EMV - the end of skimming?

Formal analysis of the EMV protocol suite

Erik Poll

Joeri de Ruiter

Digital Security group, Radboud University Nijmegen

Overview

- The EMV standard
- Attacking smartcard chips
- Some issues with EMV
- Formalisation of the EMV standard
- Future of skimming

EMV

- Started 1993 by EuroPay, MasterCard, Visa
- Common standard for communication between
  1. smartcard chip in banking or credit card (aka ICC)
  2. terminal (POS or ATM)
  3. issuer back-end
- Specs controlled by EMVCo which is owned by

- Over 1 billion cards in use
- EMV-compliance required for Single Euro Payment Area

Why EMV?

- Goal: reducing fraud by
  1. skimming
  2. stolen credit cards used with forged signatures
  3. card-not-present fraud (EMV-CAP)

- And also some transfer of liability?

EMV in the Netherlands

- EMV migration moved forward from 2013 to 2011

will disappear, to be replaced by

- Reason: increasing cost of skimming
  - 2007 : 15 M€
  - 2008 : 31 M€
  - 2009 : 38 M€

on a total >> 100 billion €, so fraud only around 0.03%

- Other countries already switched to EMV, eg. UK by 2006

The EMV protocol suite

- EMV is not a protocol, but a "protocol toolkit suite":

  many options and parameterisations (incl. proprietary ones)

  - 3 different card authentication mechanisms
    - SDA, DDA, CDA
  - 5 different card holder verification mechanisms
    - online PIN, offline plaintext PIN, offline encrypted PIN, handwritten signature, no card holder verification
  - 2 types of transactions: offline, online

  All these mechanisms again parameterised by Data Object Lists (DOLs)

- Specification public but very complex (>700 pages)

EMV CAP protocol

- use EMV chip for internet banking or e-commerce
  - EMV CAP defined on top of EMV:
    an EMV-CAP session is an *aborted* EMV session
  - internet banking
    - Mastercard : CAP (Card Authentication Program)
    - Visa : DPA (Dynamic Passcode Authentication)
  - e-commerce
    - Mastercard: SecureCode
    - Visa: Verified by Visa

- *CAP specs are secret but have been partially reverse-engineered*

Non-technical risks

- Liability shifts?
  - Client more likely to be held responsible for fraud by PIN
  - Also, merchant falling back on magstripe will be liable for fraud

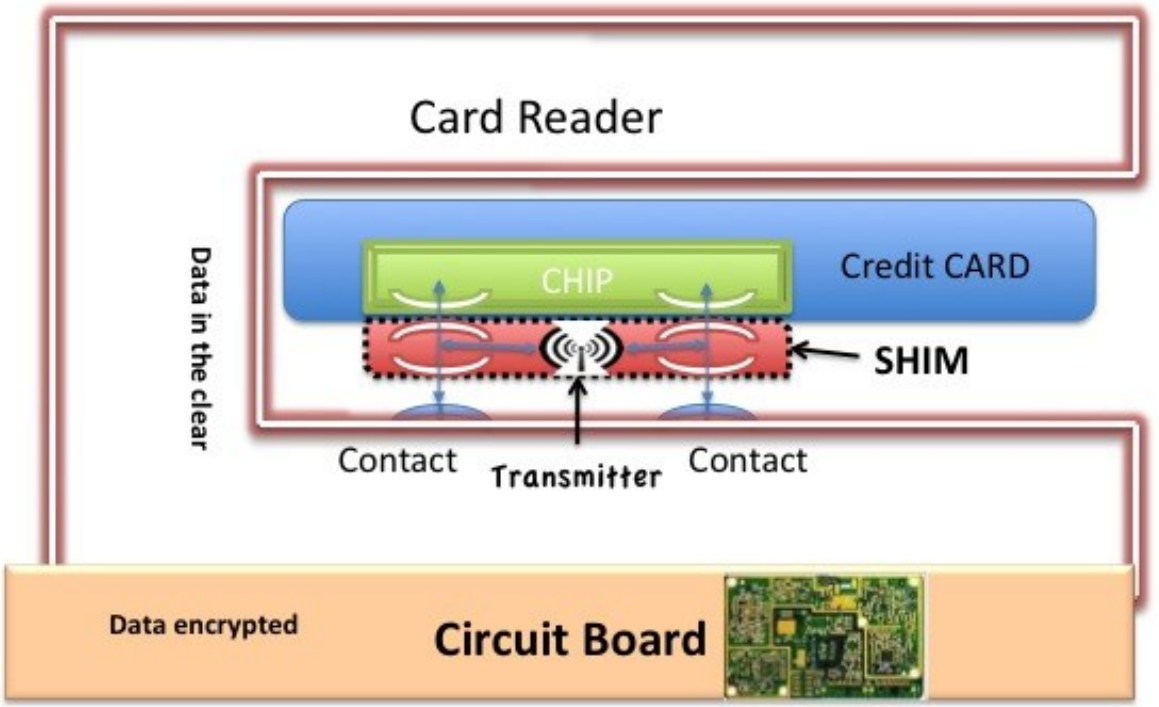
- Mugging?
  - CAP readers convenient for muggers to force people to reveal PIN

"Skimming" smartcards

het nieuwe skimmen

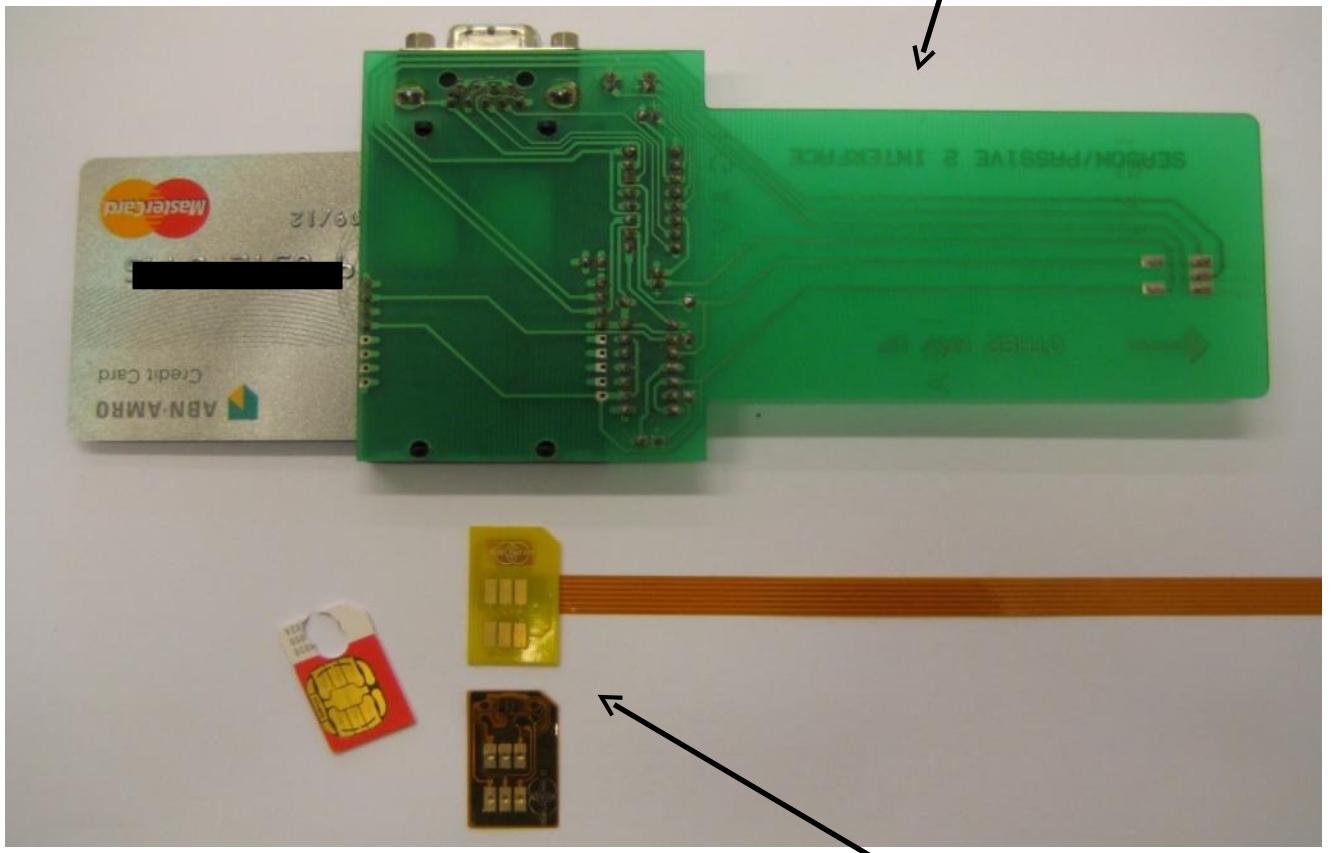
Attacking smartcard chips: passive

- chip cannot be copied like a magstripe, but communication with terminal can still be eavesdropped
  - using a shim, invisible inside terminal



Ready to use devices on sale

old-fashioned version
(mainly used for hacking pay TV)



newer, thin versions
(mainly used for SIM unlocking?)

Attacking smartcards: active

- Smarter shim allows an active man-in-the-middle attack
  - ie eavesdropping *and altering* messages
  - also called wedge attacks

- Two usage scenarios
  1. tampering with a terminal, which is then used by normal cards

     abusing access to the card and PIN code
  2. tampering with a card, which is then used at normal terminal

     acting as relay of (stolen?) genuine card to terminal

EMV weaknesses

Cloning SDA cards (for offline payments)

- SDA card authenticates by presenting digitally signed data to terminal
  - card presents $\{\text{card number, other card data}\}_{\text{signedByBank}}$
  - card cannot do asymmetric crypto, and cannot sign things itself
- SDA card can be cloned
  - clone will always say offline PIN succeeded
    and be used for any offline transactions

- *Reason for SDA: cards that cannot do asymmetric crypto are cheaper*
- *Recommendation to phase out SDA*
  - *Mastercard and Visa require DDA for offline capable cards issued after 1/1/2011. SEPA also wants them banned.*

DDA wedge attacks (for offline payments)

- DDA cards authenticates by challenge-response
  - DDA card can do symmetric and asymmetric crypto
- DDA card cannot be cloned
- but.. transaction is not tied to the card authentication
  - terminal cannot spot fake offline transactions,
    even though issuer later will

- CDA repairs this, by adding digital signature over transaction data
- *Recommendation to move to CDA, but nobody seems to be doing this yet*

"Chip & PIN is broken" [Murdoch, Drimer, Anderson, Bond, 2010]

- Terminal can be fooled into thinking a transaction was with PIN, while card & issuer know it was PIN-less
  - using a wedge attack
  - for online and offline transactions
  - root cause: terminal cannot authenticate response to offline pin verification

- This allows a stolen card to be used without PIN, but only
  - as long as the card is not reported stolen
  - if issuer allows PIN-less transactions (as is case in UK)

  or… if the issuer misses the correct checks for this in the back-end

Complexity of the EMV specs

- Moral of the story: specs too complex to understand
  - long specs, split over 4 books
  - little discussion of security goals or design choices
  - little abstraction or modularity

- Eg why not build on a notion of session level integrity & confidentiality as in SSL/TLS?
- Who really takes responsibility for ensuring these specs are secure? EMVCo, credit card companies, or banks?

Formalising EMV ?

- Lots of progress in past decade with formal, tool-supported methods to analyse security protocols.

  Can these help with protocols as complex as EMV?

- First attempt: formalising EMV in ProVerif

  Horrible! If-statements in applied pi-calculus cause huge duplication

- Second attempt: formalising EMV in F#

  Much better! F# allows sequential if-statements & functions

  F# can be translated to applied pi calculus by FS2PV tool

Formalisation of EMV in F#

Formalisation of EMV in F#

- EMV can be formalised in around 5 pages of F# code
  - including all options
  - remaining configuration (DOLs) fixed
    - we use those taken from Dutch bank/credit cards
- Translation to pi calculus explodes things a bit
  - 350 lines of F# becomes > 2.5 kloc of applied pi calculus
- But… ProVerif can still verify security properties
  - usually in minutes, but *this requires some care!*

Part of EMV model: DDA

```
// Perform DDA Authentication if requested, otherwise do nothing

let card_dda (c, atc, (sIC,pIC), nonceC) dda_enabled =

  let data = Net.recv c in

  if Data.INTERNAL_AUTHENTICATE = APDU.get_command data then

    if dda_enabled then

       begin   let nonceT = APDU.parse_internal_authenticate data in

               let signature = rsa_sign sIC (nonceC, nonceT) in

                Net.send c (APDU.internal_authenticate_response nonceC signature);

               Net.recv c

      end

    else  failwith "DDA not supported by card"

  else  data
```

Properties checked with ProVerif

Eg

- highest supported card authentication method used
- no replay of card authentication
- agreement between card and terminal on key actions, eg pin verification, transaction details, etc

Sample ProVerif query

query evinj:TerminalTransactionFinish(sda,dda,cda,pan,…)

==> evinj:CardTransactionInit(sda,dda,cda,pan,…).

No new attacks found, but all existing weaknesses are confirmed

Further work

- How far can we push ProVerif?
  - coping with SDA, DDA, and CDA in one go?
- Using F7 instead of ProVerif for verification?
  - F7 might give better response time
- Making F# model executable?
  - so that it can interact with real cards and terminals

The future of skimming

Skimming revisited (1)

- EMV chip cards still have a magnetic stripe…
  - Magstripe can still be cloned and used in countries that don't use the chip (notably USA)

- Skimming fraud with UK cards, in millions£

	2005	2006	2007	2008	2009
domestic	**79**	**46**	**31**	**36**	**25**
foreign	**18**	**53**	**113**	**134**	**56**

Skimming fraud with UK cards in Australia and Canada halved in 2009 after move to EMV there [Source: UK Payment Association]

Skimming revisited (2)

- Track 2 data on the magstripe can be read off EMV chip…
  - So a shim in a terminal can reconstruct a magstrip for (ab)use in countries that don't use chip
  - If the card uses offline plaintext PIN, shim can also eavesdrop on the PIN, and you won't need a camera
  - First incident with tampered CAP readers *inside* Dutch banks in summer 2009
  - Hot news: EMV specs are to be updated to avoid this

cross-channel attacks

- If attacker has compromised a terminal, attacking eBanking may be more lucrative than creating a counterfeit magstripe card to use in ATM
  - improved detection at ATMs has reduced fraud per skimmed Dutch bankcard from 1500€ in 2008 to €1000 in 2009

- Dually, eShopping (with EMV or IDEAL) may be much easier way to harvest card data than tampering with terminals

cross-channel possibilities

harvest data via eshopping?

traditional skimming

tampered CAP readers

use data for ebanking?

Conclusion

- Move to EMV can reduce skimming

  Main question: will skimmers move to the US?

  - for obtaining *and* using card data?
  - or just for using card data they obtain by skimming in Europe ?

- Too many eggs in the same basket?

  One card for ATMs, shops, eBanking and eShopping introduces potential for problems


4b
Tunnelv
Tickets
Bescherm uw
PIN-CODE
Werk aan het spoor
ProRail
Werk aan het spoor
NS Lenteto

Questions?