EMV

Erik Poll

Digital Security

Radboud University Nijmegen

Overview

- The EMV standard
- Known issues with EMV
- The EMV-CAP standard for internet banking

EMV

- Started 1993 by EuroPay, MasterCard, Visa
- Common standard for smart cards in banking: communication between
  1. smartcard chip in bank 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

Newer variants

- Contactless EMV

Visa payWave

MasterCard paypass

- Also via NFC phones
  - eg. Google Wallet

Google wallet

do you see anything strange?

do you see anything strange?

Skimming

- Attacker tampers with ATM to
  1. copy magnetic strip (mag-stripe)
  2. look at PIN code (with camera or fake keyboard overlay)

  Attacker can than make a cloned card and withdraw cash

example skimming equipment

compromised train ticket machine


LET OP! PIN-pas komt na invoer meteen terug.
Neem eerst uw pas uit en toets daarna uw PIN-code in.

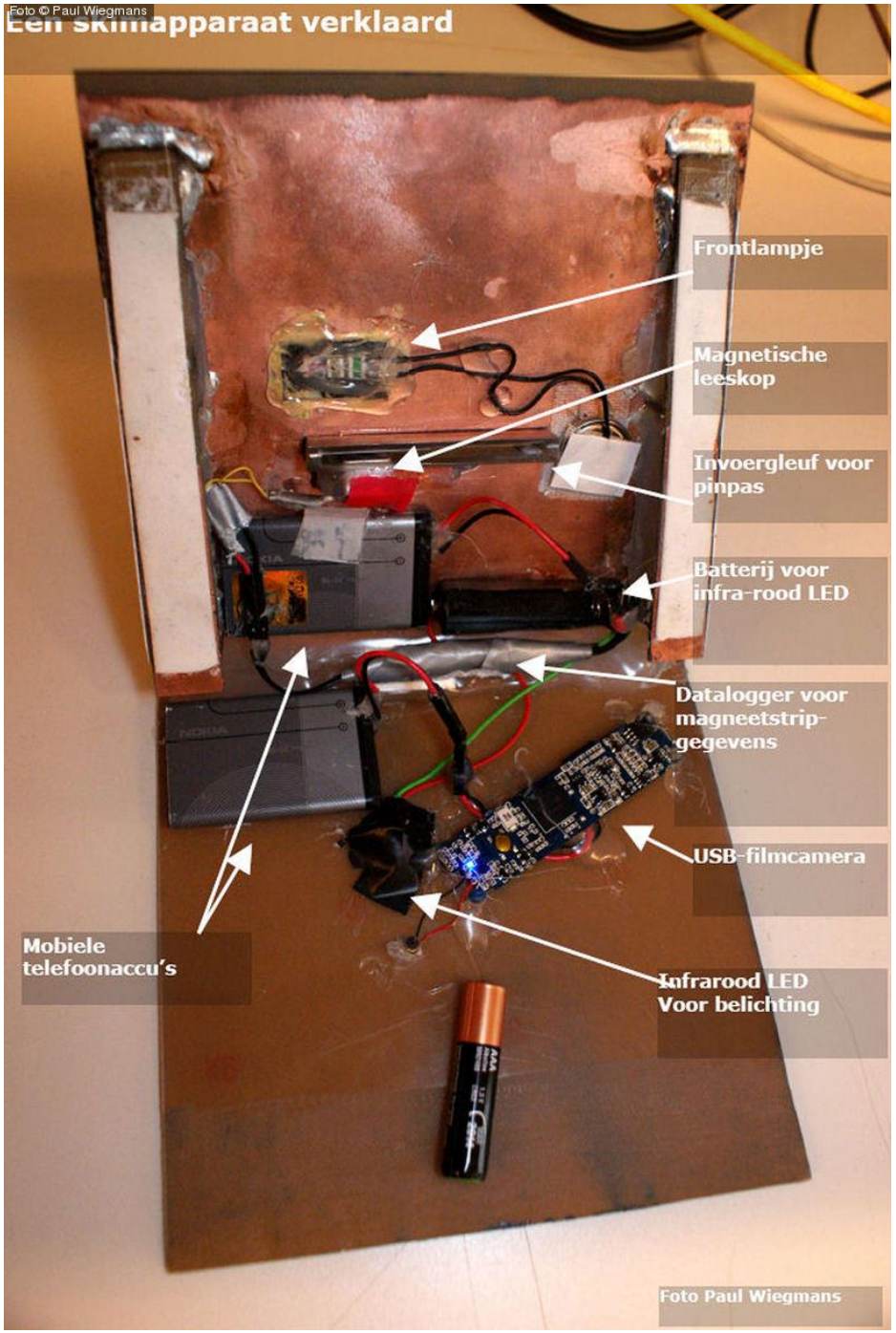
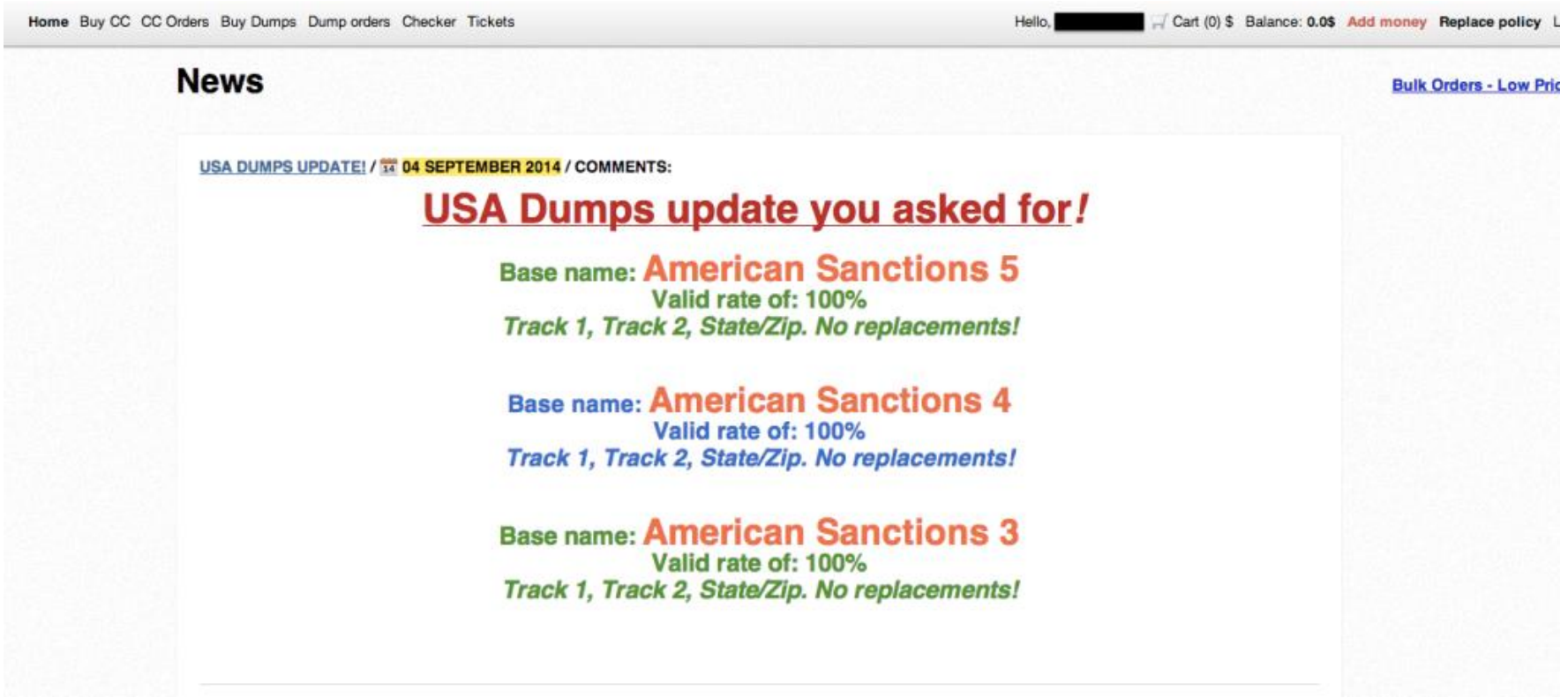


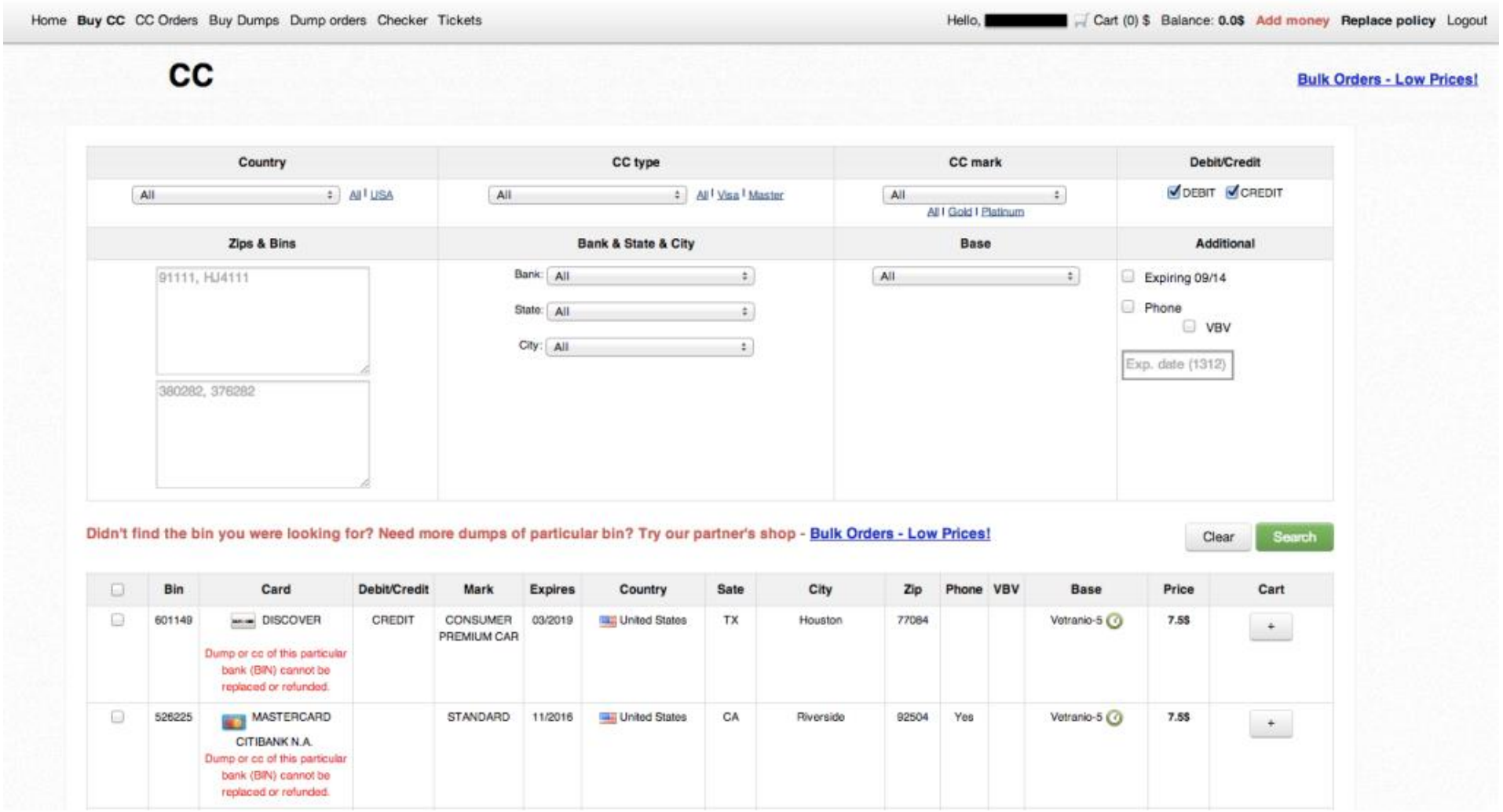
Foto © Paul Wiegmans
Een skimapparaat verklaard
Frontlampje
Magnetische leeskop
Invoergleuf voor pinpas
Batterij voor infra-rood LED
Datalogger voor magneetstrip-gegevens
USB-filmcamera
Mobiele telefoonaccu's
Infrarood LED Voor belichting
Foto Paul Wiegmans

or simply buy the data from someone else..

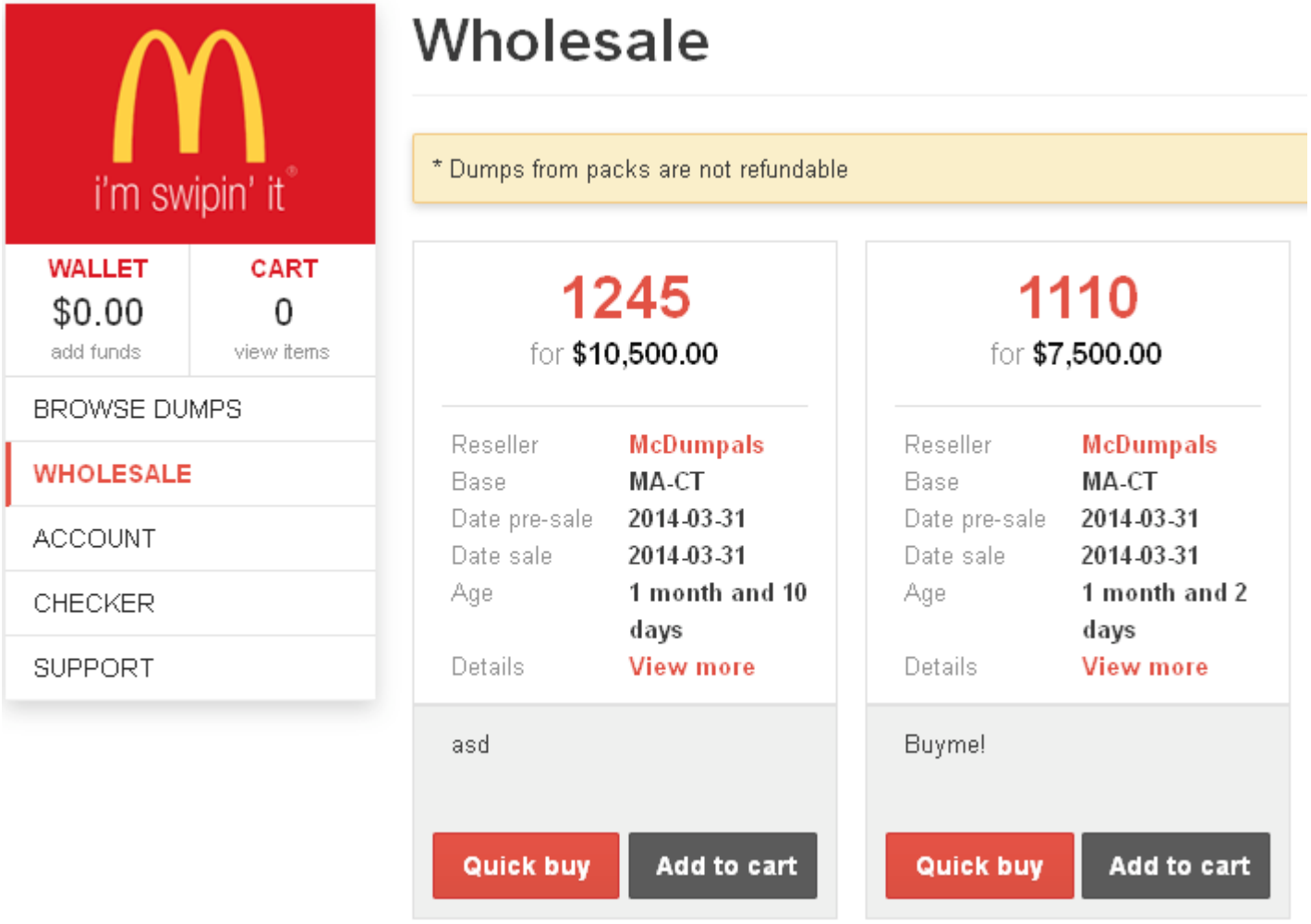
- Many web sites for trading card numbers, with/without address/post code, .., used for skimming or Card-Not-Present fraud (ie. online shopping)



web-interface for online criminal shop



Another criminal website: McDumpals



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?
  - client more likely to be held liable for fraud using PIN than fraud using signature
  - merchant falling back on magstripe instead of chip will be liable for fraud

Skimming in the Netherlands

- EMV migration moved forward from 2013 to 2011
- Reason: increasing cost of skimming
  - 2007 : 15 M€
  - 2008 : 31 M€
  - 2009 : 36 M€
  - 2010: 19.7M€
  - 2011: 38.9 M€

Cost per skimmed card went down thanks to better monitoring

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

But also: lot of hassle to customers & banks in blocking cards, issuing new cards, loss of confidence by consumers, …

Does EMV reduce skimming?

- UK introduced EMV in 2006

Skimming fraud with UK cards, in millions £

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

- Problem: magstripe can still be cloned and used in countries that don't use the chip (notably USA)
- Simple and effective approach taken in Netherlands now: by default, your bank card does not work outside Europe

The EMV standard

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 cardholder 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)
- Specs public but very complex (4 books, totalling >750 pages)

These specs do not motivate design or mention security objectives…

EMV basics: key set-up

- Card & issuer share shared symmetric key (3DES)
  Terminal (ATM or Point-Of-Sale) does *not* have this key
- Issuer has private/public keypair (RSA) used to sign data.
  Terminal *can* verify this signature, since it has the issuer's public key
- *Some* cards have a private/public keypair, used to sign data.
  Terminal can verify this signature, after card provides its certificate
  - only on (more expensive) cards that can do such asymmetric crypto; all Dutch bank cards can now do this

EMV protocol phases

I. Initialisation

Terminal reads some data from the card, incl. several DOLs

II. Card Authentication (using SDA, DDA or CDA)

III. Cardholder Verification (optional)

IV. Terminal & Card Risk Management

V. Transaction

where the card produces Application Cryptograms,

which are MACs (Message Authentication Codes) over some transaction data, encrypted with the shared symmetric key

NB terminal does not have this key, so it cannot check this MAC when it is offline

II. Card Authentication: SDA

1. SDA – Static Data Authentication

- SDA card cannot do asymmetric crypto
- Card presents static data (card no, expiry date etc) signed by issuer

  ie $\{\text{card no, expiry date, ...}\}_{\text{PUBKEY-ISSUER}}$

- Problem: can be replayed, so card can be cloned
  - of course, clone will always say offline PIN check succeeded
- Hence: *offline terminal can be fooled*
  - transaction is signed (MACed) using symmetric key, but terminal cannot check this MAC
  - issuer will spot this fraud later

II. Card Authentication: DDA

1. SDA – Static Data Authentication
2. DDA – Dynamic Data Authentication
   - card has (Pub,Priv) keypair and does challenge-response
     - requires more expensive card than SDA: one that can do asymmetric crypto
   - problem : card authenticated, but *not* the transaction
   - hence: *offline terminal can still be fooled*
     - issuer will spot fraud later

II. Card Authentication: CDA

1. SDA – Static Data Authentication
2. DDA – Dynamic Data Authentication
3. CDA – Combined Data Authentication
    - card has (Pub,Priv) keypair , as in DDA
    - signature now added over all the transaction data
        - so even an offline terminal can check authenticity

II. Card Authentication

1. SDA – Static Data Authentication
2. DDA – Dynamic Data Authentication
3. CDA – Combined Data Authentication

- Most cards in use today are SDA or DDA
- SDA is being phased out
  - eg Visa & Mastercard forbid issuance of offline capable SDA cards starting 1/1/2011

III. Cardholder Verification Mechanisms

1. PIN
   a. online: PIN checked by the issuer
   b. offline: PIN checked by the chip
      b1. unencrypted
          PIN could be eavesdropped using shim
      b2. encrypted (in keyboard of the ATM)
          requires a card that can do asymmetric crypto
2. Handwritten signature
3. Nothing

NB: only offline PIN involves the smartcard chip

PIN encryption (ie. offline encypted PIN)

- Encryption of PIN code in tamper-evident secure keypad
- Card issuers don't want to trust the entire ATM, but only
  - the Hardware Security Module (HSM)
  - this keypad

Cardholder Verification Methods (CVM)

- Range of cardholder verification methods
  - depending on card and the application
- Terminal and smartcard negotiate CVM
  - given their lists of rules, which specify allowed/supported method (in order of preference) with conditions (when this is allowed)
- Potential for trouble: forcing terminal/card to fall back to old CVM

CVM codes

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							Fail cardholder verification if this CVM is unsuccessful
	1							Apply succeeding CV Rule if this CVM is unsuccessful
		0	0	0	0	0	0	Fail CVM processing
		0	0	0	0	0	1	Plaintext PIN verification performed by ICC
		0	0	0	0	1	0	Enciphered PIN verified online
		0	0	0	0	1	1	Plaintext PIN verification performed by ICC and signature (paper)
		0	0	0	1	0	0	Enciphered PIN verification performed by ICC
		0	0	0	1	0	1	Enciphered PIN verification performed by ICC and signature (paper)
		0	x	x	x	x	x	Values in the range 000110-011101 reserved for future use by this specification
		0	1	1	1	1	0	Signature (paper)
		0	1	1	1	1	1	No CVM required
		1	0	x	x	x	x	Values in the range 100000-101111 reserved for use by the individual payment systems
		1	1	x	x	x	x	Values in the range 110000-111110 reserved for use by the issuer
		1	1	1	1	1	1	This value is not available for use

CVM condition codes

Value	Meaning
'00'	Always
'01'	If unattended cash
'02'	If not unattended cash and not manual cash and not purchase with cashback
'03'	If terminal supports the CVM [19]
'04'	If manual cash
'05'	If purchase with cashback
'06'	If transaction is in the application currency [20] and is under X value (see section 10.5 for a discussion of "X")
'07'	If transaction is in the application currency and is over X value
'08'	If transaction is in the application currency and is under Y value (see section 10.5 for a discussion of "Y")
'09'	If transaction is in the application currency and is over Y value
'0A' - '7F'	RFU
'80' - 'FF'	Reserved for use by individual payment systems

V. Transaction

- For the transaction the card generates cryptograms
  ie MAC (Message Authentication Code)
- For online transaction the card generates 2 cryptograms:
  the first is sent to the bank, and
  the card only generates second after receiving approval by the bank.
- For offline transaction the card just generates one TC cryptogram
  - A card may refuse an offline transaction, and force the terminal to go online

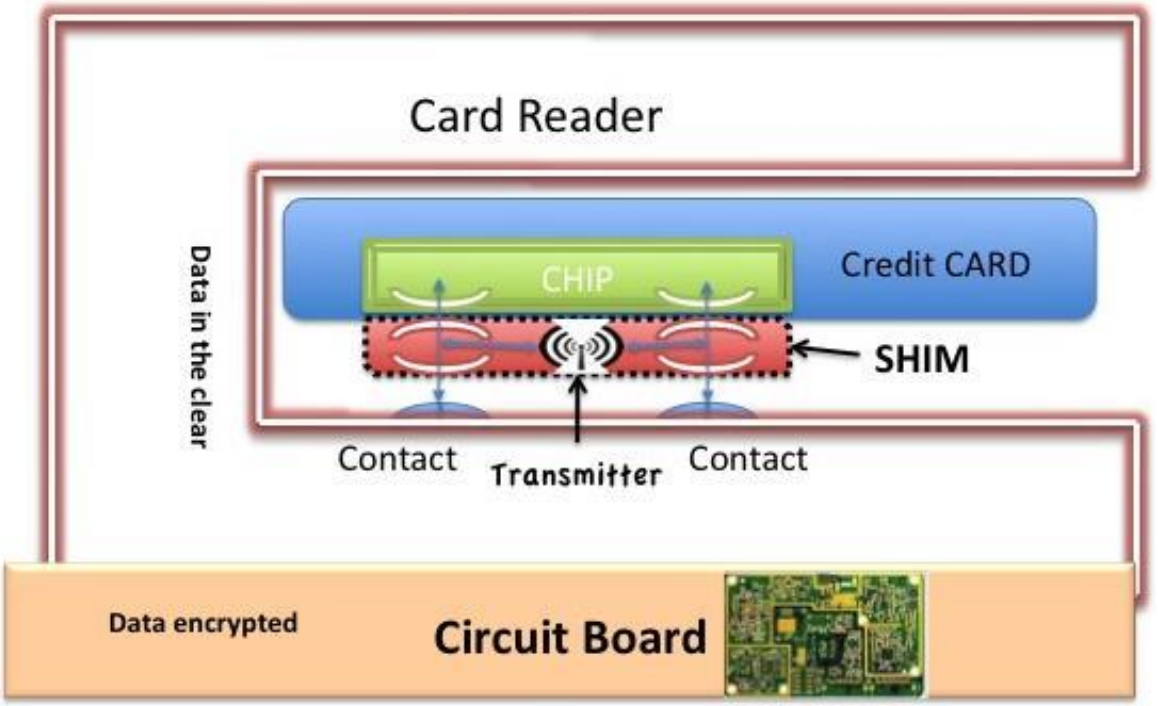
V. Transaction

- The data is included in the cryptogram is still configurable, using a DOL (Data Object List), but typically includes
  - the card's Application Transaction Counter (ATC)
    - a counter increased after each transaction
  - a terminal-generated nonce (aka Unpredictable Number)
  - the amount

Attack basics

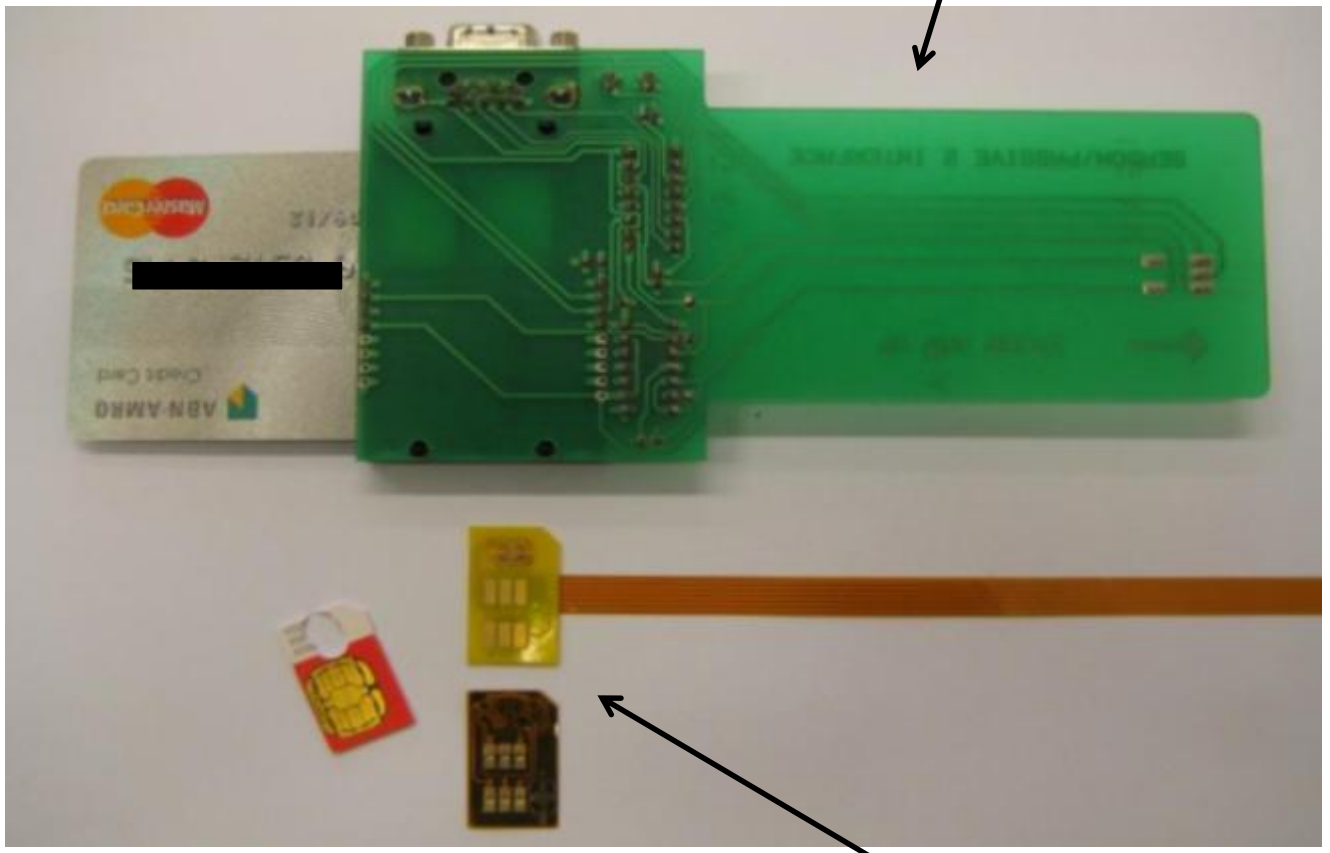
Man-in-the-Middle attacks

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



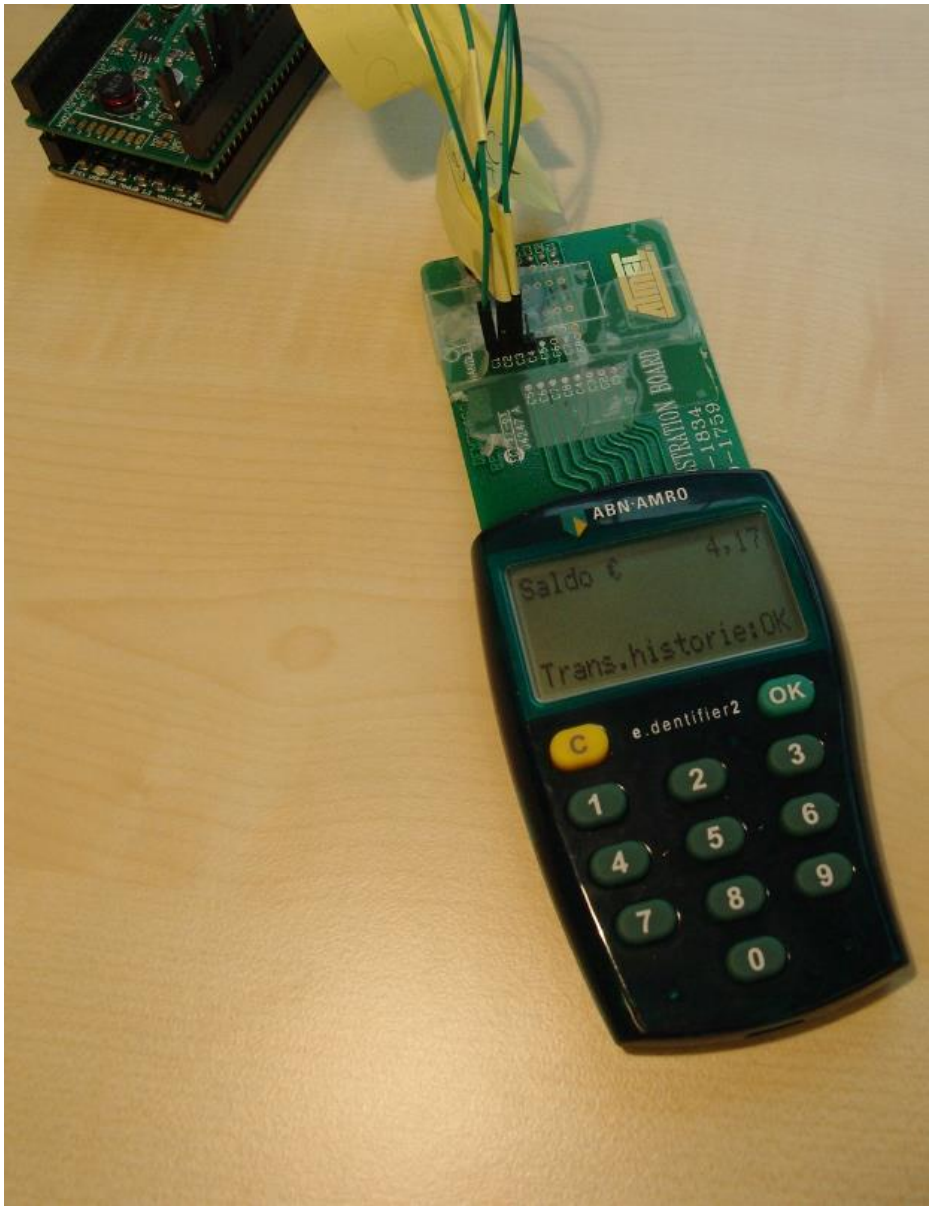
Ready to use devices on sale

old-fashioned version
(used for hacking pay TV)



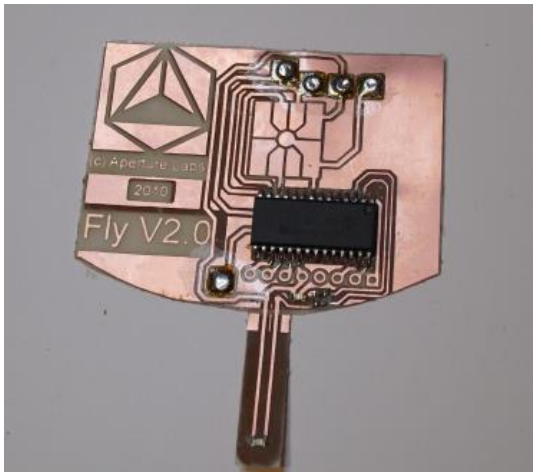
newer, thin versions
(used for studying SIM locking)

Tools for Active MitM



Smartlogic

by Gerhard de Koning Gans



Custom shim

by Inversepath.com

NB this fits inside terminal

EMV troubles…

Already discussed

1. SDA cards can be cloned

   and offline PIN with this clone will always succeed

   Fundamental limitation due to absence of asymmetric crypto

2. DDA card cannot be cloned, but with a real DDA card we can fool the terminal into accepting a bogus offline transaction

Backwards compatibility

- Bank card offers several communication channels
  - *visual, embossed and magnetic information on credit card*
- This may provide more security
  - *visual info helps against tampering with card, or faking one*

but may also allow more attacks

1. *classic example: copying your mag-stripe info onto my credit card*
2. *or: copying info from chip onto mag-stripe…*

3. Skimming revisited

Track 2 data on the magstripe can be read from the EMV chip…

- after eavesdropping with a shim in a terminal we can reconstruct a magstripe and use it in countries that don't use chip
- If the card uses offline plaintext PIN, shim can also eavesdrop on the PIN, so attacker does not need a camera
- First incident with tampered EMV-CAP readers *inside Dutch ABN-AMRO bank branche*s that did this in summer 2009; court case in 2011
- EMV specs have been updated to avoid this

4. faking “PIN ok” response

Terminal can be fooled into thinking a transaction was with PIN, while card & issuer know it was PIN-less

- using a wedge attack
- works for online and offline transactions
- root cause: terminal cannot authenticate the response to offline PIN verification (which is 0x9000 to say the PIN was ok…)

- 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 the case in UK)

  or… if the issuer misses the correct checks for this in the back-end
- Reportedly this won’t work in NL, as Dutch cards always do online PIN

[Murdoch et al., Chip & PIN is broken, FC’2010]

5. Eavesdropping PIN

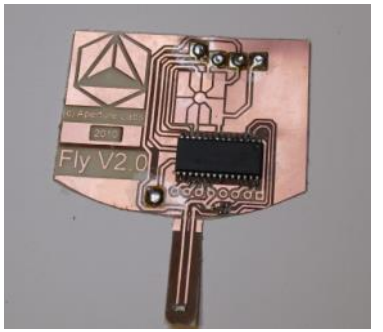
contact to eavesdrop on PIN

Problem with a common type of PED (Pin Entry Device)

- VISA claimed this was Common Criteria evaluated, but is lying?

[Drimer et al. ‚Thinking inside the box: system-level failures of tamper proofing, Security & Privacy 2008]

6. rollback to unencrypted PIN



- Shim can force a rollback to unencrypted PIN, by modifying card response to indicate the card does not support it
- Strangely, the terminal can tell the card is lying, because the signature over some card data is incorrect, but does *not* abort transaction!
- Not so serious, since
  - just having the PIN of an DDA card is useless without the card
  - the attack is detectable in the back-end
- Reportedly, most terminals in NL patched to disallow this rollback
  - but not the one terminal where we successfully tried this …

[Barisani et at, Chip & PIN is definitely broken, DEFCON 2011]

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
- Who really takes responsibility for ensuring these specs are secure? EMVCo, the credit card companies behind EMVCo, or individual banks?

Formalisation of EMV in F#

Analysis of the F# model

- F# can be translated to pi calculus by FS2PV tool and then analysed using ProVerif
- ProVerif can still verify security properties
  - usually in minutes, but *this requires some care*
  - No new attacks found, but existing attacks inevitably (re)discovered

EMV-CAP

EMV CAP protocol

- use EMV chip for internet banking or e-commerce
  - challenge-response mechanism using the bank card
- EMV CAP is defined on top of EMV
- internet banking
  - Mastercard : CAP (Card Authentication Program)
  - Visa : DPA (Dynamic Passcode Authentication)
- online shopping
  - Mastercard: SecureCode
  - Visa: Verified by Visa
- *EMV CAP specs are secret but have been largely reverse-engineered*

Reverse engineering EMV-CAP

Internet banking fraud in the Netherlands

2008	2.1 M€
2009	1.9 M€
2010	9.8 M€ (7100€ per incident)
2011	35 M€ (4500€ per incident)
2012	34.8 M€
2013	9.48 M€

[Source: NVB]

by infected computers, fake websites, or by phone

NB this is serious organised crime, not done by clever teenagers

Banks have been fighting the problem by *better detection & reaction*

Problems & limitations of EMV-CAP?

1. A non-technical risk: mugging?
   - CAP readers convenient for muggers to force people to reveal PIN
2. A serious limitation:

EMV-CAP does not protect against :Man-in-the-Browser attacks and Social Engineering attacks by telephone on customers

- Basic problem: the user types in meaningless challenges & responses, and still has to trust a PC terminal to see which transfer he is approving

internet banking

This reader can be trusted.
But can the user understand
the meaning of these numbers?

Computer display of
cannot be trusted
(despite)

→ 23459876
← 123654

internet banking

This display can be trusted & understood

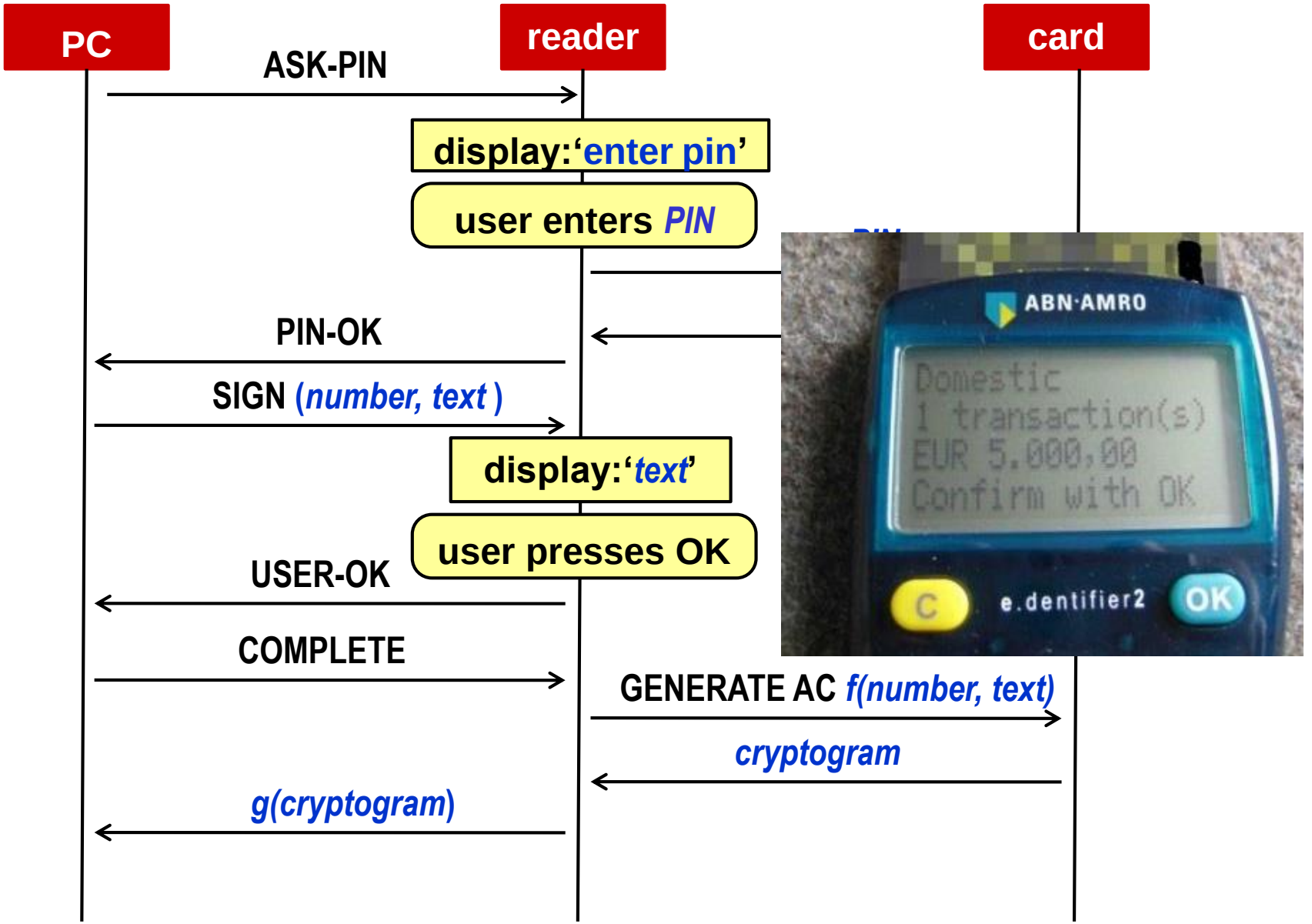
“What You Sign is What You See” (WYSIWYS)

USB

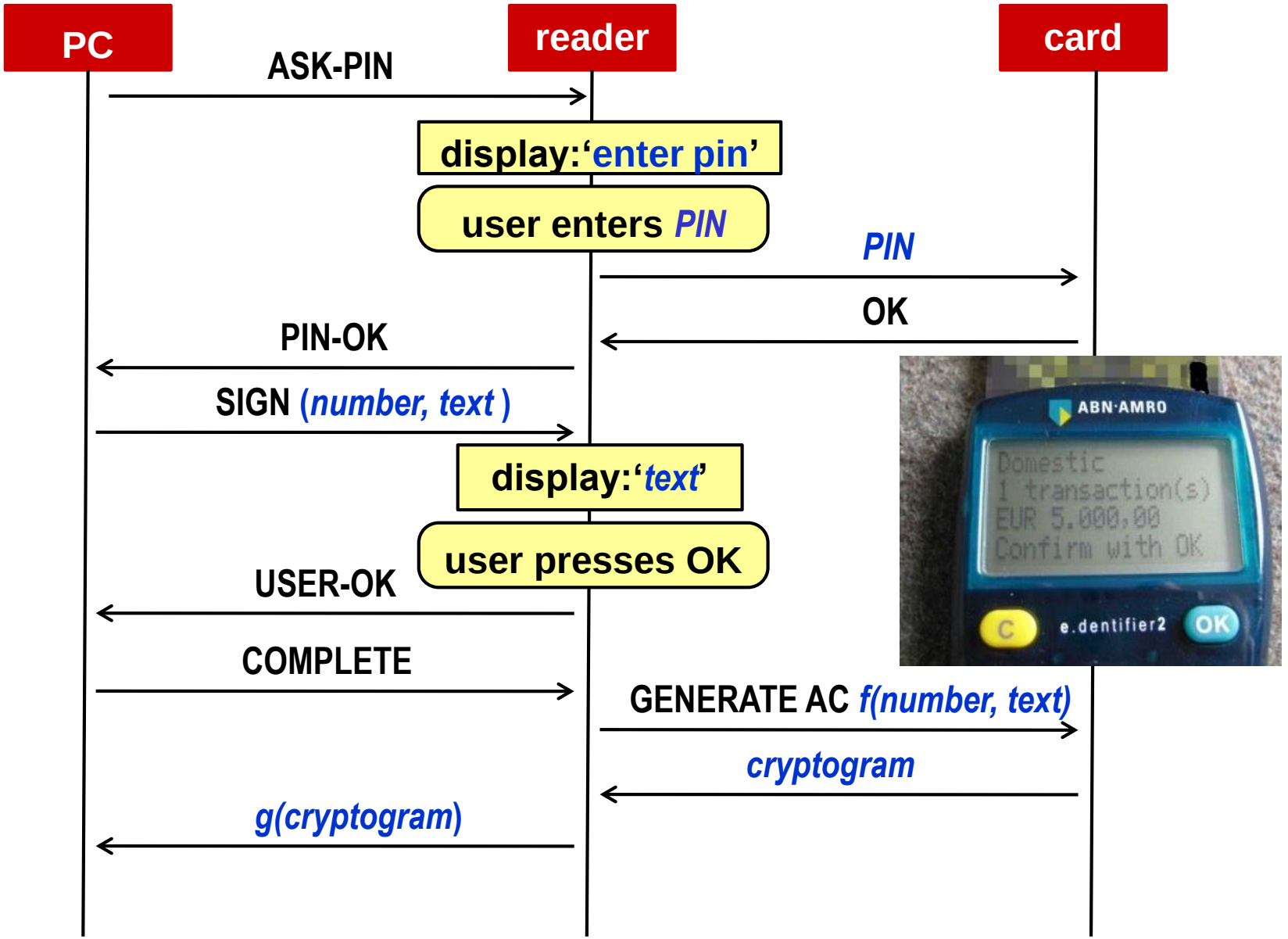
Analysis: first observation

- Text for display goes in plain-text over USB line
- So the PC can show
  - messages predefined in the e.dentifier2
  - any message that it wants to be signed

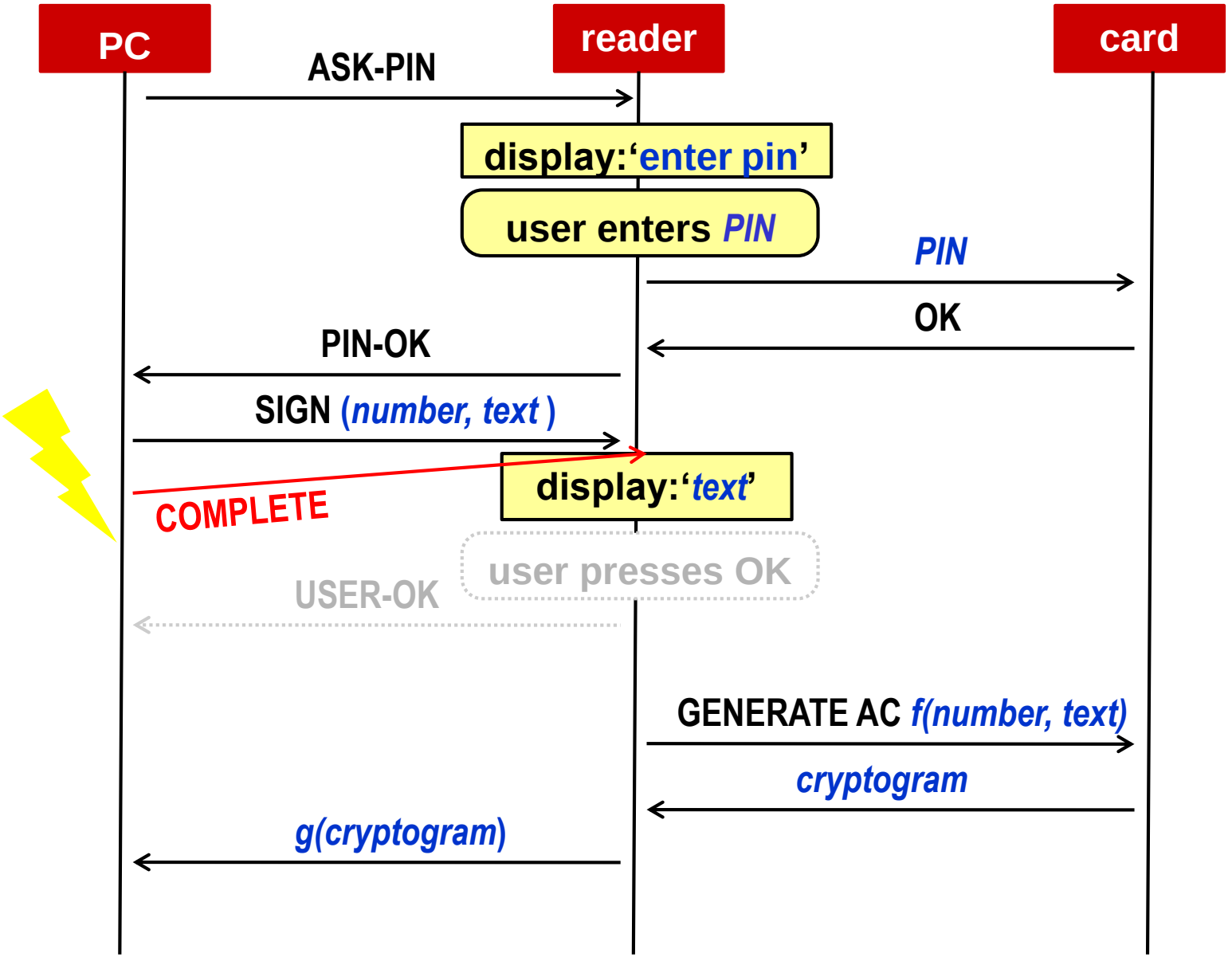
Reverse-Engineered Protocol



Reverse-Engineered Protocol



Attack!


PC
reader
card
ASK-PIN
display:‘enter pin’
user enters PIN
PIN
OK
PIN-OK
SIGN (number, text)
COMPLETE
display:‘text’
user presses OK
USER-OK
GENERATE AC f(number, text)
cryptogram
g(cryptogram)

Problem with Todos/Gemalto e.dentifier2

It’s possible to press OK via
USB cable...

Malware on an infected PC could change all the transaction details and press OK

See MSc thesis of Arjan Blom, 2012
or
[Designed to Fail: a USB-connected Reader for Online Banking, NORDSEC 2012]

Conclusion

Conclusions about EMV

- EMV protocol suite is far too complicated
  - too many options, written down in confusing way
- Move to EMV can reduce skimming

Main question: when skimmers move to the USA, will they

- skim cards in the USA *and* using card data?
- skim cards here, and use the cloned cards in USA?

- Too many eggs in the same basket?
  - One card, with one PIN code, for ATMs, shops, e-banking and e-shopping introduces potential for problems.
  - Organised crime is seriously investing in attacking internet banking

Conclusions about the banking world

- Banks routinely screw up their security protocols

- Not clear who is taking responsibility for checking security
  - MasterCard and Visa?
  - EMVCo ?
  - individual banks?
  - their payment processor ?
  - suppliers of cards and terminals ?
  - the Dutch National Bank?

Moral of the story

- Keep it simple!
- Protocols should only have *one* version/variant, namely the secure one
- Be careful what you trust; a well-known logo or supplier is no guarantee for good security