INTERNATIONAL STANDARD

ISO/IEC 19785-4

First edition
2010-08-15

Information technology — Common Biometric Exchange Formats Framework —

Part 4: Security block format specifications

Technologies de l'information — Cadre de formats d'échange biométriques communs —

Partie 4: Spécifications de format de bloc de sécurité

Reference number
ISO/IEC 19785-4:2010(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword iv

Introduction v

1 Scope 1

2 Normative references 1

3 Terms and definitions 2
3.1 Terms defined in ISO/IEC 19785-1 2
3.2 Terms defined in ISO/IEC 19784-1 2
3.3 Terms defined in ISO/IEC 24761 2
3.4 Terms defined in ISO/IEC 9798-6 2

4 Abbreviated terms 2
4.1 Abbreviated terms defined in ISO/IEC 19785-1 2
4.2 Abbreviated terms defined in ISO/IEC 24761 2
4.3 Abbreviated terms defined in ISO/IEC 9798-6 2
4.4 Abbreviated terms defined in RFC 3852 3

5 Security block format: general purpose 3
5.1 Security block format owner 3
5.2 Security block format owner identifier 3
5.3 Security block format name 3
5.4 Security block format identifier 3
5.5 ASN.1 object identifier for this security block format 3
5.6 Domain of use 4
5.7 Version identifier 4
5.8 Format specification and conformance statement 4
5.9 Encoding of abstract values 10

6 Security block format: signature only 11
6.1 Security block format owner 11
6.2 Security block format owner identifier 11
6.3 Security block format name 11
6.4 Security block format identifier 11
6.5 ASN.1 object identifier for this security block format 11
6.6 Domain of use 12
6.7 Version identifier 12
6.8 Format specification and conformance statement 12

Annex A (normative) **ASN.1 module for security block format** 13

Annex B (informative) **Difference from types defined in RFC 5911** 15

Bibliography 18

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 19785-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

ISO/IEC 19785 consists of the following parts, under the general title *Information technology — Common Biometric Exchange Formats Framework*:

— *Part 1: Data element specification*

— *Part 2: Procedures for the operation of the Biometric Registration Authority*

— *Part 3: Patron format specifications*

— *Part 4: Security block format specifications*

Introduction

Biometric verification and identification are important techniques for the authentication and/or identification of an individual. Biometric data used in biometric verification and identification has to be from a trusted source with no interference in transmission (integrity). It might or might not be necessary for it to be kept secret (encryption) depending on security policy. This part of ISO/IEC 19785 provides for both integrity and encryption of the biometric data.

To ensure interoperability, the Common Biometric Exchange Formats Framework (CBEFF) was specified in ISO/IEC 19785-1 to associate meta-data with one or more Biometric Data Blocks (BDBs). In ISO/IEC 19785-1, the options for integrity and encryption, and the concept of a security block (SB) to contain security information related to these options are defined, but the format and detailed content of security blocks (SB formats) are not specified.

There are several steps in the chain, starting from a CBEFF patron format.

First, the patron format can determine that the abstract value of the CBEFF data element CBEFF_BDB_encryption_options is fixed as NO ENCRYPTION and that the CBEFF data element CBEFF_BIR_integrity_options is fixed as NO INTEGRITY. In this case, there is no need for a security block to be required in that patron format.

If the patron format requires the inclusion of a security block in some circumstances, it can fix it as one of the security blocks defined in this part of ISO/IEC 19785 (or as some other security block), or can include the CBEFF data elements CBEFF_SB_format_owner and CBEFF_SB_format_type to identify one of these or some other security block format.

Besides the security block formats defined in this part of ISO/IEC 19785, there will be many possible CBEFF security block formats meeting different needs. For example, a security block format is specified for the ILO seafarers profile in ISO/IEC 24713-3. The security block format specified in Clause 5 is designed to be as general as possible. The security block format specified in Clause 6 is designed to provide a basic security provision and supports integrity only.

This part of ISO/IEC 19785 specifies two security block formats.

The first security block specifies a general-purpose security block format with optional elements for encryption, and for integrity, using RFC 3852 Cryptographic Message Syntax (CMS), with certain modifications to **`EnvelopedData`**, **`EncryptedData`**, **`SignedData`**, and **`AuthenticatedData`**, to meet the needs and requirements in expressing the security of biometric information in conformance with CBEFF. The second is named signature-only security block format, which is also defined using RFC 3852.

The general-purpose security block format specified in this part of ISO/IEC 19785 also contains optional Authentication Context for Biometrics (ACBio) instances specified in ISO/IEC 24761. ACBio also uses the RFC 3852 Cryptographic Message Syntax scheme. The inclusion of ACBio instances enables the security levels of the systems producing the authenticated biometric to be determined. The optional use of ACBio instances is an important part of the provision of a telebiometric authentication infrastructure (TAI) [3].

Information technology — Common Biometric Exchange Formats Framework —

Part 4: Security block format specifications

1 Scope

This part of ISO/IEC 19785 specifies security block formats (see ISO/IEC 19785-1) registered in accordance with ISO/IEC 19785-2 as formats defined by the CBEFF biometric organization ISO/IEC JTC 1/SC 37, and specifies their registered security block format identifiers.

NOTE The security block format identifier is recorded in the standard biometric header (SBH) of a patron format (or defined by that patron format as the only available security block format).

The general-purpose security block format provides for specification of whether the biometric data block (BDB) is encrypted or the SBH and BDB have integrity applied (or both), and can include ACBio instances (see ISO/IEC 24761). This security block provides all necessary security parameters, including those used for encryption or integrity.

It does not restrict the algorithms and parameters used for encryption or integrity, but provides for the recording of such algorithms and parameter values.

It is a matter for profiling to determine, for a particular application area, what algorithms and parameter ranges can be used by the generator of a security block, and hence what algorithms and parameter ranges have to be supported by the user of a security block. This is out of the scope of this part of ISO/IEC 19785.

The second security block is more limited, but simpler (and in particular cannot contain ACBio instances, and does not support encryption of the BDB).

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8824 (all parts) | ITU-T Rec. X.680–683, *Information technology — Abstract Syntax Notation One (ASN.1)*

ISO/IEC 8825 (all parts) | ITU-T Rec. X.690–693, *Information technology — ASN.1 encoding rules*

ISO/IEC 9798-6, *Information technology — Security techniques — Entity authentication — Part 6: Mechanisms using manual data transfer*

ISO/IEC 19784-1, *Information technology — Biometric application programming interface — Part 1: BioAPI specification*

ISO/IEC 19785-1, *Information technology — Common Biometric Exchange Formats Framework — Part 1: Data element specification*

ISO/IEC 24761, *Information technology — Security techniques — Authentication context for biometrics*

RFC 3852, *Cryptographic Message Syntax (CMS)*, July 2004

RFC 5911, *New ASN.1 Modules for Cryptographic Message Syntax (CMS) and S-MIME*, June 2010

3 Terms and definitions

3.1 Terms defined in ISO/IEC 19785-1

For the purposes of this document, the following terms defined in ISO/IEC 19785-1 apply:

biometric, biometrics, biometric data block (BDB), biometric information record (BIR), CBEFF biometric organization, security block (SB), security block format, security block format identifier, security block format owner, standard biometric header (SBH).

3.2 Terms defined in ISO/IEC 19784-1

For the purposes of this document, the following term defined in ISO/IEC 19784-1 applies:

BioAPI Unit.

3.3 Terms defined in ISO/IEC 24761

For the purposes of this document, the following terms defined in ISO/IEC 24761 apply:

ACBio instance, authentication context for biometrics (ACBio), biometric processing unit (BPU).

3.4 Terms defined in ISO/IEC 9798-6

For the purposes of this document, the following term defined in ISO/IEC 9798-6 applies:

message authentication code.

4 Abbreviated terms

4.1 Abbreviated terms defined in ISO/IEC 19785-1

For the purposes of this document, the following abbreviated terms in ISO/IEC 19785-1 apply:

BDB, BIR, CBEFF, SB, SBH.

4.2 Abbreviated terms defined in ISO/IEC 24761

For the purposes of this document, the following abbreviated terms in ISO/IEC 24761 apply:

ACBio, BPU.

4.3 Abbreviated terms defined in ISO/IEC 9798-6

For the purposes of this document, the following abbreviated term in ISO/IEC 9798-6 applies:

MAC.

4.4 Abbreviated terms defined in RFC 3852

For the purposes of this document, the following abbreviated term in RFC 3852 applies:

CRL.

5 Security block format: general purpose

5.1 Security block format owner

ISO/IEC JTC 1/SC 37

5.2 Security block format owner identifier

257 (0101Hex). This identifier has been assigned in accordance with ISO/IEC 19785-2 to ISO/IEC JTC 1/SC 37 as a CBEFF biometric organization.

5.3 Security block format name

ISO/IEC JTC 1/SC 37 CBEFF general-purpose security block format

5.4 Security block format identifier

1 (0001 Hex). This has been registered in accordance with ISO/IEC 19785-2 when DER encodings (see ISO/IEC 8825-1) are applied.

2 (0002 Hex). This has been registered in accordance with ISO/IEC 19785-2 when canonical PER encodings (see ISO/IEC 8825-2) are applied.

3 (0003 Hex). This has been registered in accordance with ISO/IEC 19785-2 when canonical XER encodings (see ISO/IEC 8825-3) are applied.

5.5 ASN.1 object identifier for this security block format

5.5.1 The case of DER encodings

{iso registration-authority cbeff(19785) organizations(0) jtc-sc37 (257) sb-formats(3) general-purpose(0) der-encoding(1)}

or, in XML value notation,

1.1.19785.0.257.3.0.1

5.5.2 The case of canonical PER encodings

{iso registration-authority cbeff(19785) organizations(0) jtc-sc37 (257) sb-formats(3) general-purpose(0) per-encoding(2)}

or, in XML value notation,

1.1.19785.0.257.3.0.2

5.5.3 The case of canonical XER encodings

{iso registration-authority cbeff(19785) organizations(0) jtc-sc37 (257) sb-formats(3) general-purpose(0) xer-encoding(3)}

or, in XML value notation,

1.1.19785.0.257.3.0.3

5.6 Domain of use

The general-purpose security block is designed for applications that require integrity and/or encryption, and optionally inclusion of ACBio instances.

5.7 Version identifier

This security block format specification has a version identifier of (major 0, minor 0).

5.8 Format specification and conformance statement

5.8.1 General

5.8.1.1 In this part of ISO/IEC 19785, a CBEFF security block is defined as the ASN.1 (see ISO/IEC 8824) type **CBEFFSecurityBlock** which is a sequence of the ASN.1 type **CBEFFSecurityBlockElement**.

```
CBEFFSecurityBlock ::= SEQUENCE OF CBEFFSecurityBlockElement

CBEFFSecurityBlockElement ::= CHOICE {
        elementCBEFFSB ContentInfoCBEFFSB,
        subBlockForACBio SubBlockForACBio,
        accumulatedACBioInstances ACBioInstances
        }
```

5.8.1.2 There are three alternatives for the type **CBEFFSecurityBlockElement**. These are **ContentInfoCBEFFSB**, **SubBlockForACBio**, or **ACBioInstances**. **CBEFFSecurityBlockElement** carries information about the integrity of the concatenation of the SBH and the BDB or encryption of the BDB. The latter two carry information on ACBio which is specified in ISO/IEC 24761.

5.8.1.3 The type **ContentInfoCBEFFSB** is defined as:

```
ContentInfoCBEFFSB ::= SEQUENCE {
        contentType CONTENT-TYPE.&id({ContentTypeCBEFF}),
        content [0] EXPLICIT CONTENT-TYPE.&Type
            ({ContentTypeCBEFF}{@contentType})
        }
```

NOTE This type replaces the type **ContentInfo** in RFC 5911. The first component of this type can take only four object identifiers, namely **id-envelopeRelatedData**, **id-encryptionRelatedData**, **id-signatureRelatedData**, or **id-authenticationRelatedData**, while that of the type **ContentInfo** in RFC 5911 can take other object identifiers.

This type can occur two times at most in the **CBEFFSecurityBlock** sequence, once to support integrity and once to support encryption.

The type **ContentInfoCBEFFSB** is composed of two components, **contentType** and **content**. The first component **contentType** is an object identifier, which indicates the type of content in the second component **content**. The value of **contentType** takes one of the following four object identifiers: **id-envelopeRelatedData**, **id-encryptionRelatedData**, **id-signatureRelatedData**, or

id-authenticationRelatedData. This is done by the following definition of **ContentTypeCBEFF** and that of the four **CONTENT-TYPE**s. Here type **CONTENT-TYPE** associates an object identifier with an ASN.1 type.

```
ContentTypeCBEFF CONTENT-TYPE ::= { envelopeRelatedData | encryptionRelatedData |
                    signatureRelatedData | authenticationRelatedData}

envelopeRelatedData CONTENT-TYPE ::= {
        EnvelopeRelatedData
        IDENTIFIED BY id-envelopeRelatedData
}

encryptionRelatedData CONTENT-TYPE ::= {
        EncryptionRelatedData
        IDENTIFIED BY id-encryptionRelatedData
}

signatureRelatedData CONTENT-TYPE ::= {
        SignatureRelatedData
        IDENTIFIED BY id-signatureRelatedData
}

authenticationRelatedData CONTENT-TYPE ::= {
        AuthenticationRelatedData
        IDENTIFIED BY id-authenticationRelatedData
}
```

The above listed four object identifier names are defined as follows:

```
id-envelopeRelatedData OBJECT IDENTIFIER ::= {
        iso(1) standard(0) cbeff(19785) contentType(1) envelopeRelatedData(1)
}

id-encryptionRelatedData OBJECT IDENTIFIER ::= {
        iso(1) standard(0) cbeff(19785) contentType(1) encryptionRelatedData(2)
}

id-signatureRelatedData OBJECT IDENTIFIER ::= {
        iso(1) standard(0) cbeff(19785) contentType(1) signatureRelatedData(3)
}

id-authenticationRelatedData OBJECT IDENTIFIER ::= {
        iso(1) standard(0) cbeff(19785) contentType(1) authenticationRelatedData(4)
}
```

id-envelopeRelatedData or **id-encryptionRelatedData** shall be taken in the field **contentType** of type **ContentInfoCBEFFSB** if the data element CBEFF_BDB_encryption_options (see ISO/IEC 19785-1) is present and contains the encoding for ENCRYPTION.

id-signatureRelatedData or **id-authenticationRelatedData** shall be taken in the field **contentType** of type **ContentInfoCBEFFSB** if the data element CBEFF_BIR_integrity_options (see ISO/IEC 19785-1) is present and contains the encoding for INTEGRITY.

5.8.1.4 The second alternative **subBlockForACBio** of type **SubBlockForACBio** shall be used if a BPU of a BioAPI unit generates and outputs an ACBio instance. This data shall be exchanged to the successive BPU of BioAPI unit. The type **SubBlockForACBio** is defined as follows:

```
SubBlockForACBio ::= SEQUENCE {
        bpuIOIndex INTEGER,
        acbioInstance ACBioInstance
}
```

The first component **bpuIOIndex** is the BPU IO index for the output from a BPU and is transferred to the next BPU as the BPU IO index for the input to the second BPU. The second component is the ACBio instance generated by the first BPU. For details, see ISO/IEC 24761.

5.8.1.5 The third alternative **accumulatedACBioInstances** of type **ACBioInstances** shall be used to record ACBio instances except the newest one, which is recorded in a data of type **SubBlockForACBio**. Type **ACBioInstances** is a sequence of type **ACBioInstances**.

```
ACBioInstances ::= SEQUENCE OF ACBioInstance
```

5.8.1.6 In the following, the use of SECURITY BLOCK is specified covering three options: 1) ENCRYPTION is set for CBEFF_DBD_encryption_options (or required by the patron format), 2) INTEGRITY is set for CBEFF_BIR_integrity_options (or required by the patron format), and 3) both are set (or either is required by the patron format).

5.8.2 Encryption

If the CBEFF_BDB_encryption_options abstract value in the SBH specifies ENCRYPTION, the security block shall contain a component of type **ContentInfoCBEFFSB**, the value of whose first component is **id-envelopeRelatedData** or **id-encryptionRelatedData**. As seen in the definition of **ContentInfoCBEFFSB**, the type of its second component is determined by the value of the first component, i.e., **EnvelopeRelatedData** is taken for **id-envelopeRelatedData**, and **EncryptionRelatedData** for **id-encryptionRelatedData**. The BDB contains an encrypted form of the biometric data.

NOTE 1 The selection of **EnvelopeRelatedData** and **EncryptionRelatedData** is dependent on the key management used (see RFC 3852 Cryptographic Message Syntax for a discussion of key management).

NOTE 2 Data elements in the SBH related to the BDB do not indicate the attributes of the encrypted BDB but indicate those of the original BDB (the biometric data before encryption).

5.8.2.1 envelopeRelatedData content type

5.8.2.1.1 The **envelopeRelatedData** content type associates an object identifier **id-envelopeRelatedData** with an ASN.1 type **EnvelopeRelatedData** as described in 5.8.1.3.

a) **EnvelopeRelatedData** consists of the content-encryption algorithm and encrypted content-encryption keys for one or more recipients. The encrypted biometric data is contained in the BDB. Any biometric data can be encrypted for an arbitrary number of recipients using any of the supported key management techniques for each recipient.

NOTE For details of key management, see RFC 3852.

b) A recipient decrypts one of the encrypted content-encryption keys in the data of the type **EnvelopeRelatedData** and then decrypts the encrypted biometric data stored in the BDB with the recovered content-encryption key

5.8.2.1.2 Type **EnvelopeRelatedData** is defined as follows:

```
EnvelopeRelatedData::= SEQUENCE {
        version CBEFFSBVersion DEFAULT v0,
        originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
        recipientInfos RecipientInfos,
        contentEncryptionAlgorithm ContentEncryptionAlgorithmIdentifier
}
```

a) **version** is the syntax version number of type **CBEFFSBVersion** defined as

```
CBEFFSBVersion ::= INTEGER { v0(0) } ( v0, ... )
```

b) The field **originatorInfo** of type **OriginatorInfo** optionally provides information about the originator. It is present only if required by the key management algorithm. It may contain certificates and CRLs. For details of type **OriginatorInfo**, see RFC 3852 and RFC 5911.

c) The field **recipientInfos** of type **RecipientInfos** is a collection of per-recipient information. There shall be at least one element in the collection. The type **RecipientInfos** is SET of **RecipientInfo**. For details of type **RecipientInfo**, see RFC 3852 and RFC 5911.

d) The field **contentEncryptionAlgorithm** identifies the content-encryption algorithm, and any associated parameters, used to encrypt the biometric data. The same content-encryption algorithm and content-encryption key are used for all recipients.

5.8.2.2 encryptionRelatedData content type

5.8.2.2.1 The **encryptionRelatedData** content type associates an object identifier **id-encryptionRelatedData** with an ASN.1 type **EncryptionRelatedData** as described in 5.8.1.3.

a) Unlike the **envelopeRelatedData** content type, the **encryptionRelatedData** content type has neither recipients nor encrypted content-encryption keys. Keys shall be managed by other means.

NOTE The typical application of the **encryptionRelatedData** content type will be to encrypt the biometric data for local storage, where the encryption key is derived from a password.

b) Type **EncryptionRelatedData** is defined as follows:

```
EncryptionRelatedData ::= SEQUENCE {
        version CBEFFSBVersion  DEFAULT v0,
        contentEncryptionAlgorithm ContentEncryptionAlgorithmIdentifier
}
```

5.8.3 Integrity

If CBEFF_BIR_integrity_options in the SBH is present and contains the encoding for INTEGRITY, the security block shall contain a component of type **ContentInfoCBEFFSB**, the value of whose first component is **id-signatureRelatedData** or **id-authenticationRelatedData**. As seen in the definition of **ContentInfoCBEFFSB**, the type of its second component is determined by the value of the first component, i.e., **SignatureRelatedData** is taken for **id-signatureRelatedData**, and **AuthenticationRelatedData** for **id-authenticationRelatedData**. When a digital signature is used to ensure integrity, the **signatureRelatedData** content type is used. When a MAC is used, the **authenticationRelatedData** content type is used. The digital signature or MAC is calculated on the concatenation of the SBH and the (possibly encrypted) BDB encodings.

5.8.3.1 signatureRelatedData content type

5.8.3.1.1 The **signatureRelatedData** content type associates an object identifier **id-signatureRelatedData** with an ASN.1 type **SignatureRelatedData** as described in 5.8.1.3.

a) The type **SignatureRelatedData** consists of one or more signature values. Any number of signers in parallel can sign the concatenation of the SBH and (possibly encrypted) BDB encodings. Unlike the **signedData** content type defined in RFC 3852 and RFC 5911, this content type does not contain the data that is being digitally signed.

b) The process by which **SignatureRelatedData** is constructed involves the following steps, which is illustrated as the left half of Figure 1:

1) For each signer, a message digest, or hash value, is computed on the concatenation of the SBH and (possibly encrypted) BDB encodings with a signer-specific message-digest algorithm (DA in Figure 1), and the result becomes the message digest (MD in Figure 1).

2) For each signer, the message digest is digitally signed using the signer's private key (PrK in Figure 1) with a signer-specific signature algorithm (SA in Figure 1).

3) For each signer, the signature value (DS in Figure 1) and other signer-specific information are collected into a **`SignerInfo`** value, which is defined in RFC 3852 and RFC 5911. Certificates and CRLs for each signer, and those not corresponding to any signer, are collected in this step.

4) The message digest algorithms for all the signers and the **`SignerInfo`** values for all the signers are collected together into a **`SignatureRelatedData`** value.

c) The process by which **`SignatureRelatedData`** is verified involves the following steps, which are illustrated as the right half of Figure 1. A recipient independently computes the message digest (MD' in Figure 1) of the concatenation of the SBH and (possibly encrypted) BDB encodings with the signer-specific message-digest algorithm (DA in Figure 1). This message digest and the signer's public key (PbK in Figure 1) are used to verify the signature value (DS in Figure 1) comparing the message digest calculated by the verifier (MD' in Figure 1) and the message digest calculated by the signer (MD in Figure 1), which is decrypted from the digital signature (DS in Figure 1) using the signer's public key (PbK in Figure 1) with the signer-specific signature algorithm (DS in Figure 1). The signer's public key is referenced either by an issuer distinguished name along with an issuer-specific serial number or by a subject key identifier that uniquely identifies the certificate containing the public key. The signer's certificate can be included in the field **`certificates`** in the **`SignatureRelatedData`**.

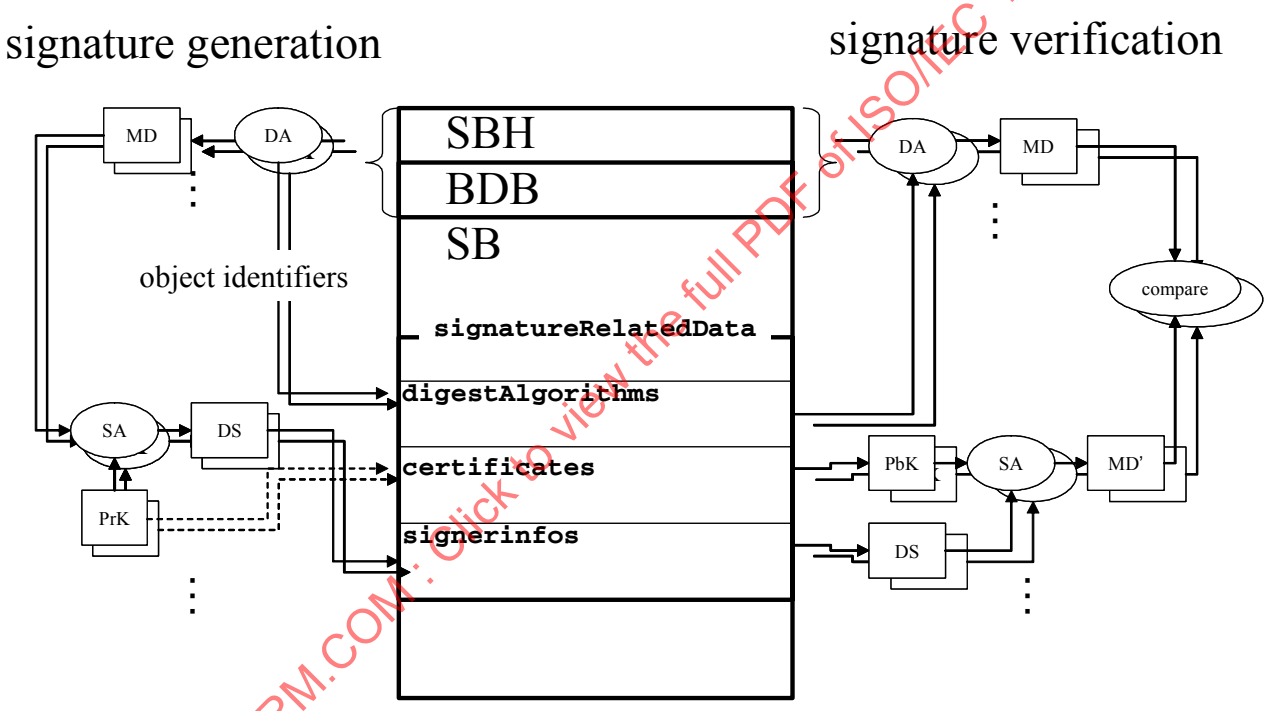

Figure 1 — Processes for signature generation and verification

In Figure 1, the dotted line from the PrK to the **`certificates`** field means the inclusion of the public key certificate corresponding to the private key PrK.

5.8.3.1.2 Type **`SignatureRelatedData`** is defined as follows:

```
SignatureRelatedData ::= SEQUENCE {
        version CBEFFSBVersion DEFAULT v0,
        digestAlgorithms SET OF DigestAlgorithmIdentifier,
        certificates [0] IMPLICIT CertificateSet OPTIONAL,
        crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,
        signerInfos SignerInfos
}
```

a) **version** is the version number of this security block specification of type **CBEFFSBVersion**, as defined in 5.8.2.1.2.

b) The **digestAlgorithms** component takes a value of type **DigestAlgorithmIdentifiers**, which is a collection of message digest algorithm identifiers. Each element identifies the message digest algorithm, along with any associated parameters, used by one or more signers. The collection is intended to list the message digest algorithms employed by all of the signers, in any order. The cryptographic hash algorithm to be supported is not specified in this part of ISO/IEC 19785.

c) **certificates** is a collection of certificates. It is intended that the set of certificates be sufficient to contain certification paths from a recognized "root" or "top-level certification authority" to all of the signers in the **signerInfos** field. There may be more certificates than necessary, and there may be certificates sufficient to contain certification paths from two or more independent top-level certification authorities. There may also be fewer certificates than necessary, if it is expected that recipients have an alternate means of obtaining necessary certificates (e.g., from a previous set of certificates). The signer's certificate may be included.

d) **crls** is a collection of revocation status information. It is intended that the collection contain information sufficient to determine whether the certificates in the **certificates** field are valid, but such correspondence is not necessary. Certificate revocation lists (CRLs) are the primary source of revocation status information. There may be more CRLs than necessary, and there may also be fewer CRLs than necessary.

e) **signerInfos** is a collection of per-signer information. There may be any number of elements in the collection. For details of the **SignerInfo** type, see RFC 3852 and RFC 5911.

5.8.3.2 **authenticationRelatedData** content type

5.8.3.2.1 The **authenticationRelatedData** content type associates an object identifier **id-authenticationRelatedData** with an ASN.1 type **AuthenticationRelatedData** as described in 5.8.1.3.

a) The type **AuthenticationRelatedData** consists of a message authentication code (MAC), and encrypted authentication keys for one or more recipients. The combination of the MAC and one encrypted authentication key for a recipient is necessary for that recipient to verify the integrity of the concatenation of The SBH and (possibly encrypted) BDB encodings. Unlike the **authenticatedData** content type of RFC 3852, this content type does not contain the data to be authenticated.

b) The process by which **AuthenticationRelatedData** is constructed involves the following steps:

1) A message-authentication key for a particular message-authentication algorithm is generated at random.

2) The message-authentication key is encrypted for each recipient. The details of this encryption depend on the key management algorithm used.

3) For each recipient, the encrypted message-authentication key and other recipient-specific information are collected into a **RecipientInfo** value (See RFC 3852 and RFC 5911 for details of the type **RecipientInfo**).

4) Using the message-authentication key, the originator computes a MAC value on the concatenation of The SBH and (possibly encrypted) BDB encodings, and the result becomes the MAC value store in the field **mac**.

5.8.3.2.2 The type **AuthenticationRelatedData** is defined as follows:

```
AuthenticationRelatedData ::= SEQUENCE {
        version CBEFFSBVersion DEFAULT v0,
        originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
        recipientInfos RecipientInfos,
        macAlgorithm MessageAuthenticationCodeAlgorithm,
        mac MessageAuthenticationCode
}
```

a) **`version`** is the syntax version number of the SB.

b) **`originatorInfo`** optionally provides information about the originator. It is present only if required by the key management algorithm. It may contain certificates, attribute certificates, and CRLs.

c) **`recipientInfos`** is a collection of per-recipient information. There shall be at least one element in the collection.

d) **`macAlgorithm`** is a message authentication code (MAC) algorithm identifier. It identifies the MAC algorithm, along with any associated parameters, used by the originator. Placement of the **`macAlgorithm`** field facilitates one-pass processing by the recipient.

e) **`mac`** is the message authentication code.

5.8.4 Encryption and integrity

5.8.4.1 If both the CBEFF_BDB_encryption_options and the CBEFF_BIR_integrity_options in the SBH are present and contain the encoding for ENCRYPTION and INTEGRITY respectively, the SB shall contain elements for encryption and integrity.

5.8.4.2 The process of encryption shall be done before that of integrity:

1) The biometric data (original BDB) is encrypted and placed in the BDB field.

2) Data of type **`EnvelopeRelatedData`** or **`EncryptionRelatedData`** is generated and stored in the SB.

3) One or more digital signatures or MACs are calculated on the concatenation of the SBH and the encrypted BDB.

4) Data of type **`SignatureRelatedData`** or **`AuthenticationRelatedData`** is generated and included in the SB.

5.8.4.3 The verification of integrity shall come before the decryption process, i.e.

1) One or more digital signatures or MACs are obtained from the data of type **`SignatureRelatedData`** or **`AuthenticationRelatedData`** in the SB.

2) The integrity of the concatenation of the SBH and the encrypted BDB be verified by the digital signature or MAC.

3) The information on encryption be obtained from the data of type **`EnvelopeRelatedData`** or **`EncryptionRelatedData`** in the SB.

4) The original BDB be extracted by decrypting the encrypted BDB.

As is seen above, the integrity can be verified without performing the decryption process.

5.9 Encoding of abstract values

The encodings which result in a security block encoding are as follows:

a) The octet encoding of an SBH is determined by the patron format in use.

b) The octet encoding of a BDB is determined by the BDB format specification.

c) The encoding of the **`CBEFFSecurityBlock`** specified in Annex A shall be as specified in 5.5.

6 Security block format: signature only

6.1 Security block format owner

ISO/IEC JTC 1/SC 37

6.2 Security block format owner identifier

257 (0101 Hex). This identifier has been assigned in accordance with ISO/IEC 19785-2 to ISO/IEC JTC 1/SC 37 as a CBEFF biometric organization.

6.3 Security block format name

ISO/IEC JTC1/SC 37 signature-only security block format

6.4 Security block format identifier

4 (0004 Hex). This has been registered in accordance with ISO/IEC 19785-2 when DER encodings (see ISO/IEC 8825-1) are applied.

5 (0005 Hex). This has been registered in accordance with ISO/IEC 19785-2 when canonical PER encodings (see ISO/IEC 8825-2) are applied.

6 (0006 Hex). This has been registered in accordance with ISO/IEC 19785-2 when canonical XER encodings (see ISO/IEC 8825-3) are applied.

6.5 ASN.1 object identifier for this security block format

6.5.1 The case of DER encodings

```
{iso registration-authority cbeff(19785) biometric-organization(0) jtc1-sc37(257) sb-formats(3) signature-only(2) der-encoding(1)}
```

or, in XML value notation,

```
1.1.19785.0.257.3.2.1
```

6.5.2 The case of canonical PER encodings

```
{iso registration-authority cbeff(19785) biometric-organization(0) jtc1-sc37(257) sb-formats(3) signature-only(2) per-encoding(2)}
```

or, in XML value notation,

```
1.1.19785.0.257.3.2.2
```

6.5.3 The case of canonical XER encodings

```
{iso registration-authority cbeff(19785) biometric-organization(0) jtc1-sc37(257) sb-formats(3) signature-only(2) xer-encoding(3)}
```

or, in XML value notation,

```
1.1.19785.0.257.3.2.3
```

6.6 Domain of use

The ISO/IEC JTC 1/SC 37 signature-only security block is designed for applications that always require integrity and never require encryption. It provides a tightly defined implementation of CMS formatted signed data by profiling to remove many options. It does not support the inclusion of ACBio instances, nor does it support multiple signatures.

NOTE This format is the same as that required for NIST PIV conformant smart card implementations [4].

6.7 Version identifier

This security block format specification has a version identifier of (major 0, minor 0).

6.8 Format specification and conformance statement

The signature-only security block shall be the encoding specified in RFC 3852 or the ASN.1 data type **`SignedData`** specified in that RFC.

NOTE RFC 3852 specifies the use of ISO 8825-1 Distinguished Encoding Rules (DER) for the encoding of **`SignedData`**.

The digital signature shall be computed over the entire CBEFF structure except the signature-only security block itself (which means that it includes the SBH and the BDB).

It shall conform to the following constraints which relate to types defined in RFC 3852:

- The **`CMSVersion`** shall be **`v3`**
- The **`encapcontentInfo`** shall omit the **`eContent`** field
- The **`certificates`** field shall include zero or a single **`certificate`** (depending on application requirements) which can be used to verify the **`signature`** in the **`SignerInfo`** field
- The **`crls`** field shall be omitted
- **`signerInfos`** shall be present and shall include only a single **`SignerInfo`**
- The **`SignerInfo`** shall
  - Use the **`issuerAndSerialNumber`** choice for **`SignerIdentifier`**
  - Include at a minimum a **`MessageDigest`** attribute containing the hash of the concatenated SBH + BDB

Annex A
(normative)

ASN.1 module for security block format

This ASN.1 module has been syntactically checked without errors by an ASN.1 tool.

```
CBEFF-GENERAL-PURPOSE-SECURITY-BLOCK
        {iso(1) standard(0) cbeff(19785) module(0) sb(16) rev(0)}

DEFINITIONS AUTOMATIC TAGS ::= BEGIN
IMPORTS

-- RFC 5911 ASN.1 Module for RFC 3852 Cryptographic Message Syntax
        ContentEncryptionAlgorithmIdentifier,
        SignerInfos, MessageAuthenticationCodeAlgorithm,
        DigestAlgorithmIdentifier, AuthAttributes, MessageAuthenticationCode,
        OriginatorInfo, RecipientInfos
        FROM CryptographicMessageSyntax2004 {
            iso(1) member-body(2) us(840) rsadsi(113549)
            pkcs(1) pkcs-9(9) smime(16) modules(0) cms-2004(24)}

-- ISO/IEC 24761 Authentication context for biometrics
        ACBioInstance, CertificateSet, RevocationInfoChoices
        FROM AuthenticationContextForBiometrics {
            iso(1) standard(0) acbio(24761) module(1) acbio(2) rev(0)} ;

CONTENT-TYPE ::= TYPE-IDENTIFIER

CBEFFSecurityBlock ::= SEQUENCE OF CBEFFSecurityBlockElement

CBEFFSecurityBlockElement ::= CHOICE {
        elementCBEFFSB ContentInfoCBEFFSB,
        subBlockForACBio SubBlockForACBio,
        accumulatedACBioInstances ACBioInstances
}

ContentInfoCBEFFSB ::= SEQUENCE {
        contentType CONTENT-TYPE.&id({ContentTypeCBEFF}),
        content [0] EXPLICIT CONTENT-TYPE.&Type
            ({ContentTypeCBEFF}{@contentType})
        }

ContentTypeCBEFF CONTENT-TYPE ::= { envelopeRelatedData | encryptionRelatedData |
                    signatureRelatedData | authenticationRelatedData}

EnvelopeRelatedData::= SEQUENCE {
        version CBEFFSBVersion DEFAULT v0,
        originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
        recipientInfos RecipientInfos,
        contentEncryptionAlgorithm ContentEncryptionAlgorithmIdentifier
}

CBEFFSBVersion ::= INTEGER { v0(0) } ( v0, ... )

EncryptionRelatedData ::= SEQUENCE {
        version CBEFFSBVersion  DEFAULT v0,
        contentEncryptionAlgorithm ContentEncryptionAlgorithmIdentifier
}

SignatureRelatedData ::= SEQUENCE {
        version CBEFFSBVersion DEFAULT v0,
        digestAlgorithms SET OF DigestAlgorithmIdentifier,
        certificates [0] IMPLICIT CertificateSet OPTIONAL,
        crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,
        signerInfos SignerInfos
}
```

```
AuthenticationRelatedData ::= SEQUENCE {
        version CBEFFSBVersion DEFAULT v0,
        originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
        recipientInfos RecipientInfos,
        macAlgorithm MessageAuthenticationCodeAlgorithm,
        mac MessageAuthenticationCode
}

SubBlockForACBio ::= SEQUENCE {
        bpuIOIndex INTEGER,
        acbioInstance ACBioInstance
}

ACBioInstances ::= SEQUENCE OF ACBioInstance

-- contentType object identifiers
id-envelopeRelatedData OBJECT IDENTIFIER ::= {
        iso(1) standard(0) cbeff(19785) contentType(1) envelopeRelatedData(1)
}

id-encryptionRelatedData OBJECT IDENTIFIER ::= {
        iso(1) standard(0) cbeff(19785) contentType(1) encryptionRelatedData(2)
}

id-signatureRelatedData OBJECT IDENTIFIER ::= {
        iso(1) standard(0) cbeff(19785) contentType(1) signatureRelatedData(3)
}

id-authenticationRelatedData OBJECT IDENTIFIER ::= {
        iso(1) standard(0) cbeff(19785) contentType(1) authenticationRelatedData(4)
}

-- ContentType objects

envelopeRelatedData CONTENT-TYPE ::= {
        EnvelopeRelatedData
        IDENTIFIED BY id-envelopeRelatedData
}

encryptionRelatedData CONTENT-TYPE ::= {
        EncryptionRelatedData
        IDENTIFIED BY id-encryptionRelatedData
}

signatureRelatedData CONTENT-TYPE ::= {
        SignatureRelatedData
        IDENTIFIED BY id-signatureRelatedData
}

authenticationRelatedData CONTENT-TYPE ::= {
        AuthenticationRelatedData
        IDENTIFIED BY id-authenticationRelatedData
}


END -- CBEFF-SECURITY-BLOCK
```