ICAO SECURITY & FACILITATION

E-Passport Validation
A Practical Experience

R Rajeshkumar

International Organization for Standardization (ISO)

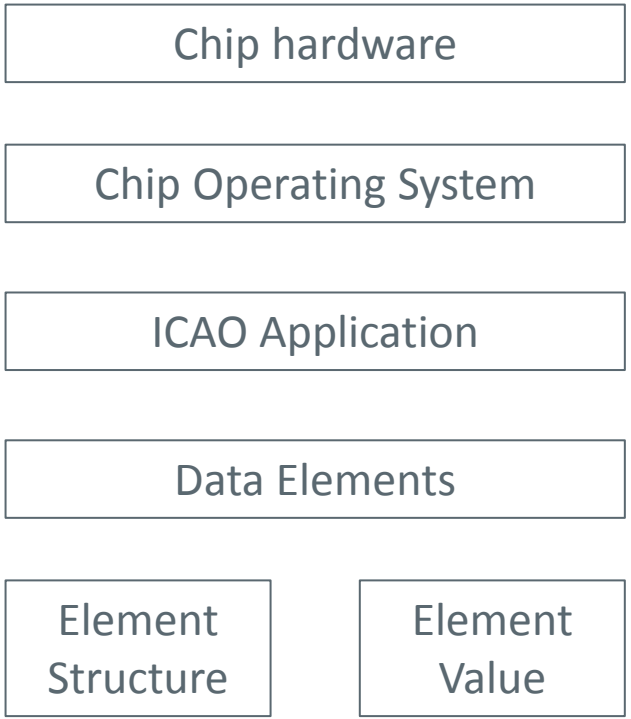
Montreal/October 24, 2018

Background

- Validating e-Passports at SG border since 2006
- Reading e-Passports from 135 countries and three organizations
- Found 23 defects from 55 countries
- Defect can lead to verification failure – depends on crypto toolkit

What is a defect?

- Chip Hardware is very stable
- Chip OS is standard – some strange behaviors , but readers know how to handle it
- ICAO application – No issues till now
- Data element (Elementary Files) all good
- Structure and Value have issues



Current status of defects

- In 2012, 34% of documents at border had defects
- States have started correcting – latest count is 16%
- Some problems may manifest in future – for example, the defect in the encoding of signature value

ICBWG

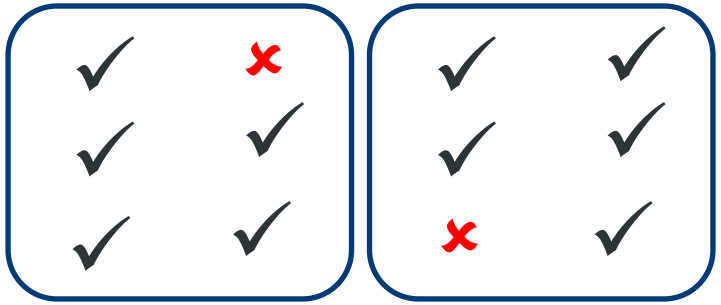
- Since 2016, Non-compliance sub group expanded scope to include chip encoding defects
- Initial focus on 6 defects – 9 state letters have been sent out and 7 have responded that they have fixed the defect
- However, defective documents are still in circulation

Handling Defects

- Currently, there are three known implementations for handling defects

Do Nothing

- In case of defect, the verification fails
- Throw the error to the officer and let them decide
- Most common mechanism in ABCs



Desensitizes officer to possible frauds

Defect Profiling

- In case of defect, the verification fails
- Check against a list of countries and known defects
- If the same defect and other parameters are okay (for example, AA passed and all hashes match), consider a valid document

Fraudster who understands defect profiling only needs to create a defective document to evade detection

Defect Handling

- Try to ensure that verification succeeds in spite of defect
- Requires that every detected defect is analyzed and a workaround is found
- Handling by defect and not by country

Defect Handling – example 1

- RFC 3852 defines Digest Algorithm and Signature algorithm.
- The digest algorithm is used to hash the contents of the eContent (DG Hashes), which is then used as the value in MessageDigest field in Signed Attributes.
- The signed attributes are then hashed using the same digest algorithm and then signed using the signature algorithm.
- One country uses SHA512 to hash the eContent and then uses SHA256 to hash the signed attributes.
- All crypto toolkits fail to verify this SOD – 78% of all E-Passports seen from this country

Defect Handling – solution

- Implement verification as a low level two step process
- Read the DigestAlgorithmIdentifier and use that algorithm to hash the encapsulated content
- Read SignatureAlgorithmIdentifier- If it contains a digest algorithm, then use that to digest the signed attributes, or else use the previous algorithm to digest the signed attributes

Defect Handling – example 2

- Wrong encoding of RSA signature value
  - RSA signature is encoded as OctetString with length of string equal to Modulus value
  - Assumed to be positive integer. Hence do not need to add 0x00 in front to make the value positive in two's complement encoding
  - 0x00 added in front of Signature value making the signature value longer than modulus

Defect Handling – solution

- Take the signature value and remove any leading zeroes that may be encoded as they have no value anyway e.g. 001 is the same as 1
- Compare with modulus after stripping the leading zeroes.

Current status

- In Singapore , we have handled all the possible defects and deployed
  - At the start 1 in 3 e-Passport could not be verified
  - Now it is 1 in 5000
- Improved the verification time to under 200 milliseconds.
- Reading is between 4 and 7.5 seconds. Hence entire process including reading now is under 8 seconds

Defect Handling – Impact

- Border Control Officers – Loss of faith in chip
- Gives rise to the opinion – PKI is broken and chip data can be falsified
- Attack based on defects is a vulnerability at the border

Contact Details

Name:
R Rajeshkumar

Email:
R.Rajeshkumar@Auctorizium.com