# Designed to Fail: A USB-Connected Reader for Online Banking

Arjan Blom, Gerhard de Koning Gans, Erik Poll, Joeri de Ruiter & Roel Verdult

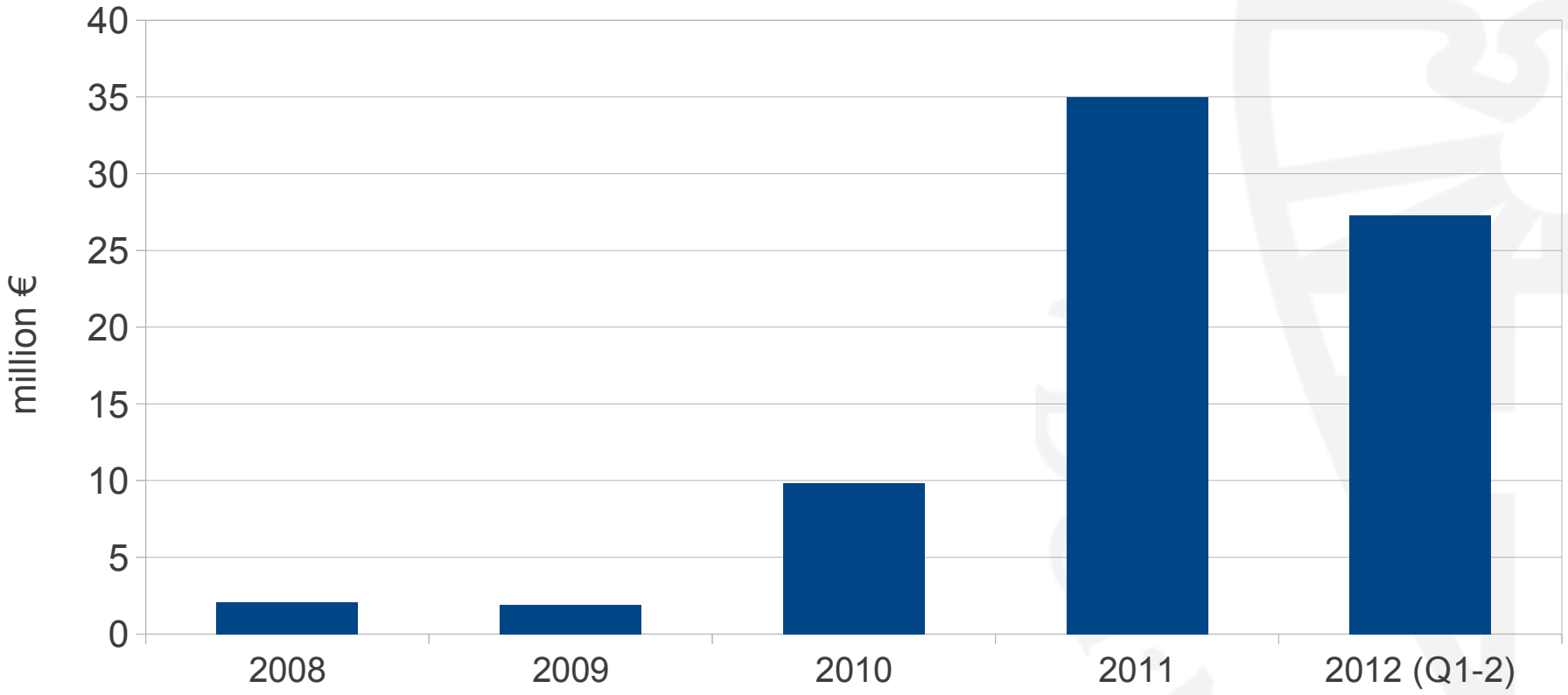
Digital Security, Radboud University Nijmegen

# E-banking in the Netherlands

- 93% of all transactions
- 3 billion transactions a year

  (€ 3.200 billion)

# Fraud figures in the Netherlands



# EMV-CAP

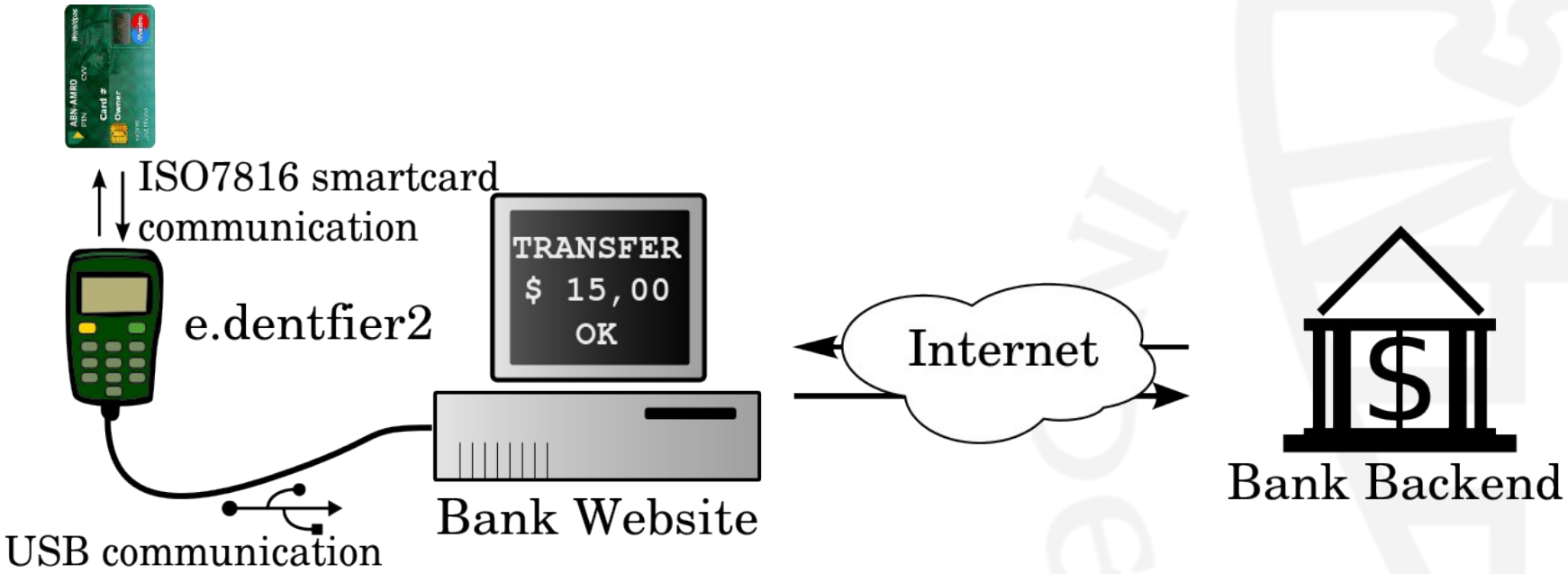
- Based on EMV
- Not public
- Reverse engineered

- Handheld readers
- Challenge-response

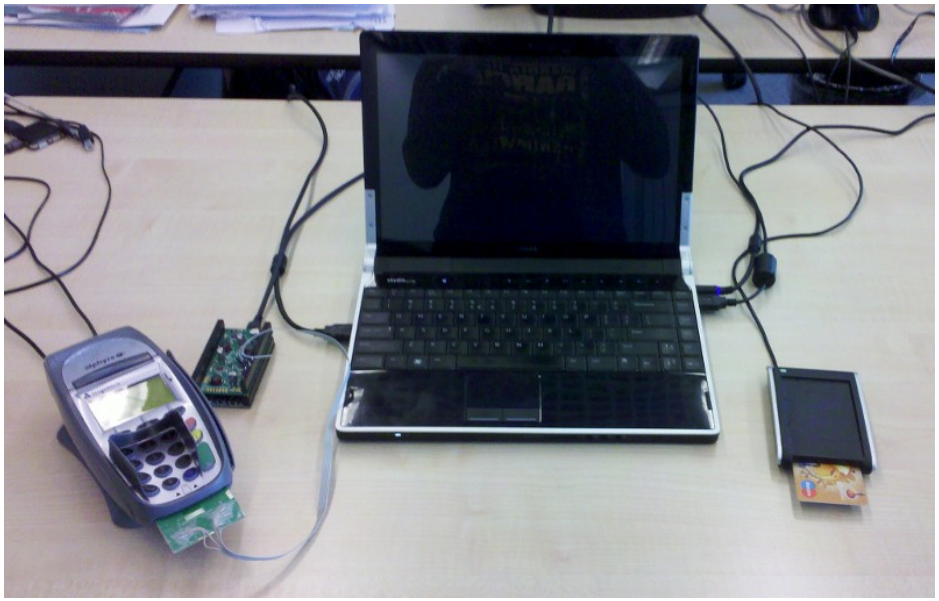
# e.dentifier2

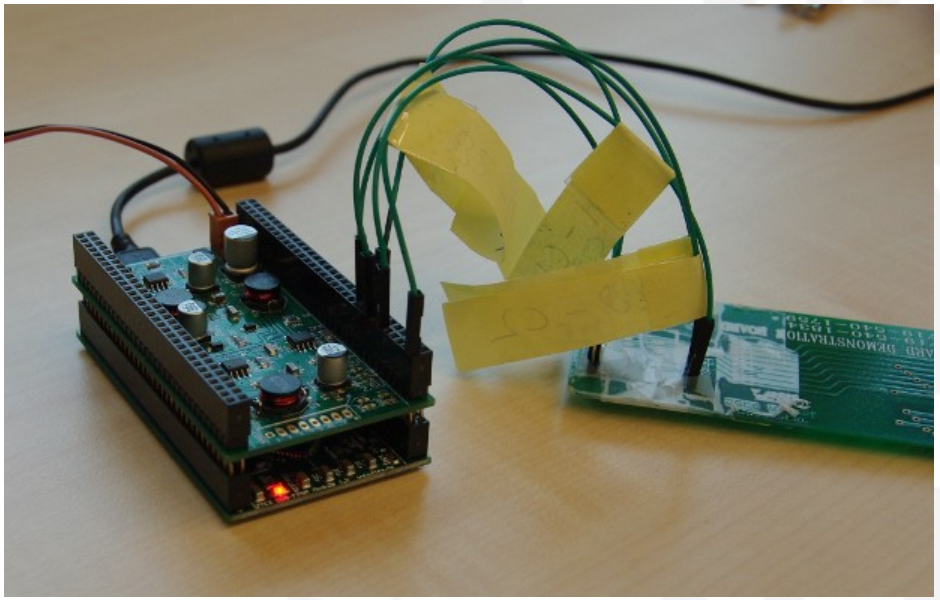
- Developed by Todos (now Gemalto)
- Can be with or without USB
- With USB:
  - See-What-You-Sign
  - “the most secure sign-what-you-see end user device ever seen”
  - Good idea!

# Analysis

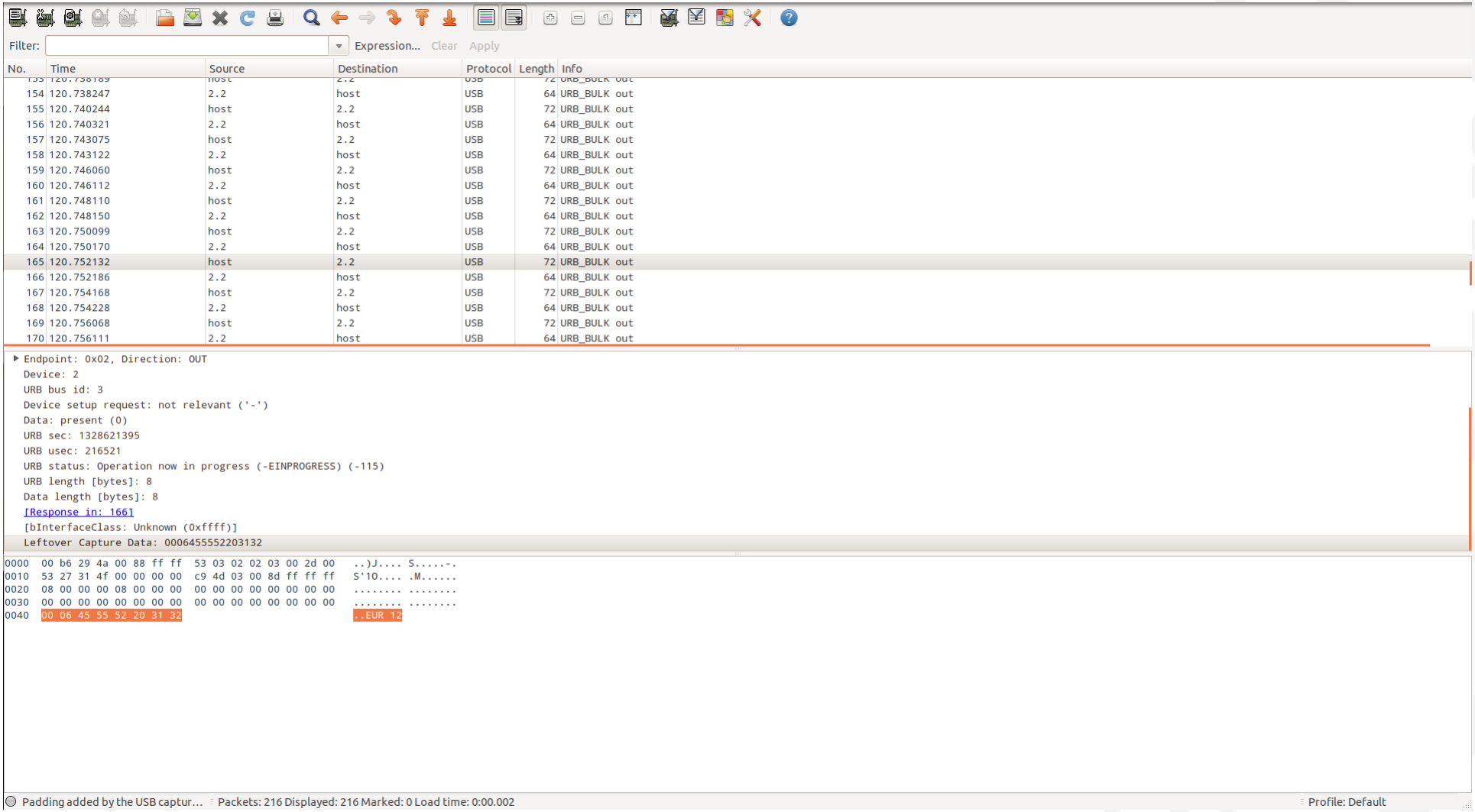

ISO7816 smartcard
communication
e.dentfier2
TRANSFER
$ 15,00
OK
Internet
Bank Backend
Bank Website
USB communication

# Smartcard ↔ Reader (ISO 7816): SmartLogic





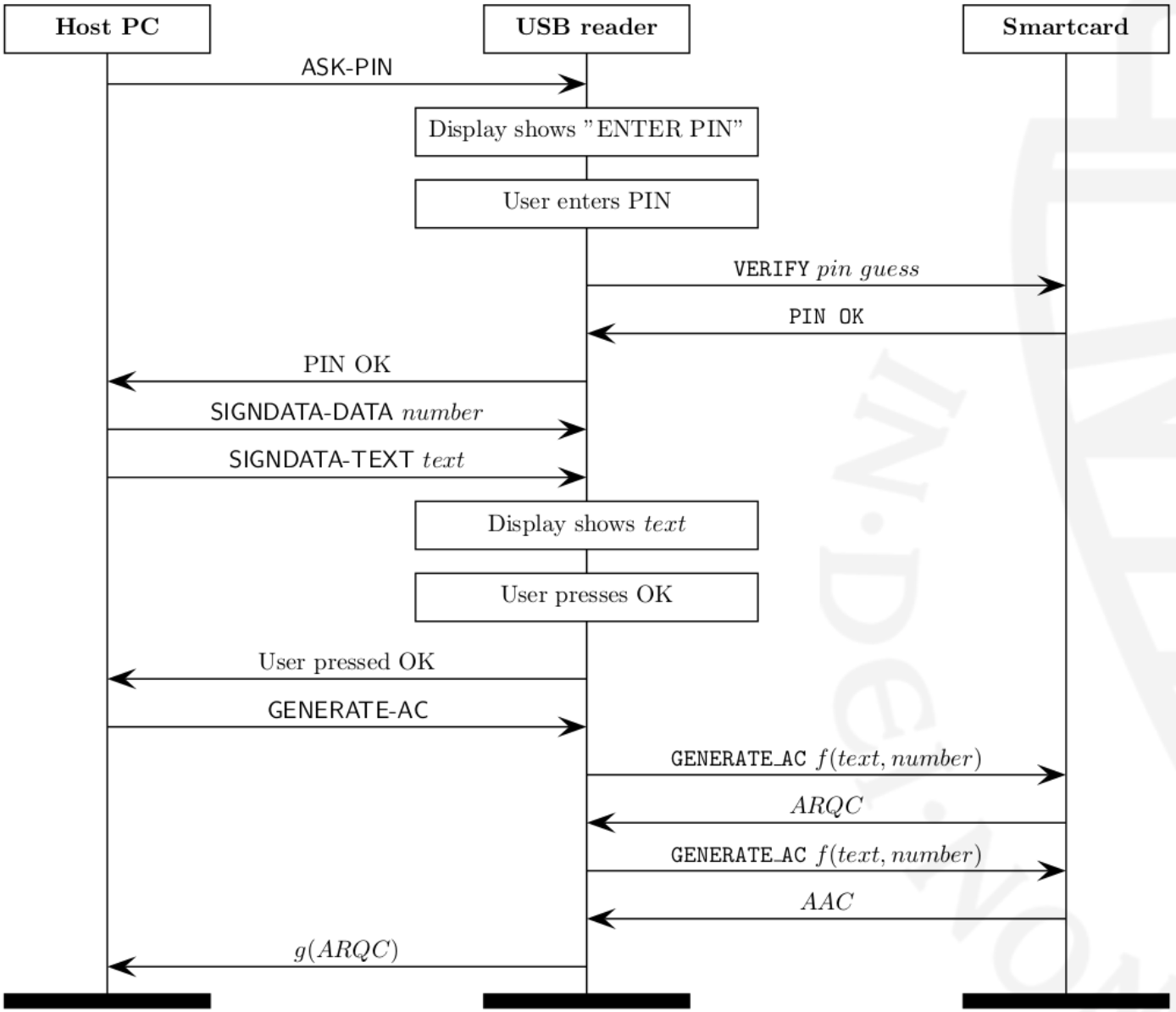
# Reader ↔ PC (USB): Wireshark



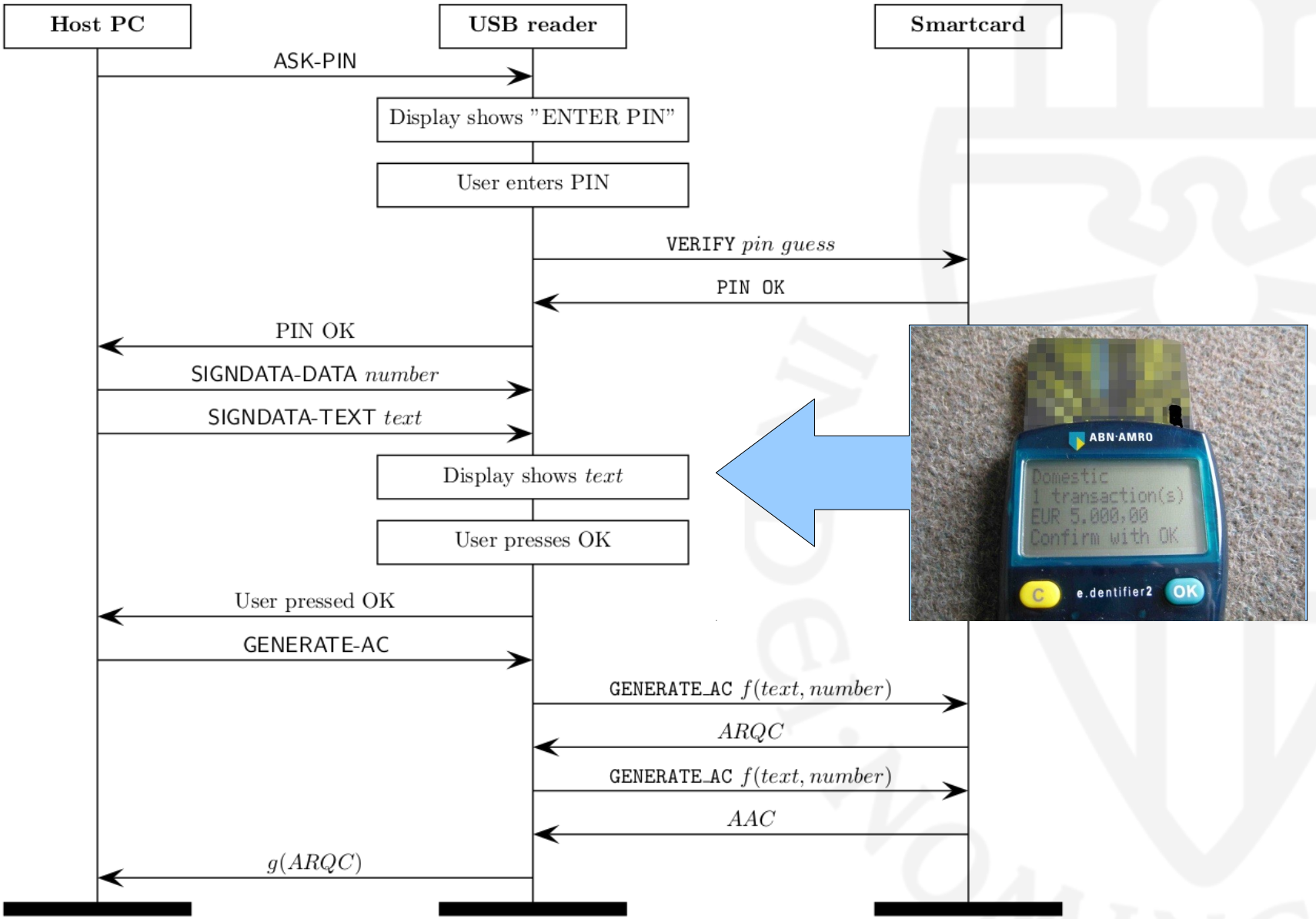
# Analysis

- PC can show
  - messages predefined in the e.dentifier2
  - any message that it wants to be signed

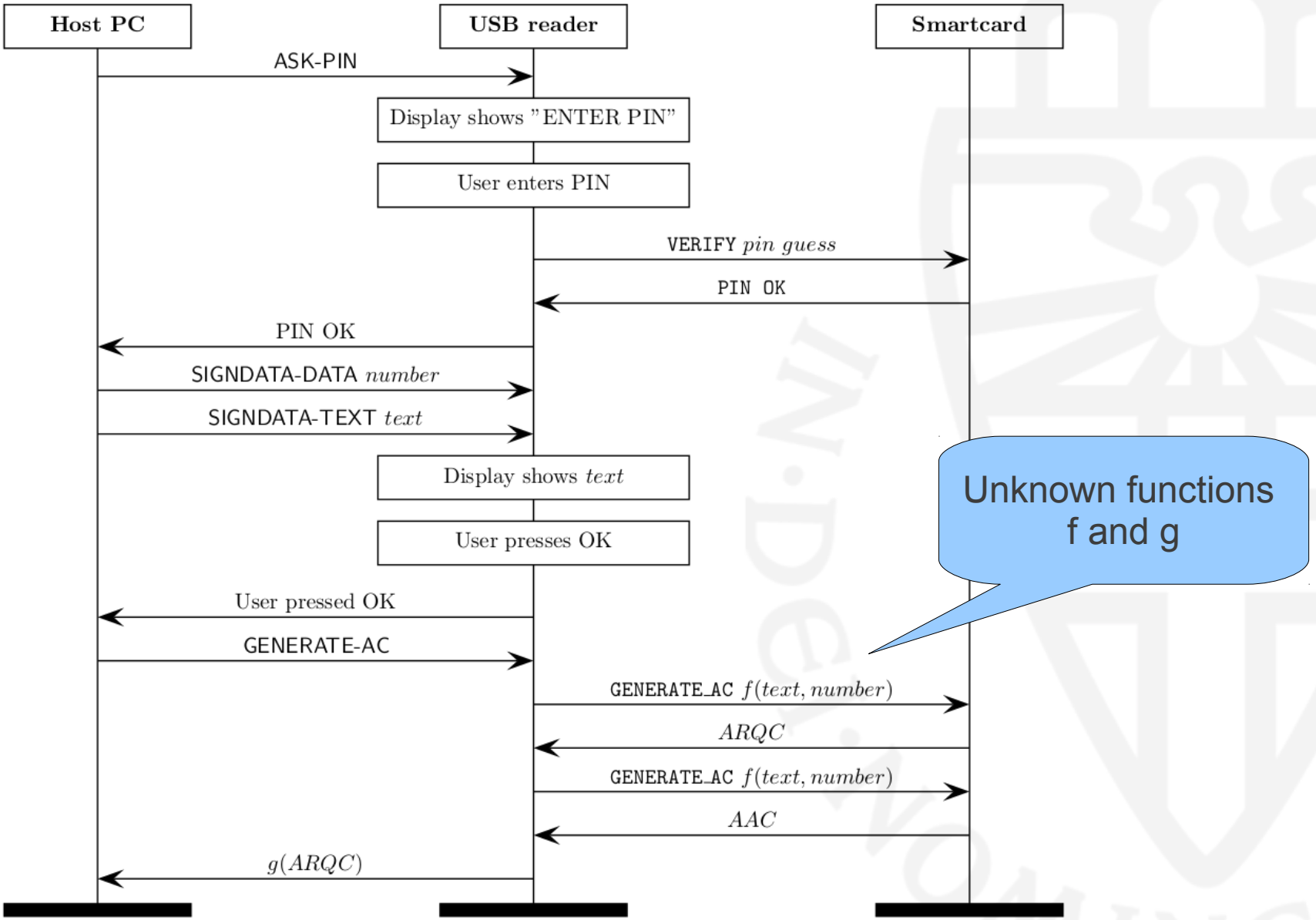
# Protocol


Host PC
USB reader
Smartcard
ASK-PIN
Display shows "ENTER PIN"
User enters PIN
VERIFY pin guess
PIN OK
PIN OK
SIGNDATA-DATA number
SIGNDATA-TEXT text
Display shows text
User presses OK
User pressed OK
GENERATE-AC
GENERATE_AC f(text, number)
ARQC
GENERATE_AC f(text, number)
AAC
g(ARQC)

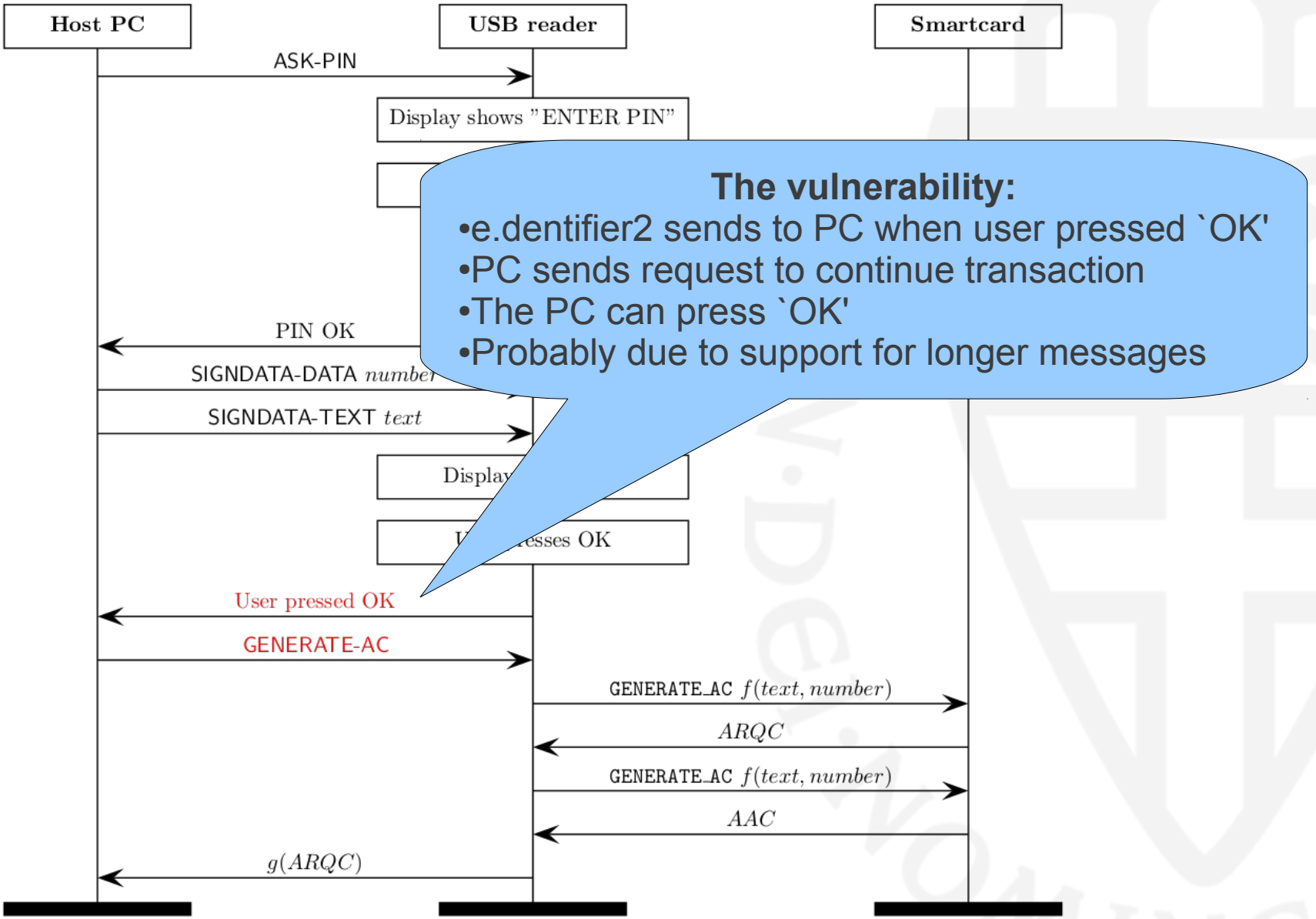
# Protocol


Host PC
USB reader
Smartcard
ASK-PIN
Display shows "ENTER PIN"
User enters PIN
VERIFY pin guess
PIN OK
PIN OK
SIGNDATA-DATA number
SIGNDATA-TEXT text
Display shows text
User presses OK
User pressed OK
GENERATE-AC
GENERATE_AC f(text, number)
ARQC
GENERATE_AC f(text, number)
AAC
g(ARQC)
ABN·AMRO
Domestic
1 transaction(s)
EUR 5.000,00
Confirm with OK
C
e.dentifier2
OK

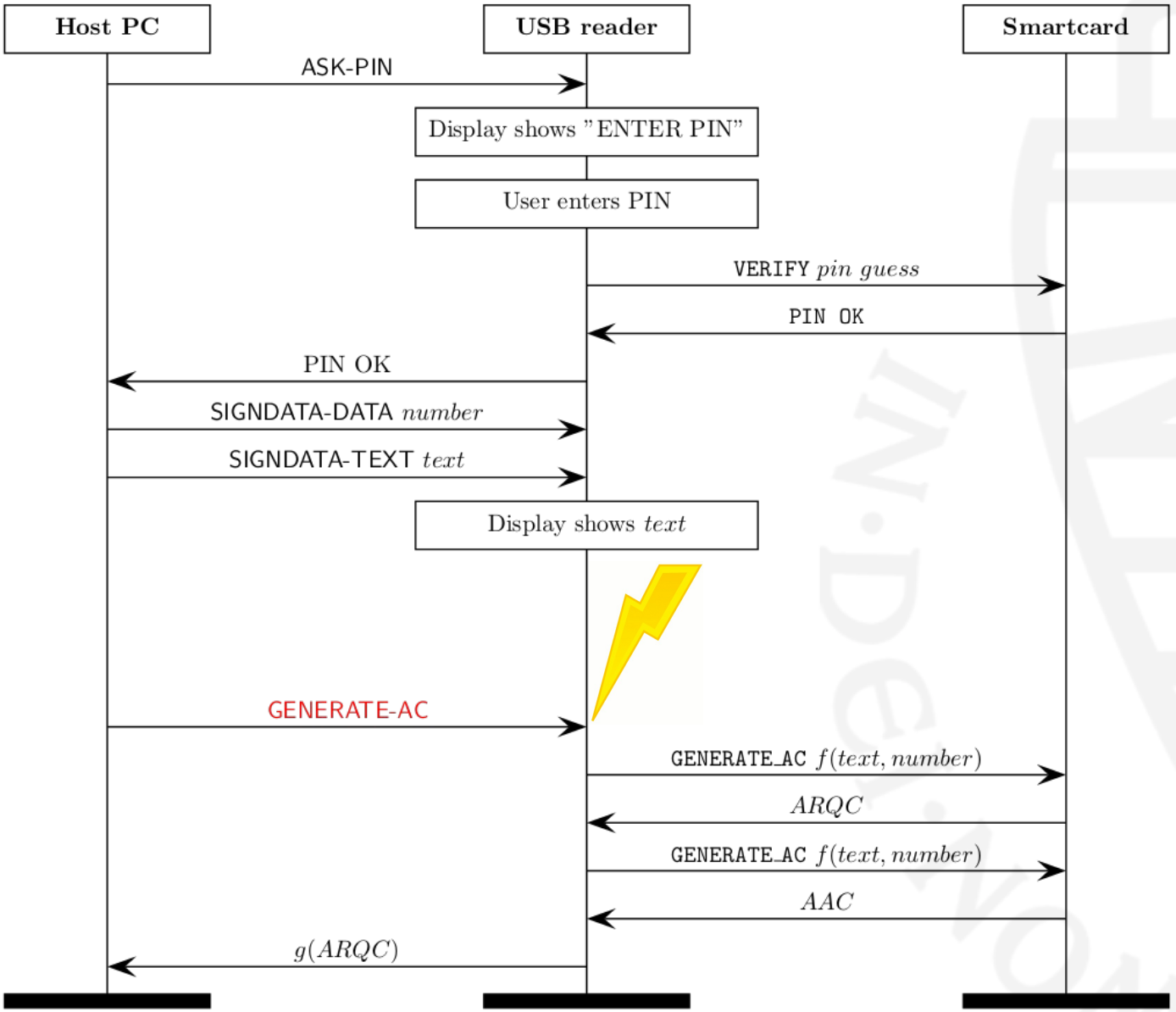
# Protocol


Host PC
USB reader
Smartcard
ASK-PIN
Display shows "ENTER PIN"
User enters PIN
VERIFY pin guess
PIN OK
PIN OK
SIGNDATA-DATA number
SIGNDATA-TEXT text
Display shows text
User presses OK
User pressed OK
GENERATE-AC
GENERATE_AC f(text, number)
ARQC
GENERATE_AC f(text, number)
AAC
g(ARQC)
Unknown functions f and g

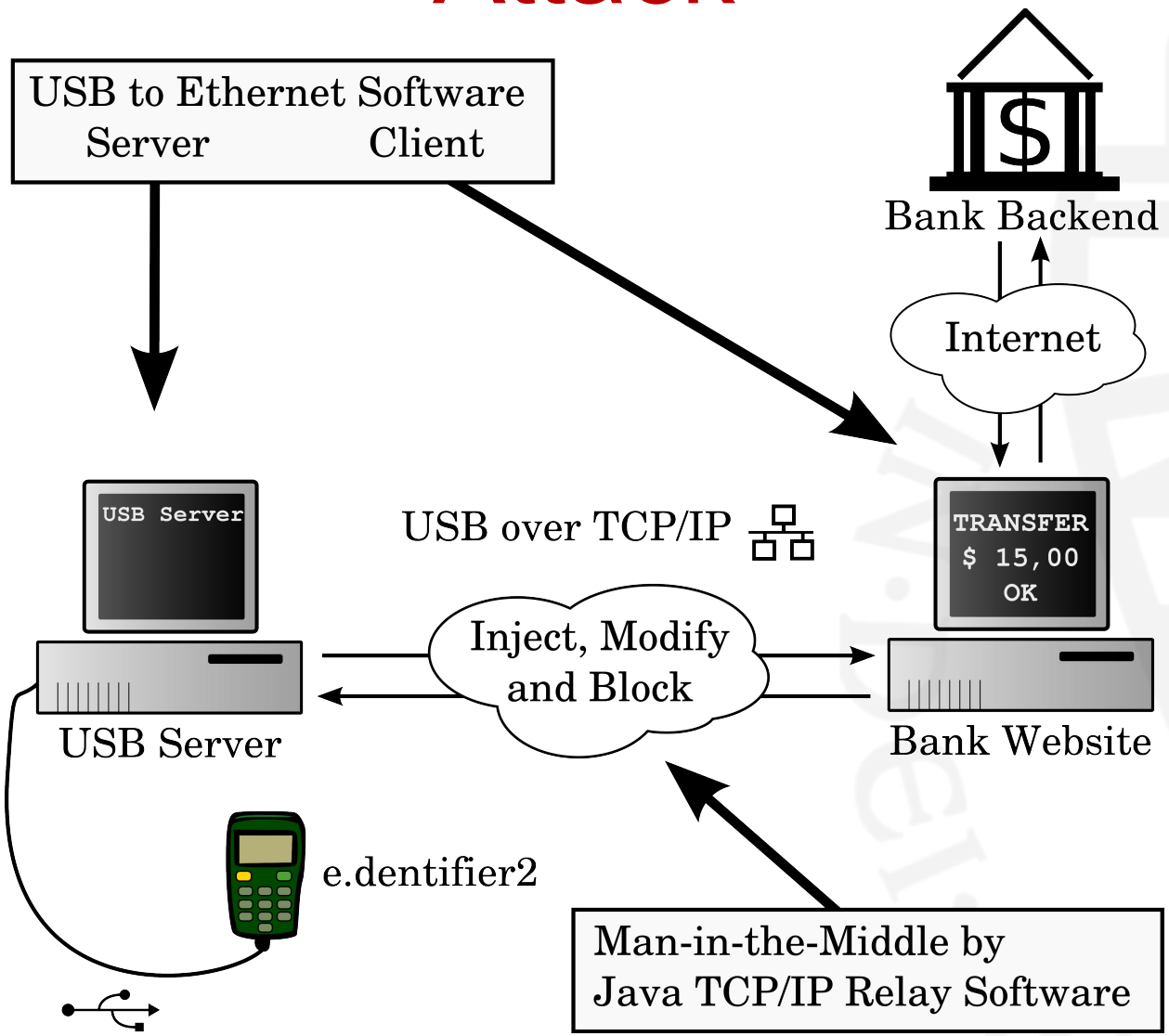
# Protocol


Host PC
USB reader
Smartcard
ASK-PIN
Display shows "ENTER PIN"
PIN OK
SIGNDATA-DATA number
SIGNDATA-TEXT text
User pressed OK
GENERATE-AC
GENERATE_AC f(text, number)
ARQC
GENERATE_AC f(text, number)
AAC
g(ARQC)
The vulnerability:
•e.dentifier2 sends to PC when user pressed `OK'
•PC sends request to continue transaction
•The PC can press `OK'
•Probably due to support for longer messages

# Protocol


Host PC
USB reader
Smartcard
ASK-PIN
Display shows "ENTER PIN"
User enters PIN
VERIFY pin guess
PIN OK
PIN OK
SIGNDATA-DATA number
SIGNDATA-TEXT text
Display shows text
GENERATE-AC
GENERATE_AC f(text, number)
ARQC
GENERATE_AC f(text, number)
AAC
g(ARQC)

# Attack


USB to Ethernet Software
Server
Client
Bank Backend
Internet
USB Server
USB over TCP/IP
TRANSFER
$ 15,00
OK
Inject, Modify
and Block
USB Server
Bank Website
e.dentifier2
Man-in-the-Middle by
Java TCP/IP Relay Software

# Video


ABN·AMRO
U bent ingelogd.
Laatste bezoek:
26 okt 11, 14:09u
OK
C
e.dentifier2
1
2
3
4
5
6
7
8
9
0
Maestro
Wereldpas

# Reaction bank

- Existing devices cannot be patched
- Visited us a few days after the discovery
- Informed their supplier
- Production was stopped and an improved version developed
- We were asked to investigate the improved version, but this proved problematic due to NDAs

# Implications

- Handelsbanken uses same device
- Is the e.dentifier2 more secure with or without USB?
  - Using USB it should have been more secure
  - Now equally “secure” depending on challenges, however USB attack requires ABN-AMRO specific malware

# Conclusion

- If implemented correctly What-you-see-is-what-you-sign is a good idea
- How could this ever have been introduced in development & missed in security review??
- Reduce control of PC over the device to minimum
- Stick to Kerckhoffs' principle (no NDAs)

# Conclusion

- If implemented correctly What-you-see-is-what-you-sign is a good idea
- How could this ever have been introduced in development & missed in security review??
- Reduce control of PC over the device to minimum
- Stick to Kerckhoffs' principle (no NDAs)

## Thanks for your attention!