Reverse engineering Internet banking

Eduardo Pablo Novella Lorente

e.novellalorente@student.ru.nl

http://ednolo.alumnos.upv.es

Institute for Computing and Information Sciences – Digital Security
Radboud University Nijmegen

24 June 2013
Nijmegen (The Netherlands)

Outline

Introduction

1. Handheld smartcard readers: USB-connected & unconnected
2. ABN-AMRO & ING Direct
3. E.dentifier2 : Attack to e.dentifier2 (2012)
4. Try the attack in the new readers
5. Additional funcionalities: Mode1 & Mode2

Background

EMV-CAP

- Based on EMV
- Reverse engineered
- EMV-CAP handheld smartcard readers
- Login & Signing using challenge-response

e.dentifier2

- ABN-AMRO EMV-CAP reader
- Reverse engineered by Digital Security
- Versions : Old (2007) & new (2012)
- Modes: USB-connected & unconnected
- Operations: Login & Signing of transactions

Background

Challenge-response

2 Application Cryptograms (AC) are created as proof of authorization from smartcard

- ARQC (Authorization Request Cryptogram). Response against the challenge sent
- AAC (Application Authentication Cryptogram). Verification

DigiPass 850

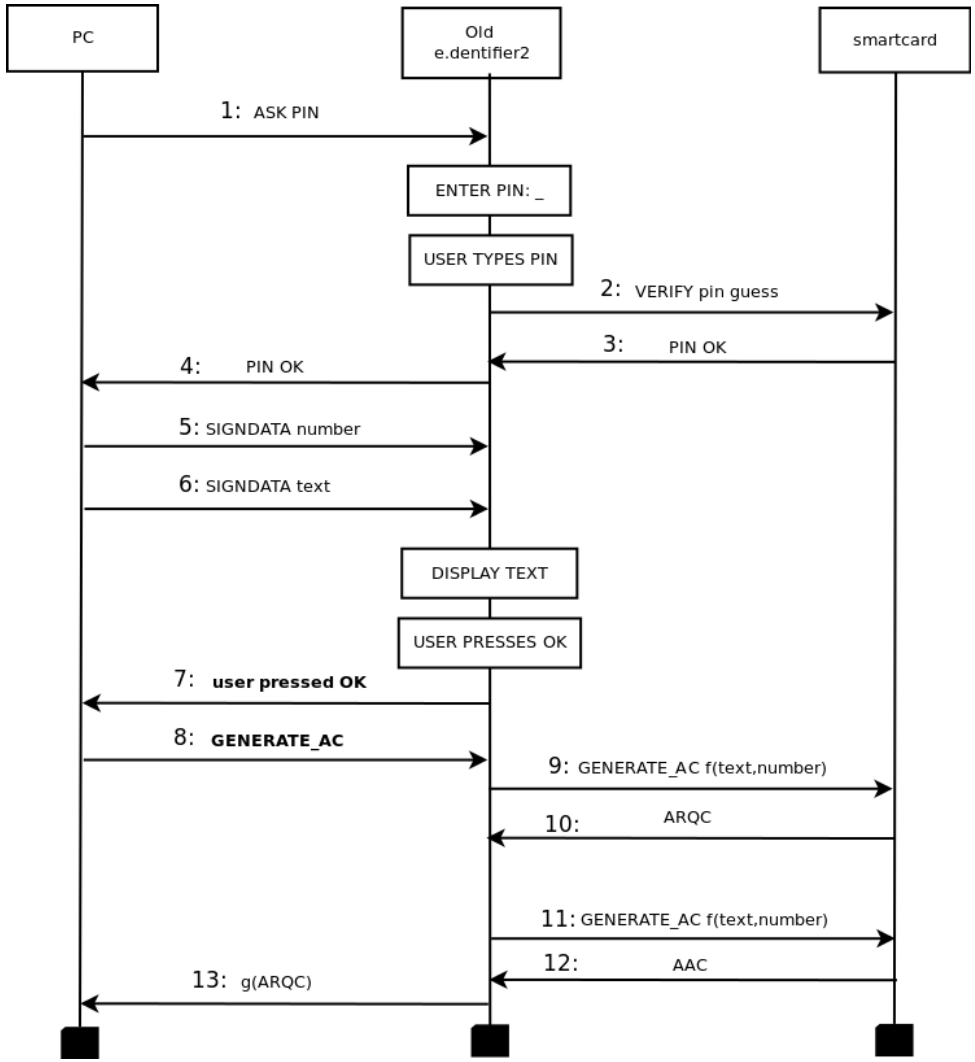
- ING Direct EMV-CAP reader
- Modes: USB-connected & unconnected
- Operations: Login & Signing of transactions

Background

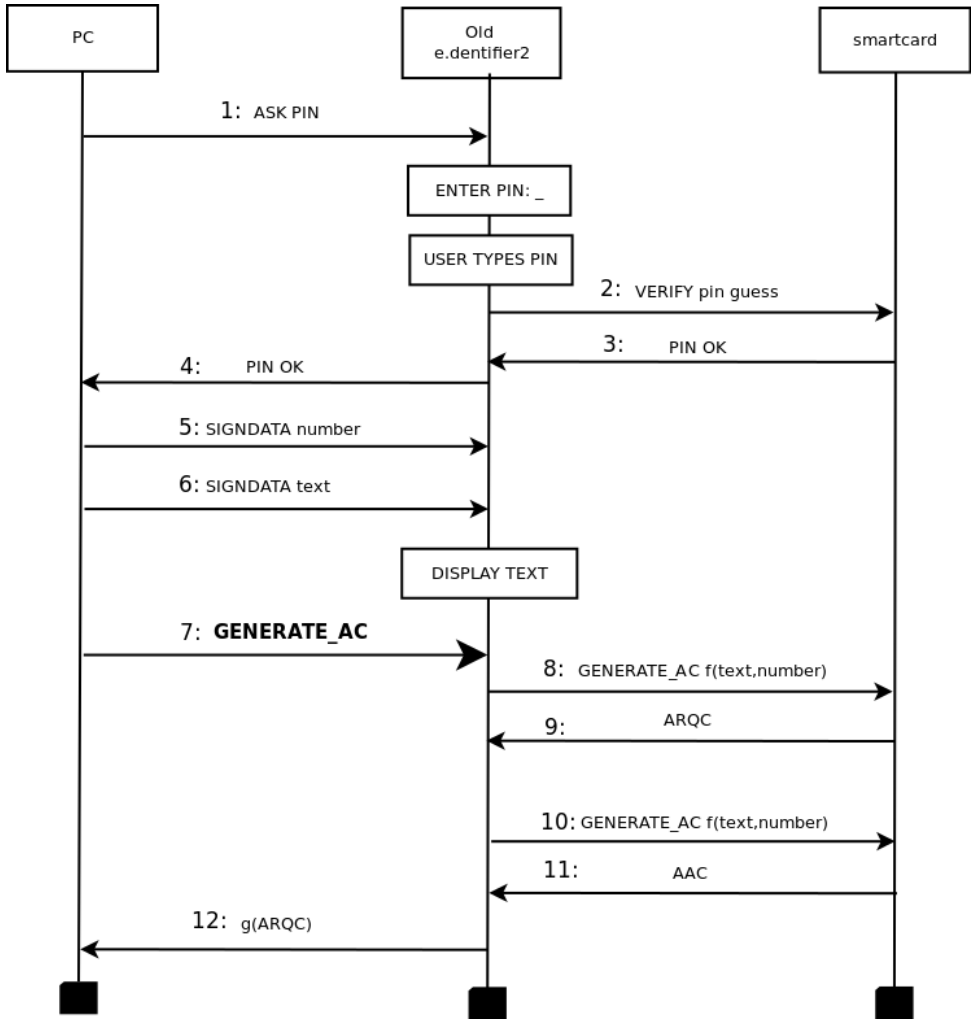
SWYS

- aka "What You Sign Is What You See" (WYSIWYS)
- Pretend to avoid Man-in-the-browser attacks
- PIN code has been entered in the reader
- Cardholder can accept/deny operations' messages
- Cardholder can understand messages
- But: Bad designed (Attack by Digital Security)

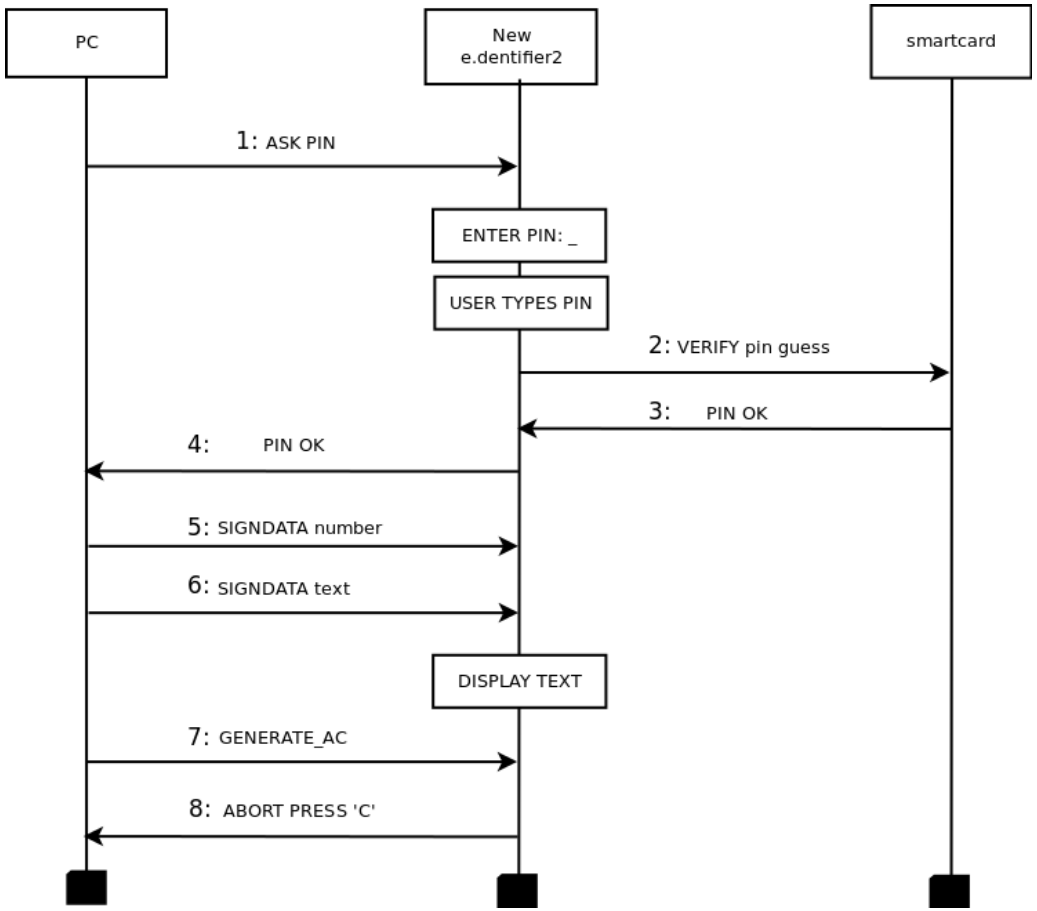
Vunerability in the old e.dentifier2



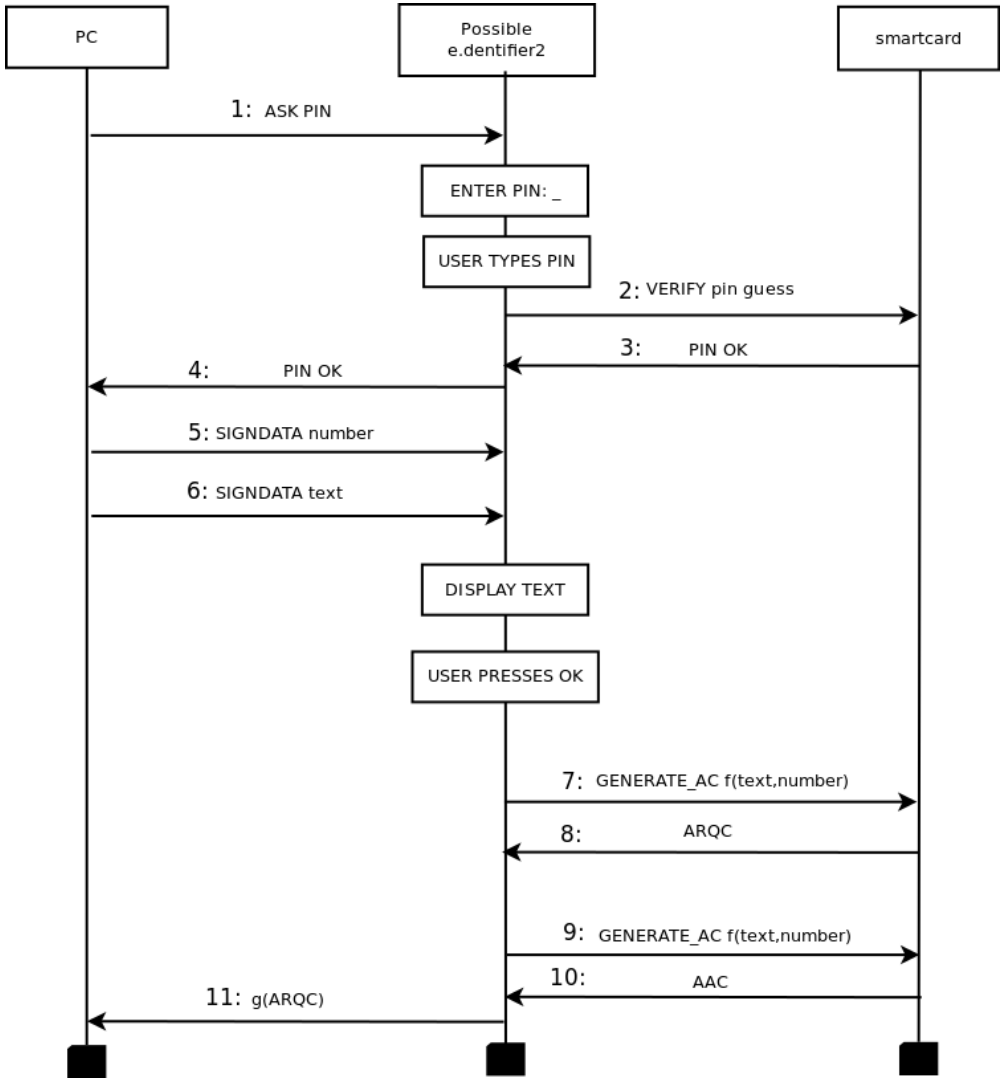
Attack in the old e.dentifier2



Patch in the new e.dentifier2



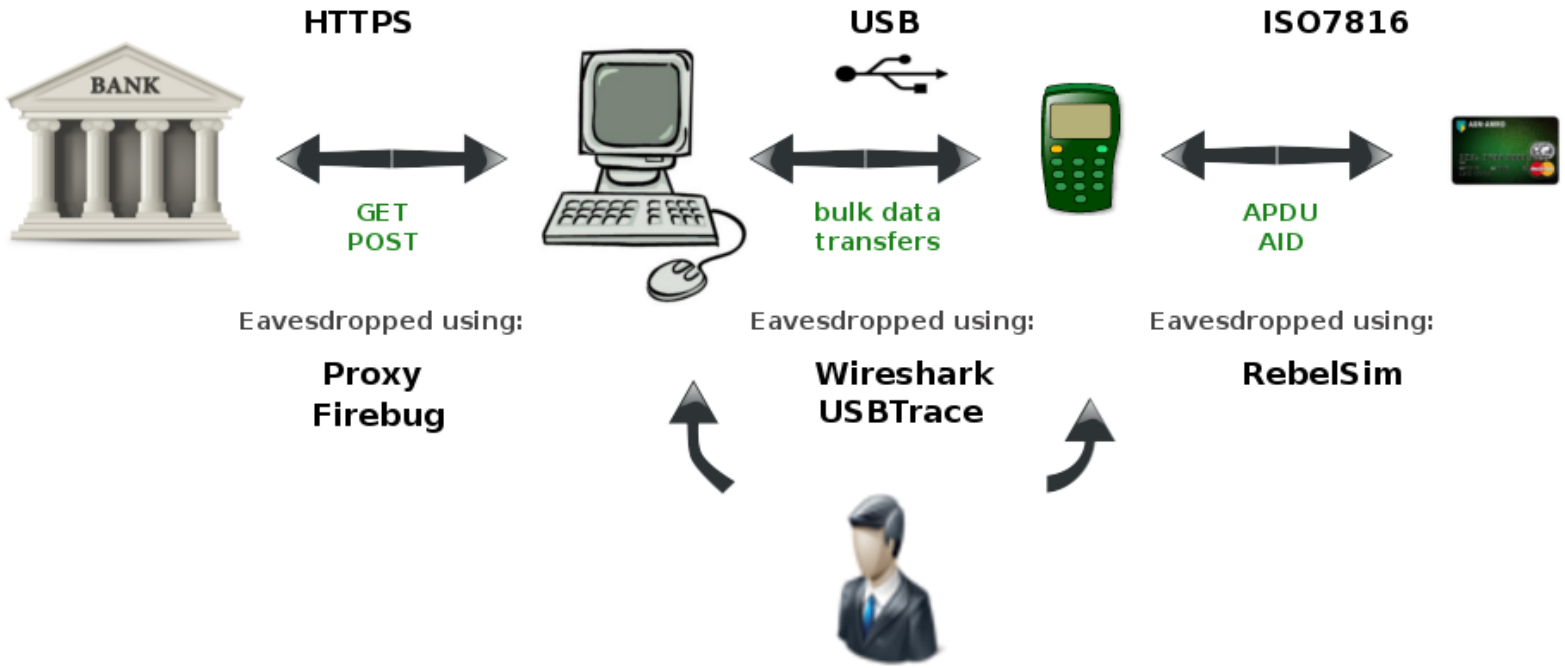
Possible correct SWYS protocol



Tools

1. Wireshark & USBTrace
2. RebelSim & RealTerm
3. Fake bankcard with Javacards
4. Own webpage
5. Python code using PyUSB library
6. Firebug Add-on

Big picture


HTTPS
USB
ISO7816
BANK
GET
POST
bulk data
transfers
APDU
AID
Eavesdropped using:
Proxy
Firebug
Eavesdropped using:
Wireshark
USBTrace
Eavesdropped using:
RebelSim

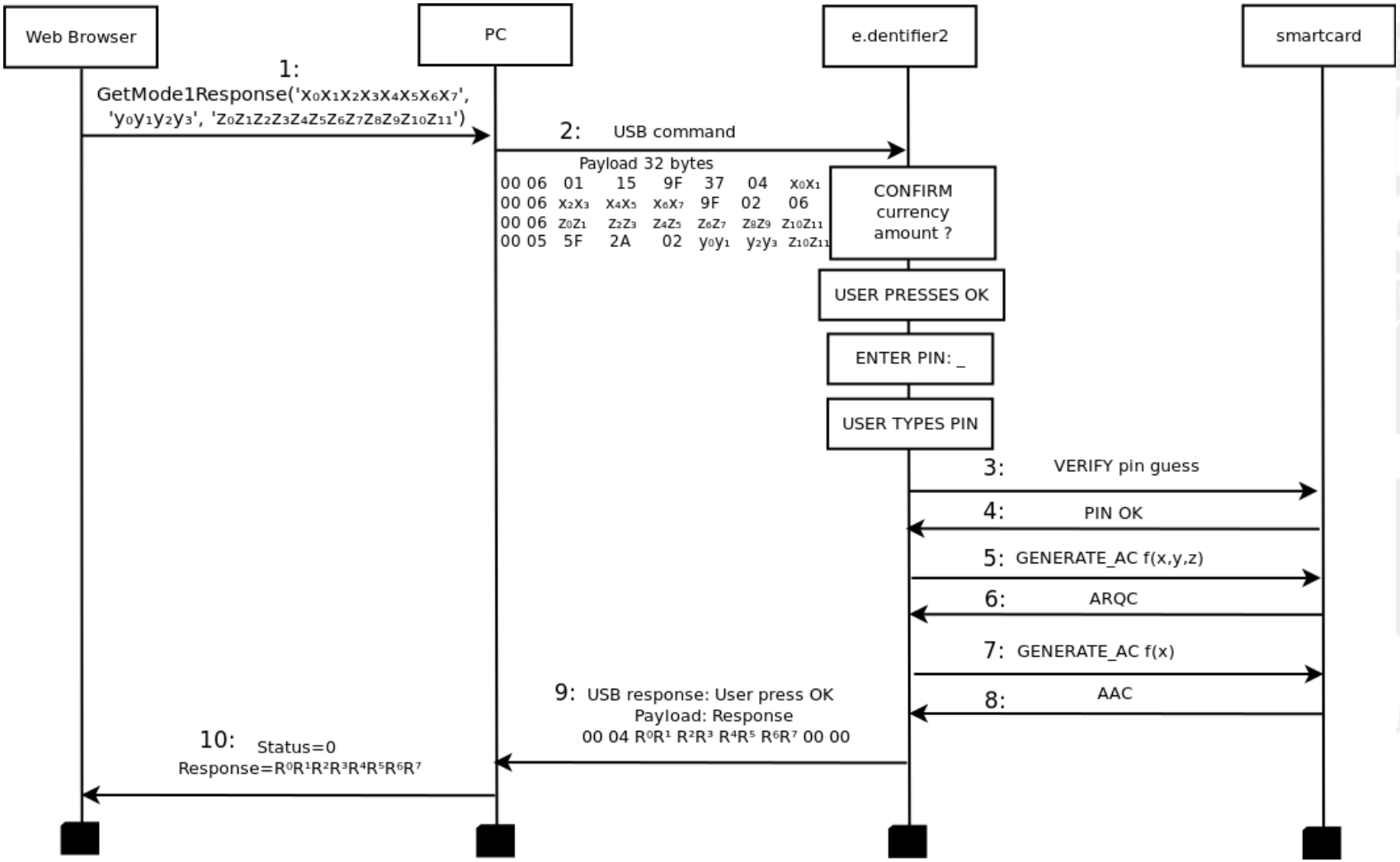
Mode1

GetMode1Response (Challenge, Currency, Amount)

JavaScript functions in ABN-AMRO website. File :*BECON.js*

- Reverse engineered
- Signing using challenge-response
- Unconnected mode has this mode
- Challenge 8 numeric digits
- Currency 4 digits for EMV code (0978 €) (0826 £) (0840 $)
- Amount 12 numeric digits between [0000.000.000,00 - 0999.999.999,99]

Protocol of GetMode1Response



Reverse engineering Mode1

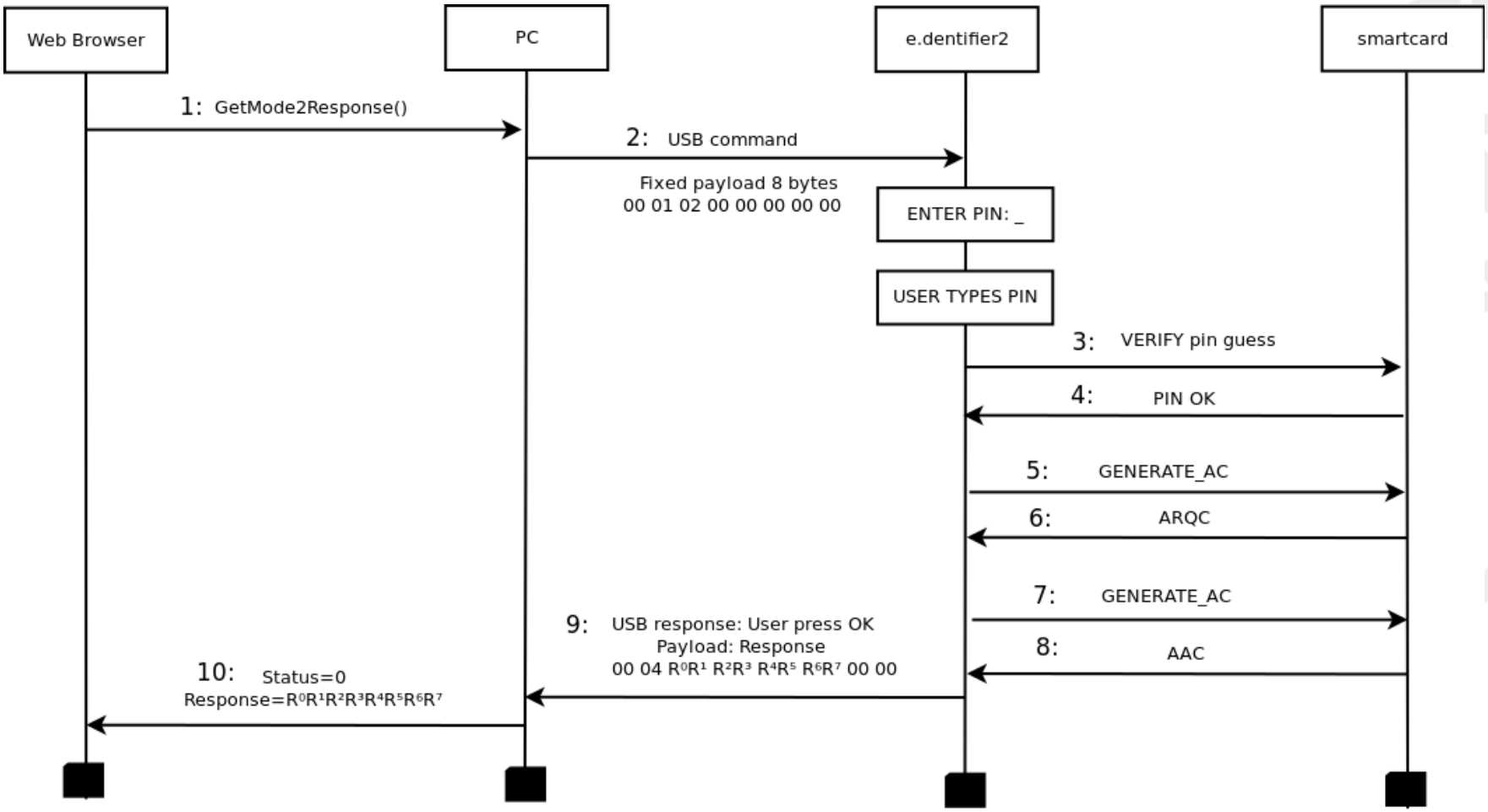
Mode2

GetMode2Response()

JavaScript functions in ABN-AMRO website. File :*BECON.js*

- Reverse engineered
- Login
- Generate a right response

Protocol of GetMode2Response



Conclusion

- if (SWYS) safe$_{++}$; else problems=true;
- Mode1 & Mode2 are more secure