

# Belgian Electronic Identity Card content

## Table of content

1. Scope.....	2
1.1. Terms and definitions.....	2
1.2. Symbols, abbreviated terms and document conventions.....	3
1.2.1. Symbols.....	3
1.2.2. Abbreviated terms.....	3
2. Versions.....	4
2.1. Applet version.....	4
2.2. Card content versions.....	4
2.3. Electrical Personalisation Versions History.....	4
3. Security Objects.....	5
3.1. Convention about PIN and key references.....	5
3.2. PIN.....	5
3.3. Keys and Certificates.....	5
3.3.1. Keys and certificates relationships.....	5
3.3.2. Keys Access Control.....	6
4. Files.....	7
4.1. File structure.....	7
4.2. PKCS#15 files.....	8
4.3. Files identifiers and read permissions.....	9
4.5. Note for non-eID cards.....	10
5. PKCS#15 information detail.....	11
5.1. PKCS#15 application selection.....	11
5.2. MF directory contents.....	11
5.2.1. EF(DIR).....	11
5.3. DF(BELPIC) Application directory contents.....	11
5.3.1. EF(TokenInfo).....	12
5.3.2. EF(ODF).....	13
5.3.3. EF(AODF).....	14
5.3.4. EF(PrKDF).....	15
5.3.5. EF(PuKDF).....	18
5.3.6. EF(CDF).....	19
6. Application information detail.....	23
6.1. TLV format.....	23
6.2. Identity data.....	24
6.2.1. DF(ID).....	24
6.2.2. EF(ID#RN).....	24
6.2.3. EF(SGN#ID).....	24
6.2.4. EF(ID#Address).....	24
6.2.5. EF(SGN#Address).....	24
6.2.6. EF(ID#Photo).....	25
6.2.7. EF(PuK#1 ID).....	25
7. Public and Private Keys detail.....	26
7.1. PrivateKey #1.....	26
7.2. Public Key #1.....	26
7.3. PrivateKey #2.....	26
7.4. Private Key #3.....	26
8. Certificates detail.....	27
8.1. Certificate #2.....	27
8.2. Certificate #3.....	27
8.3. Certificate #4.....	27
8.4. Certificate #6.....	27
8.5. Certificate #8.....	27

# 1. Scope

This standard describes the specifications of the Belgian Electronic Identity Card files and objects. Only the **DF(BeIPIC)** and the **DF(ID)** are covered in this document.

## 1.1. Terms and definitions

For the purposes of this document, the following definitions apply:

<b>authentication object directory file</b>	optional elementary file containing information about authentication objects known to the PKCS#15 application
<b>binary coded decimal</b>	Number representation where a number is expressed as a sequence of decimal digits and then each decimal digit is encoded as a four bit binary number.  Example – Decimal 92 would be encoded as the eight bit sequence 1001 0010.
<b>cardholder</b>	person for whom the card was issued
<b>card issuer</b>	organization or entity that issues smart cards and card applications
<b>certificate directory file</b>	optional elementary file containing information about certificate known to the PKCS#15 application
<b>data object directory file</b>	optional elementary file containing information about data objects known to the PKCS#15 application
<b>dedicated file</b>	file containing file control information, and, optionally, memory available for allocation, and which may be the parent of elementary files and/or other dedicated files
<b>directory (DIR) file</b>	optional elementary file containing a list of applications supported by the card and optional related data elements
<b>elementary file</b>	set of data units or records that share the same file identifier, and which cannot be a parent of another file
<b>file identifier</b>	2-byte binary value used to address a file on a smart card
<b>master file</b>	mandatory unique dedicated file representing the root of the structure  NOTE – The MF typically has the file identifier 3F00
<b>object directory file</b>	elementary file containing information about other directory files in the PKCS #15 application
<b>path</b>	concatenation of file identifiers without delimitation  NOTE – If the path starts with the MF identifier (3F00), it is an absolute path; otherwise it is a relative path. A relative path shall start with the identifier '3FFF' or with the identifier of the current DF.
<b>personal identification number (PIN)</b>	4 to 12 digit number entered by the cardholder to verify that the cardholder is authorized to use a functionality of the card
<b>private key directory file</b>	optional elementary file containing information about private keys known to the PKCS#15 application

<b><i>provider</i></b>	authority who has or who obtained the right to create the MF or a DF in the card
<b><i>public key directory file</i></b>	optional elementary file containing information about public keys known to the PKCS#15 application
<b><i>record</i></b>	string of bytes which can be handled as a whole by the card and referenced by a record number or by a record identifier
<b><i>private key directory file</i></b>	optional elementary file containing information about private keys known to the PKCS#15 application
<b><i>token</i></b>	portable device capable of storing persistent data

## ***1.2. Symbols, abbreviated terms and document conventions***

### ***1.2.1. Symbols***

***DF(x)*** Dedicated file x

***EF(x)*** Elementary file x

### ***1.2.2. Abbreviated terms***

For the purposes of this document, the following abbreviations apply:

<b><i>AID</i></b>	Application Identifier
<b><i>AODF</i></b>	Authentication Object Directory File
<b><i>BCD</i></b>	Binary-Coded Decimal
<b><i>CDF</i></b>	Certificate Directory File
<b><i>DER</i></b>	Distinguished Encoding Rules
<b><i>DF</i></b>	Dedicated File (directory)
<b><i>DODF</i></b>	Data Object Directory File
<b><i>EF</i></b>	Elementary File
<b><i>MF</i></b>	Master File
<b><i>ODF</i></b>	Object Directory File
<b><i>PIN</i></b>	Personal Identification Number
<b><i>PrKDF</i></b>	Private Key Directory File
<b><i>PuKDF</i></b>	Public Key Directory File

## 2. Versions

### 2.1. Applet version

Some objects are hard coded into the applet and are therefore linked to the version of the applet used. These objects are:

- The PINs
- The public and private keys
- The MF, DF(BELPIC) and DF(ID) directories

When applicable, we will refer in this document to “**Applet version x**”. This version can be received with the command “**GetCardData**” that can be sent to the card and that returns a.o. the applet version.

### 2.2. Card content versions

The main version of the card content is located in the file **TokenInfo** (see 5.3.1).

Two versions are available:

- **Electrical personalisation version:** this number increases at every change – even minor – in the personalisation format or personalisation options
- **Electrical personalisation interface version:** this number increases when a change in the personalisation format or personalisation options introduces an incompatibility (e.g. of the file structure, of PIN identifiers,...).

An application can thus use newer cards if the interface version will be the same.

Note that individual files may have an internal version number corresponding to the data in the file. The “**Electrical personalisation interface version**” should be used to check the file structure, the internal file version should be used to check the fields format in the file.

### 2.3. Electrical Personalisation Versions History

Version (Hexa)	Interface Version (Hexa)	Date	Description
00	00		▪ Initial version
01	00	01-01-2004	▪ New ATR: 3B 98 94 40 <b>0A</b> A5 03 01 01 01 AD 13 10 ▪ PKCS#15 files size adaptations ▪ Address file length extended to 117 bytes
02	00	13-12-2004	▪ New ATR: 3B 98 <b>13</b> 40 0A A5 03 01 01 01 AD 13 <b>11</b>
03	00	26-10-2009	▪ New chip
04	00	01-09-2020	▪ New ATR: 3B 7F 96 00 00 80 31 80 65 B0 85 04 01 20 12 0F FF 82 90 00

## 3. Security Objects

### 3.1. Convention about PIN and key references

Some keys and PIN become global in the BelPIC application. ISO 7816 imposes a strict convention for referencing global objects. As these objects are also local, they actually have two references.

ISO 7816 standardises the references as follow:

b <sub>7</sub>	b <sub>6</sub>	b <sub>5</sub>	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	Meaning
0	0	0	0	0	0	0	0	No information is given (RFU)
0	-	-	-	-	-	-	-	Global reference data
1	-	-	-	-	-	-	-	Local reference data
-	X	X	-	-	-	-	-	'00' others are RFU
-	-	-	X	X	X	X	X	Data object number

### 3.2. PIN

	PIN reference (Java Object)	Max. trials before blocked
PIN <sub>Cardholder</sub>	01	3

### 3.3. Keys and Certificates

#### 3.3.1. Keys and certificates relationships

	Private Key (Java Object)	Public Key	X.509 Certificates (Transparent file)
Basic	PrK#1	In EF(PuK#1)	
Authentication	PrK#2	In Cert#2	Cert#2
Non-repudiation	PrK#3	In Cert#3	Cert#3
Citizens CA		In Cert#4	Cert#4
Root		In Cert#6	Cert#6
Government CA		In Cert#4	Cert#4
RRN			Cert#8

Each key or certificate is indicated by means of a reference number (#). Some keys do not have a corresponding private/public key or certificate.

3.3.2. Keys Access Control

Command on key	Reference (hex)	PSO: Compute Digital Signature	Internal Authenticate
PrK#1 (basic)	81	×	ALW
PrK#2 (authentication)	82	CHV(PIN <sub>Cardholder</sub> )	×
PrK#3 (non-repudiation)	83	CHV(PIN <sub>Cardholder</sub> )	×

- ×

Not possible (forbidden by the card Operating System/applet)
- ALW

Always
- CHV(x)

Card Holder Verification with PIN 'x'

## 4. Files

All EF file types are transparent, as defined in ISO/IEC 7816–4, sub-clause 5.1.3.

Files in the EID card is organised into a hierarchical structure according to ISO/IEC 7816–4.

The electronic signature and electronic identification applications are separated in the card by means of two application directories: **DF(BELPIC)** and **DF(ID)**. Other applications DF might be added later.

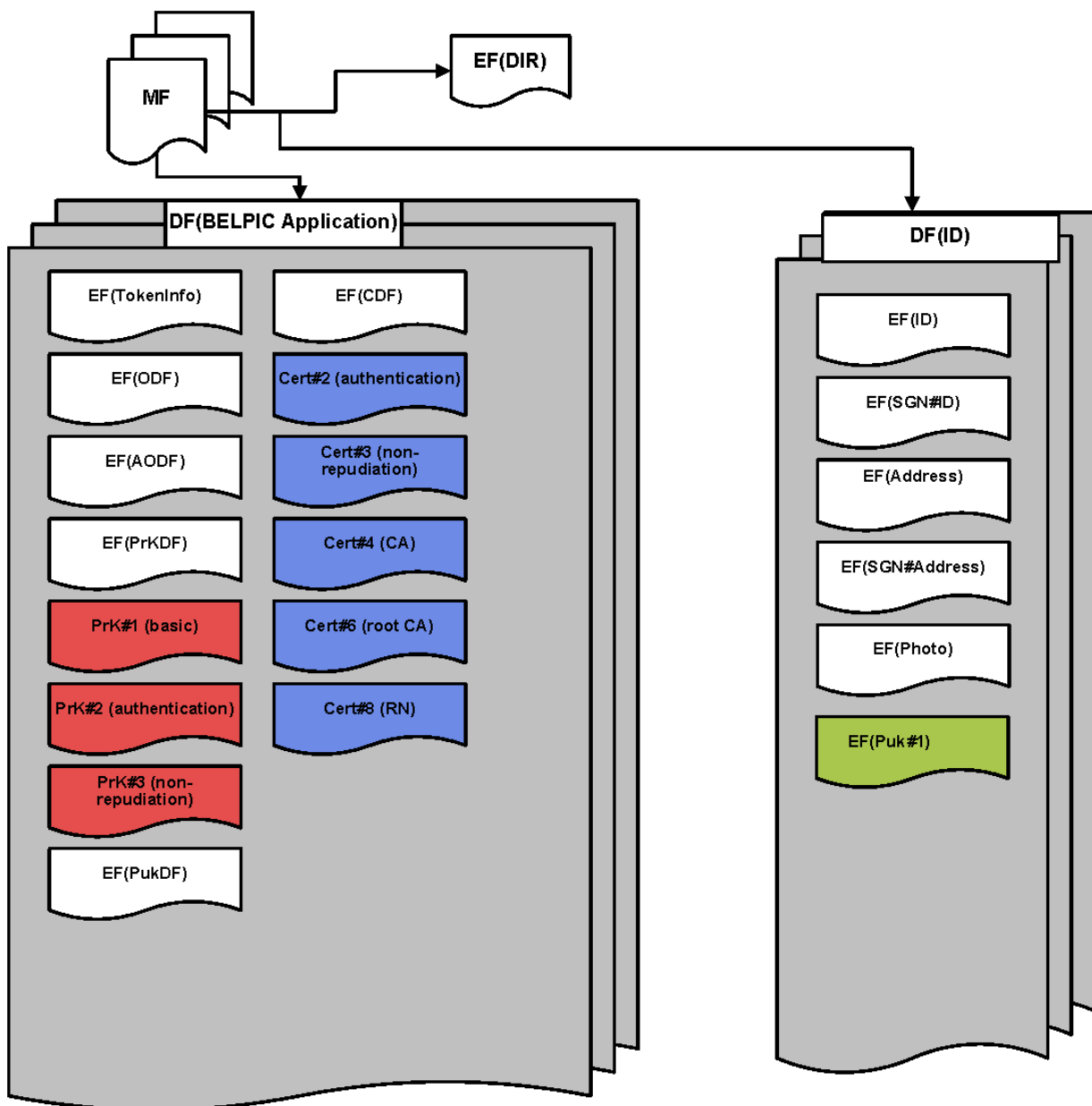
Remark: the maximum size of the files is only indicated when they can be updated later and they have to be created with a maximum size larger than the current one.

The files that can never be modified are created with the exact size to fit the content.

In case a file can be modified, its size is specified in the document.

### 4.1. File structure

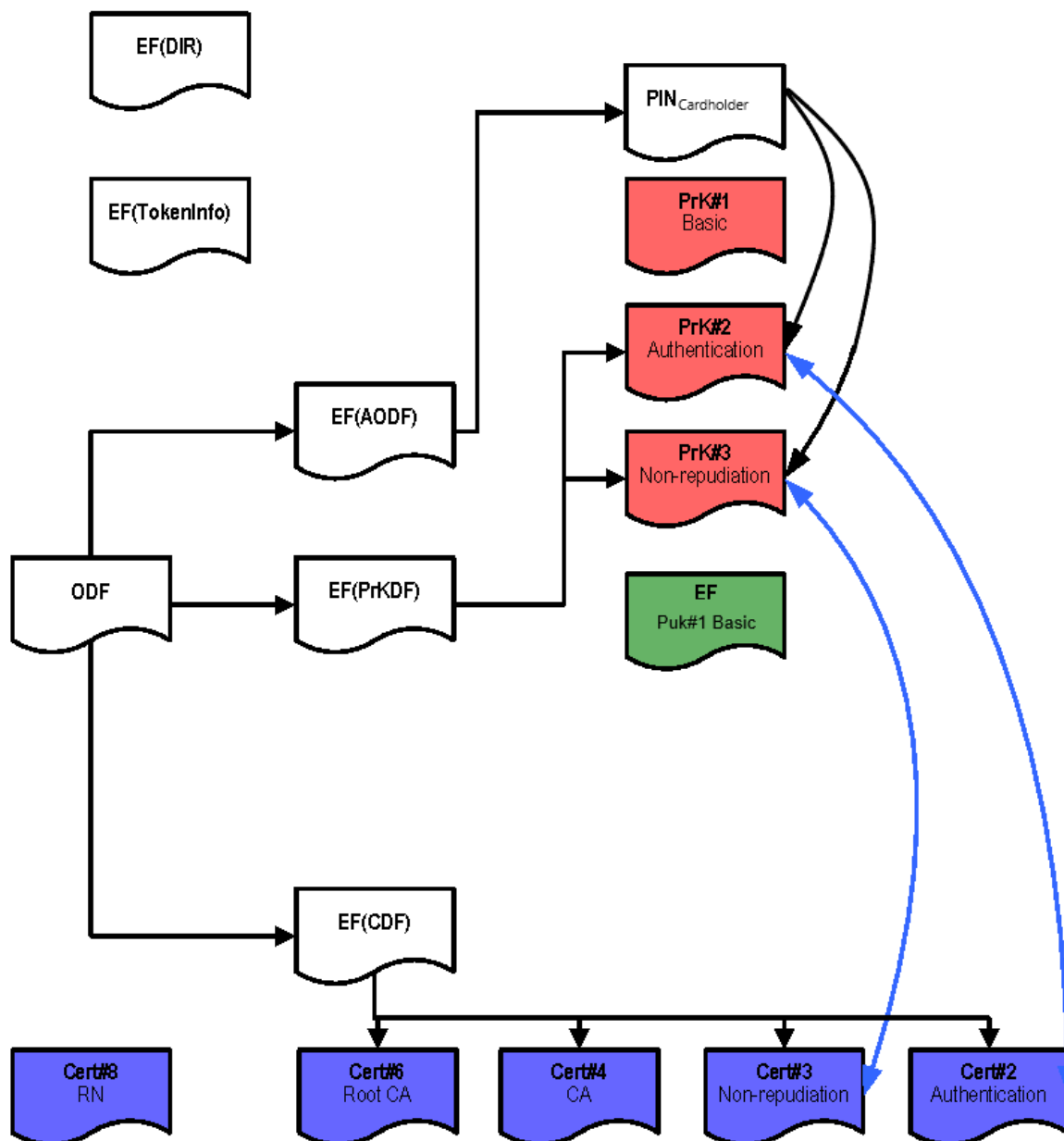
The file structure of the card is described in the figure below.



## 4.2. PKCS#15 files

The content of the **DF(BELPIC)** application directory files is compliant with PKCS#15 v1.1.

A directory file, **EF(DIR)**, containing the AID (ISO/IEC 7816–5) for each application in the EID card is present in the **Master File**.



The purpose of the figure above is to show the relationship between certain files **EF(ODF)**, **EF(AODF)**, **EF(PrKDF)** and **EF(CDF)** in the **DF(BELPIC)** Directory. **EF(ODF)** points to other EFs.

**EF(PrKDF)** contains cross-reference pointers to an authentication object (PIN) used to protect access to the keys. Arrows between PIN and Private Keys indicate this.

Some certificates (**Cert#2** & **Cert#3**) contain a public key whose private key also resides on the card, so these certificates contain the same identifier as the corresponding private key. Arrows between Certificates and Private Keys indicate this.



### 4.3. Files identifiers and read permissions

✗

Not possible (forbidden by the card Operating System/applet)

ALW

Always

	Reference (hexa)	Read Binary
MF	3F00	✗
EF(DIR)	2F00	ALW
- DF(BELPIC)	DF00	✗
EF(ODF)	5031	ALW
EF(TokenInfo)	5032	ALW
EF(AODF)	5034	ALW
EF(PrKDF)	5035	ALW
EF(CDF)	5037	ALW
EF(Cert#2) (auth)	5038	ALW
EF(Cert#3) (non-rep)	5039	ALW
EF(Cert#4) (CA)	503A	ALW
EF(Cert#6) (Root CA)	503B	ALW
EF(Cert#8) (RRN)	503C	ALW
- DF(ID)	DF01	✗
EF(ID#RRN)	4031	ALW
EF(SGN#ID#RRN)	4032	ALW
EF(ID#Address)	4033	ALW
EF(SGN#Adress#RRN)	4034	ALW
EF(ID#Photo)	4035	ALW
EF(Puk#1 Basic)	4040	ALW

#### **4.5. Note for non-eID cards**

The name Belpic does not only refer to the Belgian eID cards. All DFs, files, PINs and keys are also present on **Kids cards** and the **resident cards for EU and non-EU citizens**.

In some cases however, the contents of certain files may differ.

For example, the **EF(ID#RRN)** file contains extra fields on resident cards for EU and non-EU citizens.

Or for Kids cards, the **EF(Cert#2) (auth)** file may be empty, depending on the age of the child.

## 5. PKCS#15 information detail

### 5.1. PKCS#15 application selection

The EID card supports direct application selection as defined in ISO/IEC 7816–4, Section 9 and ISO/IEC 7816–5, Section 6 (the full AID is to be used as parameter for a ‘SELECT FILE’ command).

The operating system of the card keeps track of the currently selected application and only allows the commands applicable to that particular application while it is selected.

When several PKCS#15 applications reside on one card, they are distinguished by their object identifier in their application template in **EF(DIR)**.

### 5.2. MF directory contents

#### 5.2.1. EF(DIR)

This file contains all application templates as defined in ISO/IEC 7816–5. Each application template (tag ‘61’H) for a PKCS#15 application must at least contain the following Data Objects:

- **Application Identifier:** tag ‘4F’, UTF-8 encoded
- **Path:** tag ‘51’, DER-encoded

Other tags from ISO/IEC 7816–5 may, at the application issuer’s discretion, be present as well. In particular, it is recommended that application issuers include the following Data Objects:

- **Application Label:** tag ‘50’, UTF-8 encoded
- **Discretionary Data Objects:** tag ‘51’, DER-encoded

Encoding						ASN.1 Syntax					
						<b>Belpic (One entry per application)</b>					
						-- [APPLICATION 1] IMPLICIT SEQUENCE					
						<b>Application ID</b>					
						-- [APPLICATION 15] IMPLICIT OCTET STRING					
4F	0C										
A0	00	00	01	77	50	4B	43	53	2D	31	35
						<b>Label</b>					
						-- [APPLICATION 16] IMPLICIT UTF8 String					
50	06					-- 'Belpic'					
42	45	4C	50	49	43						
						<b>Path</b>					
						-- [APPLICATION 17] IMPLICIT OCTET STRING					
51	04					-- MF/Belpic					
3F	00	DF	00			<b>Discretionary Data Object</b>					
						-- [APPLICATION 19] IMPLICIT SEQUENCE					
						<b>ObjectID</b>					
	06	03				-- OBJECT IDENTIFIER					
	60	38	02			-- belgian citizen (2.16.56.2)					

*Remark:* the Object Identifier **2.16.56.2** was originally intended to signify “Belgian citizen”. However, since this Object Identifier is also used for **Kids cards** and resident **cards for EU and non-EU citizens**, it should now be interpreted more generally as “Belpic card”.

### 5.3. DF(BELPIC) Application directory contents

This DF is the directory of the Belpic application.

No operation is available on this data file.

### 5.3.1. EF(TokenInfo)

This file contains generic information about the token as such and its capabilities. This information includes the token’s serial number, file types for object directory files, algorithms implemented on the token, etc.

OFFSET	Encoding	ASN.1 Syntax
00	30 27	-- SEQUENCE
		Version
02	02 01	-- INTEGER
04	00	-- 0
		Serial Number
05	04 10	-- OCTET STRING
07	{16 bytes}	-- chip serial number
		Application Label
17	80 06	-- [0] Label IMPLICIT UTF8 String
19	42 45 4C 50 49 43	-- "BELPIC"
		TokenFlags
1F	03 02	-- BIT STRING
21	04 30	-- prnGeneration(2), eidCompliant (3)
23	9E 04	-- [30] BELPIC Application IMPLICIT INTEGER
25	{4 bytes}	-- Version

#### Version bytes:

- Graphical personalisation version (default = 0)
- Electrical personalisation version (default = 0)
- Electrical personalisation interface version<sup>1</sup> (default = 0)
- Reserved for future use (40)

<sup>1</sup> This is used to indicate to an application which file system organisation is used. This value only changes when a new version is no more compatible with the previous one.

### 5.3.2. EF(ODF)

The Object Directory File (**ODF**) is a transparent elementary file, which contains pointers to other elementary files (**PrKDF**, **PuKDF**, **CDF**, **AODF**) of the EID card. The information is presented in ASN.1 syntax according to PKCS #15.

An application using the EID card must use this file to determine how to perform security services with the card.

OFFSET	Encoding					ASN.1 Syntax				
00	A0	0A				-- [0] Private Keys				
						Path				
02		30	08			-- SEQUENCE				
						Path				
04			04	06		-- OCTET STRING				
06			3F 00 DF 00 50 35			-- MF/Belpic/PrKDF				
0C	A4	0A				-- [4] Certificates				
						Path				
0E		30	08			-- SEQUENCE				
						Path				
10			04	06		-- OCTET STRING				
12			3F 00 DF 00 50 37			-- MF/Belpic/CDF				
18	A8	0A				-- [8] Authentication Objects				
						Path				
1A		30	08			-- SEQUENCE				
						Path				
1C			04	06		-- OCTET STRING				
1E			3F 00 DF 00 50 34			-- MF/Belpic/AODF				

**Remark:** The AODF path might be removed in a future version, as it is the default path.

### 5.3.3. EF(AODF)

This elementary file (Authentication Object Directory File) contains generic authentication object attributes such as allowed characters, PIN length, PIN padding character, etc. It also contains the pointers to the authentication objects themselves (in the case of PINs, pointers to the DF in which the PIN file resides). The authentication objects are used to control access to other objects such as keys. The content of this file is according to PKCS#15.

02	30 0F		<i>Common Object Attributes</i>
			-- SEQUENCE
			<i>Label</i>
			-- UTF8 String
			-- "Basic PIN"
			<i>Common Object Flags</i>
			-- BIT STRING
			-- private(0), modifiable(1)
			<i>Common Authentication Object Attributes</i>
			-- SEQUENCE
			<i>Authority ID</i>
			-- OCTET STRING
			-- '01'
			-- [1] Pin Attributes
			-- SEQUENCE
			<i>Pin Flags</i>
			-- BIT STRING
			-- initialized(4), needs-padding(5)
			<i>PinType</i>
			-- ENUMERATED
			-- bcd(0)
			<i>Min Length</i>
			-- INTEGER
			-- 4
			<i>Stored Length</i>
			-- INTEGER
			-- 8 bytes
			-- [0] Pin Reference IMPLICIT INTEGER
			-- 1
			<i>Pad Char</i>
			-- OCTET STRING
			-- 'FF'
			<i>Path</i>
			-- SEQUENCE
			<i>Path</i>
			-- OCTET STRING
			-- 'MF'
04	0C 09		
06	42 61 73 69 63 20 50 49 4E		
0F	03 02		
11	06 C0		
13	30 03		
15	04 01		
17	01		
18	A1 1B		
1A	30 19		
1C	03 02		
1E	02 0C		
20	0A 01		
22	00		
23	02 01		
25	04		
26	02 01		
28	08		
29	80 01		
2B	01		
2C	04 01		
2E	FF		
2F	30 04		
30	04 02		
33	3F 00		
35			

#### **5.3.4. EF(PrKDF)**

This transparent elementary file (Private Key Directory File) contains general key attributes such as labels, intended usage, identifiers etc. It also contains the pointers to the keys themselves. The keys reside in the BELPIC application directory on the card.

OFFSET	Encoding	ASN.1 Syntax
00	A0 3A	<i>Private Authentication Key</i> -- [0] Private EC Key Attributes
02	30 17	<i>Common Object Attributes</i> -- SEQUENCE
04	0C 0E	<i>Label</i> -- UTF8 String
06	41 75 74 68 65 6E 74 69 63 61 74 69 6F 6E	-- "Authentication"
14	03 02	<i>Common Object Flags</i> -- BIT STRING
16	06 C0	-- private(0), modifiable(1)
18	04 01	<i>Authority ID</i> -- OCTET STRING
1A	01	-- '01'
1B	30 0F	<i>Common Key Attributes</i> -- SEQUENCE
1D	04 01	<i>Identifier</i> -- OCTET STRING
1F	02	-- '02'
20	03 02	<i>KeyUsageFlags</i> -- BIT STRING
22	05 20	-- Sign(2)
24	03 02	<i>Key Access Flags</i> -- BIT STRING
26	03 B8	-- sensitive(0) alwaysSensitive(2) neverextractable(3) local(4)
28	02 02	<i>KeyReference</i> -- INTEGER
2A	00 82	-- '82'
2C	A1 0E	-- [1] Private EC Key Attributes
2E	30 0C	<i>Path</i> -- SEQUENCE
30	30 06	<i>Path</i> -- SEQUENCE
32	04 04	<i>Path</i> -- OCTET STRING
34	3F 00 DF 00	-- MF
38	02 02	<i>Key Info</i> -- INTEGER
3A	01 80	-- 384 bit length



OFFSET	Encoding						ASN.1 Syntax					
3C	A0 39						<i>Private Non-repudiation Key</i>					
							-- [0] Private EC Key Attributes					
3E	30 15						<i>Common Object Attributes</i>					
							-- SEQUENCE					
40		0C 09					<i>Label</i>					
42		53 69 67 6E 61 74 75 72 65					-- UTF8 String					
							-- "Signature"					
4C		03 02					<i>Common Object Flags</i>					
4E		06 C0					-- BIT STRING					
							-- private(0), modifiable(1)					
50		04 01					<i>Authority ID</i>					
52		01					-- OCTET STRING					
							-- '01'					
53		02 01					<i>UserConsent</i>					
55		01					-- INTEGER					
							-- 15					
56	30 10						<i>Common Key Attributes</i>					
							-- SEQUENCE					
58		04 01					<i>Identifier</i>					
5A		03					-- OCTET STRING					
							-- '03'					
5B		03 03					<i>KeyUsageFlags</i>					
5D		06 00 40					-- BIT STRING					
							-- NonRepudiation(9)					
60		03 02					<i>Key Access Flags</i>					
62		03 B8					-- BIT STRING					
							-- sensitive(0)					
							alwaysSensitive(2)					
							neverextractable(3)					
							local(4)					
64		02 02					<i>KeyReference</i>					
66		00 83					-- INTEGER					
							-- '83'					
68	A1 0E						-- [1] Private EC Key Attributes					
6A		30 0C					<i>Path</i>					
							-- SEQUENCE					
6C			30 06				<i>Path</i>					
							-- SEQUENCE					
6E				04 04			<i>Path</i>					
70				3F 00 DF 00			-- OCTET STRING					
							-- MF					
74			02 02				<i>Key Info</i>					
76			01 80				-- INTEGER					
78							-- 384					

#### **5.3.5. EF(PuKDF)**

This transparent elementary file (Public Key Directory File) can be regarded as directories of public keys known to the PKCS #15 application. They contain general key attributes such as labels, intended usage, identifiers, etc. When applicable, it contains cross-reference pointers to authentication objects used to protect access to the keys. Furthermore, they contain pointers to the keys themselves. Private keys corresponding to public keys must share the same identifier. The keys reside in the BELPIC application directory on the card.

As no public keys are used through the PKCS#15 interface, this file does not exist.

### 5.3.6. EF(CDF)

This transparent elementary file contains attributes and pointers to the authentication certificate (Cert #2), non-repudiation signature certificate (Cert #3), CA certificate (Cert#4) and root certificate (Cert #6). Information in this file contains certificate attributes such as labels, key identifiers, pointers to certificate files etc. The format of the file is specified in PKCS#15.

Depending on the citizen's choice or the type of card, there can be 3 cases:

- **All certificates are present:** In this case, the **EF(CDF)** is exactly as show below.
- **No Non-repudiation certificate is present.** In this case, the information about the *Non-repudiation certificate* (bytes 30 27 30 12 ... DF 00 50 29) is not present. The information about the *Intermediate CA certificate* immediately follows the information about the *Authentication certificate*, and the remainder of the file is filled with zero bytes. Additionally, the *Non-repudiation certificate* file is filled with 2500 zero bytes.
- **No Authentication and Non-repudiation certificates are present.** In this case, the information about the *Authentication* and *Non-repudiation certificates* is not present. The file starts with the information about the *Intermediate CA certificate* (bytes 30 23 30 3B ...), and the remainder of the file is filled with zero bytes. Additionally, the *Authentication* and *Non-repudiation certificate* files are filled with 2500 zero bytes.

OFFSET	Encoding	ASN.1 Syntax
00	30 32	<b>Authentication Certificate</b> -- SEQUENCE
02	30 17	<b>Common Object Attributes</b> -- SEQUENCE
04	0C 0E	<b>Label</b> -- UTF8String
06	41 75 74 68 65 6E 74 69 63 61 74 69 6F 6E	-- "Authentication"
14	03 02	<b>Common Object Flags</b> -- BIT STRING
16	06 40	-- modifiable(1)
18	04 01	<b>AuthID</b> -- OCTET STRING
1A	01	-- '01'
1B	30 06	<b>Common Certificate Attributes</b> -- SEQUENCE
1D	04 01	<b>Identifier</b> -- OCTET STRING
1F	02	-- '02'
20	83 01	--[3] ImplicitTrust IMPLICIT BOOLEAN
22	00	-- False
23	A1 0C	-- [1] 509CertificateAttributes
25	30 0A	<b>Path</b> -- SEQUENCE
27	30 08	<b>Path</b> -- SEQUENCE
29	04 06	<b>Path</b> -- OCTET STRING
2B	3F 00 DF 00 50 38	-- MF/Belpic/Cert#2(auth)

OFFSET	Encoding	ASN.1 Syntax
		<b>Non-Repudiation Certificate</b>
31	30 2A	-- SEQUENCE
		<i>Common Object Attributes</i>
33	30 12	-- SEQUENCE
		<i>Label</i>
35	0C 09	-- UTF8String
37	53 69 67 6E 61 74 75 72 65	-- "Signature"
		<i>Common Object Flags</i>
40	03 02	-- BIT STRING
42	06 40	-- modifiable(1)
		<i>AuthID</i>
44	04 01	-- OCTET STRING
46	01	-- '01'
		<i>Common Certificate Attributes</i>
47	30 06	-- SEQUENCE
		<i>Identifier</i>
49	04 01	-- OCTET STRING
4B	03	-- '03'
4C	83 01	--[3] ImplicitTrust IMPLICIT BOOLEAN
4E	00	-- False
4F	A1 0C	-- [1] X509CertificateAttributes
		<i>Path</i>
51	30 0A	-- SEQUENCE
		<i>Path</i>
53	30 08	-- SEQUENCE
		<i>Path</i>

OFFSET	Encoding	ASN.1 Syntax
		<b>Certification Authority Certificate</b>
5D	30 26	-- SEQUENCE
		<b>Common Object Attributes</b>
5F	30 0B	-- SEQUENCE
		<b>Label</b>
61	0C 02	-- UTF8String
63	43 41	-- "CA"
		<b>Common Object Flags</b>
65	03 02	-- BIT STRING
67	06 40	-- modifiable(1)
		<b>AuthID</b>
69	04 01	-- OCTET STRING
6B	01	-- '01'
		<b>Common Certificate Attributes</b>
6C	30 09	-- SEQUENCE
		<b>Identifier</b>
6E	04 01	-- OCTET STRING
70	04	-- '04'
		<b>Authority</b>
71	01 01	-- BOOLEAN
73	FF	-- True
74	83 01	-- [3] ImplicitTrust IMPLICIT BOOLEAN
76	00	-- False
77	A1 0C	-- [1] X509CertificateAttributes
		<b>Path</b>
79	30 0A	-- SEQUENCE
		<b>Path</b>
7B	30 08	-- SEQUENCE
		<b>Path</b>

OFFSET	Encoding	ASN.1 Syntax
		<b>Root Certificate</b>
85	30 28	-- SEQUENCE
		<b>Common Object Attributes</b>
87	30 0D	-- SEQUENCE
		<b>Label</b>
89	0C 04	-- UTF8String
8B	52 6F 6F 74	-- "Root"
		<b>Common Object Flags</b>
8F	03 02	-- BIT STRING
91	06 40	-- modifiable(1)
		<b>AuthID</b>
93	04 01	-- OCTET STRING
95	01	-- '01'
		<b>Common Certificate Attributes</b>
96	30 09	-- SEQUENCE
		<b>Identifier</b>
98	04 01	-- OCTET STRING
9A	06	-- '06'
		<b>Authority</b>
9B	01 01	-- BOOLEAN
9D	FF	-- True
9E	83 01	-- [3] ImplicitTrust IMPLICIT BOOLEAN
A0	00	-- False
A1	A1 0C	-- [1] X509CertificateAttributes
		<b>Path</b>
A3	30 0A	-- SEQUENCE
		<b>Path</b>
A5	30 08	-- SEQUENCE
		<b>Path</b>

## 6. Application information detail

### 6.1. *TLV format*

Some files are encoded in a simplified **TLV** format:

- ☐ a tag identifying the data: 1 byte
- ☐ the length of the data<sup>2</sup>:
  - < 255: 1 byte
  - >= 255: multiple bytes:
    - FF x – 255
    - FF FF x – 510
    - FF FF FF x – 765
    - ...
- ☐ the data: x bytes

#### Encoding type:

- ☐ All data is either pure binary, or UTF-8 containing Unicode characters.
- ☐ UTF-8 strings are not null-terminated.
- ☐ UTF-8 containing multi-byte characters is referred as **UTF-8**.
- ☐ When data contains only 7-bits characters, it is referred as **ASCII**, although they are fully compatible with the **UTF-8/Unicode** conventions.
- ☐ The actual data may be followed by padding bytes ('0'). They have to be ignored.

---

<sup>2</sup> Nor the tag, nor the length are counted in this length.

## 6.2. Identity data

### 6.2.1. *DF(ID)*

This transparent data file contains all files related to the citizen and to information that is managed by the National Register.

### 6.2.2. *EF(ID#RN)*

This transparent elementary file contains all permanent information about the citizen, such as issuing country, issuing authority, issuing date, validity period, name, address, birth date, etc. This is known as the 'ID file'.

This file contains most of the information that is graphically personalised on the card plastic.

It is formatted in simplified **TLV** format (see 6.1).

The file structure version corresponding to this document is 2.

The contents of this file are documented in a separate document.

### 6.2.3. *EF(SGN#ID)*

This transparent elementary file contains the signature of the ***EF(ID#RN)*** by the National Register.

As the ***EF(ID#RN)*** file contains the hash of the picture, the picture is also implicitly signed.

Signature format: ECDSA P-384 with SHA-2-384.

### 6.2.4. *EF(ID#Address)*

This transparent elementary file contains the information about the citizen's residence.

It is formatted in simplified **TLV** format (see 6.1).

The file structure version corresponding to this document is 2

The contents of this file are documented in a separate document.

### 6.2.5. *EF(SGN#Address)*

This transparent elementary file contains the signature of the ***EF(ID#Address)*** by the National Register.

***EF(SGN#ID)*** is first appended to ***EF(ID#Address)*** before signing, in order to ensure the consistency with the file ***EF(ID#RN)***. If zero bytes are present at the end of ***EF(ID#Address)***, they need to be removed first.

Signature format: ECDSA P-384 with SHA-2-384

***EF(SGN#Address)*** has a fixed length. If the EC signature (in asn.1 format) it contains is smaller than this fixed length, it will be padded by zero bytes.

**Remark:** The length of the asn.1 encoded signature can be found by adding 2 to the value of its second byte.



### 6.2.6. EF(ID#Photo)

This transparent elementary file contains the citizen’s picture in the standard JPEG format.

As the **EF(ID#RN)** file contains the hash of the picture, the picture is also implicitly signed when signing this file.

The picture resolution will change in 2021:

- Old picture resolution: width: 140 pixels, height: 200 pixels, grey levels: 8 bits
- New picture resolution: width: 150 pixels, height: 195 pixels, grey levels: 8 bits

**Remark:** The resolution and colour encoding are included in the JPEG format. It is advisable to dynamically use these, as they could change in the future.

### 6.2.7. EF(PuK#1 ID)

This transparent elementary file contains the public card key. As the **EF(ID#RN)** file contains the hash of the public card key, the public card key is also implicitly signed when signing this

OFFSET	Encoding	ASN.1 Syntax
		<b>Authentication Certificate</b>
00	30 76	-- SEQUENCE
02	30 10	-- SEQUENCE
		<b>Label</b>
04	06 07	-- OBJECT_ID
06	2A 86 48 CE 3D 02 01	EcPublicKey (1 2 840 10045 2 1)
0D	06 05	-- OBJECT_ID
0F	2B 81 04 00 22	Secp384r1 (1 3 132 0 34)
14	03 62	-- BIT_STRING (98 bytes)
16	00	-- no bits unused in the final byte
17	04	compression byte
18	{48 bytes}	-- X coordinate
48	{48 bytes}	-- Y coordinate

file.

## 7. Public and Private Keys detail

### 7.1. *PrivateKey #1*

This file contains the private **Basic Key**. It is involved in the **Internal Authentication** process.

### 7.2. *Public Key #1*

This file contains the public **Basic Key**. It can be used to check the signature of the **Internal Authentication** process.

### 7.3. *PrivateKey #2*

This file contains the private RSA **Authentication Key**.

### 7.4. *Private Key #3*

This file contains the private **Non-Repudiation Key**.

The userConsent element in **PrKDF** contains value 1 for this key i.e. the cardholder must manually enter the corresponding PIN for each private key operation.

## 8. Certificates detail

All certificates stored in the card are DER encoded (not Base 64).

### 8.1. Certificate #2

This file contains the citizen's X.509 **Authentication Certificate** containing the public key corresponding to the private **Authentication Key** (Private Key #2). When the file is created and written to during personalisation, 1100 zero bytes are appended to it. If **no authentication certificate** is issued for this person, this file consists of 2500 zero bytes.

### 8.2. Certificate #3

This file contains the citizen's X.509 **Non-Repudiation Certificate** containing the public key corresponding to the private '**Non-Repudiation Key**' (Private Key #3). When the file is created and written to during personalisation, 1100 zero bytes are appended to it. If **no non-repudiation certificate** is issued for this person, this file consists of 2500 zero bytes.

### 8.3. Certificate #4

This file contains the X.509 **Citizen's CA Certificate** or **Foreigner's CA Certificate** used to sign the **Authentication Certificate** (#2) and the **Non-Repudiation Certificate** (#3). When the file is created and written to during personalisation, 1100 zero bytes are appended to it. If **no Citizen's CA Certificate** or **Foreigner's CA Certificate** is written to the card, this file consists of 2500 zero bytes.

In the case of a resident **card for an EU or non-EU citizen**, this file contains the **Foreigner CA certificate**.

### 8.4. Certificate #6

This file contains the X.509 **ROOT CERTIFICATE** used to sign the **Citizen's CA certificate** (#4) or the **Foreigner CA certificate** (#4) and the **RRN certificate** (#8). When the file is created and written to during personalisation, 1100 zero bytes are appended to it. If **no ROOT certificate** is written to the card, this file consists of 2500 zero bytes.

### 8.5. Certificate #8

This file contains the X.509 **RRN Certificate**. This certificate corresponds to the private key used to sign the files **EF(ID#RN)** and **EF(ID#Address)**. When the file is created and written to during personalisation, 1100 zero bytes are appended to it.