)

Generic problems

Each device involved in the EMV transaction process protects the interest of its controller:

Merchant: Terminal

Acquiring bank: Terminal accreditation

Issuing bank: Smart card

Customer: ???

The customer does not have anything to protect themselves, other than the smart card which is controlled by their bank. This relationship is highly asymmetric and potentially adversarial.

The “electronic attorney”

Exploit the fact that EMV is not resistant to man-in-the-middle attacks:

- Customer inserts a shim between their card and the terminal
- This shim is purchased by the customer and produced by a 3rd party, so protects the customer’s interests
- It can have a button, display, and secure storage
- It can decode, block, delay and alter commands and responses
  - Relay attack can be defeated by displaying value before permitting transaction to proceed
  - PIN recovery, whether electronically or by camera, can be defeated by never entering the real one into a untrustworthy terminal (use one-time and/or value conditional ones instead)
  - The customer can unilaterally select which new security features to adopt (e.g. biometrics)
  - Disputes can be resolved though a secure audit log

Conclusions

- EMV has a number of weaknesses, especially SDA – some to reduce costs, others apparently unintentional
- These may be mitigated with other mechanisms (back-end controls, online authentication)
- So whether flaws in a protocol matter critically depends on who is liable for failures
- In EMV's case, the party who can improve matters (the banks) do not have the incentive to deploy defences
- Non-traditional defences, such as the electronic attorney, can put customers in a better position

For further information, see "Keep Your Enemies Close: Distance bounding against smartcard relay attacks" (USENIX Security '07)

Thanks to Ben Adida, Ross Anderson, Mike Bond, Saar Drimer and Markus Kuhn for their contributions to this project