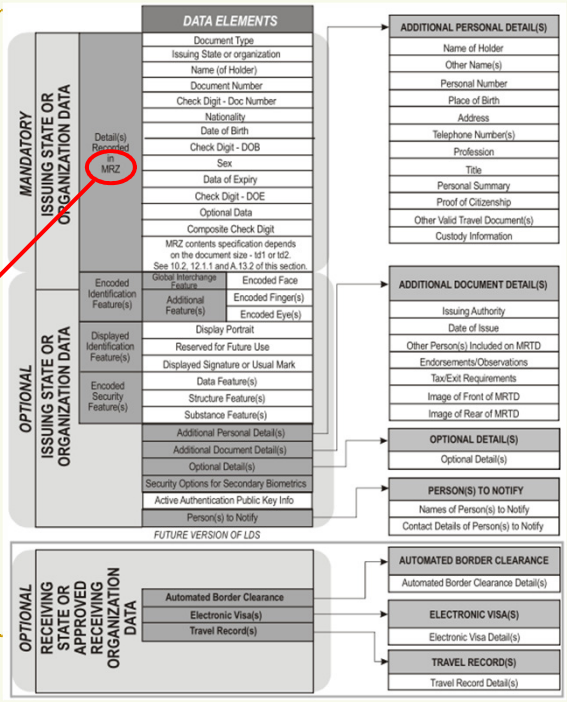
E-Passports

E-Passport: Overview

- Passport with contactless smartcard
  - RFID interface
  - Passive tag
- Smartcard
  - Contains data about the passport holder
    - Name, date of birth, passport number, etc.
    - Biometrics (picture, finger prints, etc.)
- Chip content based on a ICAO standard
  - Machine Readable Travel Documents offering ICC read-only access

ICAO standard: Chip content

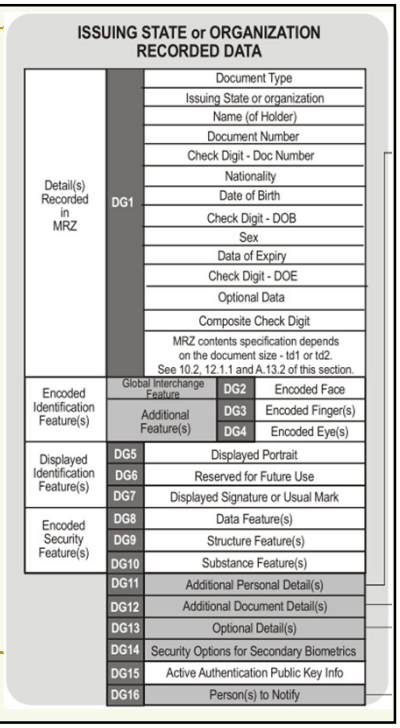
- Standardized Logical Data Structure (LDS)



© André Zúquete

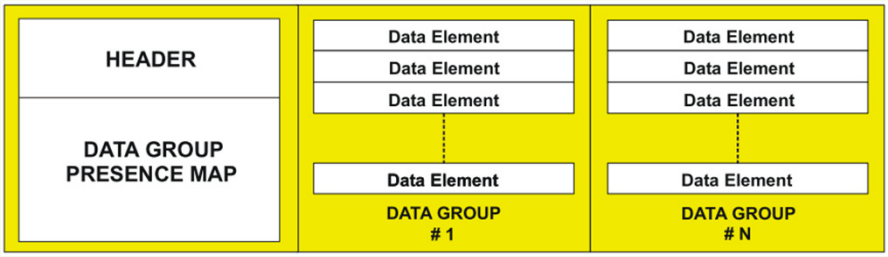
ICAO standard: LDS

- Data Elements packed in Data Groups
  - Each DG stored in a separate Elementary File (EF)
  - EF is a common element of a smartcard file system
- Special EFs
  - EF.COM:
    - Version info, tag list
  - EF.SOD:
    - Security Object Data
    - Safeguarding integrity value
    - Signed hash of all present DGs



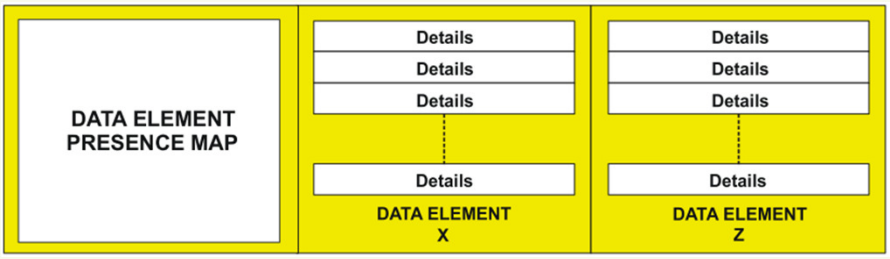
© André Zúquete

EF.COM



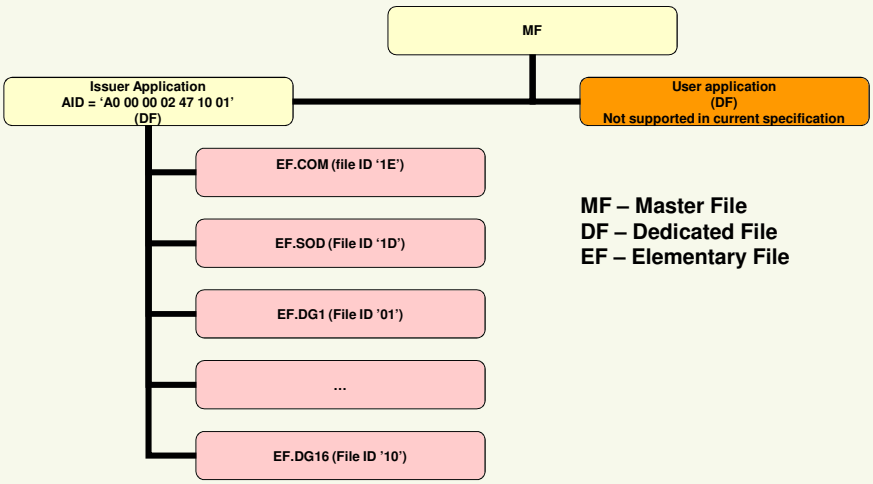
- Header
  - Application IDentifier (AID)
  - LDS version
  - Unicode version
- Data Group presence map
  - List of present/not present tags for Data Groups

Data Group



- Data Element presence map
  - List of present/not present tags for Data Elements

E-Passport file structure: Follows ISO/IEC 7816-4 rules



MF – Master File
DF – Dedicated File
EF – Elementary File

E-Passport file manipulation commands: Follows ISO/IEC 7816-4 rules

- SELECT (file)
  - Chooses a working DF/EF
- READ BINARY
  - Reads contents from a selected EF

E-Passport-reader authentication (1/2)

- Passive authentication (PA)
  - Mandatory
  - EF.SOD authenticates contents of DGs
    - Signed hash of present DG[1-15]
    - Public key is also part of the EF.SOD
- Basic Access Authentication (BAC)
  - Optional
  - Confidential data can only be read after reader authentication
    - Authentication password = passport number + date of birth + date of expiry
    - Data encryption with 3DES
    - Message integrity with MACs

E-Passport-reader authentication (2/2)

- Active Authentication (AA)
  - Optional
  - Prevent cloning and copying
  - EF.DG15 contains a public key
    - Corresponding private key in inaccessible memory
    - Chip proves authenticity by signing a reader challenge with the private key

ISO/IEC 11770-2:
Key establishment mechanism 6

Get challenge →

← r_T

Generate random 64-bit r_T

Generate random 64-bit r_R
Generate random 64-bit k_R
$S_R = r_R \| r_T \| k_R$
$C_R = E_{Kenc}(S_R)$
$M_R = MAC_{Kmac}(C_R)$

→ $C_R \| M_R$

Generate random 64-bit k_T
$S_T = r_T \| r_R \| k_T$
$C_T = E_{Kenc}(S_T)$
$M_T = MAC_{Kmac}(C_T)$

E: 3DES CBC, IV=0
MAC: ISO/IEC 9797, alg. 3

← $C_T \| M_T$

$SK_{enc} = SHA\text{-}1(k_R \| k_T \| 1)$
$SK_{mac} = SHA\text{-}1(k_R \| k_T \| 2)$

ISO/IEC 14443-3 details:
Type A initialization and anticolision

- The use of random UIDs is RECOMMENDED, but States MAY choose to apply unique UIDs