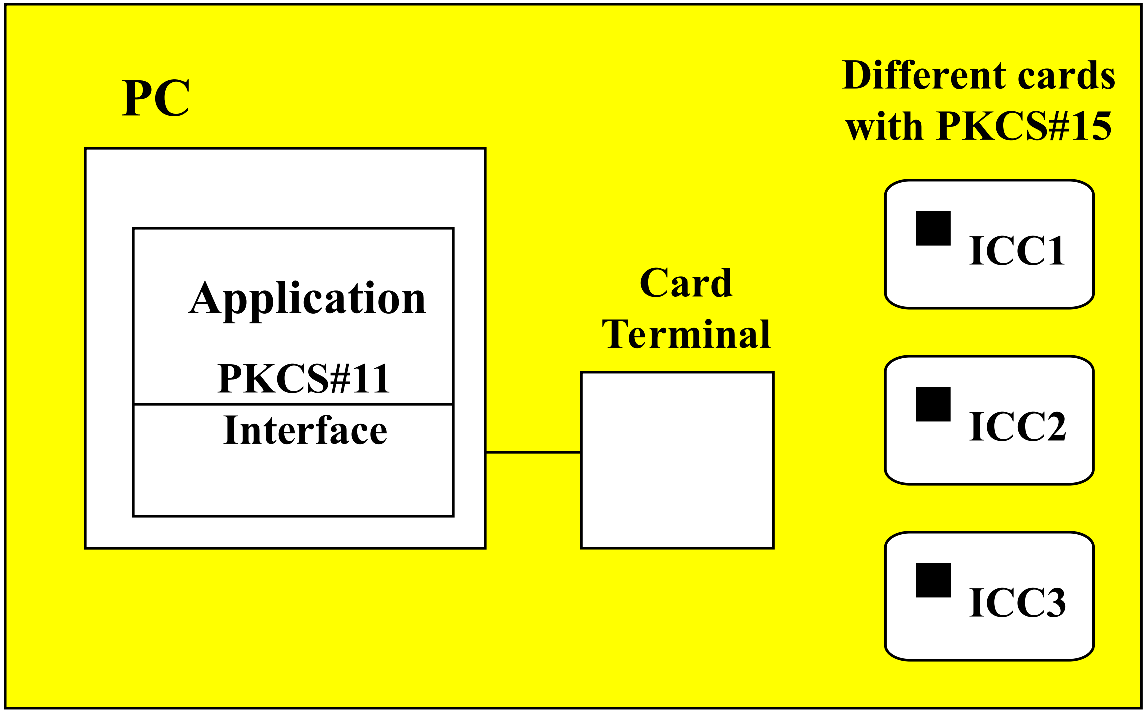
German Digital Signature Card and Office Identity Card and PKCS #15

Bruno Struif
GMD
German National Research Center for Information Technology
Darmstadt

General Configuration



Smartcards
providing the
same service,
but possibly
in a different way.

- If a PC application knows to deal with a card application, no directory files are necessary
- If a PC application does not know how to deal with a card application, it needs information

Is PKCS#15 powerful enough?

Some challenges:

- cards may have a hash function or not

- cards may support different signature algorithms

- cards may support a different set of Digital Signature Input formats

- a card may be configurated in such a way that it allows

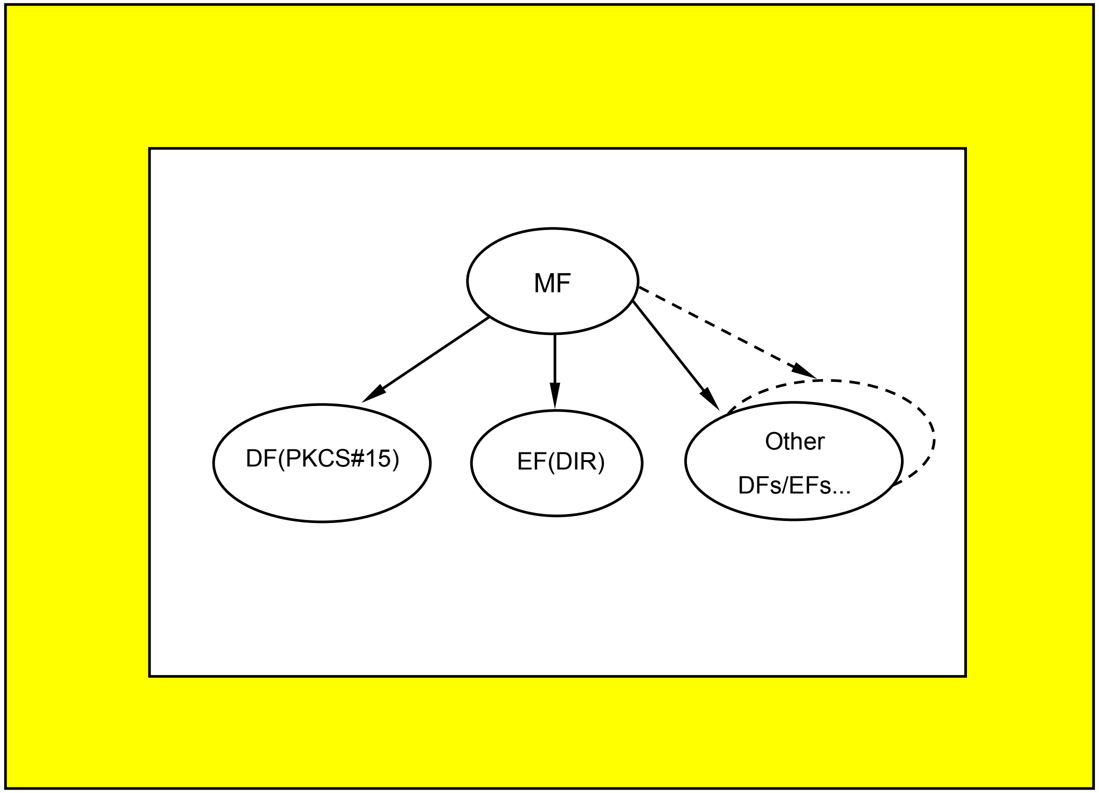
- either after PIN presentation an unlimited number of DS

- or requires PIN presentation before each DS

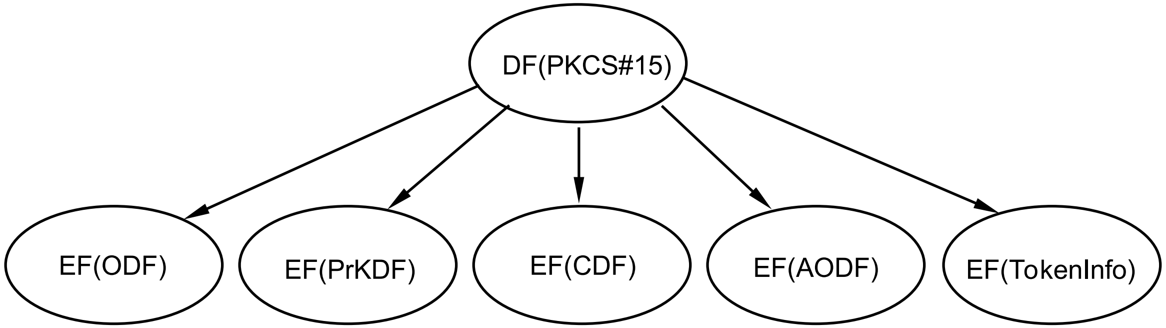
- a card may support ETSI PIN management commands instead of ISO-commands

- a card may support a proprietary command for a certain security service

Card File Structure (1)

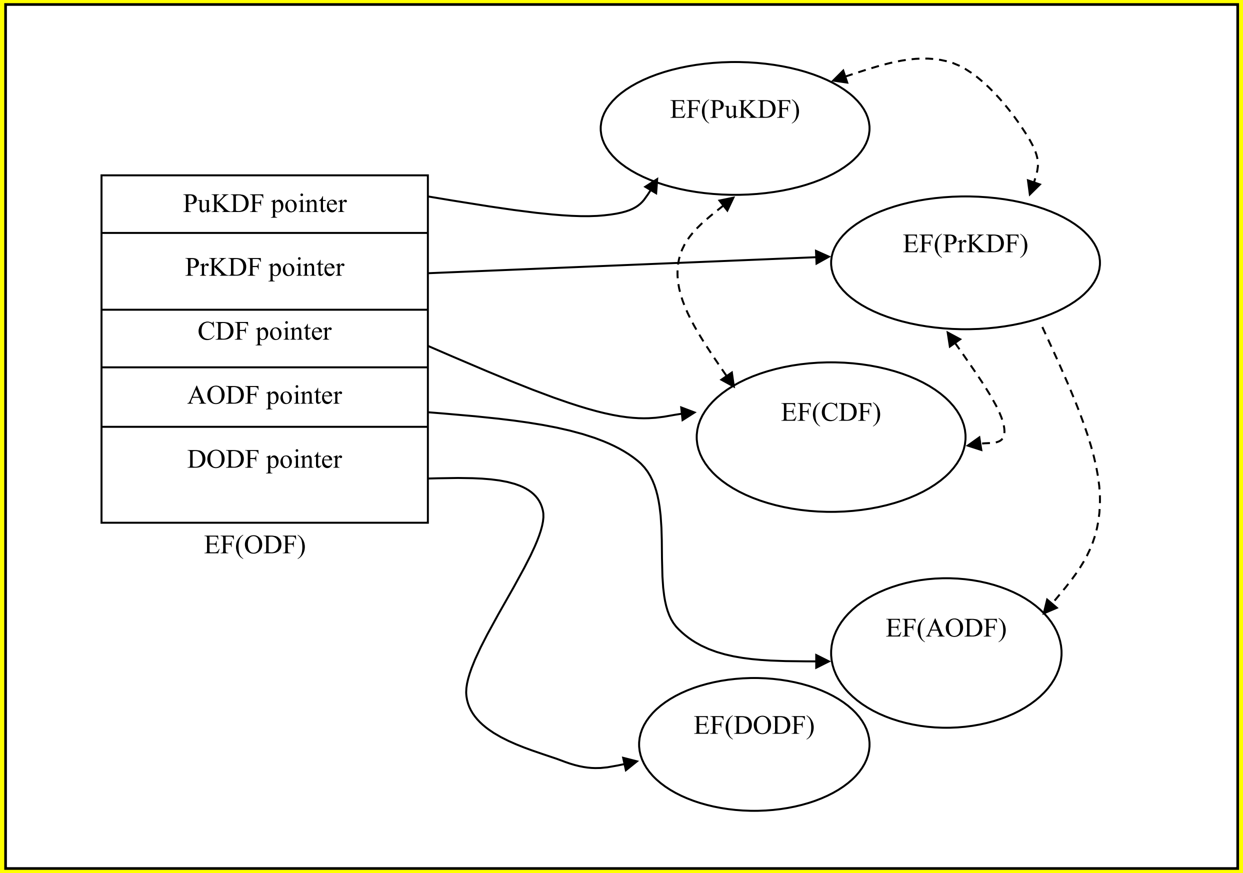


Card File Structure (2)

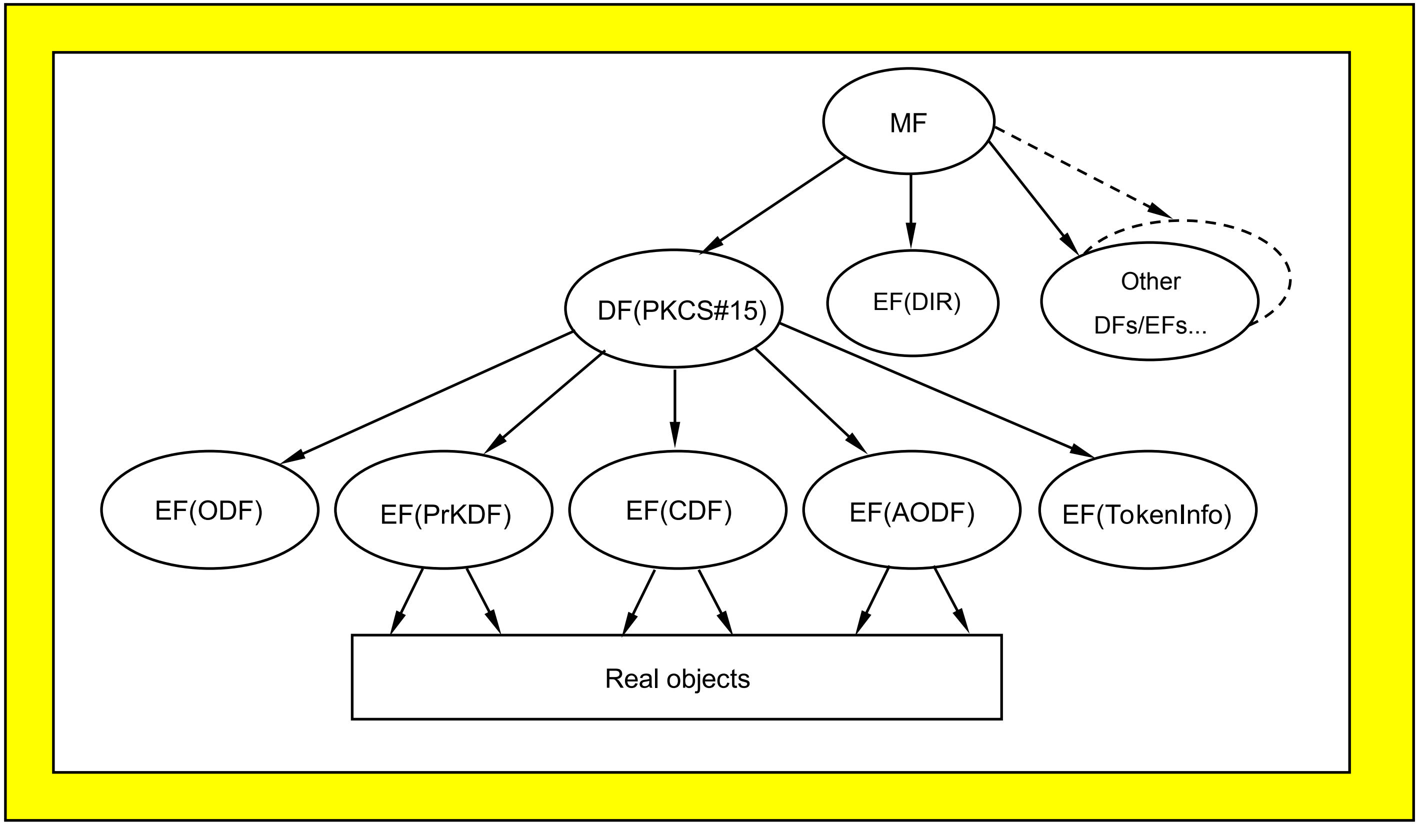


EF(ODF) - Object Directoty File - points to
- EF (PrKDFs) - Private Key Directory Files
- EF (PuKDFs) - Public Key Directory Files
- EF (SKDFs) - Secret Key Directory Files
- EF (CDFs) - Certificate Directory Files
- EF (DODFs) - Data Object Directory Files

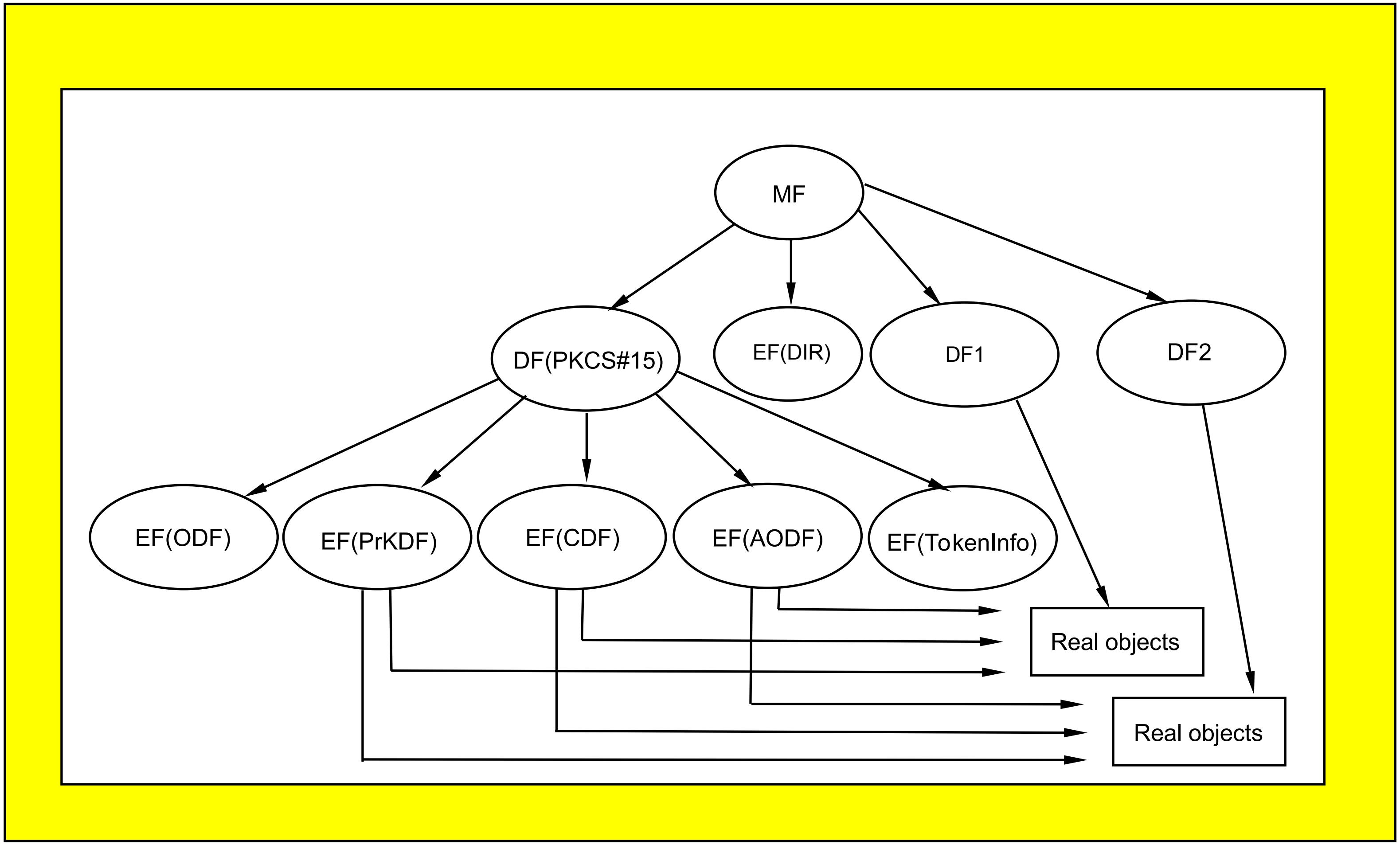
Cross-References



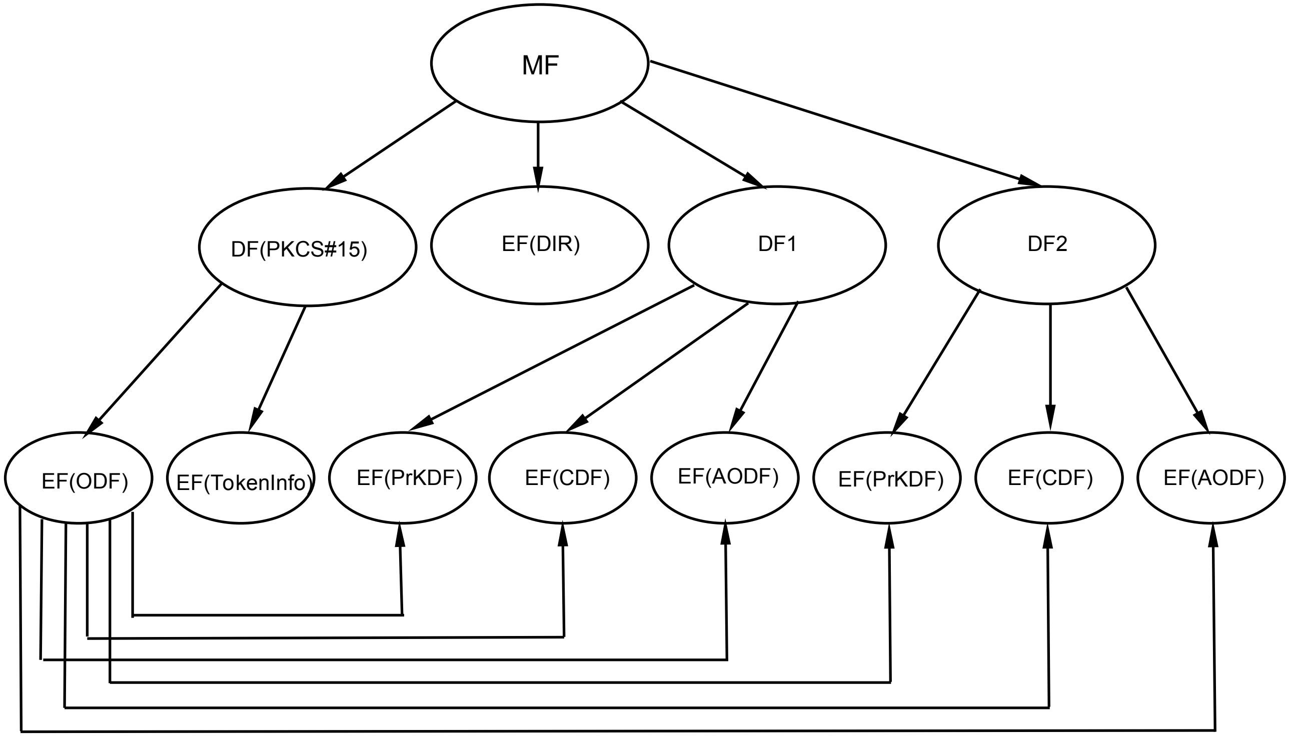
Card File Structure (3)



Card File Structure (4)



Card File Structure (5)



User Authentication

- PKCS15 describes PINs and passwords, but no biometric user authentication
- The German Digital Signature law allows biometric user authentication
- It is technically already feasible to implement biometric feature matching algorithms in cards
- ISO/IEC will add an amendment to 7816-4 with respect to biometric user authentication

VERIFY Command

VERIFY

CLA	As defined in ISO/IEC 7816-4 and -8
INS	´20´ = VERIFY
P1	´00´
P2	´81´ = PIN/PW reference ´91´ = Biometrical data reference
Lc	´xx´ = Length of subsequent data field
Data field	If P2 = ´81´: PIN or PW (min 6, max 8 ASCII characters) If P2 = ´91´: Biometrical verification data
Le	Empty

(DIN.SIG-Version 1.0, Table 11)

- If a digital signature is made on a private PC, then the PIN is presented as plain value

- If a digital signature is made on a public customer service terminal, then the PIN shall be presented as cryptogram followed by a cryptographic checksum

Proposal for integration of bio objects (1)

```
PKCS15Authentication  ::= CHOICE {
pin    PKCS15AuthenticationObject { PKCSPinAttributes },
bio   PKCS15AuthenticationObject { PKCSBioAttributes },
                                  }

PKCS15BioAttributes  ::= SEQUENCE {
        bioFlags           PKCS15BioFlags,
        bioSubject         PKCS15BioSubject,
        bioType            PKCS15BioType,
        bioReference       [0] PKCSReference DEFAULT 0,
        lastBioChange      GeneralizedTime  OPTIONAL,
        path               PKCS15Path OPTIONAL,
        ...  -- For future extensions }
```

Proposal for integration of bio objects (2)

```
PKCS15BioFlags  ::= BIT STRING {
        reserved          (0),
        local             (1),
        change-disabled   (2),
        unblock-disabled  (3),
        initialized       (4),
        reserved          (5),
        reserved          (6),
        reserved          (7),
        disable-allowed   (8),
        authentic         (9),
        enciphered        (10),
        }
```

Proposal for integration of bio objects (3)

```
PKCSBioSubject  ::= CHOICE {
        fingerPrint            [0] FingerPrint,
        voicePrint             [1] VoicePrint,
        irisPrint              [2] IrisPrint,
        facePrint              [3] FacePrint,
        retinaPrint            [4] RetinaPrint,
        handGeometry           [5] HandGeometry,
        writeDynamics          [6] WriteDynamics,
        keystrokeDynamics      [7] KeystrokeDynamics,
        lipDynamics            [8] LipDynamics,
        ... -- For future extensions
        }
```

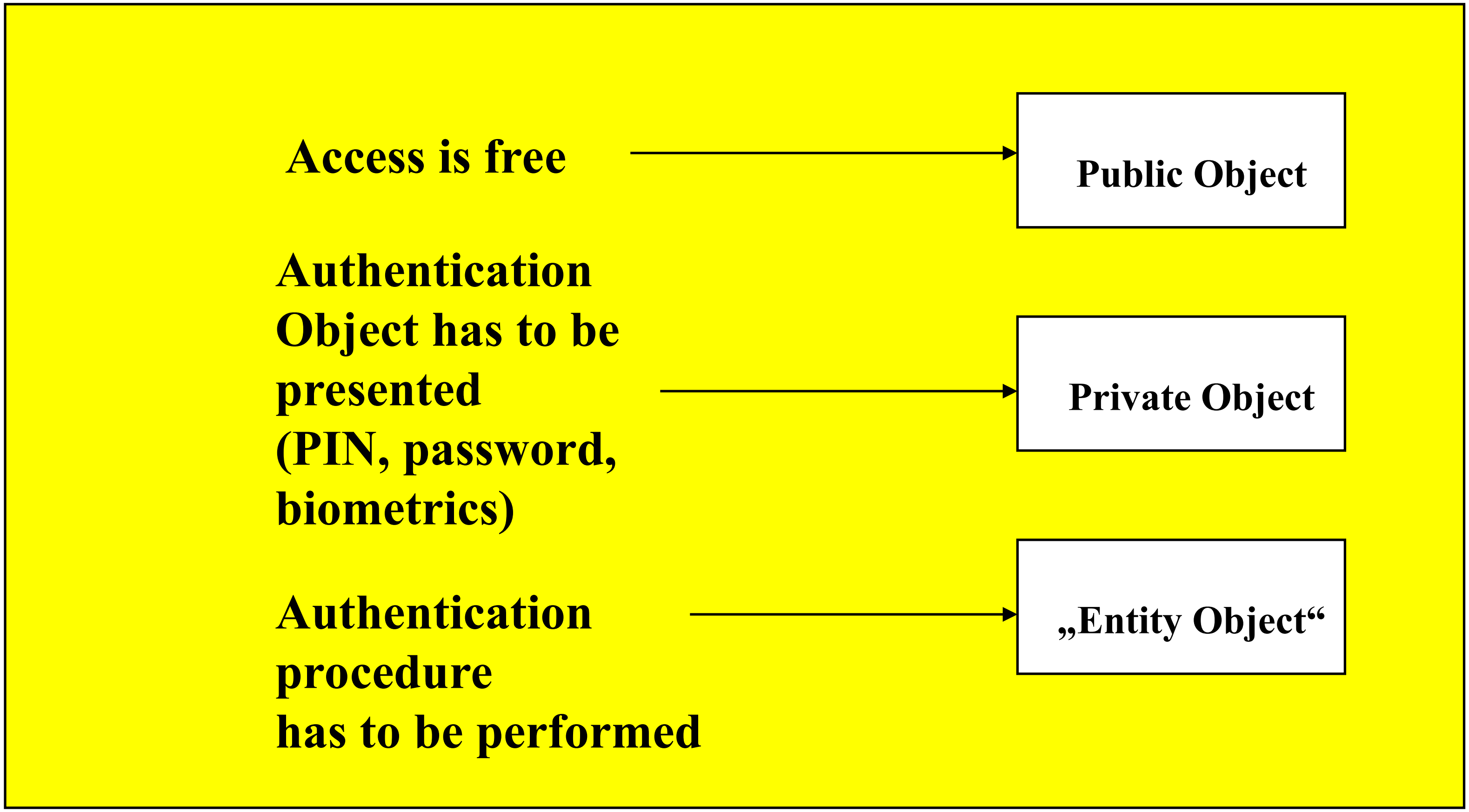
Proposal for integration of bio objects (4)

```
FingerPrint ::=  SEQUENCE {
          handID      HandID,
          fingerID    FingerID
          }

HandID ::= ENUMERATED {righthand (0), lefthand (1) }

FingerID ::= ENUMRATED { thumb(0), pointer finger (1),
middle finger (2), ring finger (3), little finger (4) }
```

Access to objects



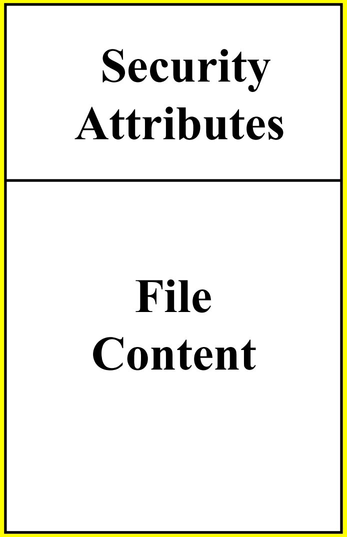
Access is free → Public Object

Authentication Object has to be presented (PIN, password, biometrics) → Private Object

Authentication procedure has to be performed → „Entity Object“

Management of Access Rights

Elementary File



Security Attributes

File Content

Example:

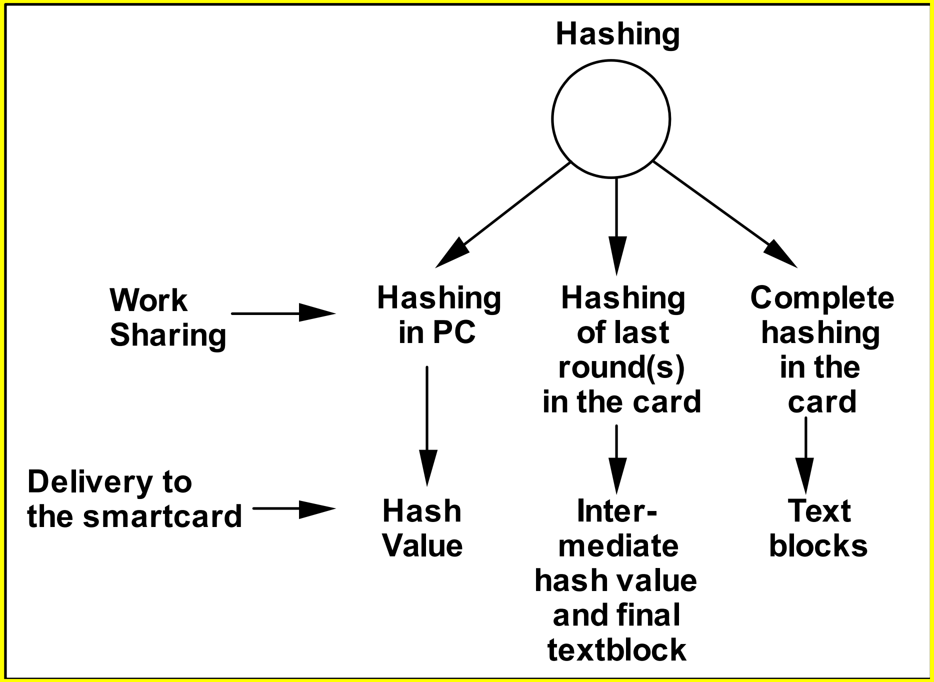
AM = Read
SC = EXT AUTH (asym) with
CHA = ´x.01´ or ´x.02´ and User AUTH

AM = Update
SC = EXT AUTH (asym) with
CHA = ´x.01´ and SM

X = Prefix denoting the AID or the entity assigning the role ID

AM = Access Mode
SC = Security Conditions
CHA = Cert. Holder Authorisation (Prefix, Role ID)
SM = Secure Messaging

Hashing


Hashing
Work Sharing
Hashing in PC
Hashing of last round(s) in the card
Complete hashing in the card
Delivery to the smartcard
Hash Value
Inter-mediate hash value and final textblock
Text blocks

Certificates

- PKCS15 distinguishes
  - x509Certificates
  - x509Attribute Certificates
  - spkiCertificates
  - pgpCertificates
  - wtlsCertificates
  - x9-68Certificates

  but no cvCertificates!!

Card Verifiable Certificates

CPI	CAR	CHR	CHA	OID	PK	SIG.CA

- CPI = Certificate Profile Identifier

- CAR = Certification Authority Reference (Authority Key Identifier)

- CHR = Certificate Holder Reference (Subject Key Identifier)

- CHA = Certificate Holder Authorisation (Authority || Role Identifier)

- OID = Object Identifier of PK Algorithm

- PK = Public Key of Certificate Holder

- SIG.CA = Signature of Certificate Issuing CA

Security Service Descriptor

- Template tags for all security services
  (e.g. user authentication service, digital signature service,
  entity authentication service, key cipherment service)

- DO Instruction set mapping ISM (regular command)
- DO Command to perform (if command is different form that in ISM)

- DO Object Id of the algorithm
- DO Algorithm reference (as used by the card)

- DO Key reference (as used by the card)
- DO Key file id (some cards select the key file containing the key to be used)

- DO Certificate file id (if present then the file contains the certificate)
- DO Certificate reference (used e.g. if the certificate is not stored in the card)
- DO Certificate qualifier (e.g. X.509 certificate, ICC certificate)

- DO PIN usage policy (present if the security service is PIN protected)

Security Service Descriptors

- Indication of supported algorithms, DSI schemas, hash functions
- Indication of user authentication method
- Indication where to find certificates
- Indication of implementation variants
- Support of migration

SSD construction (1)

- For each security service provided by the card exists one or more SSD templates
- Inside an SSS template is one DO mandatory: the DO „command to perform“

- Use e.g. for VERIFY:
  - command class is present
  - PIN reference is present
  - PIN length is present possibly with padding
  - presentation form is present: plain value or with SM

- Use e.g. for CHANGE RD:
  - command class is present
  - PIN reference is present
  - usage option is present, e.g. old PIN required/not required in the command
  - PIN length is present possibly with padding
  - presentation form is present: plain value or with SM

SSD construction (2)

- Use e.g. for digital signature function:
  - the MANAGE SECURITY ENVIRONMENT to perform is presented
  - the HASH command, if needed, is presented
  - The PERFORM SECURITY OPERATION command is presented for the digital signature compuation
- Different methods for Dig. Sig. Input constructions can be denoted by the DO OID or the DO AlgID E.g. PKCS#1 or ISO 9796-2 rnd
- The FIDs of related certificate files are given

Working with PKCS#15 (1)

- The usage of PKCS#15 requires
  - selection of DF(PKCS15)
  - selection of EF(ODF) for getting the pointer information
  - reading EF(ODF)
  - selection of EF(AODF) for getting the PIN information
  - reading EF(AODF)
  - selection of EF (PrKDF) for getting the signature key information
  - reading EF(PrKDF)
  - selection of EF(CDF) for getting the certificate information
  - reading EF(CDF)
  - selection of EF(PuKDF) for getting the root CA PuK information
  - reading EF(PuKDF)

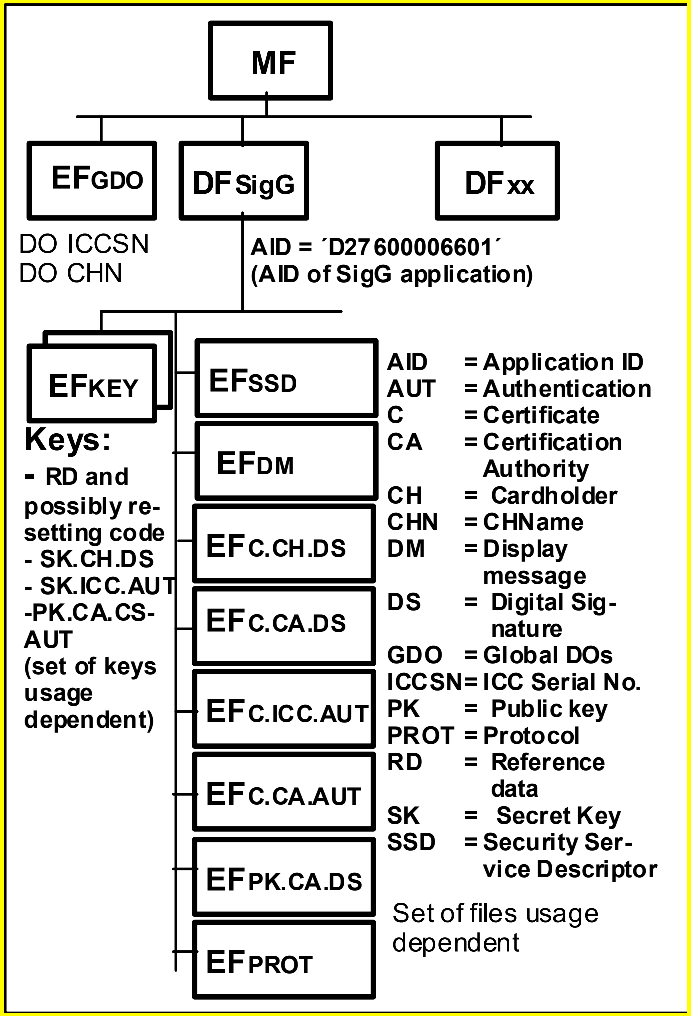
Working with PKCS#15 (2)

- To do this all is not very efficient. Therefor:
  - Read the information once from the card and
    store it under a card reference, e.g. the ICC Serial Number ICCSN

  or
  - keep the information outside the card and store in the card the
    card profile identifier pointing to the outside information

- Open problem: there is no indication whether the PKCS15 files are
  - reocrd-oriented or
  - transparent.

File Structure of DF.SIG



Keys:
- RD and possibly re-setting code
- SK.CH.DS
- SK.ICC.AUT
-PK.CA.CS-AUT
(set of keys usage dependent)

AID	= Application ID
AUT	= Authentication
C	= Certificate
CA	= Certification Authority
CH	= Cardholder
CHN	= CHName
DM	= Display message
DS	= Digital Signature
GDO	= Global DOs
ICCSN	= ICC Serial No.
PK	= Public key
PROT	= Protocol
RD	= Reference data
SK	= Secret Key
SSD	= Security Service Descriptor

Set of files usage dependent