ePassports reloaded

Jeroen van Beek

BlackHat USA 2008, Las Vegas

Where will we go today?

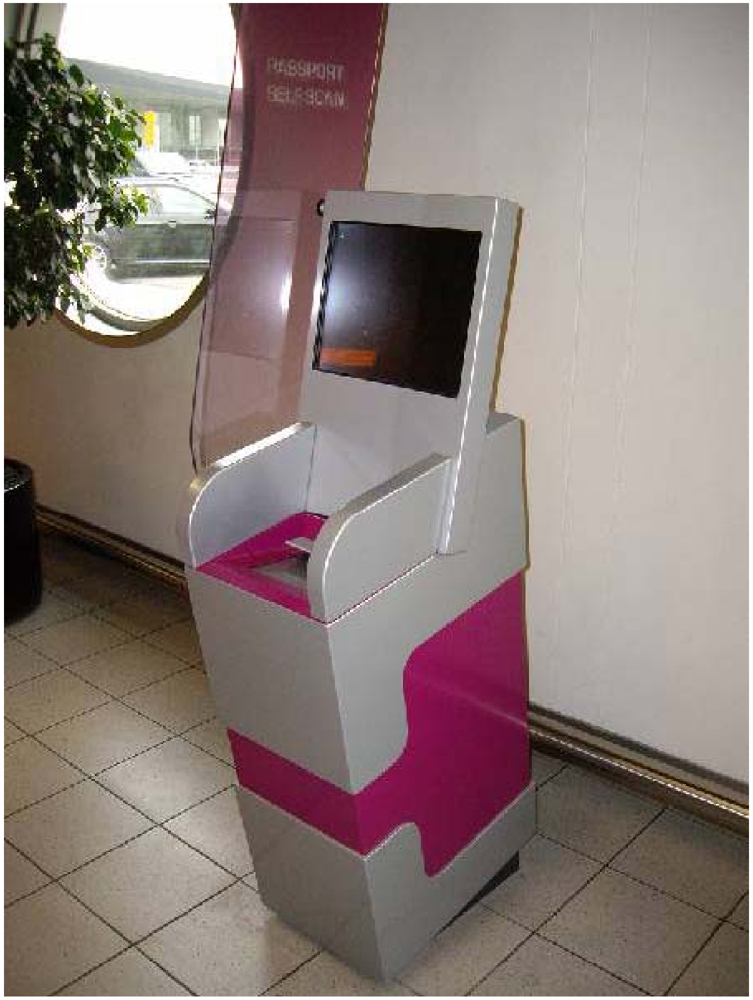
- Technology overview
- Attacks
  - – The ICAO standard
  - – Known attacks
  - – Verification process
  - – Finding new flaws
- Root causes
- Solutions
- The future(?|!)
- Questions

Technology overview

- An ePassport contains a chip
- The chip contains data about the passport holder
  - Name, date of birth, passport number, etc.
  - Biometrics (picture, finger prints, iris scan)
- Chip content is based on a standard by the International Civil Aviation Organization (ICAO)
  - See http://www.mrtd.icao.int/images/stories/Doc/ePassports/PKI_for_Machine_Readable_Travel_Documents_offering_ICC_read-only_access_v1.1.pdf for details
- *Chip content is accessible using a wireless interface (RFID)*
- ePassports are enrolled on a global scale
- Not widely used for real-life applications (yet)

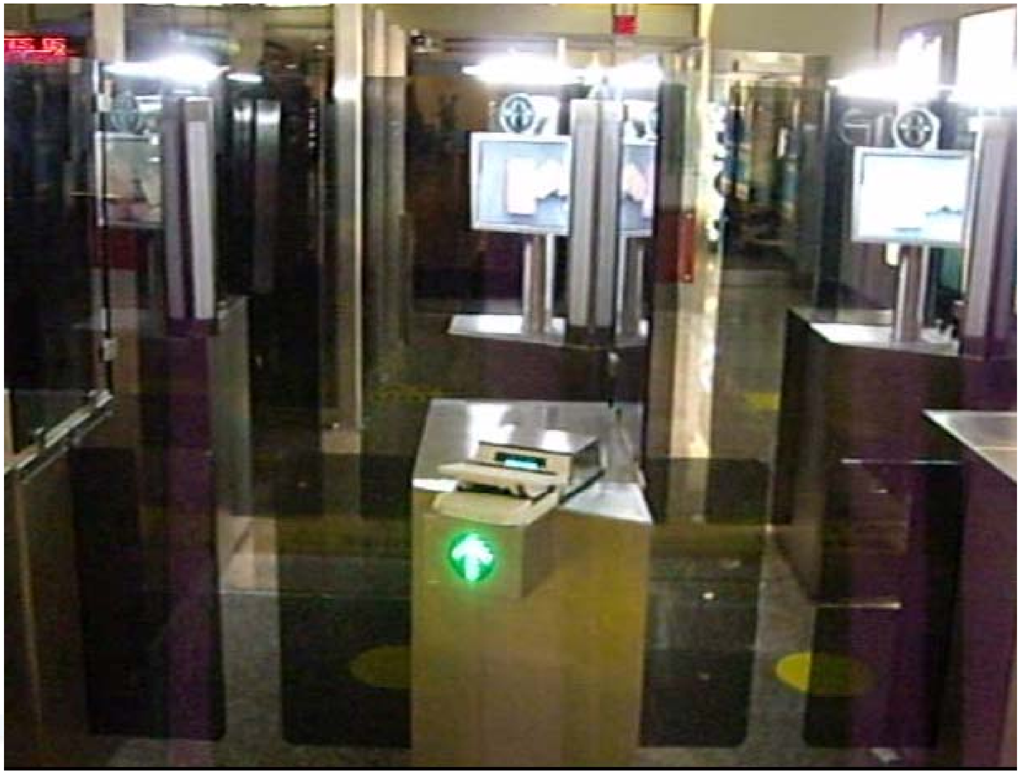
Technology overview, ct.

- So what does it look like? Test setup at Amsterdam Airport (always broken or switched off):



Technology overview, ct.

- So what does it look like? At the airport:



The ICAO standard: chip content

- Chip contains files (“Elementary Files”, EFs):
  - EF.DG1: personal information (required)
  - EF.DG2: picture, JPG/JPG2000 (required)
  - EF.DG[3-14]: finger prints, iris scans and other files for future use (optional)
  - EF.DG15: anti-cloning crypto (optional)
  - EF.SOD: safeguarding integrity of the files above (required)
  - EF.COM: index of available files (required)
  - **Demo!**

The ICAO standard: security

- The standard describes protection mechanisms:
  - Passive authentication (PA) (required):
    - Safeguard integrity of data
    - EF.SOD stores hashes of EF.DG[1-15] and a public key, hashes are signed with a private key
  - Basic Access Authentication (BAC) (optional):
    - Safeguard confidentiality of data
    - Authentication is required before reading files
    - KEY = DOCUMENT NUMBER + DATE OF BIRTH + DATE OF EXPIRY
    - After authentication data is encrypted (3DES) and messages contain MACs (MAC8)
  - Active Authentication (AA) (optional):
    - Prevent cloning and copying
    - EF.DG15 contains a public key. The private key of this key pair is in inaccessible chip memory. Authenticity of the chip can be checked by letting the chip sign a reader's challenge and verifying the result with the public key

Known attacks

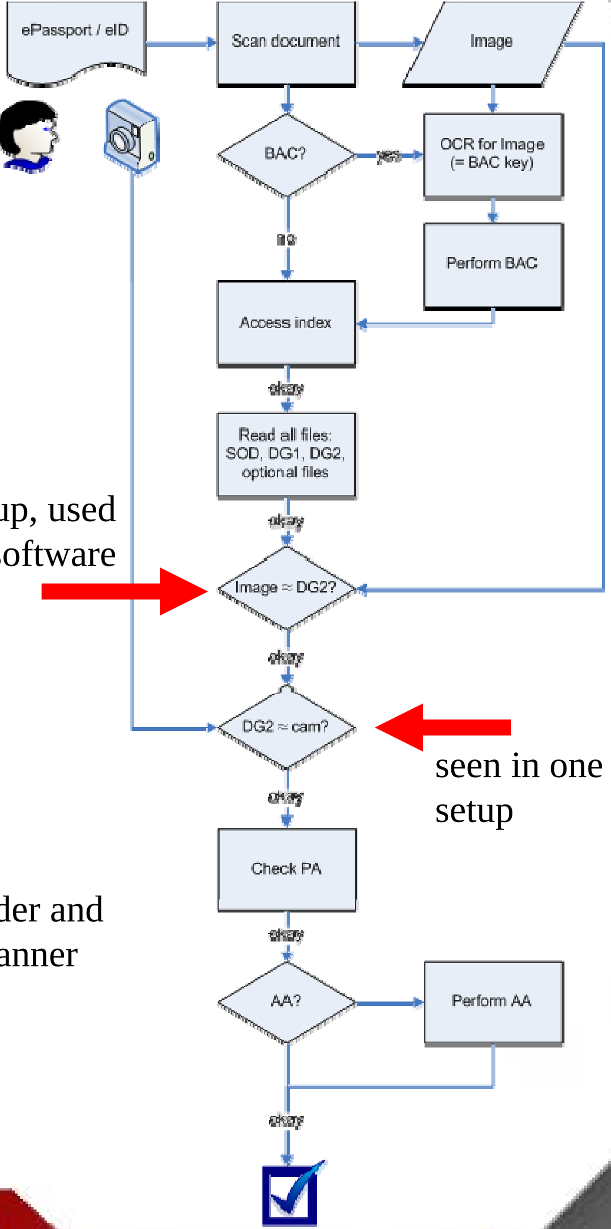
- Real life attacks, the past:
  - Cloning ePassports without Active Authentication
    - Lukas Grunwald @ BlackHat, USA, 2006
    - http://www.wired.com/science/discoveries/news/2006/08/71521
    - Bit by bit copy of content in a self-written ePassport applet
    - Can be prevented by using Active Authentication
  - Cloning ePassports with Active Authentication enabled
    - Marc Witteman @ What The Hack, The Netherlands, 2005
    - http://wiki.whatthehack.org/images/2/28/WTH-slides-Attacks-on-Digital-Passports-Marc-Witteman.pdf
    - Using Differential Power Analysis to retrieve AA private key
    - Can be prevented by using proper hardware

Known attacks, ct.

- Real life attacks, the past:
  - Read ePassports with predictable document numbers
    - Adam Laurie reads BAC protected UK ePassport of a Guardian reporter, UK, 2006
    - http://www.computerweekly.com/Articles/2006/11/21/219995/expert-cracks-biometric-passport-data.htm
    - An educated guess (sequential document numbers), also see Witteman's slides
    - Can be prevented by using non-sequential document numbers (though effective key length is still only ~72 out of 128 bits)
  - Fingerprint ePassports without authenticating
    - Radboud University / Lausitz University team @ NLUUG, The Netherlands, 2008
    - http://www.cs.ru.nl/~erikpoll/papers/nluug.pdf
    - Characteristics of APDU responses show the origin of the applet
    - Can be prevented by using standard response codes ("status words")

Verification process

- Two steps seem so be optional:
  - Scannend image versus chip image
  - Chip image versus camera image



ePassport / eID

Scan document

Image

BAC?

yes

OCR for Image (= BAC key)

Perform BAC

no

Access index

okay

Read all files: SOD, DG1, DG2, optional files

okay

Image ≈ DG2?

okay

DG2 ≈ cam?

okay

Check PA

okay

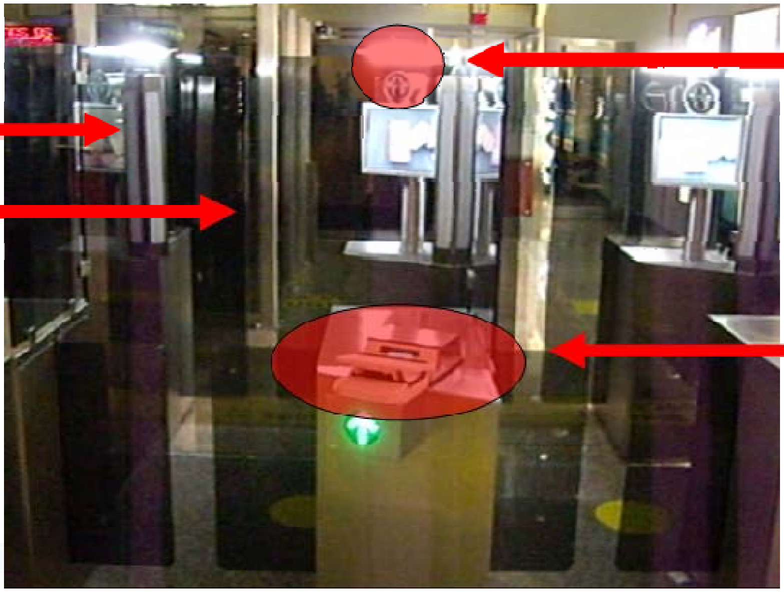
AA?

Perform AA

okay

seen in one setup, used in non-public software

seen in one setup



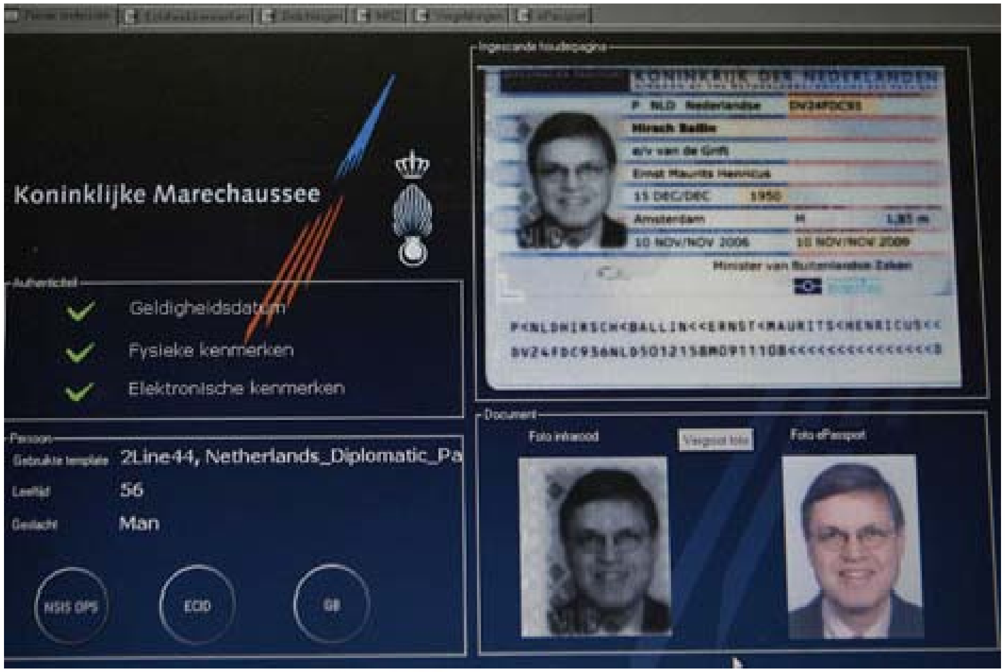
camera

doors

RFID reader and optical scanner

Verification process, ct.

- Dutch immigration seems to use (test) software which uses scan↔chip checks
  - And the minister of justice proudly shares his passport data on the net :)



Authenticiteit

✓ Geldigheidsdatum

✓ Fysieke kenmerken

✓ Elektronische kenmerken

Persoon

Gebruikte template 2Line44, Netherlands_Diplomatic_Pa

Leeftijd 56

Geslacht Man

Ingescande houderpagina

KONINKRIJK DER NEDERLANDEN

P NLD Nederlandse DV24FDC93

Hirsch Ballin

e/v van de Grift

Ernst Maurits Henricus

15 DEC/DEC 1950

Amsterdam M 1,81 m

10 NOV/NOV 2006 10 NOV/NOV 2009

Minister van Buitenlandse Zaken

P<NLDHIRSCH<BALLIN<<ERNST<MAURITS<HENRICUS<<

DV24FDC936NLD5012158M0911108<<<<<<<<<<<<<<<8

Document

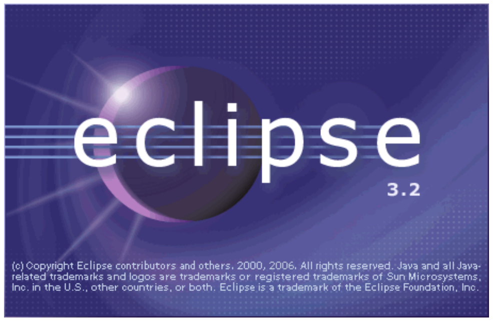
Foto infrarood

Foto ePaspoort

Finding new flaws

- First we need a test platform

RFID reader, ~ $75


eclipse
3.2
(c) Copyright Eclipse contributors and others. 2000, 2006. All rights reserved. Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S., other countries, or both. Eclipse is a trademark of the Eclipse Foundation, Inc.

Eclipse & JCOP plug-in, ~ $0

All-in-one printer, ~$75

laptop computer, ~ $750

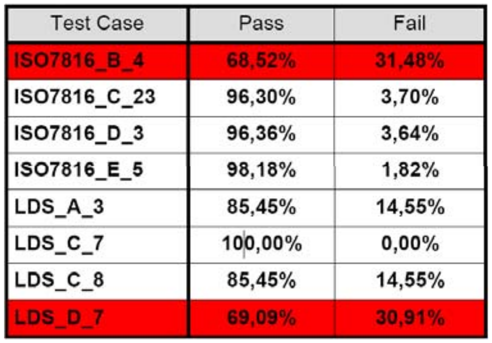
JCOP smartcard, ~$20

Finding new flaws, ct.

- Then we need code that emulates the ePassport applet
  - Just follow the specs, check ICAO’s “worked example”
  - Add function to write data to the applet
- Your applet can be tested quite easily
  - Clone data from a non-AA protected ePassport
  - Perform a read-out with Adam Laurie’s excellent RFIDIOt tools http://rfidiot.org/
  - Change both mrpkey’s and your applet code to make a Debian style random number generator
  - Fix bugs :)
  - **Code snippets!**

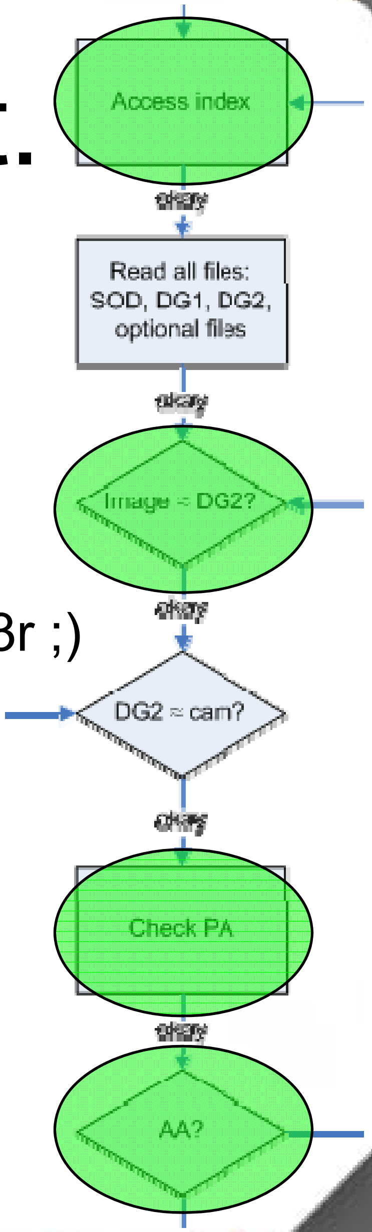
Finding new flaws, ct.

- Find interesting checks
- ePassport programmers:
  - Are only human
  - Are not (all) security aware
  - Make the same mistakes I do as an amateur c0d3r ;)
  - Check results of interoperability tests:



Test Case	Pass	Fail
ISO7816_B_4	68,52%	31,48%
ISO7816_C_23	96,30%	3,70%
ISO7816_D_3	96,36%	3,64%
ISO7816_E_5	98,18%	1,82%
LDS_A_3	85,45%	14,55%
LDS_C_7	100,00%	0,00%
LDS_C_8	85,45%	14,55%
LDS_D_7	69,09%	30,91%

  - http://www.interoptest-berlin.de/pdf/Munde_Seidel_-_Preliminary_Test_Results.pdf



Finding new flaws, ct.

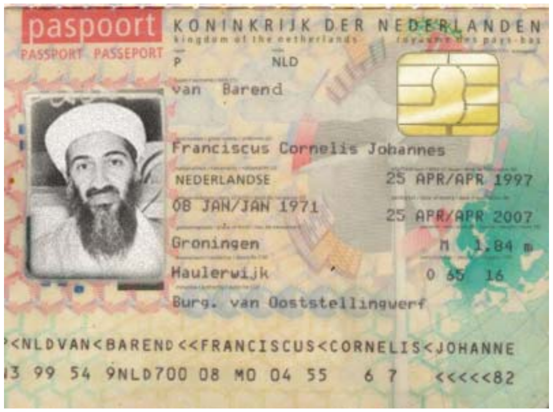
- Implement an attack and test it
- Implement an attack and test
- Implement an attack and test i
- Implement an attack and test i
- Implement an attack and te
- Implement an attack and t
- Implement an attack an

Finding new flaws, ct.

- To get a working copy / new ePassport we need to:
  - Get reference implementations:
    - Golden Reader Tool, referenced in ICAO documentation
    - Real-life test setups
  - Pass “image scan = image chip” test
  - Pass “Passive Authentication” tests
  - Pass “Active Authentication” test (enabled on e.g. Dutch documents)

Finding new flaws, ct.

- Pass “image scan = image chip” test
  - Get an updated image you would like to use
  - Get OCR-B fonts for MRZ (= BAC key)
  - Copy/paste the picture and MRZ in the right place
  - *Advanced equipment is on the market*
    - *IR scans*
    - *UV scans*
    - *Systems are as strong as the weakest link*
  - **Demo included later on!**



paspoort PASSPORT PASSEPORT

KONINKRIJK DER NEDERLANDEN
kingdom of the netherlands royaume des pays-bas

P NLD

van Barend

Franciscus Cornelis Johannes

NEDERLANDSE 25 APR/APR 1997

08 JAN/JAN 1971 25 APR/APR 2007

Groningen M 1.84 m

Haulerwijk 0 65 16

Burg. van Ooststellingwerf

```
P<NLDVAN<BAREND<<FRANCISCUS<CORNELIS<JOHANNE
3 99 54 9NLD700 08 M0 04 55   6 7   <<<<<82
```

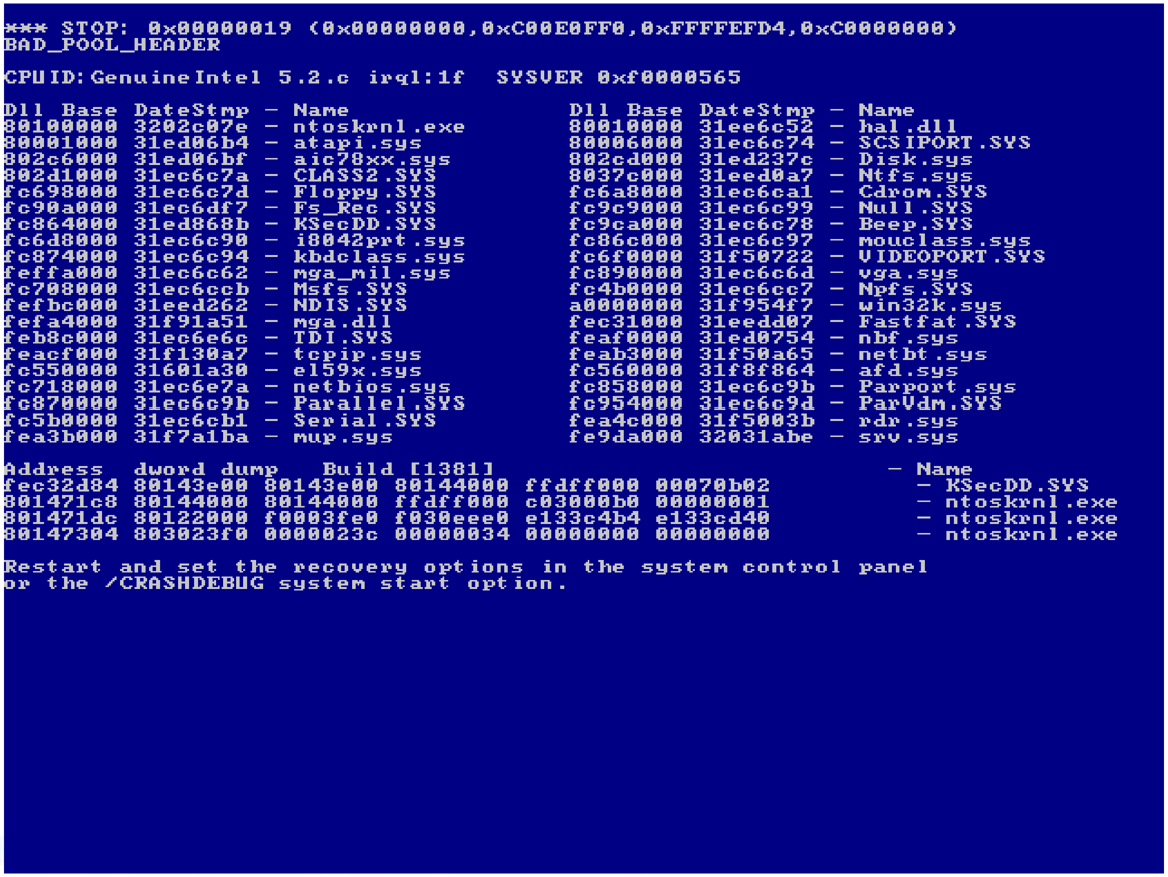
Finding new flaws, ct.

- Pass “Passive Authentication” tests
  - Hashes of all data groups are stored
  - Signing of the hashes
    - Public key is in SOD for chip-only authentication
    - Authorized public keys (KPuDS) of all countries *should be* in *all* read-out equipment
    - ICAO Public Key Directory (PKD) should facilitate this
      - ICAO, May 2008: *“The ICAO PKD has grown to nine participants”*
      - 36 participants at the interoperability tests 2006
      - What about the other 27(+)? And e.g. exchange Israel ↔ Iran?
    - Create self-signed key pairs, thanks to Peter Gutmann http://www.cs.auckland.ac.nz/~pgut001/
  - PA checks are covered by the ICAO standard. What about the implementation?

Warning or error?

Warning or error?

Warning or error?

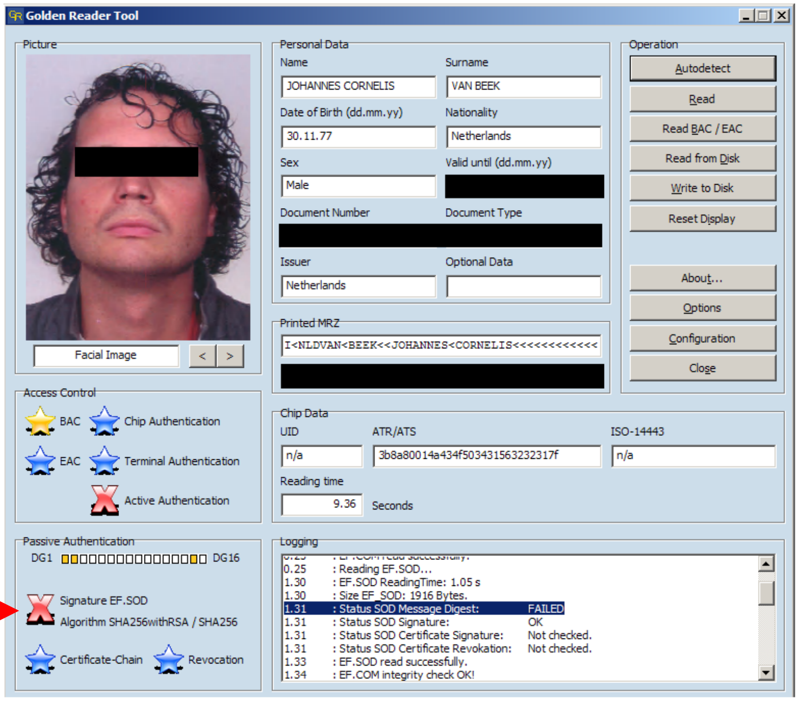


```
*** STOP: 0x00000019 (0x00000000,0xC00E0FF0,0xFFFFEFD4,0xC0000000)
BAD_POOL_HEADER

CPUID:GenuineIntel 5.2.c irql:1f  SYSVER 0xf0000565

Dll Base DateStmp - Name               Dll Base DateStmp - Name
80100000 3202c07e - ntoskrnl.exe       80010000 31ee6c52 - hal.dll
80001000 31ed06b4 - atapi.sys          80006000 31ec6c74 - SCSIPORT.SYS
802c6000 31ed06bf - aic78xx.sys        802cd000 31ed237c - Disk.sys
802d1000 31ec6c7a - CLASS2.SYS         8037c000 31eed0a7 - Ntfs.sys
fc698000 31ec6c7d - Floppy.SYS         fc6a8000 31ec6ca1 - Cdrom.SYS
fc90a000 31ec6df7 - Fs_Rec.SYS         fc9c9000 31ec6c99 - Null.SYS
fc864000 31ed868b - KSecDD.SYS         fc9ca000 31ec6c78 - Beep.SYS
fc6d8000 31ec6c90 - i8042prt.sys       fc86c000 31ec6c97 - mouclass.sys
fc874000 31ec6c94 - kbdclass.sys       fc6f0000 31f50722 - VIDEOPORT.SYS
feffa000 31ec6c62 - mga_mil.sys        fc890000 31ec6c6d - vga.sys
fc708000 31ec6ccb - Msfs.SYS           fc4b0000 31ec6cc7 - Npfs.SYS
fefbc000 31eed262 - NDIS.SYS           a0000000 31f954f7 - win32k.sys
fefa4000 31f91a51 - mga.dll            fec31000 31eedd07 - Fastfat.SYS
feb8c000 31ec6e6c - TDI.SYS            feaf0000 31ed0754 - nbf.sys
feacf000 31f130a7 - tcpip.sys          feab3000 31f50a65 - netbt.sys
fc550000 31601a30 - el59x.sys          fc560000 31f8f864 - afd.sys
fc718000 31ec6e7a - netbios.sys        fc858000 31ec6c9b - Parport.sys
fc870000 31ec6c9b - Parallel.SYS       fc954000 31ec6c9d - ParVdm.SYS
fc5b0000 31ec6cb1 - Serial.SYS         fea4c000 31f5003b - rdr.sys
fea3b000 31f7a1ba - mup.sys            fe9da000 32031abe - srv.sys

Address  dword dump   Build [1381]                            - Name
fec32d84 80143e00 80143e00 80144000 ffdff000 00070b02           - KSecDD.SYS
801471c8 80144000 80144000 ffdff000 c03000b0 00000001           - ntoskrnl.exe
801471dc 80122000 f0003fe0 f030eee0 e133c4b4 e133cd40           - ntoskrnl.exe
80147304 803023f0 0000023c 00000034 00000000 00000000           - ntoskrnl.exe

Restart and set the recovery options in the system control panel
or the /CRASHDEBUG system start option.
```

Warning or error?

Warning or error?

- The signature value is incorrect
  - A) Do nothing
  - B) Warning
  - C) Non-critical error
  - D) Critical error



[redacted]

Personal Data

Name: JOHANNES CORNELIS

Surname: VAN BEEK

Date of Birth (dd.mm.yy): 30.11.77

Nationality: Netherlands

Sex: Male

Valid until (dd.mm.yy): [redacted]

Document Number / Document Type: [redacted]

Issuer: Netherlands

Optional Data:

Printed MRZ

```
I<NLDVAN<BEEK<<JOHANNES<CORNELIS<<<<<<<<<<<<
```

[redacted]

Chip Data

UID: n/a

ATR/ATS: 3b8a80014a434f503431563232317f

ISO-14443: n/a

Reading time: 9.36 Seconds

Signature EF.SOD

Algorithm SHA256withRSA / SHA256

Logging

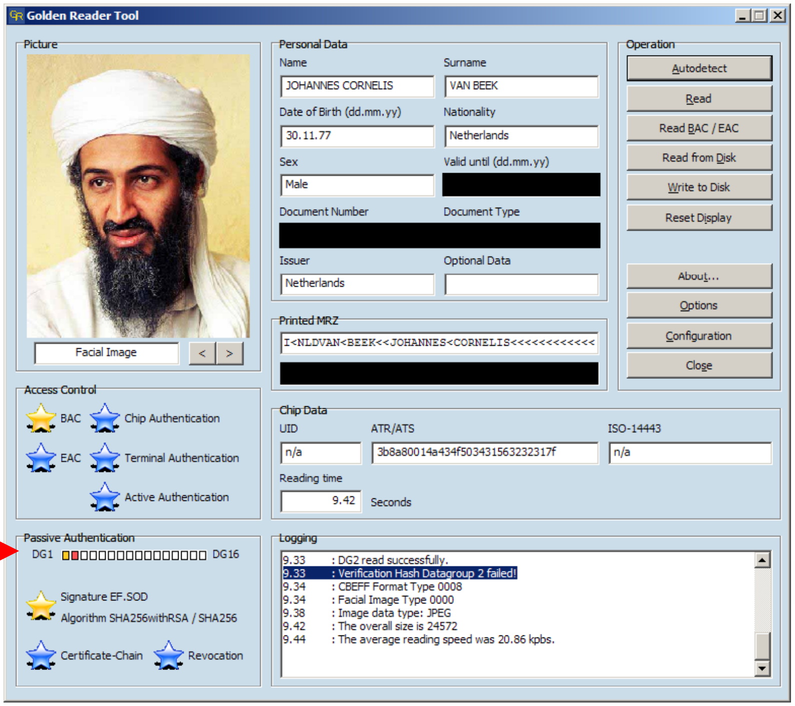
```
0.25     : Reading EF.SOD...
1.30     : EF.SOD ReadingTime: 1.05 s
1.30     : Size EF_SOD: 1916 Bytes.
1.31     : Status SOD Message Digest:            FAILED
1.31     : Status SOD Signature:                 OK
1.31     : Status SOD Certificate Signature:     Not checked.
1.31     : Status SOD Certificate Revokation:    Not checked.
1.33     : EF.SOD read successfully.
1.34     : EF.COM integrity check OK!
```

read-out continues and successfully finishes after detection of invalid SOD

Warning or error?

- A hash value is incorrect
  - A) Do nothing
  - [✓ B) Warning]
  - C) Non-critical error
  - D) Critical error

find the difference



Name: JOHANNES CORNELIS
Surname: VAN BEEK
Date of Birth (dd.mm.yy): 30.11.77
Nationality: Netherlands
Sex: Male
Valid until (dd.mm.yy): [redacted]
Document Number / Document Type: [redacted]
Issuer: Netherlands
Optional Data:

Printed MRZ:

```
I<NLDVAN<BEEK<<JOHANNES<CORNELIS<<<<<<<<<<<<
```

[redacted]

UID: n/a
ATR/ATS: 3b8a80014a434f503431563232317f
ISO-14443: n/a
Reading time: 9.42 Seconds

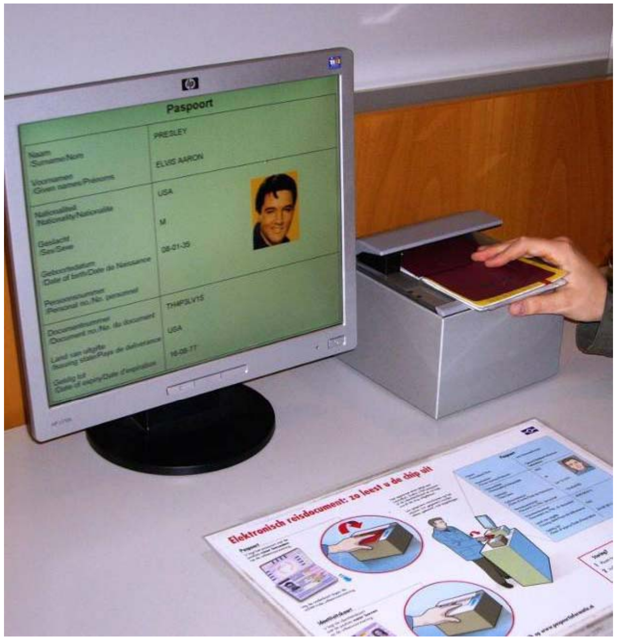
DG1 🟨🟥□□□□□□□□□□□□□□ DG16

Algorithm SHA256withRSA / SHA256

```
9.33    : DG2 read successfully.
9.33    : Verification Hash Datagroup 2 failed!
9.34    : CBEFF Format Type 0008
9.34    : Facial Image Type 0000
9.38    : Image data type: JPEG
9.42    : The overall size is 24572
9.44    : The average reading speed was 20.86 kpbs.
```

Finding new flaws, ct.

- This is all very strange… If the reference implementation is not that strict, what about real test setups?
  - Let’s try some publicly accessible test equipment
  - **Demo!**
  - *Note that the intended use for this setup is unclear: abuse is not possible (yet?)*


Paspoort
Naam /Surname/Nom
Voornamen /Given names/Prénoms
Nationaliteit /Nationality/Nationalité
Geslacht /Sex/Sexe
Geboortedatum /Date of birth/Date de Naissance
Persoonsnummer /Personal no./No. personnel
Documentnummer /Document no./No. du document
Land van uitgifte /Issuing state/Pays de délivrance
Geldig tot /Date of expiry/Date d'expiration
PRESLEY
ELVIS AARON
USA
M
08-01-35
TH4P3LV15
USA
16-08-77
Elektronisch reisdocument: zo leest u de chip uit

Finding new flaws, ct.

- Pass “Active Authentication” test
  - Not writing the file (DG15) doesn’t work
  - But what about manipulating read-out?
  - **Demo!**
  - *This attack is also applicable to new security features!*

Finding new flaws: summary

Test	Design ok	Impl. ok	Risk
Images scan = Image chip check	?	? / ✗	Illegally entering / leaving a country using low-tech scan and cloned chip
Incorrect hash	✓ *	✗	Identity theft / identity creation
Incorrect signing	✓ *	✗	Identity theft / identity creation
AA not required	✗ **	✗	Cloning cannot be prevented (use the weakest link)
AA present, check not supported	✓	✗	Cloning cannot be prevented (use the weakest link)
Index manipulation	✗	✗	Cloning cannot be prevented (use the weakest link)

* *"If both verifications in step 3 and 4 are correct, then this ensures that the contents of SOD can be trusted and SHOULD be used in the inspection process."*

** *"When a MRTD with the OPTIONAL Data Group 15 is offered to the inspection system, the Active Authentication mechanism **MAY** be performed…"*

Root causes

- Design (ICAO standard):
  - *Some key security features are optional: if one party doesn't use a feature the security level of the entire system (globally!) depends on compensating measures*
  - PA does not protect against index manipulation
- Tested implementations:
  - *Do not follow the ICAO standard!*
  - Every country is reinventing the wheel
    - Reinventing applet (fingerprinting nationalities)
    - Reinventing reader bugs (Elvis lives!)
    - Reintroducing hardware problems (DPA attacks etc.)

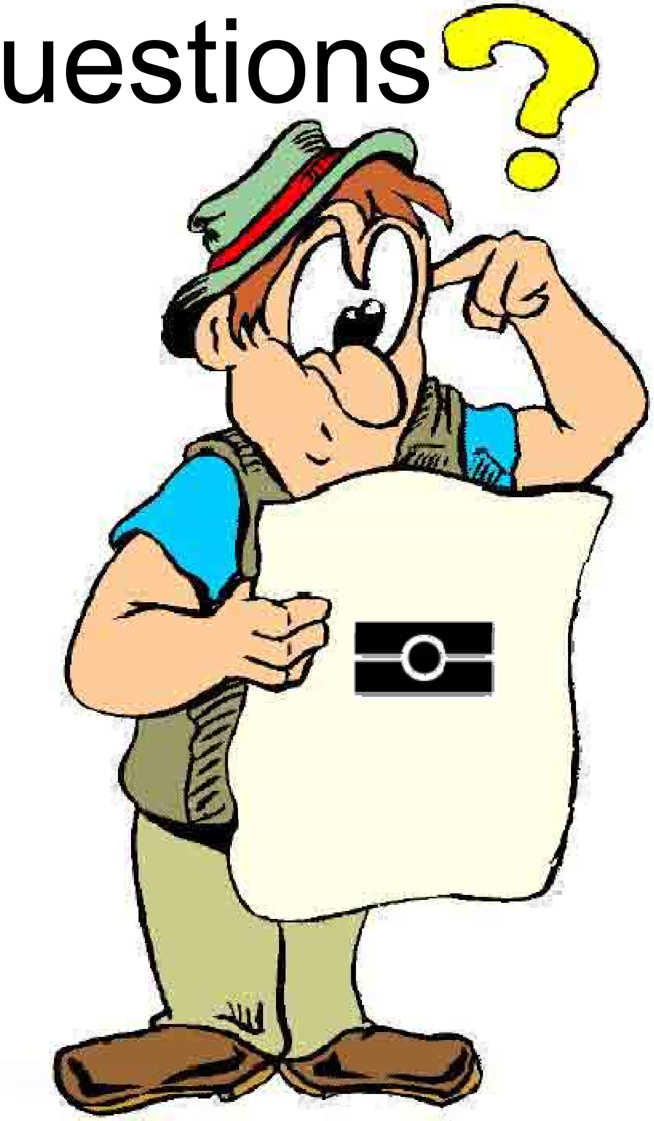
Solutions

- Design (ICAO standard):
  - Require all security features by default
  - Protect the integrity of *all* files
- Implementation:
  - Enable all security features by default
  - Use automated border control for chips with *all* security features enabled only
- Global coordination (e.g. United Nations):
  - Provide standard implementation for ePassport applets and readers
    - The more (black box) implementations, the higher the risk of a serious problem
    - *Open standards and implementations, no security by obscurity!*
  - Provide countries with a list of authorized hardware and hardware lifetimes
    - Think about the Mifare Classic chip family
    - History might repeat itself with ePassports: e.g. German ePassports are valid for 10 years. In 10 years the hardware is most probably outdated (DPA attacks etc.)
  - Provide countries with a trusted PKI environment
    - E.g. automated KPuDS & CRL distribution *before* enrolling eApplications

The future(?|!)

- More biometrics will be added:
  - June 2009: EU adds fingerprints
  - Later: Iris? DNA? Footprints?
- If implemented correctly (…), the system heavily relies on PKI
  - Let’s take a job at customs!
  - Let’s check their network security!
  - In my professional ‘ethical hacker’ career we’ve got a 100% hit rate on p0wning networks
  - I guess unethical hackers got a similar hit rate…
- In the end it’s just another software product
  - Same bugs, same exploits. Exploit the terminals to hop on to the backend
  - E.g. GRT uses CxImage for JPGs, spl0it writers, please contact me…
- Happy traveling :)

Questions



Thank you!