Cyber bank robbery

Erik Poll

Digital Security group

Radboud University, Nijmegen, the Netherlands

Biggest cyber bank robbery to date

$ 951 million stolen via SWIFT global payment system from the Bangladesh Central Bank

- **Most of the money recuperated**
- **'Only' $ 81 million really lost, via casinos on the Philippines**
- **Attackers installed custom malware on computers at bank & clearly had insider knowledge**
  - **malware removed transactions from local database & physical print outs**

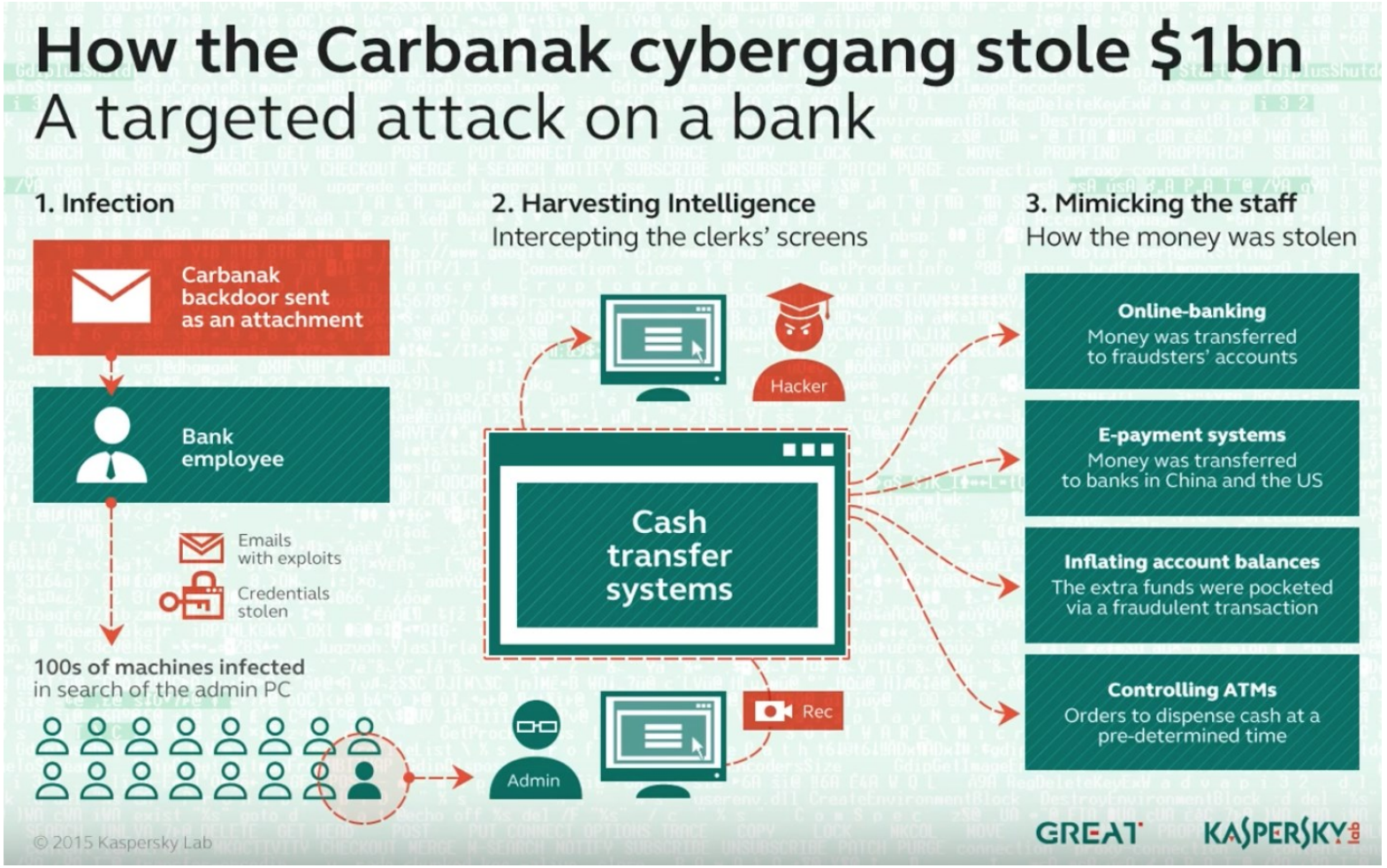
These are no script kiddies, but serious organised crime

[http://baesystemsai.blogspot.com/2016/04/two-bytes-to-951m.html]

[http://www.reuters.com/assets/iframe/cmsyovideo?videoId=370707923]

[https://www.nettitude.com/wp-content/uploads/2016/12/Nettitude-SWIFT-Threat-Advisory-Report-client.pdf]

Carbanak hack



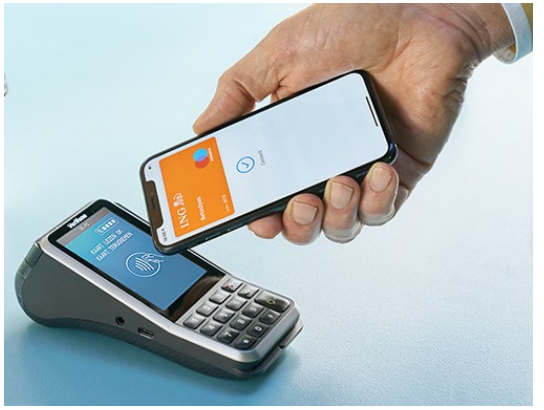
Darknet Diaries podcast **https://darknetdiaries.com/episode/35**

Overview

1. Skimming

2. EMV and complexities of EMV

3. Online banking



4. Contactless payments

Things that go wrong: Complexity, backward compatibility

Techniques to combat this: formal specification using finite state machines, fuzzing

Skimming

Skimming

Magnetic-stripe (mag-stripe) on bank card contains digitally signed information

but… this info can be copied

Do you see anything suspicious?

Skimming

Camera to see
PIN being entered

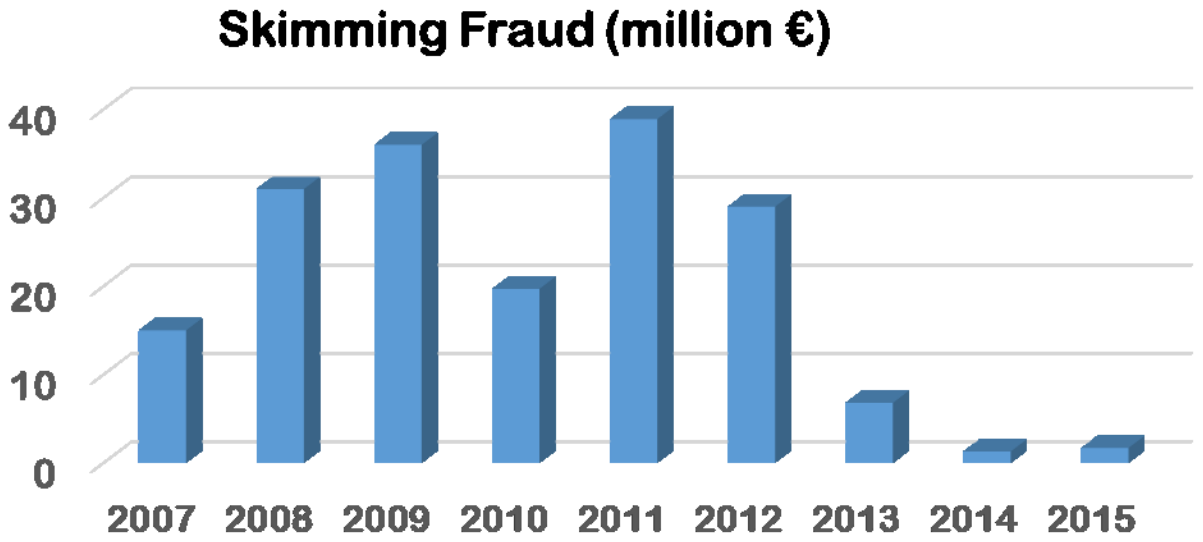
Fake cover
that makes
copy of the
magnetic stripe

More skimming equipment

Fake keyboard
to intercept PIN code

Fake cover
that copies magnetic stripe

Skimming fraud in the Netherlands



[Source: NVB & Betaalvereniging]

Fraud under control thanks to

- better monitoring & response (incl. blocking cards)
- replacing of mag-stripe by chip  in 2012

EMV (Europay-Mastercard-Visa)

- Standard used by all chip cards for banking
- Specs controlled by EMVCo which is owned by

- Unlike magstripe, a smartcard cannot be cloned

challenge c

c encrypted with K

- Payment terminal sends a different challenge c every time, so card gives a different response each time
- Card proves it knows the secret key K without revealing it

Does EMV chip  reduce skimming?

- UK introduced EMV in 2006

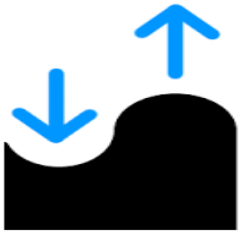
	2005	2006	2007	2008
domestic	79	46	31	36
foreign	18	53	113	134

Skimming fraud with UK cards, in millions £

Copied magstripes can still be used in countries that don't use the chip

- Blocking cards for use outside EU (geoblocking) helps a lot!
- Skimmers have now moved to the US, and the US is now migrating to EMV

Such water bed effects are a recurring phenomenon

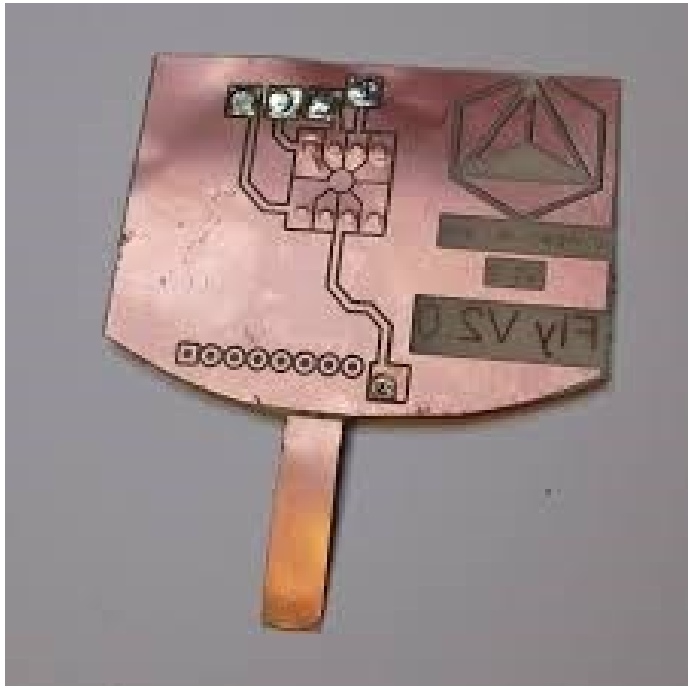


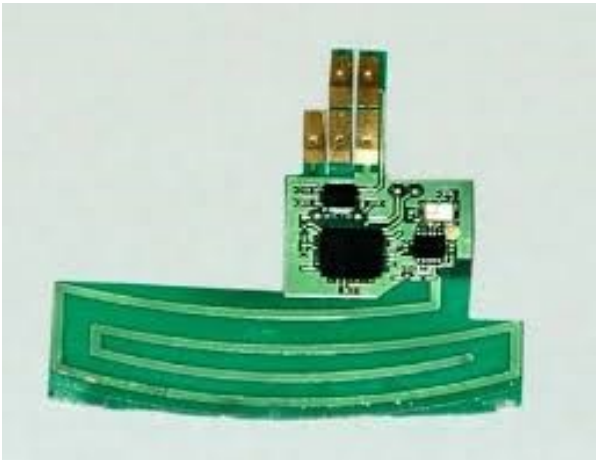
Recurring problem: BACKWARD COMPATIBILITY

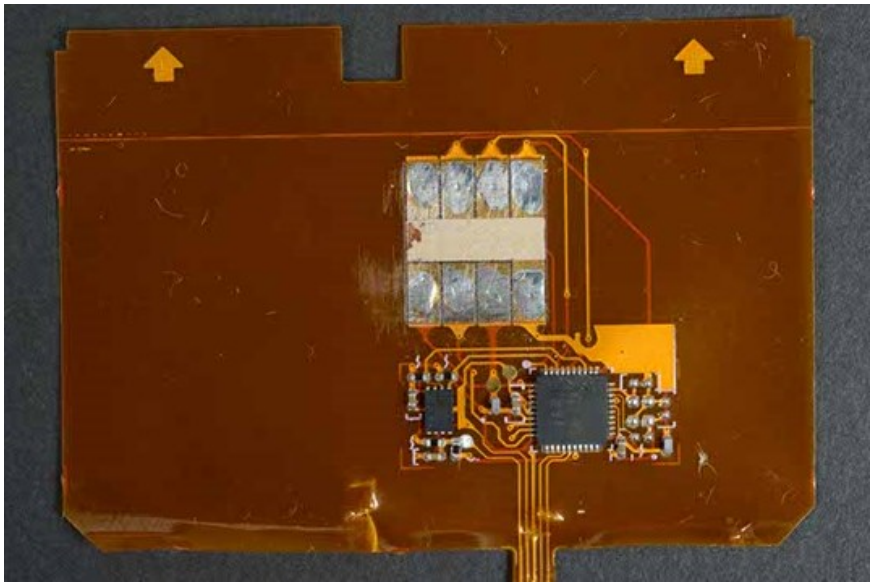
- In 2009, criminals put tampered card readers *inside* Dutch bank branches to skim cards
  - For *backwards compatibility*, the chip reports the mag-stripe data…
  - Both mag-stripe data and PIN code sent unencrypted from card to this reader
  - Criminals caught & convicted in 2011

- Cards have been improved to avoid this:
  mag-stripe data should now be different from info on the chip

Shims to eavesdrop on communication







https://krebsonsecurity.com/tag/atm-shimming/

Problem: COMPLEXITY

EMV is not a protocol, but a 'protocol toolkit suite' with *lots* of configuration options

- **Original EMV specs : 4 books, > 700 pages**
  - **3 types of cards (SDA,DDA, CDA), 5 authentication mechanism (online PIN, online PIN, offline encrypted PIN, signature, none), 2 types of transactions (offline, online), ….**
- **Contactless EMV: 7 books, > 2000 pages**

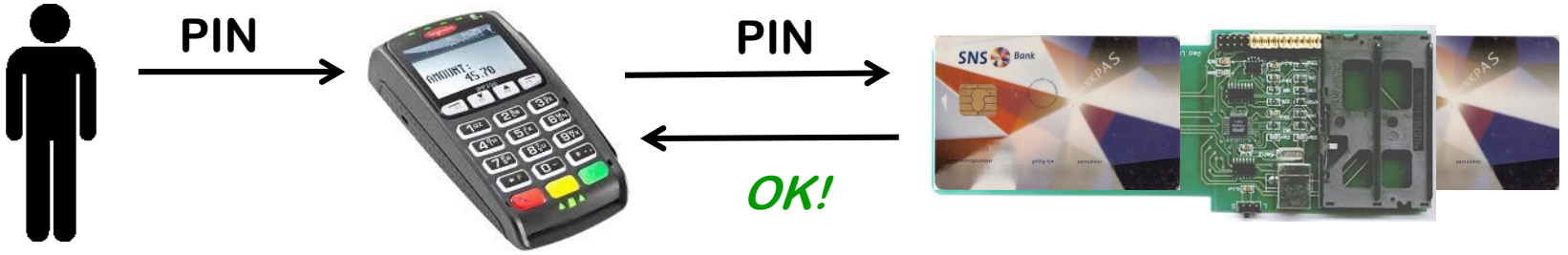
Sample sentence

> **"If the card responds to GPO with SW1 SW2 = x9000 and AIP byte 2 bit 8 set to 0, and if the reader supports qVSDC and contactless VSDC, then if the Application Cryptogram (Tag '9F26') is present in the GPO response, then the reader shall process the transaction as qVSDC, and if Tag '9F26' is not present, then the reader shall process the transaction as VSDC."**

Complexity: example protocol flaw

Terminal can choose to do offline PIN

- ie. terminal asks the card to check the PIN code



The response of the card is *not authenticated*

(not cryptographically signed)

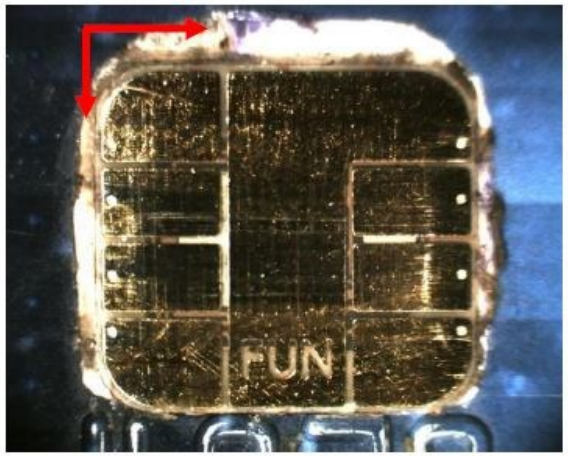
so terminal can be fooled by a Man-in-the-Middle attack

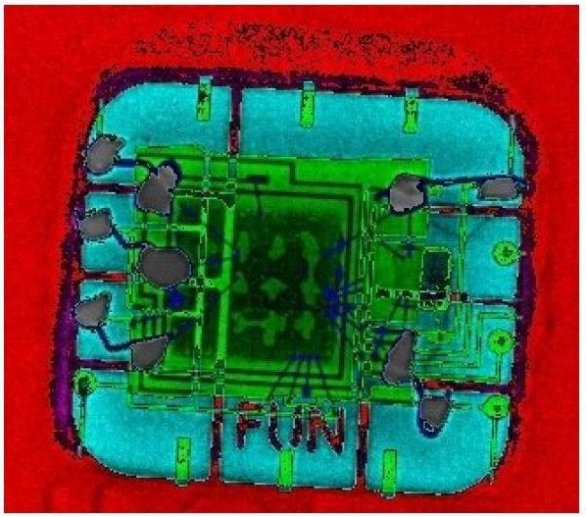
The transaction data will reveal the transaction was PIN-less,
so the bank back-end will know the PIN was *not* entered

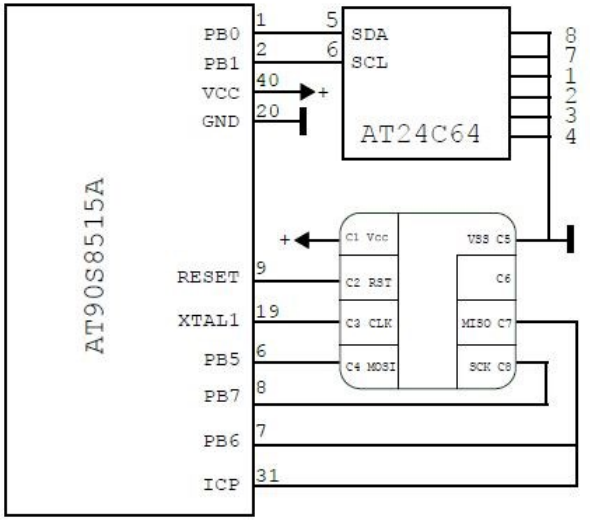
[Stephen Murdoch et al., *Chip & PIN is broken*, FC'2010]

Criminal Man-in-the-Middle set-up

Chips from stolen cards inserted under another chip, which faked the PIN OK response







xray reveals green stolen chip under blue microcontroller

[Houda Ferradi et al., *When Organized Crime Applies Academic Results: A Forensic Analysis of an In-Card Listening Device*, Journal of Cryptographic Engineering, 2015]

Complexity of EMV specs

- Specifications very complex to understand
  - long documents
  - no 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?

- Can we provide some scientific rigour?

Formalisation of EMV in F#

Essence of EMV in functional programming language F#

Formal Analysis of EMV

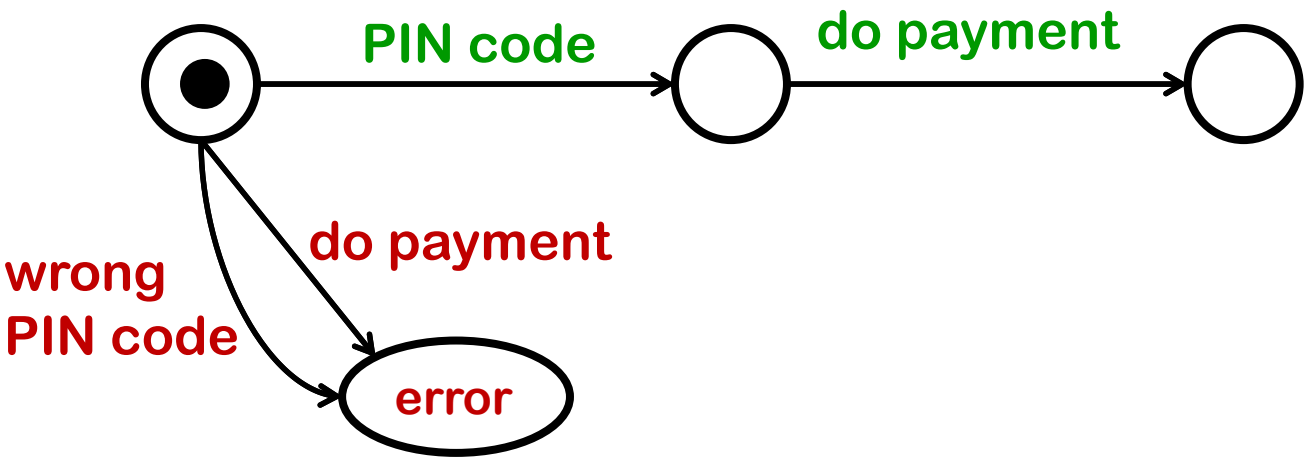
- Essence of EMV (all variants) can be formalized in less than 700 lines of F# code
- This model be analysed for security flaws using ProVerif tool
- No new attacks found, but existing attacks inevitably (re)discovered

[Joeri de Ruiter and Erik Poll, *Formal Analysis of the EMV protocol suite*, TOSCA 2012]

This still leaves the question if the software implementing these standards is correct!

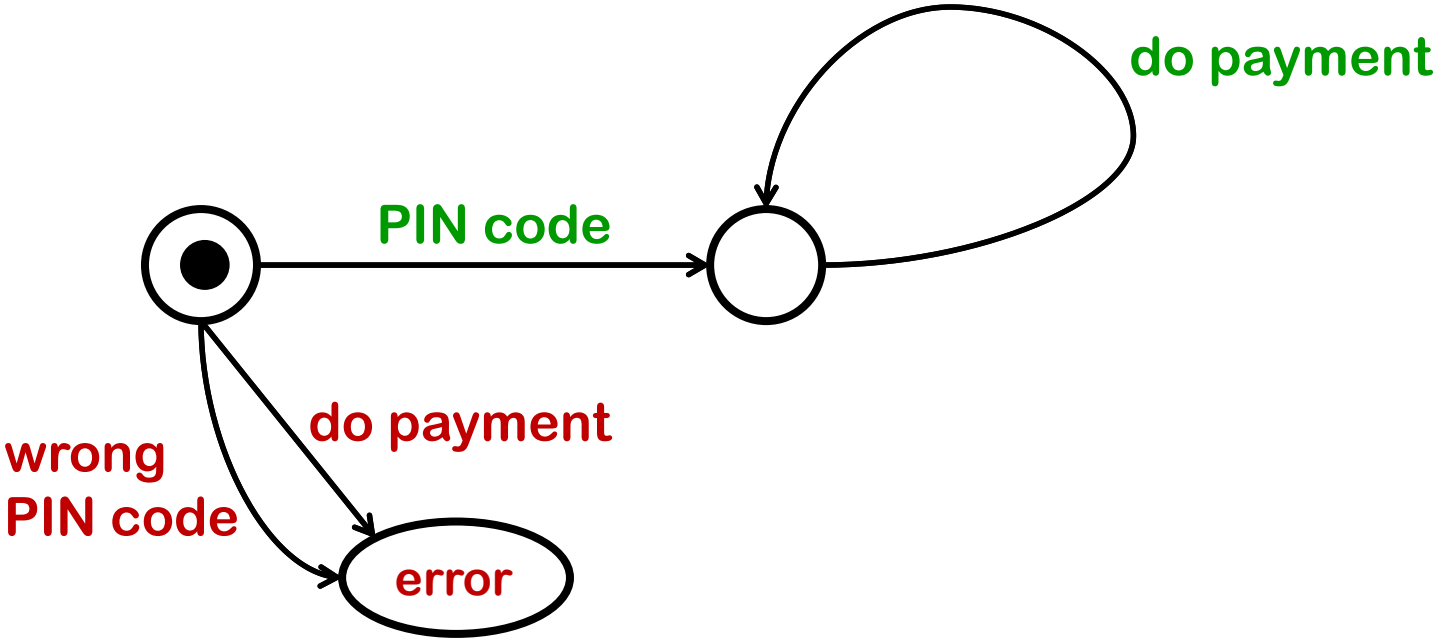
Finite State Machines

A bank card, like any program implementing a protocol, implements a finite state machine

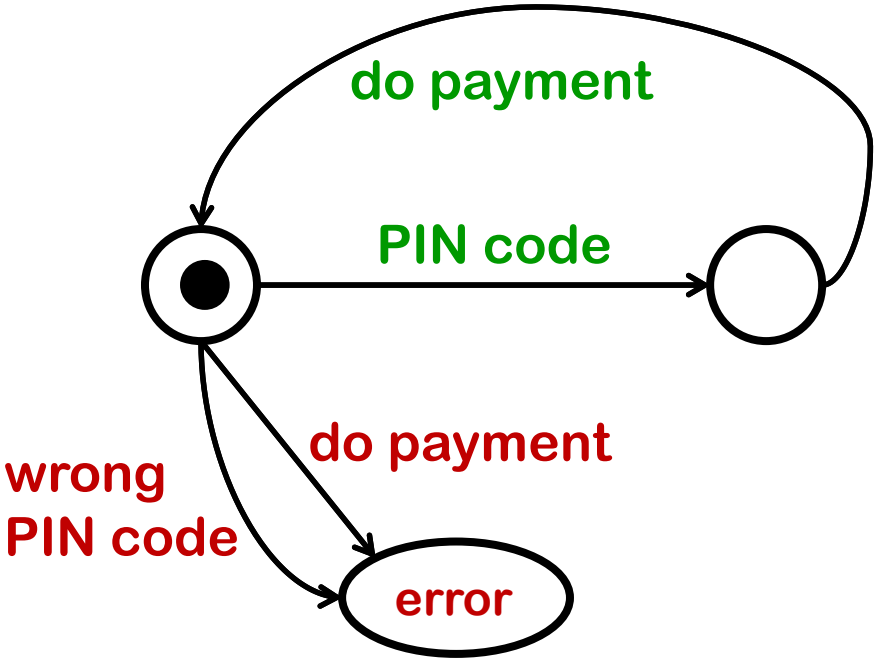


Such a state machine specifies (dis)allowed sequence of actions

Finite State Machines



Finite State Machines

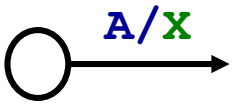


Finite state machines is a great formalism!
Easy to understand & precisely
captures subtle differences

Automated inference of state machines

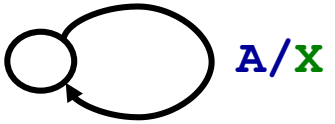
Just try out many sequences of inputs, and observe outputs

Suppose input A results in output X



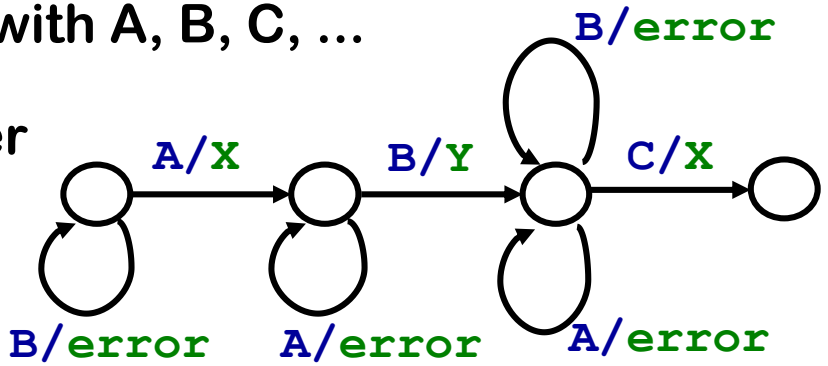
- If second input A results in *different* output Y

- If second input A results in the *same* output X



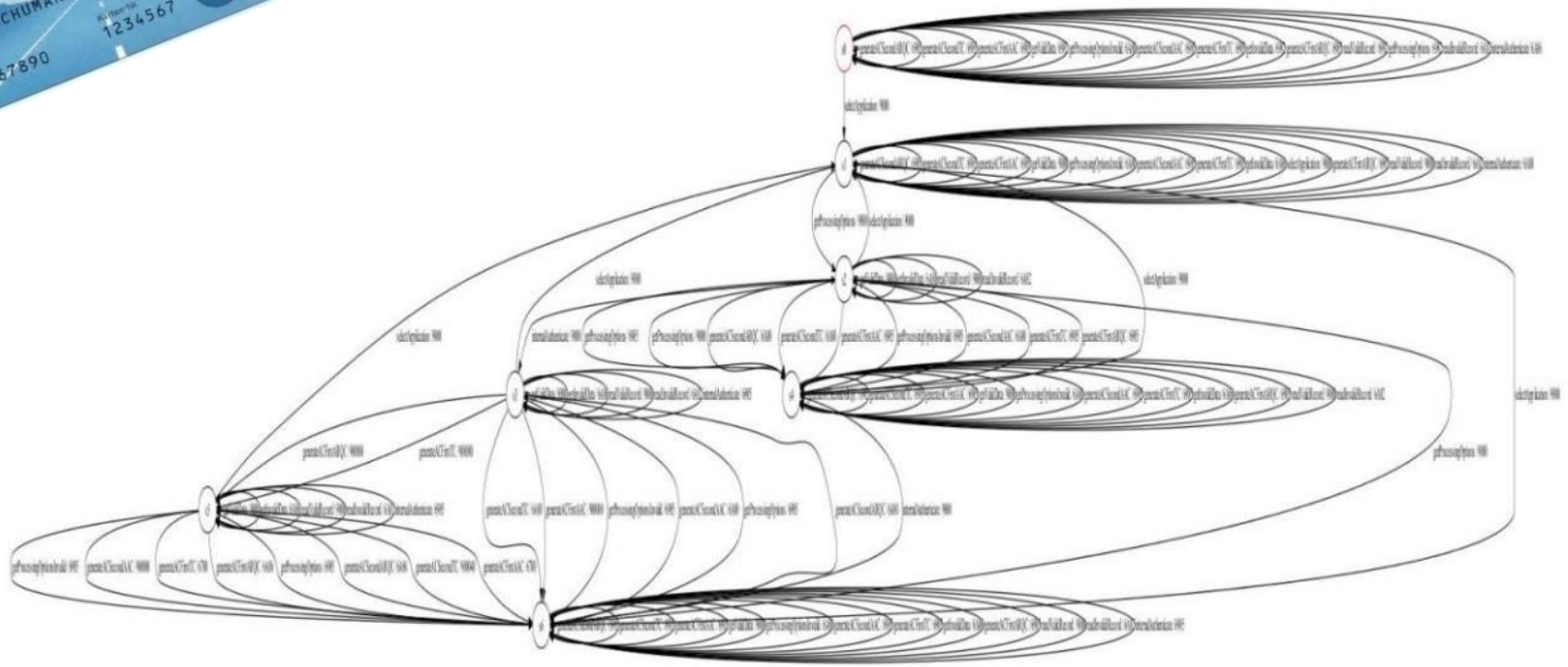
Now try more sequences of inputs with A, B, C, ...

to e.g. infer



The inferred state machine is under-approximation of real system

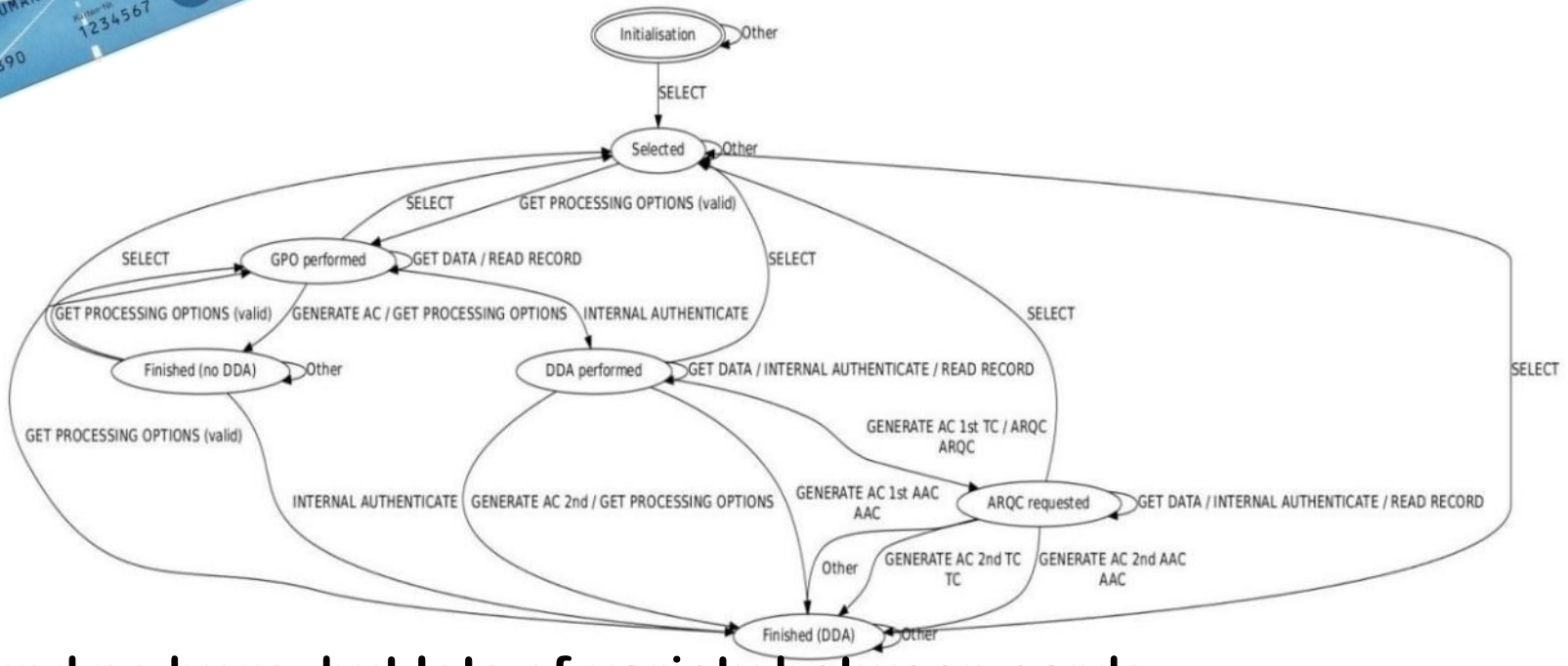
State machine inference of Maestro card



Using the LearnLib tool

State machine inference of Maestro card

merging arrows resulting in error response

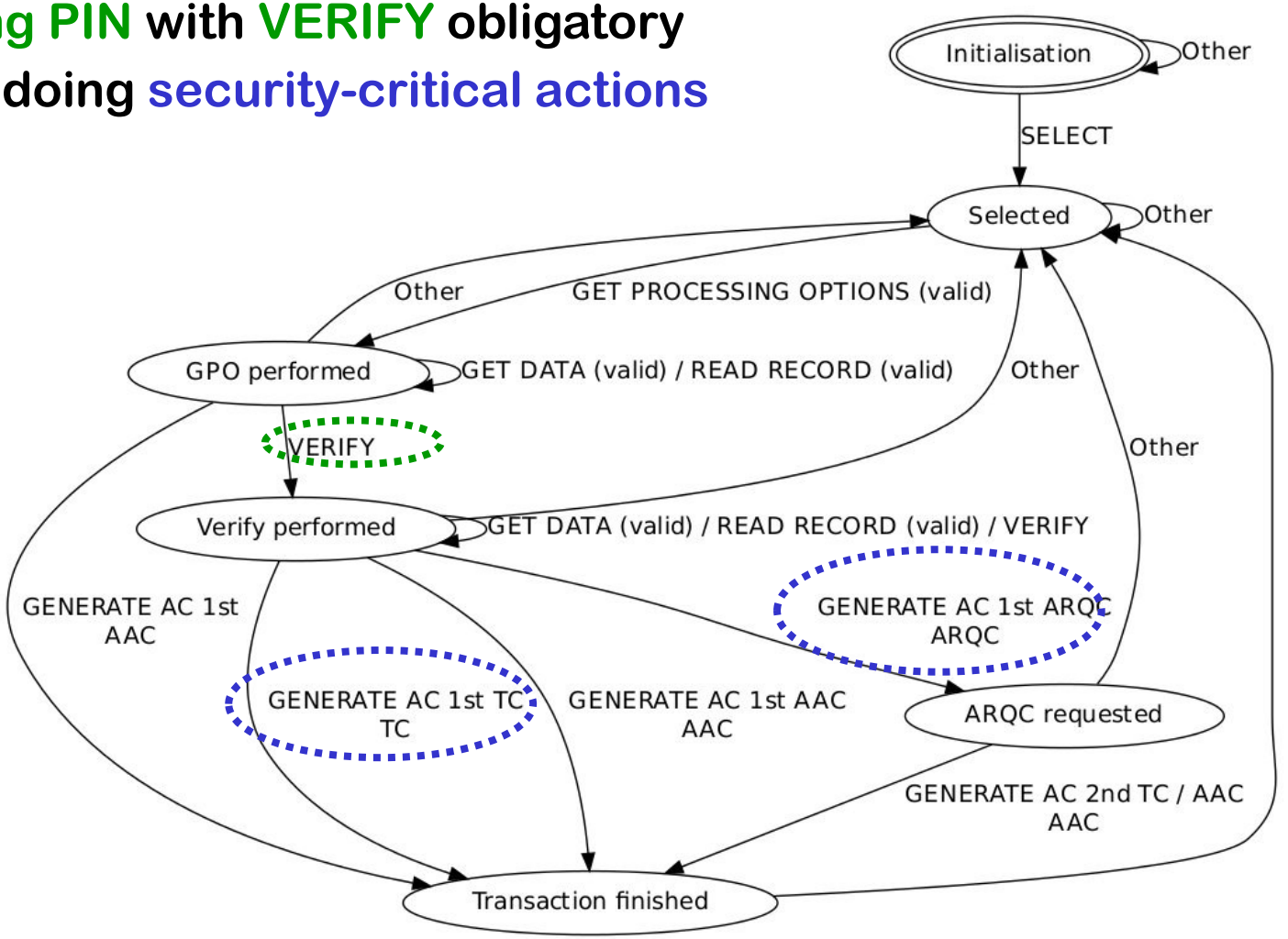


We found no bugs, but lots of variety between cards.

[Fides Aarts et al., Formal models of bank cards for free, SECTEST 2013]

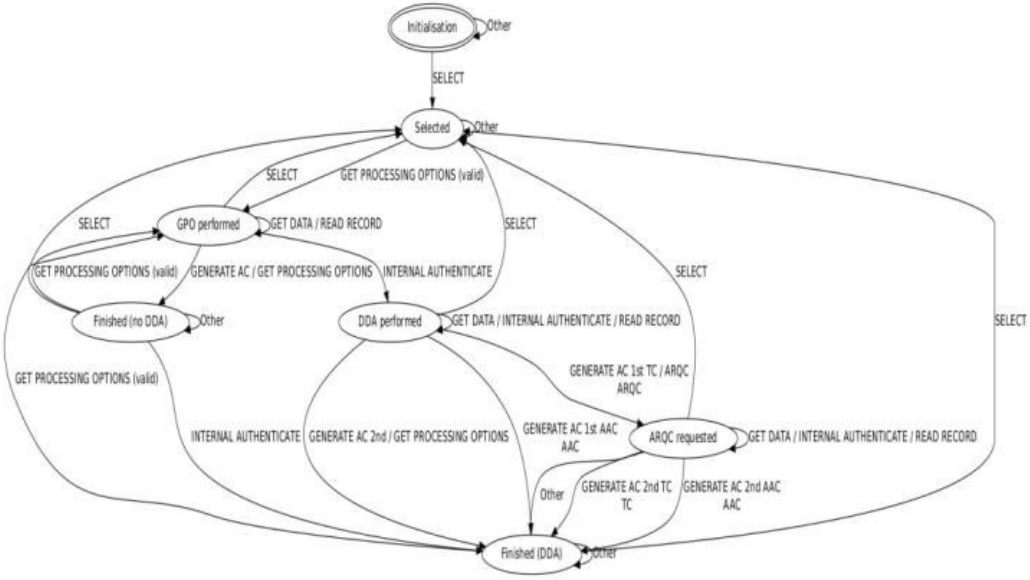
Using state machine to check security properties

entering PIN with VERIFY obligatory before doing security-critical actions

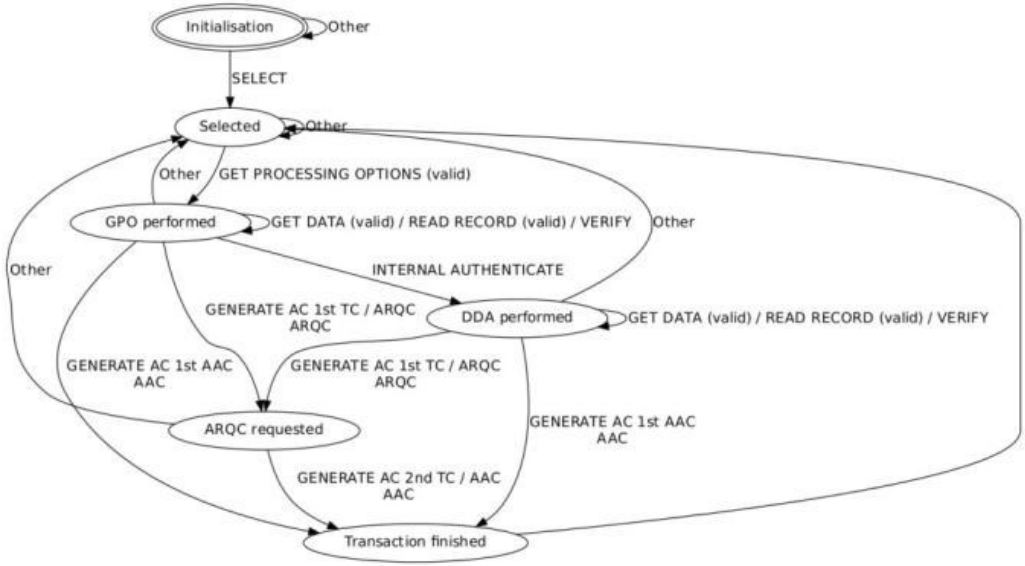


State machine of SecureCode application on Rabobank card

Understanding & comparing EMV implementations



Volksbank Maestro implementation



Rabobank Maestro implementation

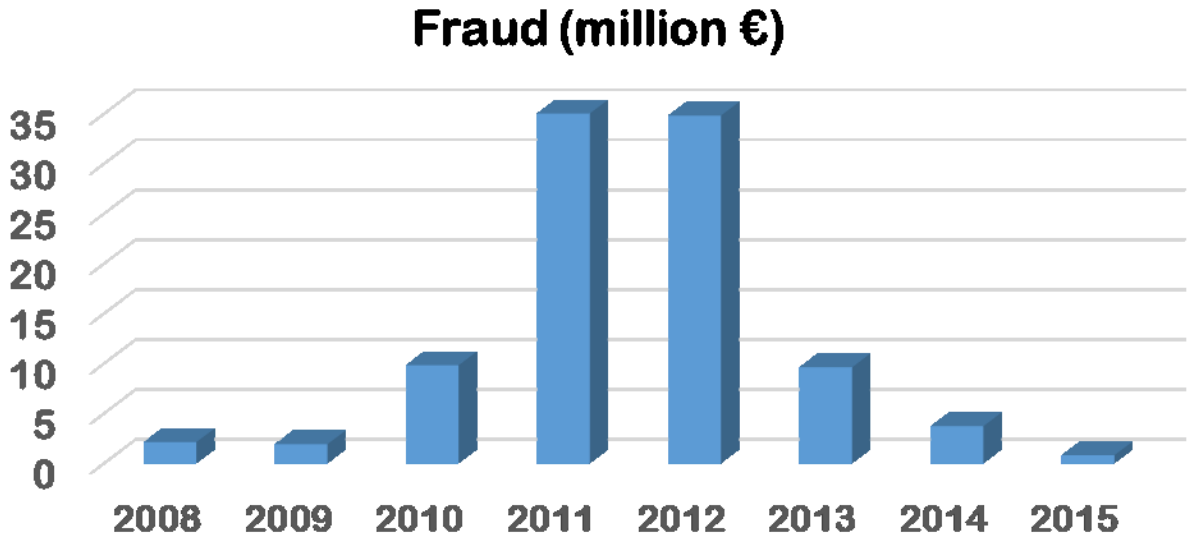
Are both implementations correct & secure? And compatible?

Presumably they both pass a Maestro compliance test-suite...

So some paths (and maybe some states) are superfluous?

Internet banking

Fraud with internet banking in NL



[Source: NVB & Betaalvereniging]

Fraud under control thanks to

- better monitoring - for suspicious transactions & money mules
  - finding money mules, to extract money from the system without being caught, is the bottleneck for attackers
- awareness compaigns
- criminal switching to ransomware as better business model?

Strong authentication for online banking

- For authentication, most Dutch banks use stronger mechanisms than just username & password
  - TAN codes: one time passwords on a printed list
  - m-TAN: one time password received by SMS
  - hand-held reader that generates one-time code using bank card

  aka two-factor authentication

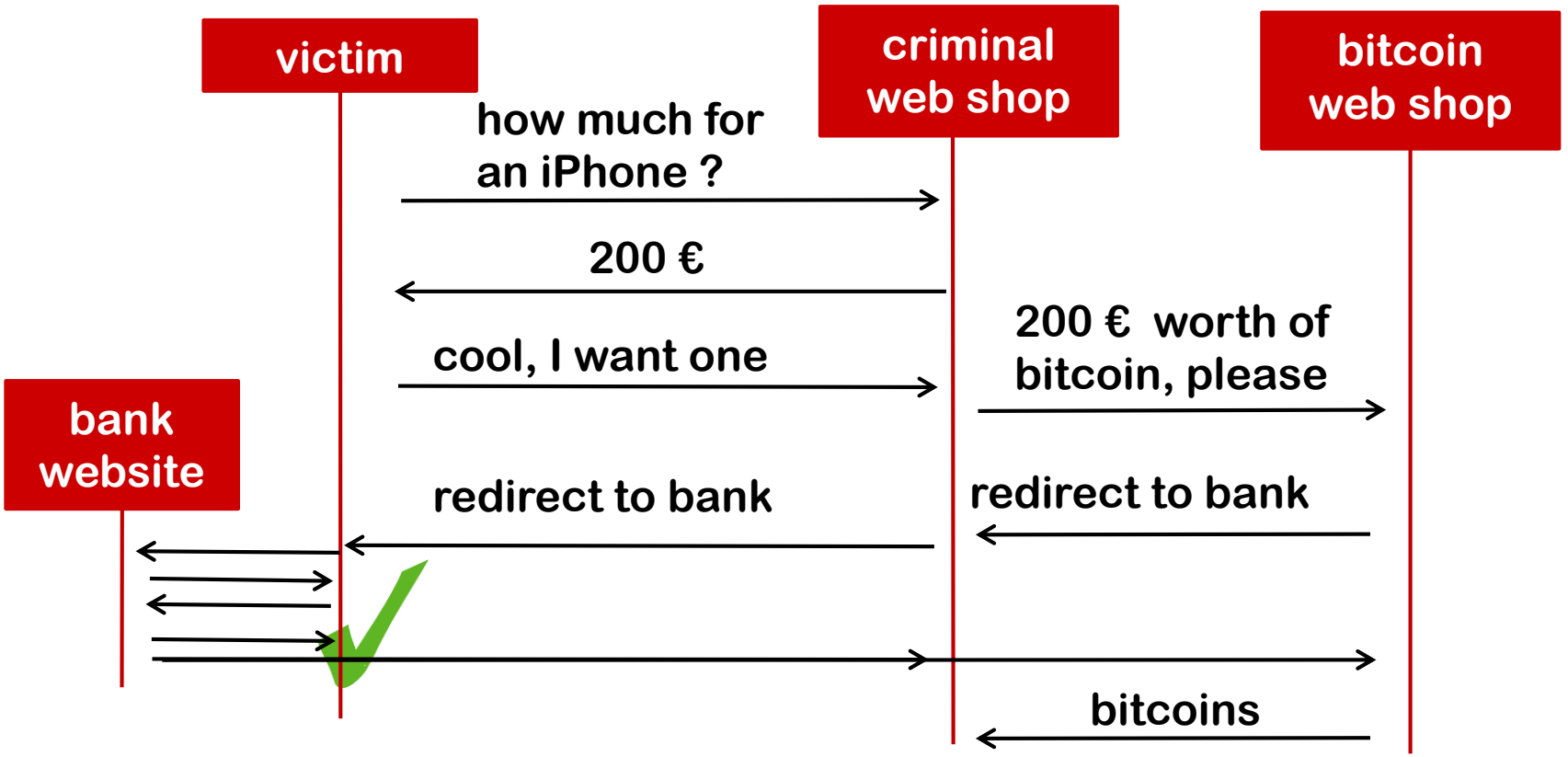
- Still, these mechanisms are not fool-proof…
  - eg. criminals have resorted to phoning people, pretending to be from the bank, to obtain these one-time codes

Example attack on internet banking (1)

- Your online bank statement shows you received 3000 euro from some company you never heard of
- You get a phone call from the bank, saying that this is a mistake and asking you to transfer the money back

- You never received 3000 euro, but malware in your browser inserts the fake transaction
  - this is a so-called Man-in-the-Browser attack
- When you transfer the money back, that is not a fake transaction…

Example attack on internet banking (2)



- Problem: messages to user not very informative, so user does not spot the attack
- Solution: better monitoring, and banks impose extra rules on bitcoin shops & online casinos for allowing internet payments

Example attack on internet banking (3)

- **For banks that use m-TANs, ie. one time passwords sent by SMS, criminals can obtain a second SIM card for your phone number aka SIM swapping**
- ***How?***
  - ***bribe someone at the Vodaphone shop!***
- **Typical countermeasure:**
  - ***banks make deal with telco to be told about re-issued SIMs, and then block internet payments for that SIM***

Internet banking

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

Computer display of cannot be trusted (despite )

→ 23459876
← 123654

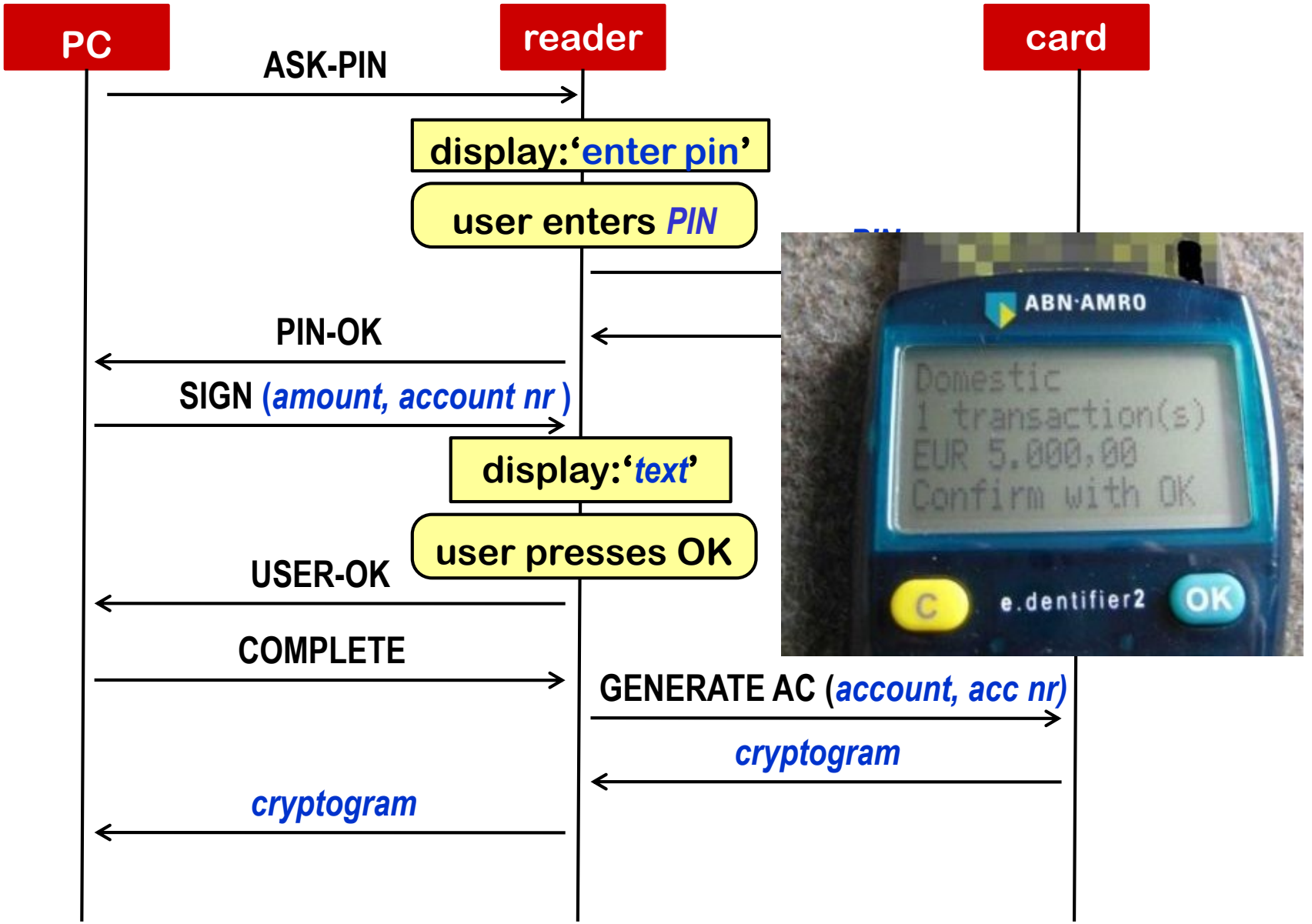
Internet banking

This display can be trusted & understood

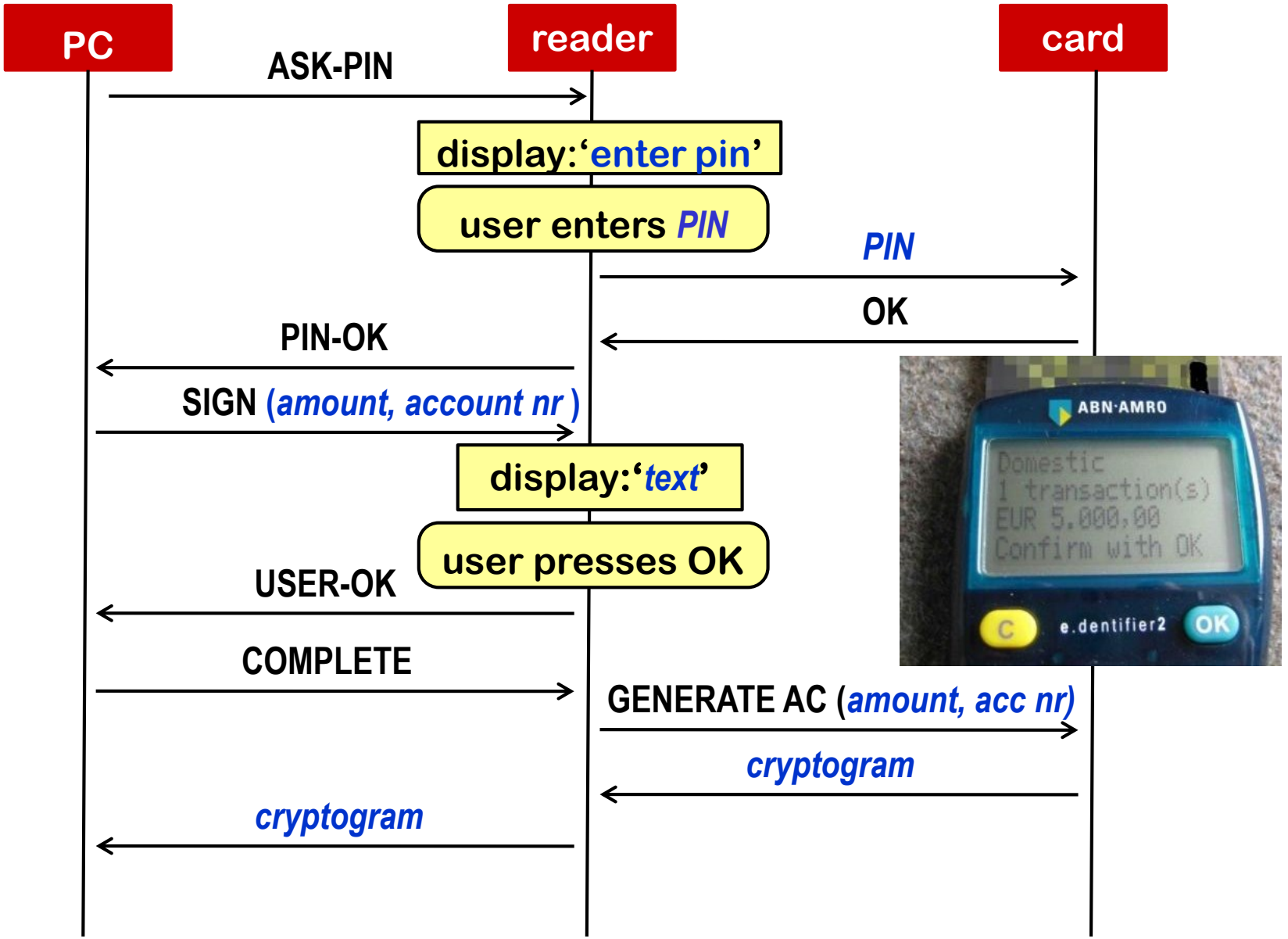
"What You Sign is What You See" (WYSIWYS)

USB

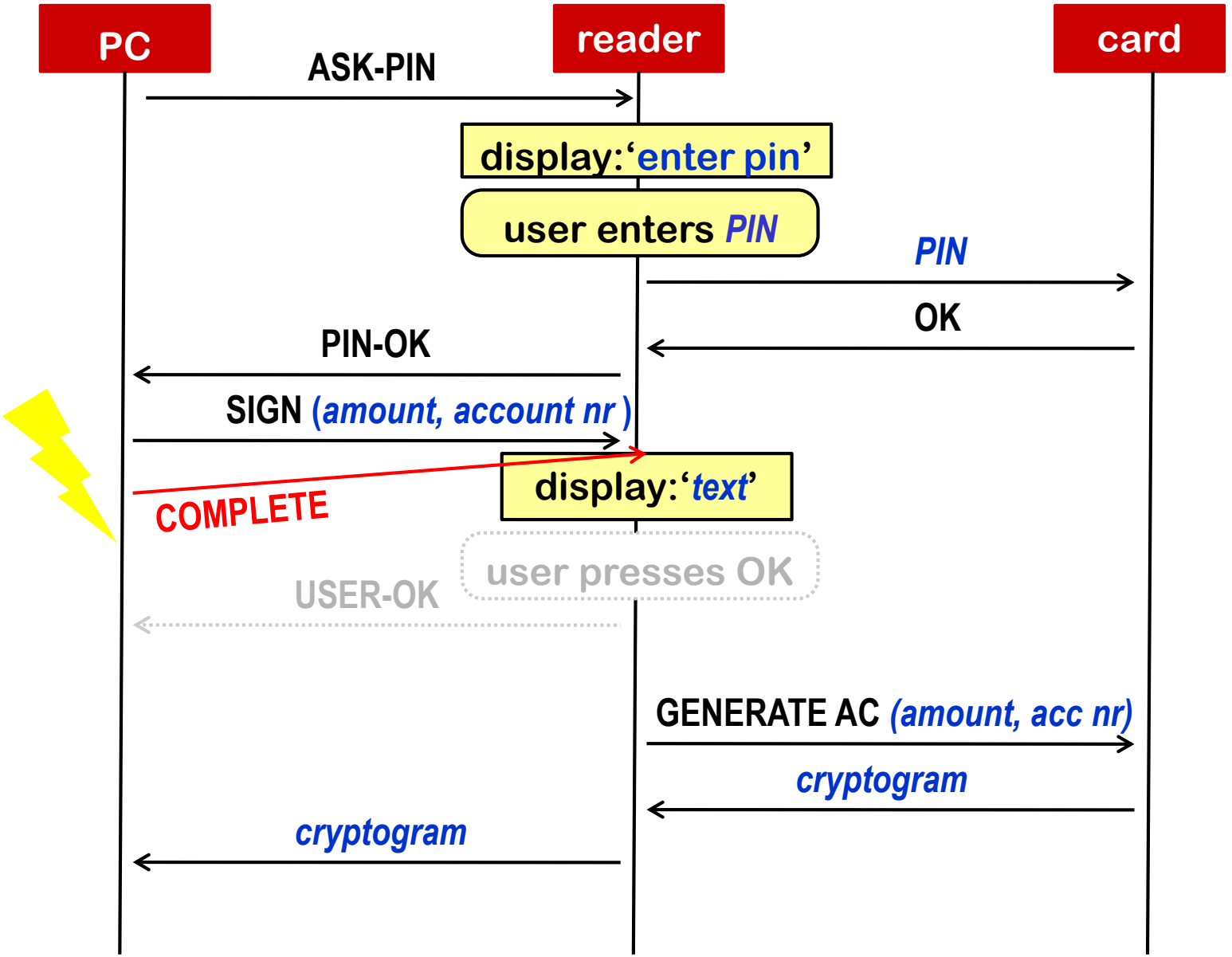
Reverse-Engineered Protocol



Reverse-Engineered Protocol



Attack!


PC
reader
card
ASK-PIN
display:'enter pin'
user enters PIN
PIN
OK
PIN-OK
SIGN (amount, account nr)
COMPLETE
display:'text'
user presses OK
USER-OK
GENERATE AC (amount, acc nr)
cryptogram
cryptogram

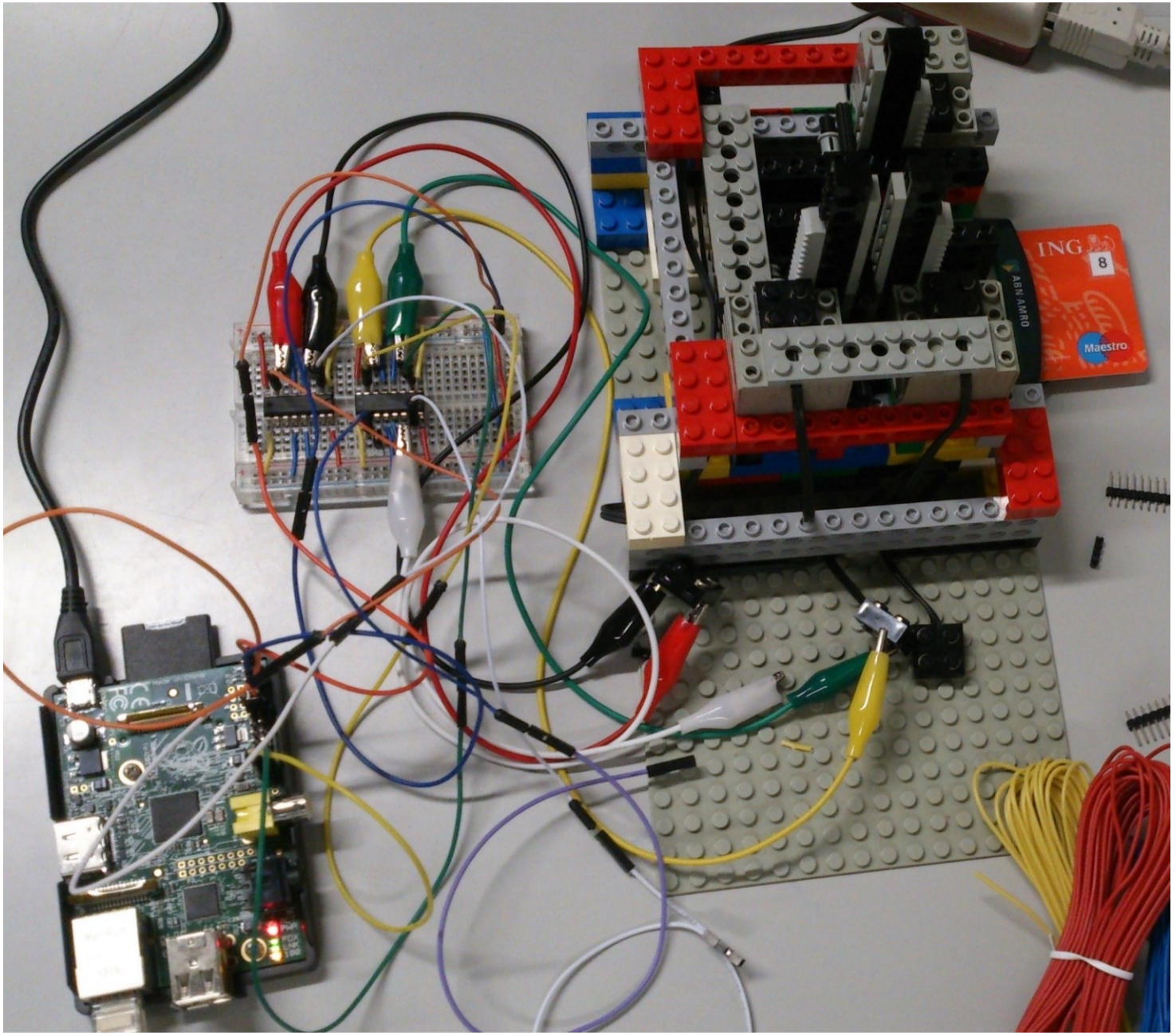
Problem with Todos/Gemalto e.dentifier2

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

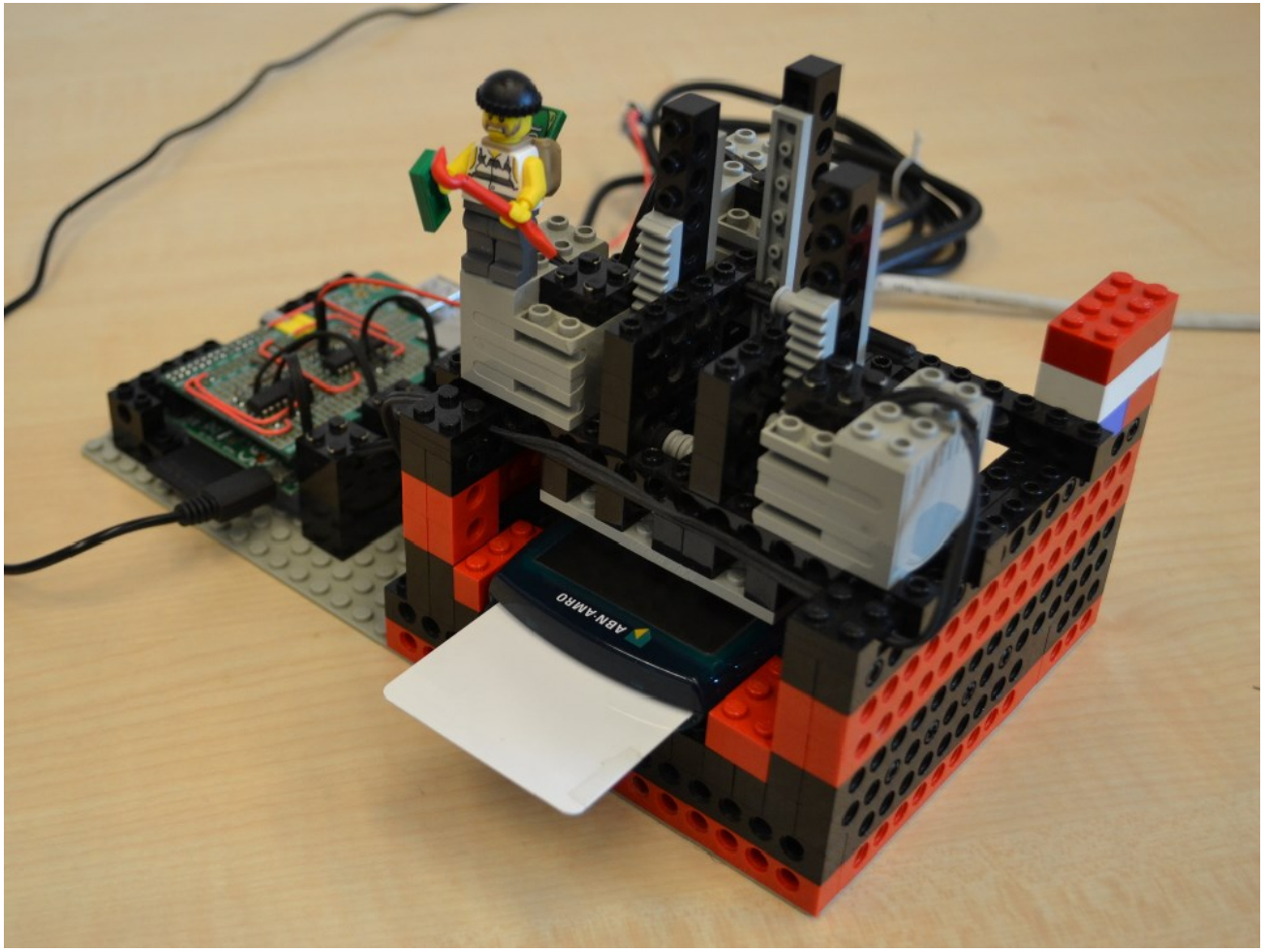
Malware on an infected PC could change all the transaction details and press OK

This is a flaw in the state machine! Can we find it automatically?

Operating the keyboard using LEGO

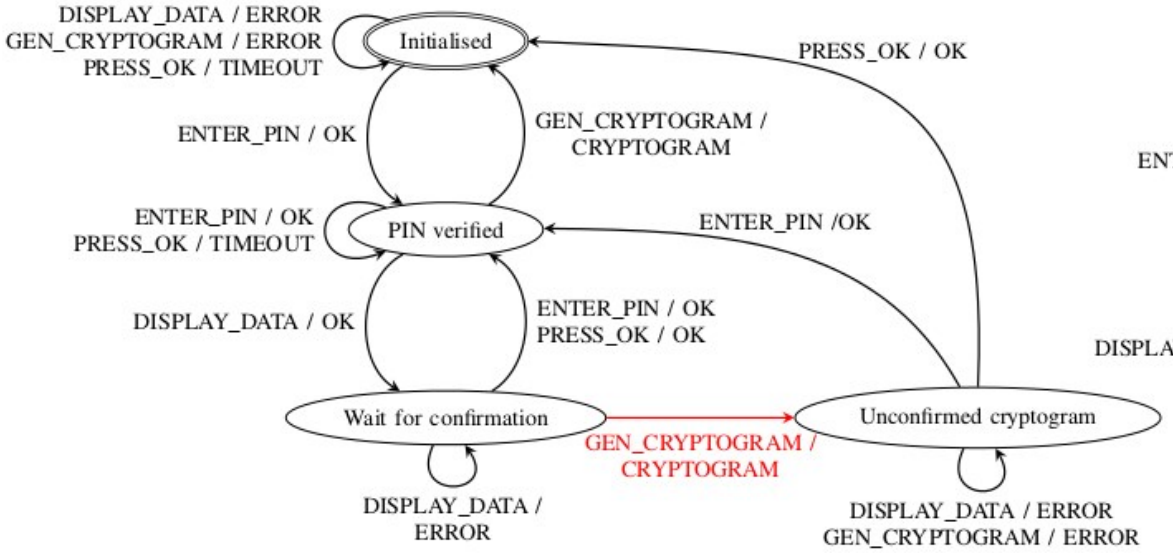

ING
8
ABN AMRO
Maestro

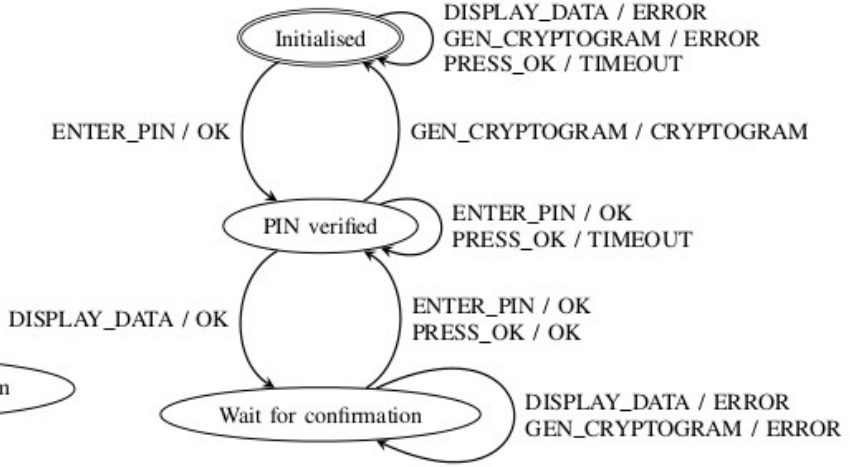
Our Lego movie online: https://tinyurl.com/legolearning

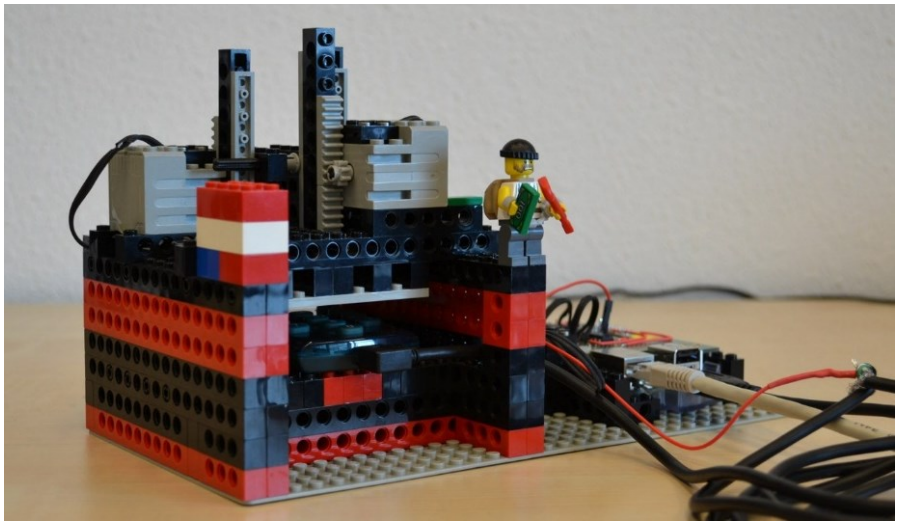


[Automated Reverse Engineering using LEGO, WOOT 2014]

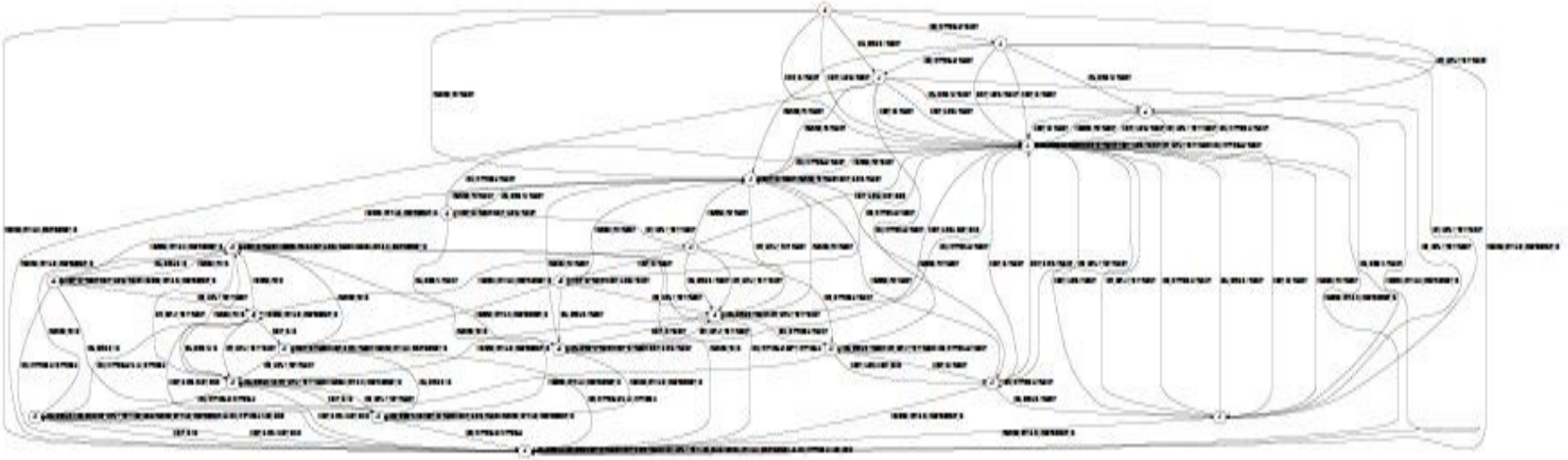
State machines of old vs new e.dentifier2







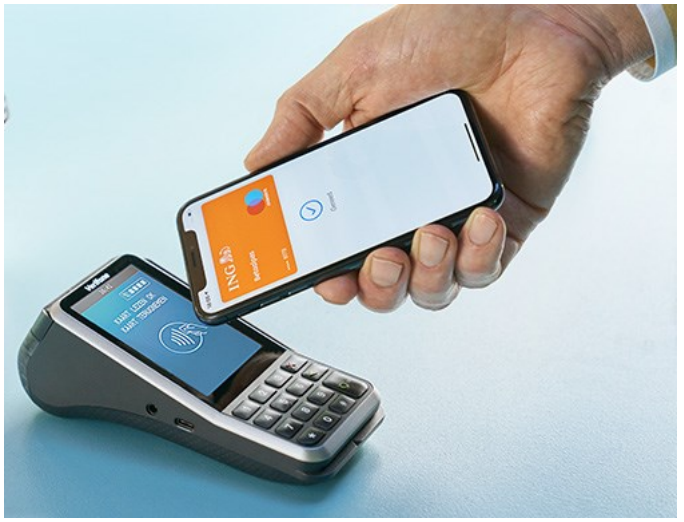
Would you trust this to be secure?



Full state machine of the handheld card reader

Do you think whoever designed or implemented this is confident that this is secure?

Or that all this behaviour is necessary?



Contactless payments

Contactless payments

Contactless version of EMV with bank card or NFC smartphone

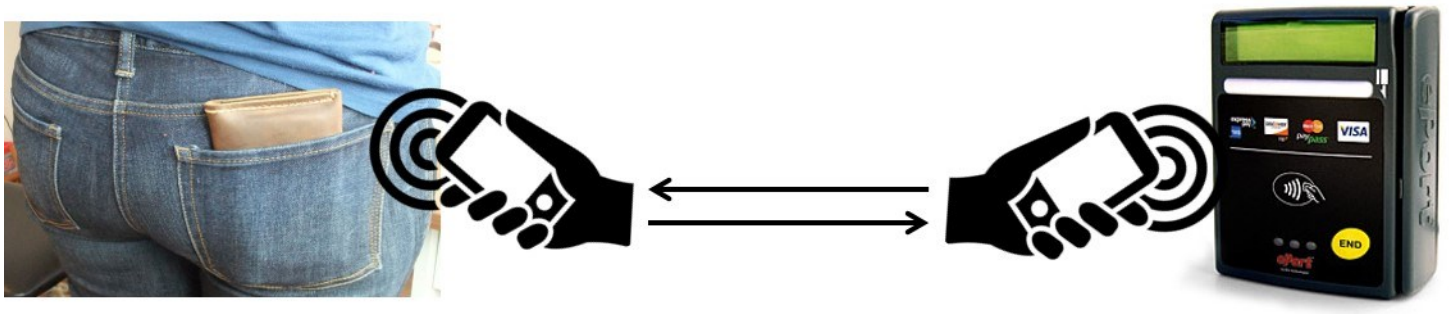
In Netherlands, for a maximum of 25 euro per individual transaction and a cumulative total of 50 euro until PIN has to be entered again.

Contactless payments

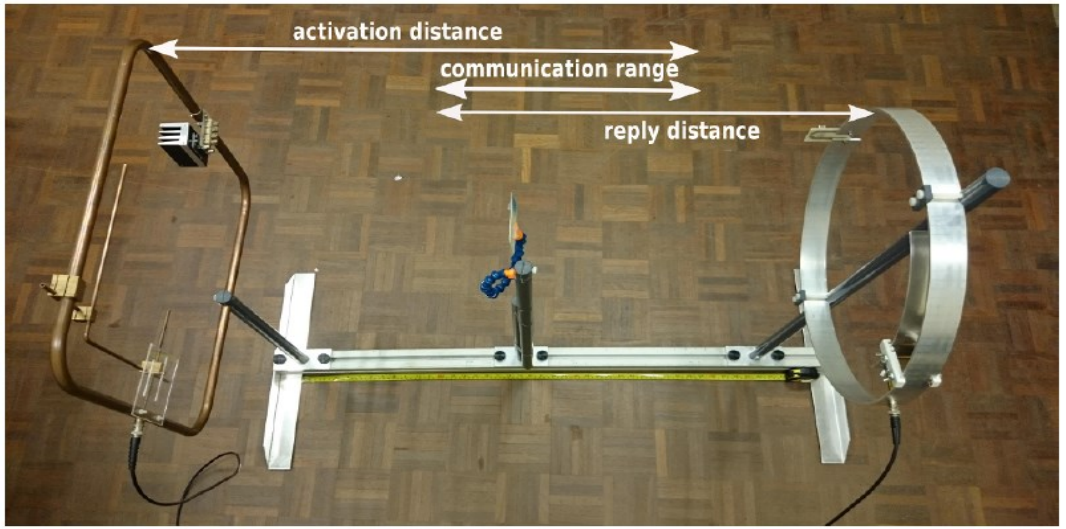
- ***Who thinks that contactless payments without PIN are less secure than contact payment with PIN?***

Security of contactless payments

- It is not possible to clone a contactless card
- It is possible to do a relay attack



  - But is there a good criminal business model? Probably not…
  - Max. distance to activate card ≈ 40 cm



[René Habraken et al., An RFID Skimming Gate Using Higher Harmonics, RFIDSec 2015]

Risks of contactless payments

1. Risks of contactless payment *without* PIN

   a) You loose max. € 50 if your card is stolen

   b) You loose max. € 25 euro if you fall victim to a relay attack

   Dutch banks typically cover these losses.

2. Risks of contact payment *with* PIN

   a) You don't loose any money if your card is stolen

   b) You can loose €1000 or more if your card is stolen after attacker snooped on your PIN code

   Banks will typically not cover these losses…

So the 'extra security' of the PIN probably *in*creases risk for customers.

Note: technical weakness in the security ≠ risk

where risk = likelihood x impact

Configuration flaws

Mistake on the first generation contactless cards issued in the Netherlands:

- **functionality to check the PIN code,**

  which should only be accessible via the contact interface 

  was also accessible via the contactless interface 

Possible risk for DoS attacks, rather than financial fraud?

Flaw discovered by Radboud students Anton Jongsma, Robert Kleinpenning, and Peter Maandag.

Software bugs

Contactless payment terminals of one manufacturer could be crashed with a legal – but unusual – input

- **Probably due to a buffer overflow**
- **Bug might be used for more interesting attacks than just Denial-of-Service, eg. to get malware on the terminal**

[Jordi van den Breekel, A security evaluation and proof-of-concept relay attack on Dutch EMV contactless transactions, MSc thesis, 2014]

- **The input format of smartcard terminals is very simple:**
  - **how can you implement this wrong?**
  - **if you do, how do you miss it in testing?**

Command APDU						
Header (required)				Body (optional)		
CLA	INS	P1	P2	Lc	Data Field	Le

Apparently, certification schemes for terminals such as are not testing for such basic flaws…

Side-channel attacks on smartcards

Side-channel attacks

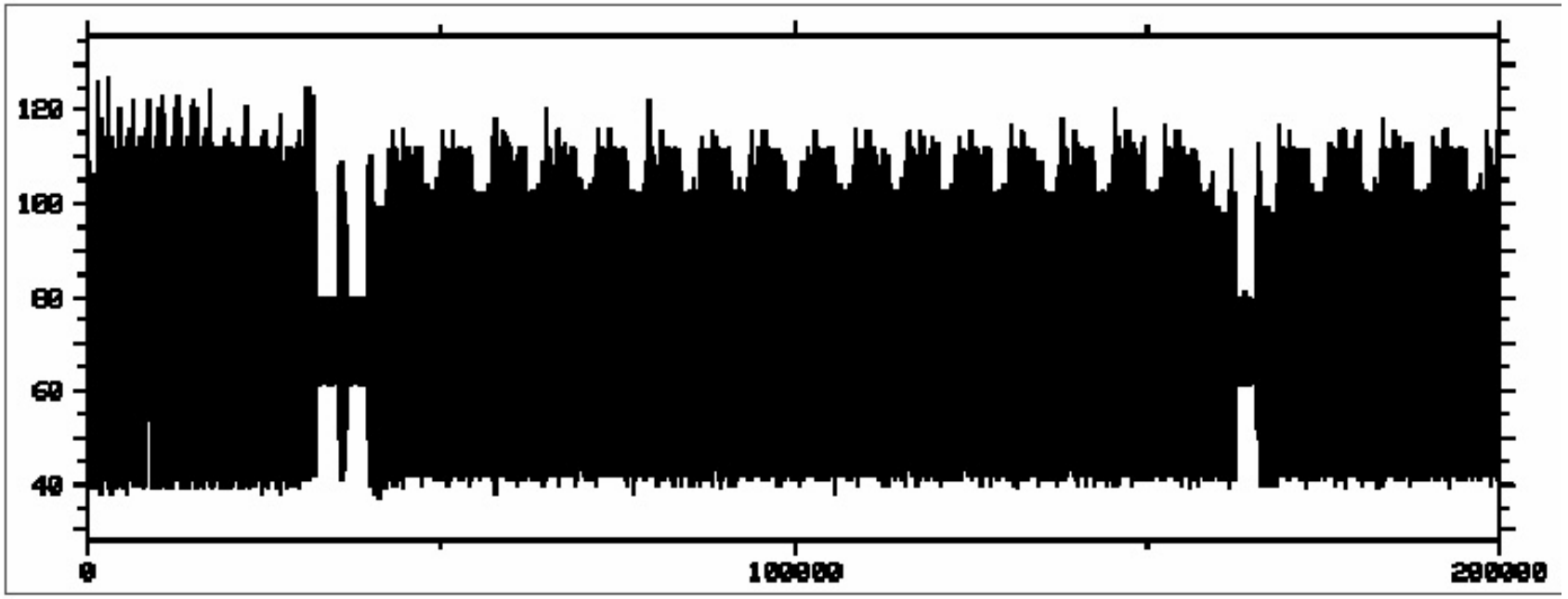
- Side-channel = any other channel than the normal input-output channel that may be observed
- Possible side-channels
  - sound
  - timing
  - power consumption
  - electro-magnetic radiation

Side channels are very hard to avoid!

Prof. Lejla Batina in our group does research into side-channels

With Spectre & Meltdown, these attacks are no longer just a concern for smartcards & other cryptographic hardware

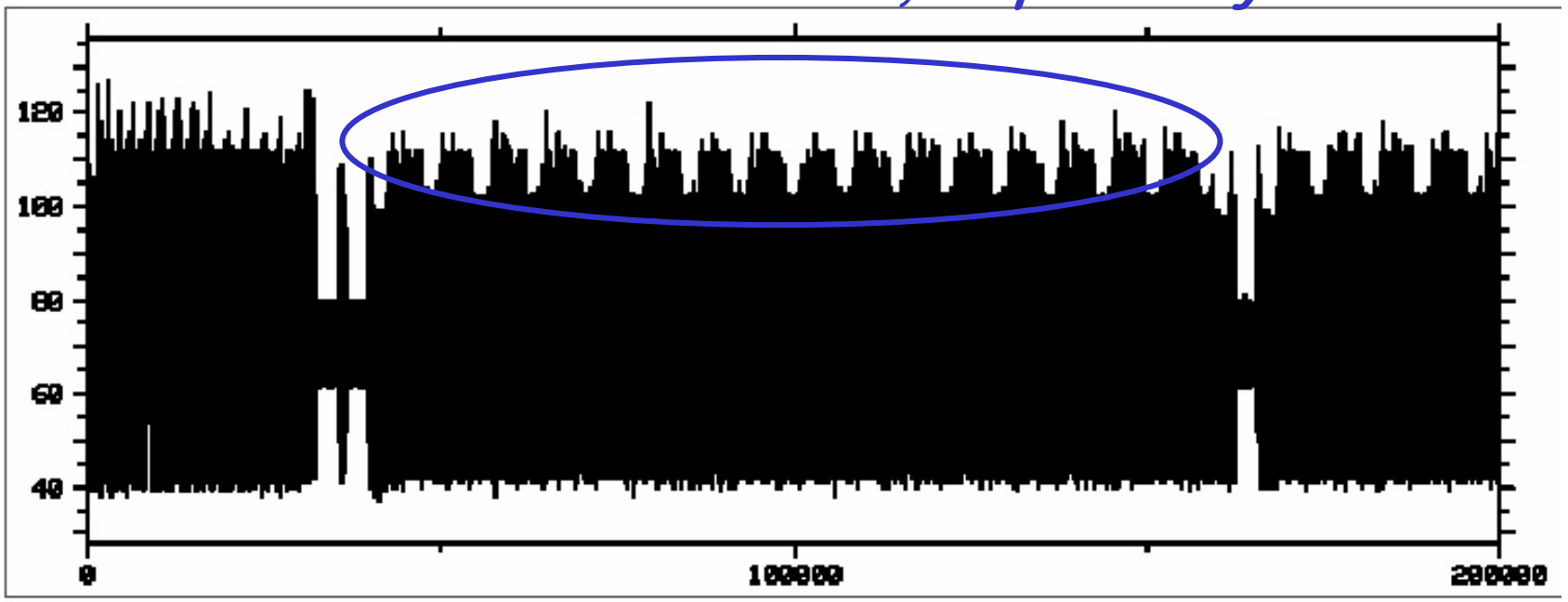
Power consumption of a smartcard



What is this card doing?

This is a DES encryption!

16 rounds, so probably DES

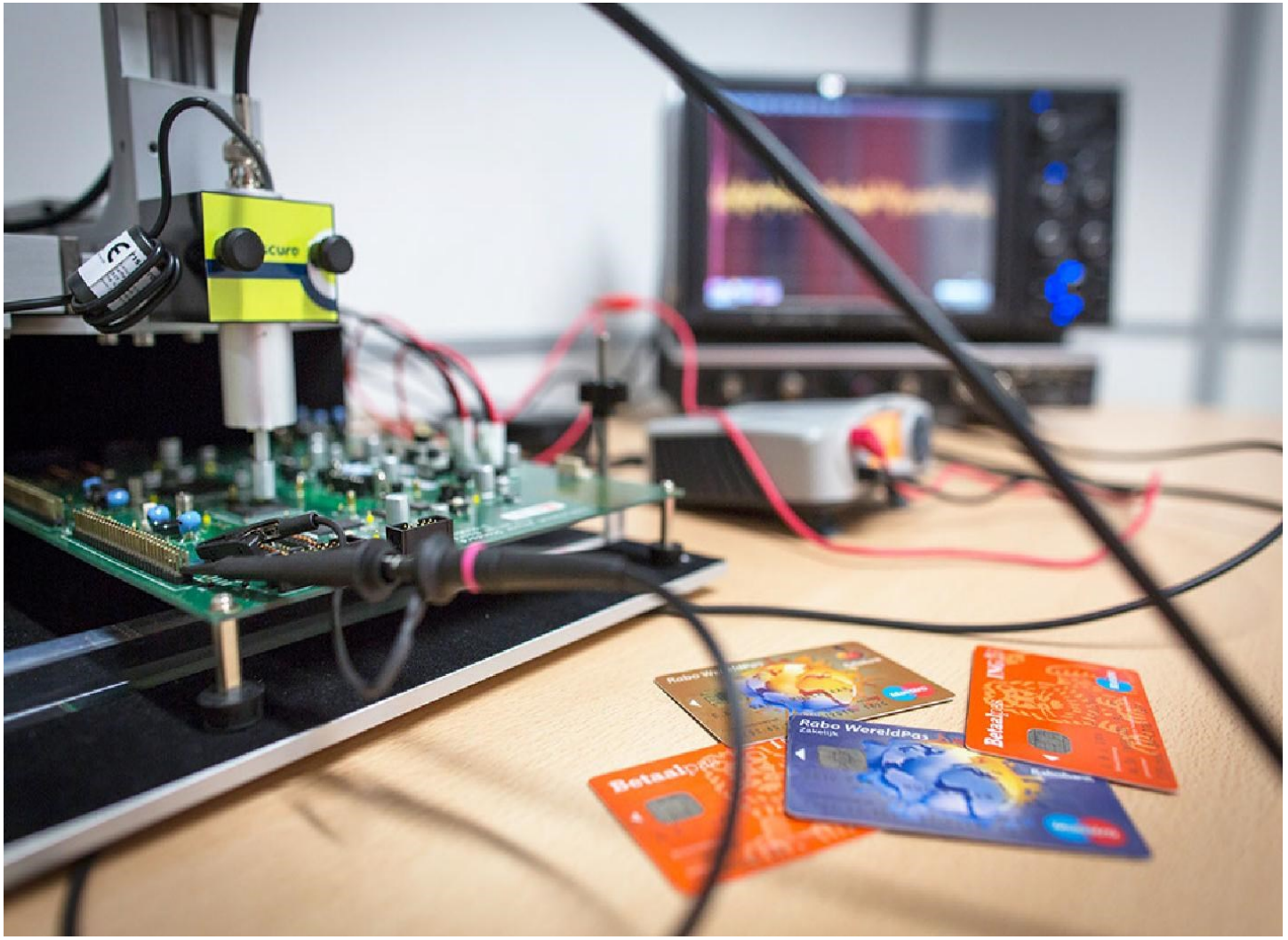


What is the key?

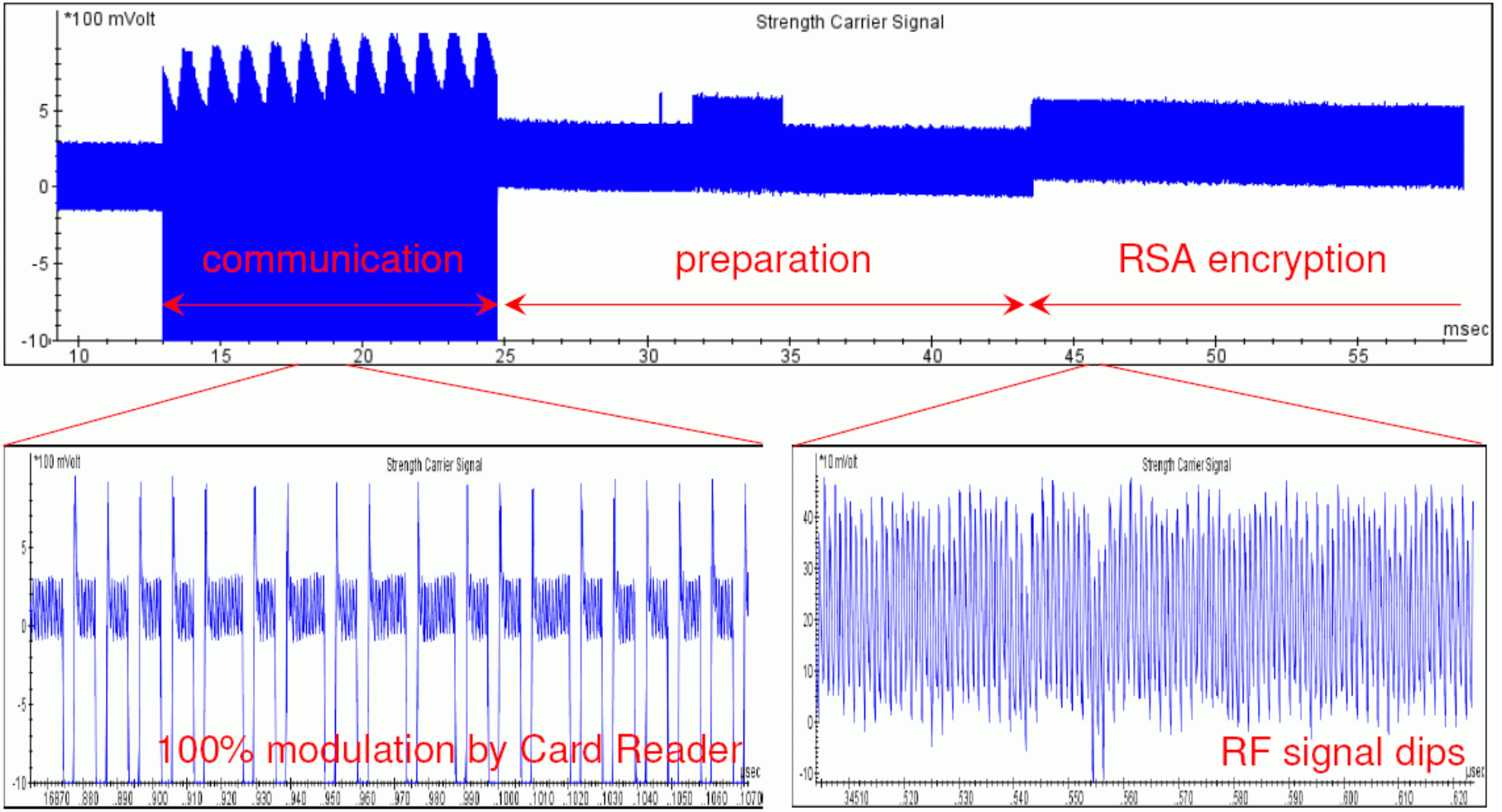
In a really poor implementation, you might be able to read it off…

Power analysis

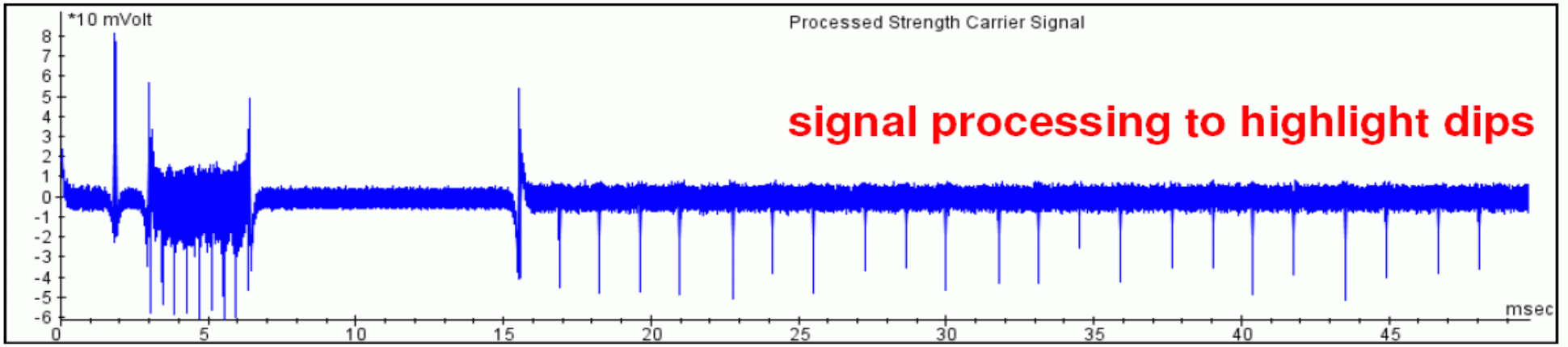
Side channel analysis using EM radiation

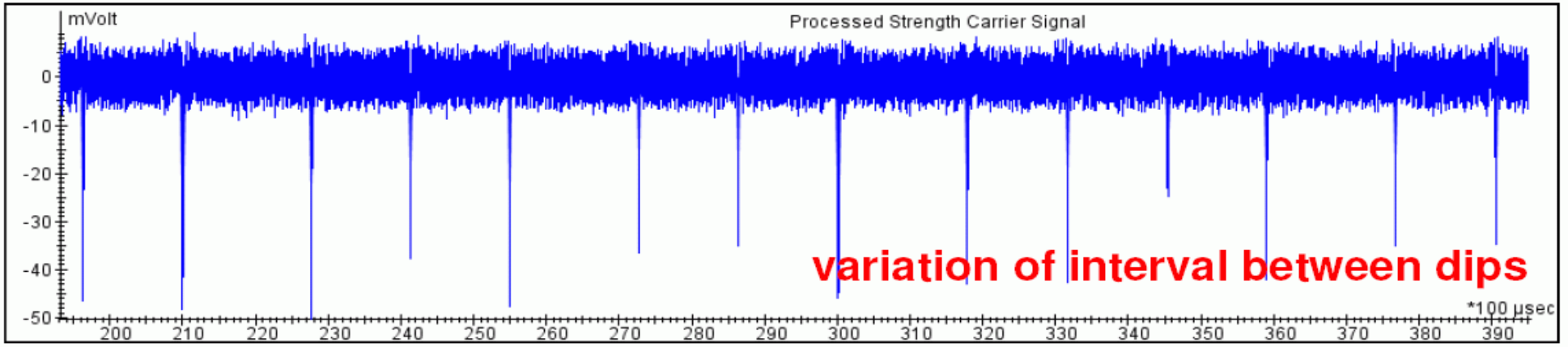


Power trace


*100 mVolt
Strength Carrier Signal
communication
preparation
RSA encryption
msec
100% modulation by Card Reader
RF signal dips

Power trace detail of RSA encryption





RSA implementation

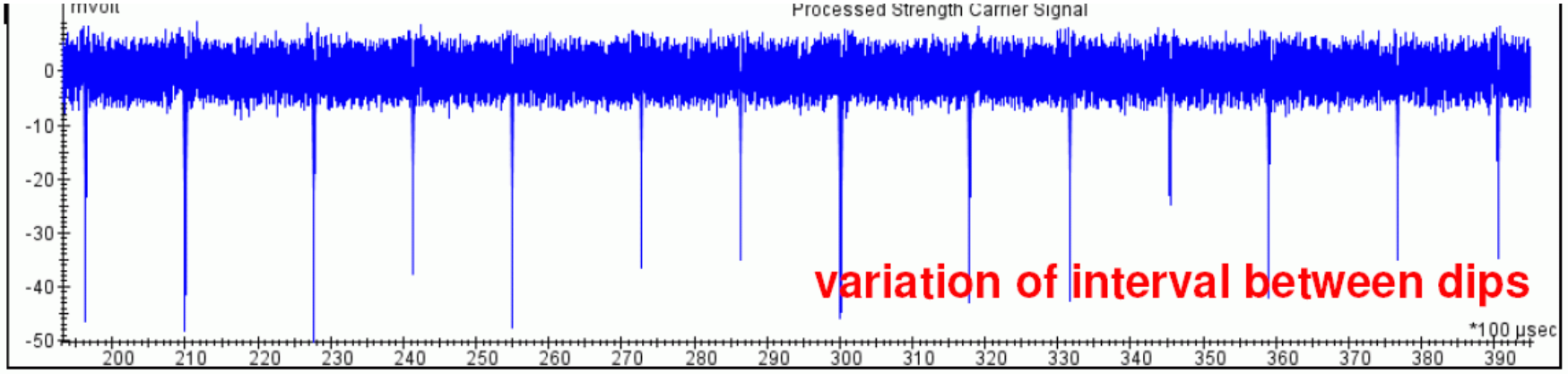
RSA involves exponentiation $s = (x^y) \bmod n$

Fast implementation with square & multiply:

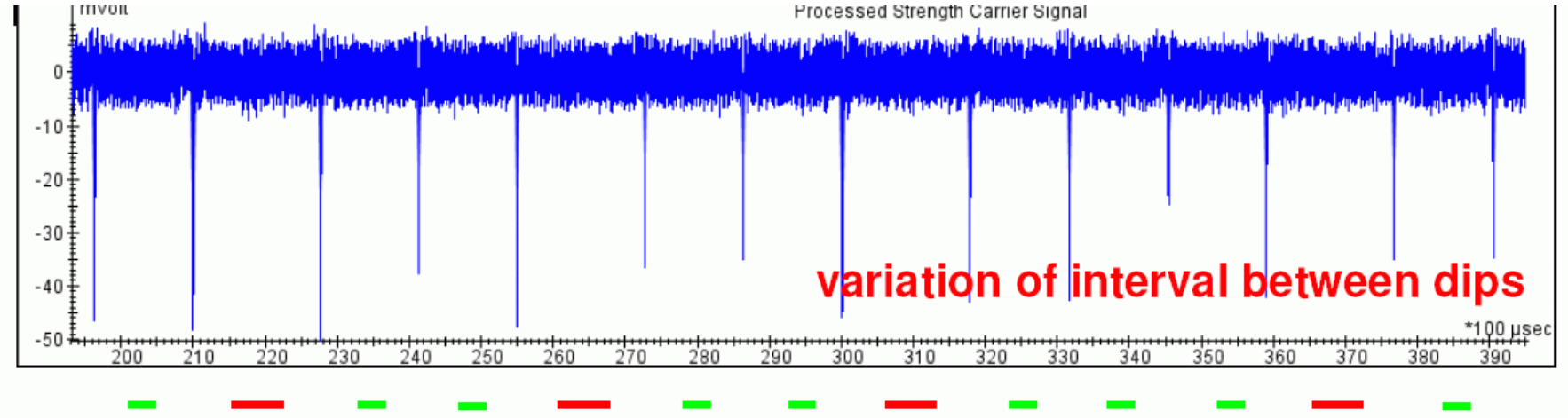
eg $x^{27} = x^1 * x^2 * x^8 * x^{16}$ in log(y) steps

```
result = 1; xi = x;
for i = 1 to length(key) do
  { if (bit i of key is 1) then result = result*xi ;
     xi = xi*xi;
  }
return result;
```

Reading key of the power trace?

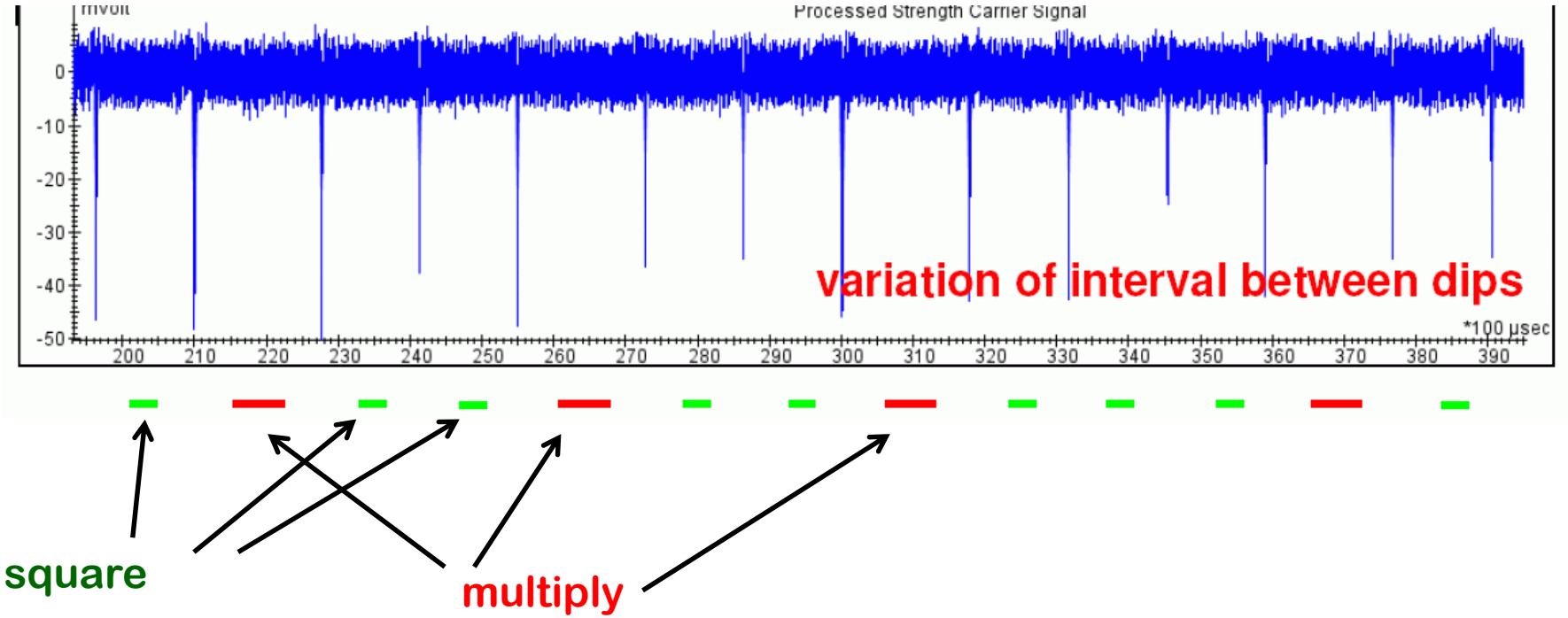


Reading key of the power trace?

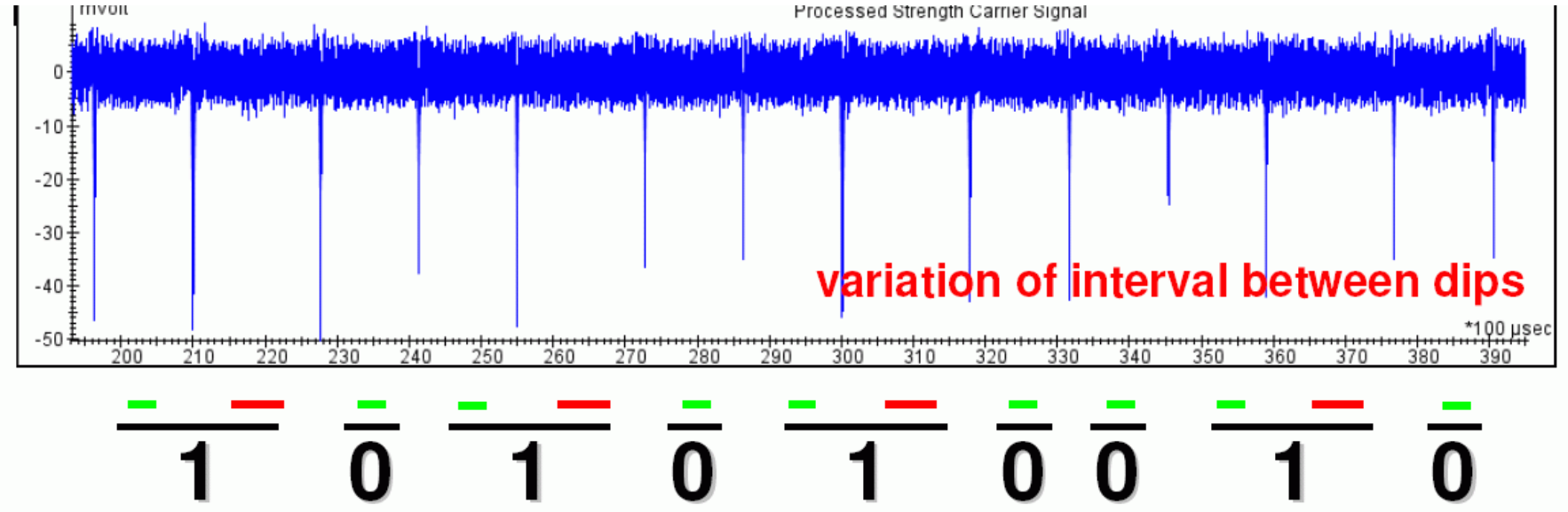


What are the squares & what are the multiplications?

Reading key of the power trace?



Reading key of the power trace!



Easier attacks?

PHISHING

Criminals have sent emails asking people to return their bank card & pin code by post to the bank

14 september 2015 13:55

Rabobank waarschuwt voor nieuwe phishing: stuur nooit je pas op

De Rabobank waarschuwt voor een nieuwe vorm van phishing, waarbij het slachtoffer wordt gevraagd om een zogenaamd verlopen betaalpas op te sturen.

Stealing credit card numbers on web sites

Ticketmaster Blames Third Party Over Data Breach

Hotel websites infected with skimmer via supply chain attack

Sep 19, 2019
NEWS by Bradley Barth

LILY HAY NEWMAN SECURITY 09.11.2018 03:00 AM

How Hackers Slipped by British Airways' Defenses

Security researchers have detailed how a criminal hacking gang used just 22 lines of code to steal credit card data from hundreds of thousands of British Airways customers.

BRIAN BARRETT SECURITY 07.11.2019 06:00 AM

Hack Brief: A Card-Skimming Hacker Group Hit 17K Domains—and Counting

Magecart hackers are casting the widest possible net to find vulnerable ecommerce sites—but their method could lead to even bigger problems.

https://www.wired.com/story/magecart-amazon-cloud-hacks/

Supply chain attack by Magecart

Trick used by criminal Magecart group

1. Look for misconfigured S3 buckets in Amazon cloud that are world-readable & world-writeable
2. Add malicious code to any JavaScript files in that bucket
   - namely code that looks for credit card numbers being entered & reports this back to criminal
3. Sit back & wait for credit cards to be reported

https://www.riskiq.com/blog/category/magecart

Weak/no authentication

RFID cards used at EV charge points

- are Mifare Classic cards (with uses a flawed self-designed crypto algorithm instead of standard solution like AES)
- worse still: authentication does not use challenge-response mechanism, but only uses the card's serial number

  This serial number can be eavesdropped and

  can be replayed (eg. with an NFC smartphone)

  so cards are trivial to clone

Conclusions

Conclusions

- General trend: from prevention to better detection & response
- A *technical* security flaw not always a serious security risk.
  The real issue: can attackers find a good business model?
  - The bad news here: ransomware is a great business model for almost any security weakness
- Some silly security flaws by reputable companies & vendors
  - Who is *really* taking responsibility for the security ?
    - Individual banks? Their suppliers? 3rd parties doing certification? MasterCard & Visa, who also approve vendors & certifiers?
  - How much security is just Cover-Your-Ass security?

Why banking security is easy!

- Banks can measure attacks & quantify their costs euros, so
  - Trends in attacks can be monitored
  - Success of defensive measures can be measured
  - This provides a rational basis for security decisions

- In other industries this is MUCH harder
  - Eg for critical infrastructures or hospitals:
    - How much can cyber protection of the electricity grid cost?
    - How much can patient privacy cost?
  - Ransomware may play a 'useful' role here…

Thanks for your attention