PKCS #15 Introduction

Ken Asnes

RSA Laboratories

kasnes@rsasecurity.com

July 2001

Agenda

- What is PKCS #15?
- How does it work?
- Conclusion

What is PKCS #15?

- It is a specification for organizing cryptographic data onto an authentication object.
  - Smart cards
  - Other token devices (ISO/IEC 7816)
  - Soft PSD’s (eventually)

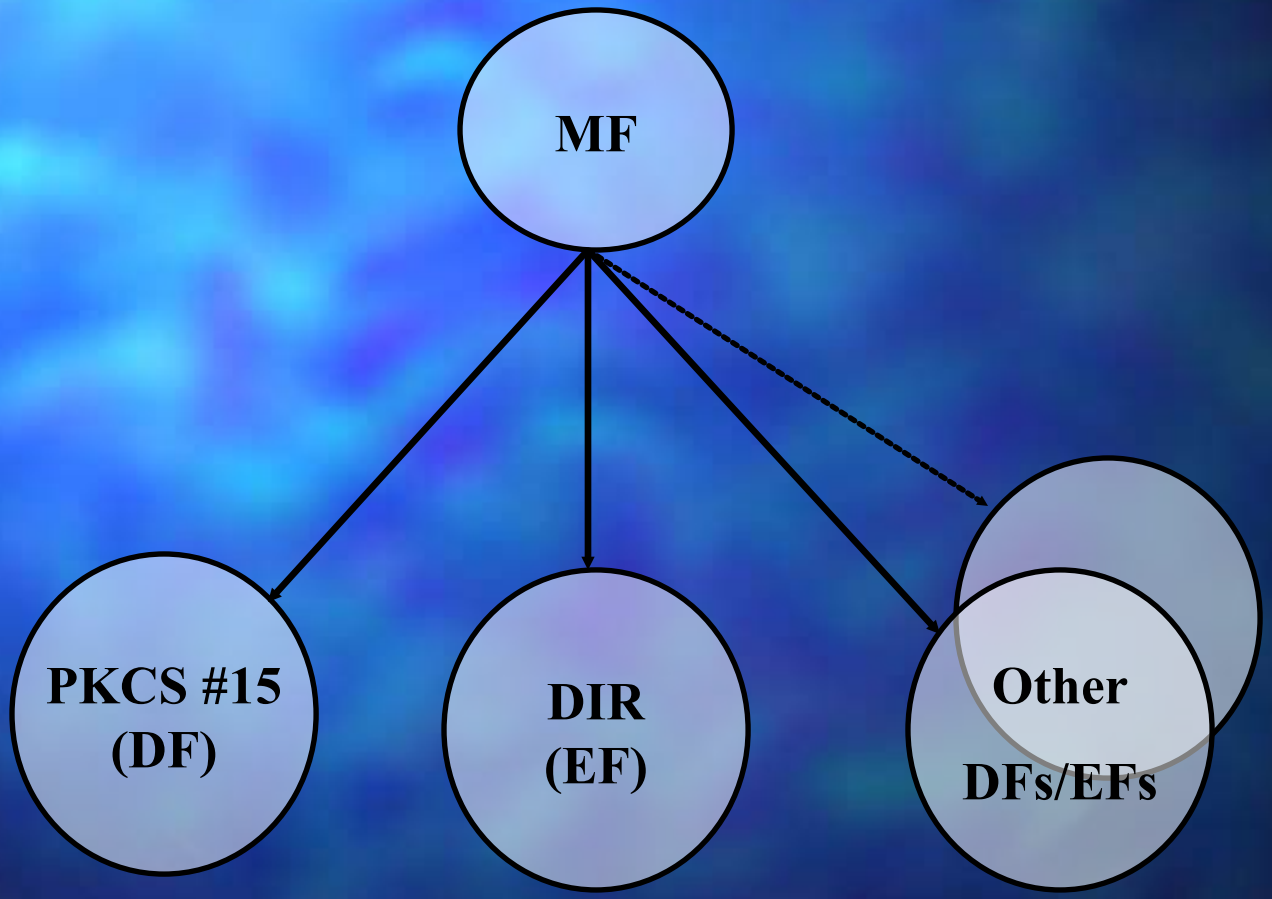
What is PKCS #15? Cont...

- It builds off of PKCS #11
- It allows for Multiple PKCS #15 aware applications to live on the same card
- It will take the place of SSEID1
  - We can still support EIDAB1 and SSEID1 of course

How Does it work?

(Are you SURE you want to know?)

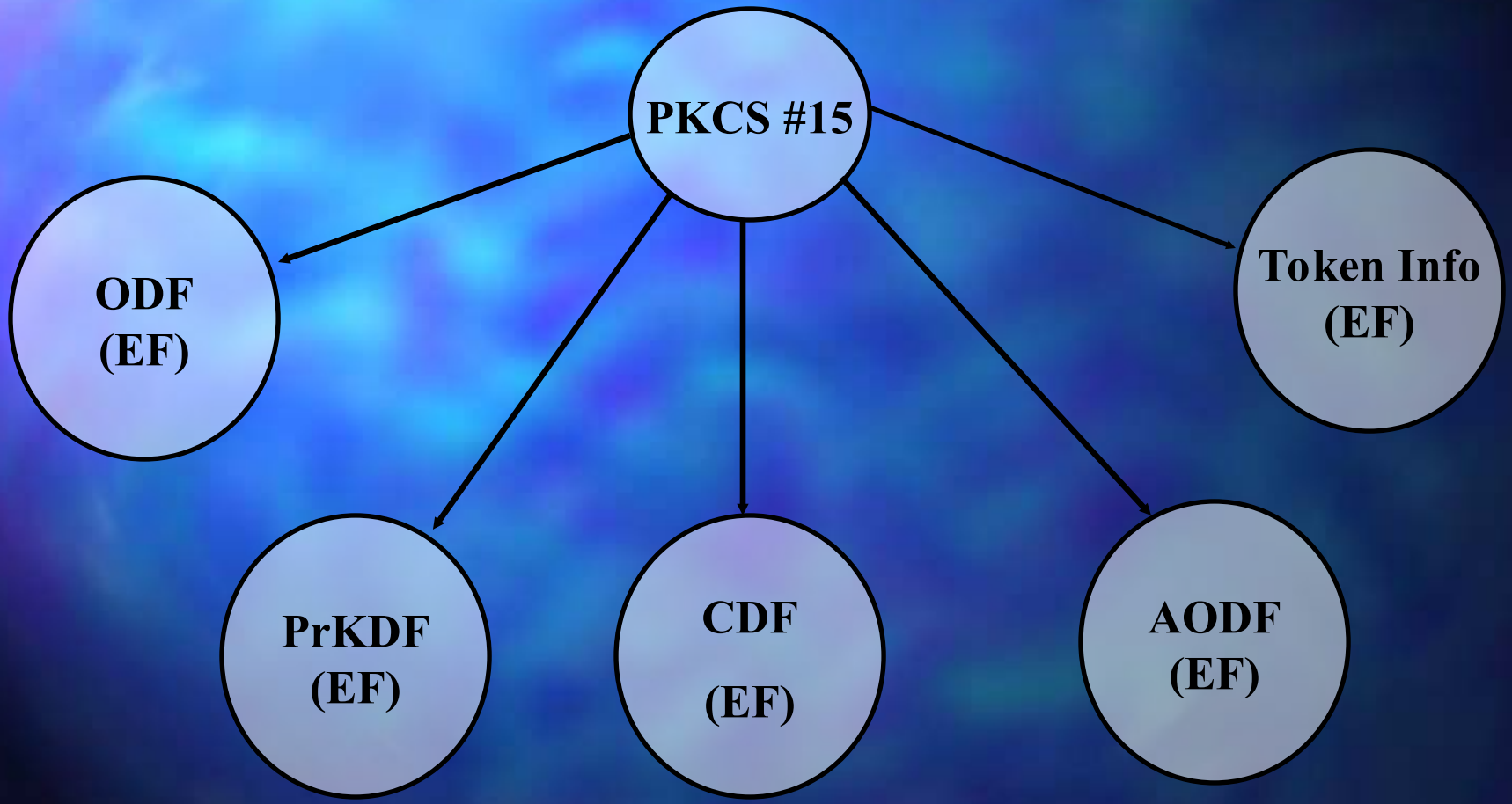
Directory Structure



DIR

- Optional File
  - Mandatory if direct application file selection is not supported OR if Multiple PKCS #15 Applications reside on the card
- Contains the following DOs:
  - OID (mandatory)
    - Unique ID for the implementation
  - ODF Path (mandatory)
    - Path to the Object Directory File
  - Token Info Path (optional)
    - Points to the TokenInfo file.
    - The Token Info file contains info that may be shared between PKSC #15 applications.
  - Unused Path (optional)
    - Points to a file that points to unused space

PKCS #15
Application Directory



ODF
Object Directory File

- Mandatory File
- Contains pointers to other EF's
  - PrKDF
  - PuKDF
  - SKDF
  - CDF
  - DODF
  - AODF

PrKDF

Private Key Directory File

- Optional File
- Holds the following:
  - Private keys or references to them
    - Keys May reside anywhere on the card
  - Key attributes
    - Labels
    - Intended Usage
    - Identifiers
  - Cross references to pointers for Authentication Objects

PuKDF
Public Key Directory File

- Optional File
- Holds the following:
  - Public keys or references to them
    - Keys May reside anywhere on the card
  - Key attributes
- If a corresponding private key exists they must share the same Identifier

SKDF

Secret Key Directory File

- Optional File
- Holds the following:
  - Symmetrical keys or references to them
    - Keys May reside anywhere on the card
  - Key attributes
  - Cross references to authentication objects

CDF
Certificate Directory File

- Optional File
- Holds the following:
  - Certificates or references to them
    - May reside anywhere on the card
  - Certificate attributes
  - Cross references to authentication objects
- If a corresponding private key exists they must share the same Identifier

DODF

Data Object Directory File

- Optional File
- For any data object other than keys or certificates
- Holds the following:
  - Data objects or references to them
  - Data object attributes
  - Cross references to authentication objects

AODF
Authentication Object Directory File

- Optional File
- For any authentication object, such as PIN's, that restrict access to other PKCS #15 objects (eg Keys)
- Holds the following:
  - Authentication objects or references to them
  - Authentication object attributes
    - What object the AO is protecting
    - Others will vary according to the auth object type

Token Info

- Mandatory File
- For Generic information about the token
- Holds the following:
  - Token Serial Number
  - Supported file types
  - Algorithms implemented
- Info can be shared between PKCS #15 Applications

Unused Space

- Optional File
- Keeps track of unused space in other EF's
- Holds the following:
  - A Path field that points to an unused area
    - Index/Offset and length must be present
  - An authID component that signals the unused space is protected by an Authentication object
- Can be shared between PKCS #15 Applications

File Identifiers

File	DF	File Identifier (relative to nearest DF)
MF	X	0x3F00 (ISO/IEC 7816-4)
DIR		0x2F00 (ISO/IEC 7816-4)
PKCS15	X	Decided by application issuer (AID is RID || "PKCS-15")
ODF		0x5031 by default (but see also Section 6.4.1)
TokenInfo		0x5032 by default (but see also Section 6.4.1)
UnusedSpace		0x5033 by default (but see also Section 6.4.1)
AODFs		Decided by application issuer
PrKDFs		Decided by application issuer
PuKDFs		Decided by application issuer
SKDFs		Decided by application issuer
CDFs		Decided by application issuer
DODFs		Decided by application issuer
Other EFs		Decided by application issuer
- (Reserved)		0x5034 - 0x5100 (Reserved for future use)

From the PKCS #15 Specification

Application Identifier

- Concatenation of the Registered Identifier (RID) and the Proprietary application Identifier eXtension (PIX) for PKCS #15
- And the Winner is:
  - A0 00 00 00 63 50 4B 43 53 2D 31 35

Adding new Objects

- Must have sufficient privileges
- Unused space file (if used)
  - Points to first free byte
  - New object is written
  - New record is added to ODF
    - APPEND RECORD for Linear Record
    - UPDATE BINARY for transparent object file
  - May need garbage collection

Adding new Objects cont.

- Must create a new EF then Update ODF as required
  - Any time you add/remove a pointer which resides in the ODF
- Locate a new EF at a record with a tag of ‘00’ (denotes free space)
  - ‘00’ is not a valid ASN.1 tag and can be used to mark empty space
  - Consistent with ISO/IEC 7816-4 annex D (Because you needed to know)

Removing Objects

- Must have sufficient privileges
- Either Tag the record to be remove with ‘00’
  - Leave the length bytes to make recalculating available space easier
- OR Rewrite the entire file
  - Still leaving the length bytes

Modifying Objects

- Must have sufficient privileges
- Remove the old file and Create a new one
  - Expensive
- OR Update the record with the new information
  - Cheaper
  - Be careful with space requirements

Strengths and Weaknesses

- Strengths
  - Multiple Applications
  - Multiple keys
  - Multiple certificates
  - Other data may be stored
- Weakness
  - One extra step for Read/Write operations
    - Slower
  - Updates can be complicated

Conclusions

- PKCS #15 has a dynamic structure well suited for interoperability.
  - Supports multiple applications
  - Can store any PKCS #11 or CAPI object
    - Keys, Certificates, Data
  - Allows Multiple Token Types
  - Allows Multiple PIN’s
    - If the token Supports them
    - PKCS #11 can be made to handle multiple PINS