ISO/IEC JTC 1/SC 27 N 4834

Date: 2006-02-15

ISO/IEC WD 24761.2

ISO/IEC JTC 1/SC 27/WG 2

Secretariat: DIN

Information technology — Security techniques — Authentication context for biometrics

Technologies de l'information — Techniques de sécurité — Élément complémentaire

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International Standard
Document subtype:
Document stage: (20) Preparatory
Document language: E

Contents

Page

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 24761 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

Introduction

Result of biometric verification is dependent on the quality of the device used. The better quality device is used, the finer result we get. In the Internet environment, verifier of the biometric verification may not know the quality of the biometric device that was used. If the verifier knows the information such as that of the biometric device, the verifier can do a better decision. We specify Authentication context for biometrics (ACBio) in this International Standard to give a solution to the above issue by informing both the static (independent of the real-time execution) and real-time information of the biometric verification to the verifier.

In general, a biometric verification consists of the following five subprocesses: data capture, signal processing, storage, comparison, and decision. ACBio is designed to be applied to five-subprocess-model but is also applicable to other biometric verification models.

ACBio is a data format for the data generated by BPUs, such as sensor, smartcard, and comparison device, to give enough information with which the verifier can verify the validity of the biometric verification.

ACBio instance also includes the information to certify the validity of the following :

(a) ACBio instance is generated by the BPU requested from the verifier.

(b) ACBio instance(s) related to the target biometric verification process interrelates correctly.

(c) ACBio instance keeps its integrity.

ACBio is designed with privacy issue in mind. That is, ACBio is defined so that the verifier can verify the validity of biometric verification process without receiving privacy data such as biometric sample and biometric template.

Information technology — Security techniques — Authentication context for biometrics

1 Scope

This International Standard defines the structure and the data elements of Authentication context for biometrics (ACBio). The structure is designed based on:

(1) the five-subprocess-model of biometric verification which consists of data capture, signal processing, storage, comparison, and decision.

(2) the concept of biometric process unit, the subject that executes subprocess(es) of a biometric verification with a uniform level of security performance.

This International Standard specifies cryptographic syntax that can be used to give biometric enrolment and processing context information to the verifier of biometric verification. The cryptographic syntax is based on an abstract Cryptographic Message Syntax (CMS) schema whose concrete values can be represented using either a compact binary encoding or a human-readable XML markup.

Cryptographic messages are represented using the XML Encoding Rules (XER), or the Basic Encoding Rules (BER) of ASN.1 commonly supported by cryptographic tool kit vendors. The syntax is algorithm independent and supports provision of data integrity and data origin authentication. The cryptographic algorithms specified by ISO/IEC JTC 1/SC27 IT Security Techniques are recommended, though any algorithm appropriate for use by a given community may be used.

This International Standard does not define protocols intercommunicated among all the subject such as biometric process units, claimant, and verifier. Some existing standards that are available for the data elements are specified.

2 Normative references

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

Data format

ISO/IEC FDIS 19785-1: 2005, Information Technology - Common Biometric Exchange Formats Framework – Part 1: Data Element Specification

ISO/IEC FCD 19785-3: 2006, Information Technology - Common Biometric Exchange Formats Framework – Part 3: Patron Format Specifications

Evaluation of biometric system

ISO/IEC CD 19792: 2006, Information technology -- Security techniques -- Security evaluation of biometrics

Evaluation of tamper resistance

ISO/IEC 19790:2006, Information technology -- Security techniques -- Security requirements for cryptographic modules

Evaluation of performance, accuracy and quality of biometric system and device

ISO/IEC 19795 (all parts), Information Technology - Biometric Performance Testing and Reporting

Quality of biometric sample

ISO/IEC WD 29794-1, Information Technology – Biometric Sample Quality Standard — Part 1: Framework

Others

ISO/IEC JTC1/SC 37 Standing Document 2 - Harmonized Biometric Vocabulary

ISO/IEC FDIS 19784-1:2005, Information technology - Biometric application programming interface - Part 1: BioAPI specification

ISO/IEC FCD 24713-1:2005, Biometrics - Biometric Profiles for Interoperability and Data Interchange - Part 1: Biometric Reference Architecture

ISO DIS 19092-2:2005, Financial Services - Biometrics - Part 2: Message syntax and cryptographic requirements

3 Terms and definitions

For the purposes of this International Standard, the following terms and definitions apply.

3.1 biometric
of or having to do with biometrics

3.2 biometrics
automated recognition of individuals based on their behavioural and biological characteristics

3.3 biometric data
biometric sample at any stage of processing, biometric reference, biometric feature or biometric property

3.4 biometric feature
concise representation of information extracted from an acquired or intermediate biometric sample by applying a mathematical transformation

3.5 biometric model
biometric reference consisting of a stored function (dependent on the individual) generated from a biometric sample(s)

3.6 biometric property
descriptive attributes of the individual estimated or derived from the biometric sample

3.7 biometric reference
one or more stored biometric samples or biometric models attributed to an individual and used for comparison

3.8 biometric sample
data obtained from a biometric device, either directly or after processing

3.9 acquired biometric sample
analog or digital representation of biometric characteristics directly taken from an individual by a sensor

3.10 intermediate biometric sample
biometric sample that is obtained by modifying an acquired biometric sample to allow better feature extraction, and that is not suitable as yet for automated matching in the biometric system under consideration

3.11 processed biometric sample
biometric sample comprising biometric features derived from an acquired or intermediate biometric sample, and suitable for automated matching

3.12 post-processed biometric sample
manipulated biometric features

3.13 biometric template
biometric reference consisting of a set of stored biometric features comparable directly to biometric features of a presented biometric sample using a function not dependent on an individual

3.14 acquired biometric template
acquired biometric sample stored as a biometric template

3.15 intermediate biometric template
intermediate biometric sample stored as a biometric template

3.16 processed biometric template
processed biometric sample stored as a biometric template

3.17 post-processed biometric template
post-processed biometric sample stored as a biometric template

3.18 comparison result
value of “match”, “non-match” or possibly “undetermined” resulting from a decision based on a comparison score, a decision policy including threshold, and possibly other inputs

3.19 comparison score
numerical value (or set of values) resulting from a comparison

3.20 one-to-one comparison
process in which a biometric sample set from one individual is compared to biometric reference(s) from one individual to produce a comparison score, perhaps using additional data from the enrolment database

3.21 verification (biometric system function)
biometric system function that performs a one-to-one comparison

3.22 enrolment
process of creating and storing, for an individual, a data record containing biometric and, typically, non-biometric data

3.23 biometric application decision
making a conclusion or resolution based on the application decision policy after consideration of one or more comparison results, comparison scores and possibly other non-biometric data

3.24 comparison
estimation, calculation or measurement of similarity or dissimilarity between biometric sample(s) and biometric reference(s)

3.25 on-card matching
comparison done on IC card

NOTE – The term "matching" is deprecated and replaced with the term "comparison" in ISO/IEC JTC 1/SC 37 SD 2. But the term "on-card matching" is considered to be a tem of the field of ISO/IEC JTC 1/SC 17, and is used in the subcommittee. Therefore we use this term in this International Standard.

3.26 claimant (of biometric verification)
object of biometric verification

3.27 verifier (of biometric verification)
subject of biometric application decision

3.28 biometric process unit (BPU)
subject that executes subprocess(es) of a biometric verification process with a uniform level of security performance

3.29 biometric process unit certificate (BPU certificate)
X.509 certificate issued to a biometric process unit

3.30 BPU certification organization
organization in which BPU certificate is issued

3.31 biometric process unit report (BPU report)
report on a biometric process unit, which consists of BPU function report and BPU security report

3.32 biometric process unit function report (BPU function report)
report on the function of a biometric process unit, which contains evaluation reports(s) on function

3.33 biometric process unit security report (BPU security report)
report on the security of a biometric process unit, which contains evaluation reports(s) on security

3.34 biometric template certificate (BT certificate)
certificate issued to a biometric template by an enrolment organization with which the biometric process verifier can verify the authenticity of the biometric template

3.35 evaluation organization
organization which evaluates biometric process units on function or security

3.36 enrolment organization
organization to which biometric templates are created, enrolled, and certified

3.37 authentication context for biometrics (ACBio)
data format of information generated by a biometric process unit for the verifier of the biometric verification to show the validity of the execution of subprocess(es) in the biometric process unit

3.38 ACBio instance
data generated by a biometric process unit of the ACBio format

4 Symbols (and abbreviated terms)

4.1 ACBio
Authentication Context for Biometrics

4.2 ASN.1
Abstract Syntax Notation One

4.3 BER
Basic Encoding Rules

4.4 BIR
Biometric Information Record

4.5 BPU
Biometric Process Unit

4.6 CMS
Cryptographic Message Syntax

4.7 DER
Distinguished Encoding Rules

4.8 OCM
On-Card Matching

4.9 PKI
Public Key Infrastructure

4.10 SOC
System On Card

4.11 STOC
STore On Card

4.12 UUID
Universal Unique Identifier

4.13 XER
XML Encoding Rules of ASN.1

4.14 XML
Extensible Markup Language

5 Model, framework, and requirements

5.1 Biometric verification process model

The specification of ACBio is based a biometric verification process which consists of the following five subprocesses:

a) data capture

This subprocess captures the biometric information from the claimant and converts the information to the acquired biometric sample. The acquired biometric sample is transmitted to the signal processing subprocess.

b) signal processing

This subprocess receives the acquired biometric sample from the data capture subprocess, transforms the acquired biometric sample into the processed biometric sample of the form required by the comparison subprocess. The processed biometric sample is transmitted to the comparison subprocess.

c) storage

This subprocess maintains the biometric template, of the form of the acquired biometric sample or of the processed biometric sample, for the claimant. The biometric template is transmitted to the signal processing subprocess or the comparison subprocess respectively.

d) comparison

This subprocess receives the processed biometric sample generated by the signal processing subprocess and the biometric template from the storage subprocess or from the signal processing subprocess which processed the acquired biometric template originally maintained in the storage, compares the two data, and scores the similarity of the data, which is called comparison score. The comparison score is transmitted to the decision subprocess.

e) decision

This subprocess receives the comparison score from the comparison subprocess, evaluates the score under certain rules, decides the validity of the claimant's identity, and outputs the resulting binary match/no-match to the verifier.

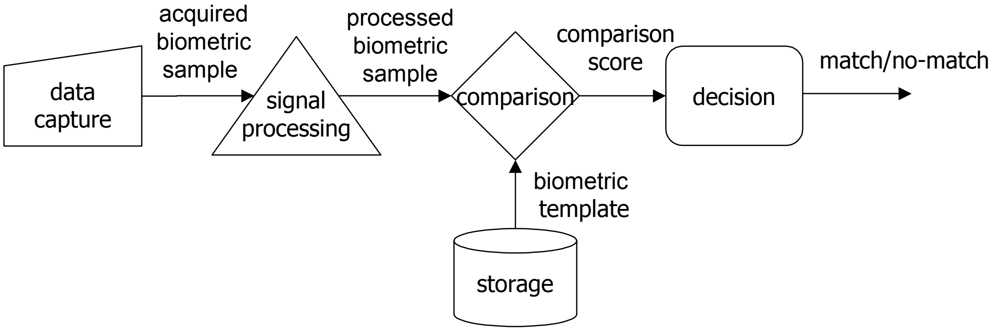

Figure 1- Biometric verification process model

One or more subprocesses are executed on a biometric process unit.

One biometric verification process is accomplished by one or more biometric process units.

NOTE - A biometric process unit is an abstract concept of a security domain such as a sensor, a smart card, a comparison device, and a software running on a personal computer.

5.2 Framework and requirements for the use of ACBio

5.2.1 Preparation for use of ACBio

To use biometric verification, one needs certain preparation such as enrolment of biometric templates of claimants (users). To verify a biometric verification processes with ACBio, additional preparation are necessary because ACBio needs to contain static information (independent of the real-time execution) on the BPU and on the biometric template stored in BPU.

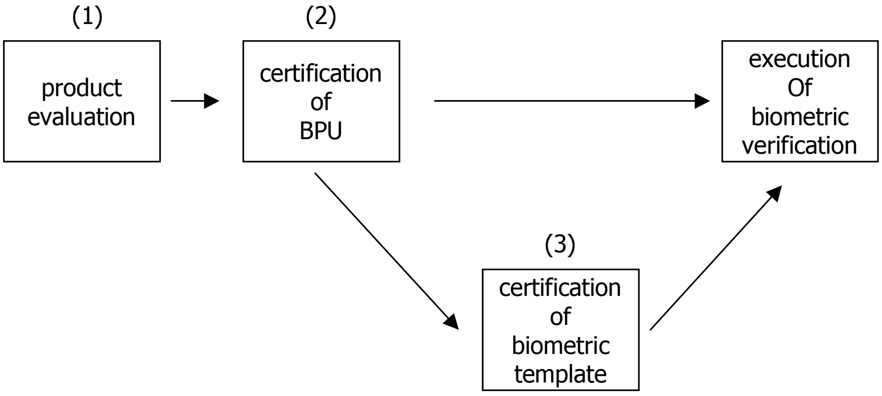
(1)
product evaluation
(2)
certification of BPU
execution Of biometric verification
(3)
certification of biometric template

Figure 2 - Preparation for use of ACBio

A series of preparation for the use of ACBio is drawn in Figure2. Static information on the BPU such as specification and quality of the BPU should be certified (2) after evaluation of the BPU (1).

ACBio requires that each BPU obtains its BPU report from the evaluation organization after the evaluation. The BPU report contains BPU function report and BPU security report. The former includes the information on the functional specification and function quality, such as the accuracy of the BPU, the quality of the data generated by the BPU. The latter includes the information on security (tamper-resistance) of the BPU.

A BPU may store its BPU report in itself or contain the referrer to the BPU report, such as URI, in itself.

ACBio also requires that each BPU obtains its X.509 certificate (BPU certificate) from a BPU certification organization

A BPU may store its BPU certificate in itself or contain the referrer to the BPU certificate, such as URI, in itself.

Static information on a biometric template should be certified by a certain enrolment organization (3) on the enrolment of the biometric template.

ACBio requires that a biometric template is enrolled to a certain organization and obtains its BT certificate. BT certificate includes information on the biometric template such as the enrolment organization, the validity period, and information on the biometric device(s) used on the enrolment.

A BPU with the function of storage may store the BT certificate in itself or contain the referrer to the BT certificate, such as URI, in itself.

5.2.2 Generation and transmission of ACBio instance(s)

To each BPU, an ACBio instance is generated and transmitted to the verifier so that the verifier can verify the validity of the subprocesses executed on the BPU. ACBio requires that each BPU has means to generate a digital signature with which the verifier can verify the integrity of the ACBio instance.

5.2.3 Biometric application decision using ACBio

The verifier verifies biometric verification process by checking information given by ACBio instance. To check an ACBio instance, the verifier needs to verify the BPU certificate, BPU report, and BT certificate in the ACBio instances, connecting to certain relevant organizations such as the BPU certification organization, the evaluation organization, and the enrolment organization. The following Figure3 draws the relation between the verifier and the relevant sites.

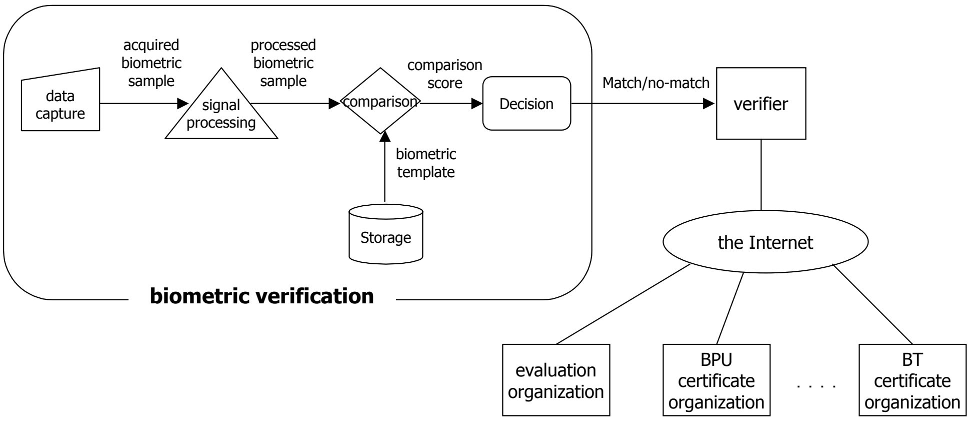

Figure 3- Relation between the verifier and relevant sites

6 Structure of ACBio

The definition of ACBio is a `SignedData` of the data of the type `ACBioContentInformation`, whose structure is drawn in Table 1. The signature shall be generated with the private key of the BPU. The `ACBioContentInformation` consists of version and three blocks. Each of three blocks is specified in 6.1 to 6.3. The definition for data elements in `BPUInformation` is specified in section 7.

Table 1 - Content of ACBio to be signed

ACBioContentInformation		
	Version	
	BPU Information Block	
		BPU Certificate Referrer Information
		BPU Report Information
		BT Certificate Information
	Verifier Controlling Block	
		Verifier Controlling Value
	Biometric Process Block	
		Input Information[1]
		. . .
		Input Information[N]
		Output Information[1]
		. . .
		Output Information[N]

In ASN.1 notation, the structure of ACBio is described as the following:

```
AuthenticationContextForBiometrics ::= SignedData { EncodedACBioContextInformation }


EncodedACBioContextInformation::= ENCODED-VALUE-OF.&Type(EACBioContextInformation)


ACBioContextInformation::= SEQUENCE {
   version            Version DEFAULT v0,
   bpuInformation     BPUInformation,
   verifierControl    VerifierControl,
   biometricProcess   BiometricProcess
}


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


EACBioContextInformation  ::= OCTET STRING( CONTAINING ACBioContextInformation )
```

6.1 BPU Information Block

This block is designed for static information of BPU, determined in advance and independent of the real-time execution. Data about the BPU, such as its function, security facility or/and tamper resistance, quality of the function implemented on the BPU, can be referred directly or indirectly with the information in this block. Definition for ACBio data elements BPU Report and BT Certificate is specified in section 7.

In ASN.1 notation, the structure of BPU Information Block is described as the following:

```
BPUInformation ::= SEQUENCE {
  bpuCertificateReferrerInformation     BPUCertificateReferrerInformation OPTIONAL,
  bpuReportInformation     BPUReportInformation,
  btCertificateInformation BTCertificateInformation
}
```

6.2 Verifier Controlling Block

This block is designed for information needed to distinguish whether the ACBio instance is generated by the verifier's requested or not. Data such as challenge generated by the verifier are put in this block.

In ASN.1 notation, the structure of Verifier Controlling Block is described as the following:

```
VerifierControl ::= SEQUENCE {
   controlValue  ControlValue
}
ControlValue ::= OCTET STRING
```

6.3 Biometric Process Block

This block is designed for information related to the real-time execution of the subprocesses on BPU.

This block includes information on the input/output data processed in the BPU. If the BPU sends/receives a data to/from other BPU, then the corresponding data element in this block is mandatory.

In ASN.1 notation, the structure of Biometric Process Block is described as the following:

```
BiometricProcess ::= SEQUENCE {
   inputInformationList     InputInformationList,
   outputInformationList     OutputInformationList
}


InputInformationList ::= SEQUENCE OF InputOutputInformation
OutputInformationList ::= SEQUENCE OF InputOutputInformation
```

`InputInformationList`/`OutputInformationList` has as many components as the inputs/outputs which the BPU has. For example, if a BPU consists of data capture and signal processing subprocesses, `InputInformationList` has no element and `OutputInformationList` has one element. For the case of BPU with only comparison subprocess, `InputInformationList` has two elements and `OutputInformationList` has one element.

`InputOutputInformation` consists of two components, type of data and the hash value of the data:

```
InputOutputInformation ::= SEQUENCE {
   typeData       TypeData,
   hashValue       Hash
}
```

The type `TypeData` is defined in 7.2.1.1.3.

The definition of `Hash` is a pair of the identifier of the hash algorithm and the hash value:

```
 Hash ::= SEQUENCE {
    digestAlgorithm    DigestAlgorithmIdentifier,
    hashValue           OCTET STRING
 }
```

7 Data elements and their types of BPU information block

7.1 BPU certificate

BPU certificate is X.509 certificate for the public key of BPU. The structure of X.509 certificate is described in Table 2.

Table 2 – Fields of X.509 Certificate

field		content
tbsCertificate	Version	as ordinary
	serialNumber	as ordinary
	signature	as ordinary
	validity	as ordinary
	issuer	a trusted third party or a public CA in the vendor which produces/ sells the product of the entity
	subject	identifier of the subject including the information such as the serial number of the product, the name of the product of the entity, and the name of the vendor of the product
	subjectPublicKeyInfo	as ordinary
	issuerUniqueID	as ordinary
	subjectUniqueID	as ordinary
	extensions	
signatureAlgorithm		as ordinary
signatureValue		as ordinary

The basic part of X.509 certificate consists of nine fields; Version, serialNumber, signature, validity, issuer, subject, subjectPublicKeyInfo, issuerUniqueID, and subjectUniqueID. Here the issuer is a public CA in the vendor which produces/sells the product of the BPU. The subject field is the identifier whose description is subject to X.500 and shall include the serial number of the product, the name of the product of the BPU, and the name of the vendor of the product. The serial number of the product in the subject field shall be the leaf entry of the identifier. The product name, also in the subject field, shall be the entry next to the leaf. Other seven attributes in the basic field are used as ordinary.

The BPU certificate shall be either in the `certificates` in `SignedData` of ACBio or referred from `bpuCertificateReferrer` in `BPUCertificateReferrerInformation`. In the former case, `bpuCertificateReferrerInformation` in `BPUInformation` may be omitted. In the case in which the renewal of X.509 certificate is difficult, such as the case of a BPU of biometric sensor, **`bpuCertificateReferrerInformation`** should be used.

`crlsReferrer` is optional.

```
BPUCertificateReferrerInformation ::= SEQUENCE {
   bpuCertificateReferrer     URI,
   crlsReferrer     URI OPTIONAL
}
```

7.2 BPU report

BPU report is a report which describes the specification and evaluation of the function and security of BPU. In **`BPUReportInformation`**, BPU report is stored directly or referred to as URI as the following:

```
BPUReportInformation ::= CHOICE {
   bpuReport       BPUReport,
   bpuReportReferrer     URI
}
```

BPU report consists of two components, **`bpuFunctionReport`** and **`bpuSecurityReport`**.

```
BPUReport ::= SEQUENCE {
   bpuFunctionReport     BPUFunctionReport,
   bpuSecurityReport     BPUSecurityReport
}
```

The overview of the structure of BPU report is described in Table 3. Each components of BPU report is specified in 7.2.1 to 7.2.2.

Table 3 – Structure of BPU report

BPUReport						
	SignedData					
		BPUFunctionReportInformation				
			BPUFunctionReportSubprocess[1]			
				BPUFunctionReportSubprocess		
					DefinitionFunction	
						NameFunction
						DescriptionFunction
					QualityEvaluation	
						QualityBiometricProcess
						QualityExtension
			. . .			
			BPUFunctionReportSubprocess[N]			
				BPUFunctionReportSubprocess		
					DefinitionFunction	
						NameFunction
						DescriptionFunction
					QualityEvaluation	
						QualityBiometricProcess
						QualityExtension
			Modality			
			InputType1			
			InputType2			
			OutputType			
	BPUSecurityReport					
		SecLevelCryptoModule				
		SecBiometricProcess				
		SecExtension				

7.2.1 `BPUFunctionReport`

In ASN.1 notation, the definition of `BPUFunctionReport` is described as the following:

```
 BPUFunctionReport ::= SignedData { EncodedBPUFunctionReportInformation }


 EncodedBPUFunctionReportInformation ::=
  ENCODED-VALUE-OF.&Type( EBPUFunctionReportInformation )
```

The type `BPUFunctionReport` is `SignedData` of the data of the type `BPUFunctionReportInformation`, with the signature generated using the private key of the vendor of the product of the BPU.

7.2.1.1 `BPUFunctionReportInformation`

The components of `BPUFunctionReportInformation` are as the following:

```
 BPUFunctionReportInformation ::= SEQUENCE {
    bpuFunctionReportSubprocesses   BPUFunctionReportSubprocesses,
    modality                Modality,
    inputType1            TypeData OPTIONAL, -- if data capture, none
```

```
   inputType2          TypeData OPTIONAL,  -- unless comparison, none
   outputType          TypeData
}

BPUFunctionReportSubprocesses ::= SEQUENCE OF BPUFunctionReportSubprocess
```

`bpuFunctionReportSubprocesses` describes the specification and evaluation of the subprocesses contained in the BPU.

`bpuFunctionReportSubprocesses` contains elements of type `BPUFunctionReportSubprocess` as many as the number of the subprocesses implemented on the BPU.

`modality` indicates the modality of the biometric data which the BPU processes.

`inputType1` indicates the type of input data to the BPU. If the BPU contains the data capture subprocess and does not contain the comparison subprocess, `BPUFunctionReportInformation` shall contain the component `inputType1`.

If the BPU contains the comparison subprocess and does not contain either the signal processing subprocess or the storage subprocess, then `BPUFunctionReportInformation` shall contain the component `inputType2`.

`outputType` indicates the type of output data from the BPU.

7.2.1.1.1 BPUFunctionReportSubprocess

The type `BPUFunctionReportSubprocess` contains information about the function of the subprocess and the evaluation report to the subprocess:

```
BPUFunctionReportSubprocess ::= SEQUENCE {
   function          DefinitionFunction,
   qualityEvaluation    QualityEvaluation OPTIONAL
}
```

7.2.1.1.1.1 DefinitionFunction

The type `DefinitionFunction` gives information of the subprocess, with the name of the function of the subprocess `NameFunction` and the detail of the function of the subprocess `DescriptionFunction`:

```
DefinitionFunction ::= SEQUENCE {
   functionName          NameFunction,
   functionDescription   DescriptionFunction OPTIONAL
}
```

The type `NameFunction` indicates function of the subprocess:

```
NameFunction ::= ENUMERATED {
   data-capture(1),
   signal-processing(2),
   storage(3),
   comparison(4),
    decision(5),
  ...
  }
```

The type `DescriptionFunction` describes the detail of the function of the subprocess:

```
  DescriptionFunction ::= CHOICE {
    descriptionDataCapture   DescriptionDataCapture,
    descriptionSignalProcessing  DescriptionSignalProcessing,
    descriptionStorage  DescriptionStorage,
    descriptionComparison  DescriptionComparison,
    descriptionDecision  DescriptionDecision,
  ...
  }
```

The definition of description of functions of subprocessess is as the following:

/ Editor's Note: The following shall be defined after the discussion with SC 37. **/**

```
  DescriptionDataCapture ::= OCTET STRING (SIZE(1..MAX))
  DescriptionSignalProcessing ::= OCTET STRING (SIZE(1..MAX))
  DescriptionStorage ::= OCTET STRING (SIZE(1..MAX))
  DescriptionComparison ::= OCTET STRING (SIZE(1..MAX))
  DescriptionDecision ::= OCTET STRING (SIZE(1..MAX))
```

7.2.1.1.1.2 QualityEvaluation

The type `QualityEvaluation` consists of components of the type `QualityBiometricProcess` and the type `QualityExtension`. The former is the evaluation report of the subprocess by a certain evaluation organization. The latter is for extension.

```
  QualityEvaluation ::= SEQUENCE {
    qualityBiometricProcess    QualityBiometricProcess   OPTIONAL,
    qualityExtension           QualityExtension  OPTIONAL
  }
```

/ Editor's Note: The `QualityBiometricProcess` shall be defined after ISO/IEC 19795 which is under standardization in SC 37. **/**

```
  QualityBiometricProcess ::= OCTET STRING (SIZE(1..MAX))
  QualityExtension ::= OCTET STRING (SIZE(1..MAX))
```

7.2.1.1.2 Modality

The type `Modality` indicates the modality which the BPU processes. This type is defined as the following;

```
Modality ::= SEQUENCE {
   bioTypeACBio   BioTypeACBio,
   bioSubtypeACBio    BioSubtypeACBio  OPTIONAL
}
```

Two types are defined as the following. In either definition, the former is from ISO/IEC 19785-3, and the latter from ISO 19092-2. The choice shall be determined on the purpose of the application which uses this International Standard.

```
BioTypeACBio ::= CHOICE {
   bioType37          BiometricType,
   bioType68          BiometricTypes,
   ..
}

BioSubtypeACBio ::= CHOICE {
   bioSubtype37       BiometricSubtype,
   bioSubtype68       Subtypes,
   ..
}
```

7.2.1.1.3 TypeData

The type `TypeData` indicates the type of the data which the BPU handles as its input/output data:

```
TypeData ::= ENUMERATED {
   acquired-biometric-sample(1),
   intermediate-biometric-sample(2),
   processed-biometric-sample(3),
   post-processed-biometric-sample(4),
   acquired-biometric-template(5),
   intermediate-biometric-template(6),
   processed-biometric-template(7),
   post-processed-biometric-template(8),
   comparison-score(9),
   comparison-result(10),
...
}
```

For example, the `TypeData` of the output data of a BPU of STOC card which stores a biometric template already processed by a signal processing subprocess is `processed-biometric-template`.

7.2.2 BPUSecurityReport

The type **BPUSecurityReport** contains three components, namely **secLevelCryptoModule**, **secBiometricProcess**, and **secExtension**. The last component is for extension.

```
BPUSecurityReport ::= SEQUENCE {
   secLevelCryptoModule   SecLevelCryptoModule  OPTIONAL,
   secBiometricProcess    SecBiometricProcess  OPTIONAL,
   secExtension          SecExtension  OPTIONAL
}
```

Criteria for evaluation on security function is specified in ISO/IEC 19790. ISO/IEC 19790 should be applied to **SecLevelCryptoModule**.

```
SecLevelCryptoModule ::= OCTET STRING (SIZE(1..MAX))
```

Criteria for evaluation on security of biometric process is to be specified in ISO/IEC 19792. ISO/IEC 19792 should be applied to **SecBiometricProcess**.

```
SecBiometricProcess ::= OCTET STRING (SIZE(1..MAX))
```

```
SecExtension ::= OCTET STRING (SIZE(1..MAX))
```

7.3 BT certificate

BT certificate shall be contained in the component **btCertificate** of the type **BTCertificateInformation** or referred from the **btCertificateReferrer**.

```
BTCertificateInformation ::= CHOICE {
   btCertificate          BiometricTemplateCertificate,
   btCertificateReferrer     URI
}
```

The type **BiometricTemplateCertificate** for BT certificate is defined as:

```
BiometricTemplateCertificate ::= SignedData { EncodedTemplateCertificateInformation }

EncodedTemplateCertificateInformation ::=
 ENCODED-VALUE-OF.&Type(ETemplateCertificateInformation)

ETemplateCertificateInformation ::=
 OCTET STRING( CONTAINING TemplateCertificateInformation)
```

The type **BiometricTemplateCertificate** is **SignedData** of the data of the type **TemplateCertificateInformation** whose content describes the enrolment event that created the biometric template, with the signature generated using the private key of the enrolment organization.

Since only the hash of the biometric template is included in the **BiometricTemplateCertificate**, the BT certificate can be made public without revealing the biometric template information.

The type **TemplateCertificateInformation** is defined as:

```
TemplateCertificateInformation ::= SEQUENCE {
   version           Version DEFAULT v0,
   hashBTC           Hash,
   indexBTC          IndexBTC,
   validityPeriodBTC ValidityPeriodBTC,
   enrolmentOrgBTC   EnrolmentOrgBTC,
   dataQuality       DataQuality,
   acbioPartials     AuthenticationContextForBiometricsPartials  OPTIONAL
}


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


DataQuality ::= OCTET STRING (SIZE(1..MAX))
```

/ Editor's Note: The DataQuality shall be defined after ISO/IEC 29794 which is under standardization in SC 37. **/**

The **version** component is the version of the **TemplateCertificateInformation** schema.

The component **hashBTC** is the hash value of the biometric template for which BT certificate is issued.

The components **indexBTC**, **validityPeriodBTC**, and **enrolmentOrgBTC** are of types **IndexBTC**, **ValidityPeriodBTC**, and **EnrolmentOrgBTC** respectively. Three types are defined as the following. In either definition, the former is from ISO/IEC 19785-3, and the latter from ISO 19092-2. The choice shall be determined on the purpose of the application which uses this International Standard.

```
IndexBTC ::= CHOICE {
   index37        Index,   -- ISO/IEC 19785-3
   index68        UUID,  -- ISO 19092-2
   ..
}


ValidityPeriodBTC ::= CHOICE {
   validityPeriod37        BDBValidityPeriod,   -- ISO/IEC 19785-3
   validityPeriod68        ValidityPeriod,  -- ISO 19092-2
   ..
}


EnrolmentOrgBTC ::= CHOICE {
   enrolmentOrg37        Creator,   -- ISO/IEC 19785-3
   enrolmentOrg68        Enroler,  -- ISO 19092-2
   ..
}
```

The `dataQuality` comoponent is of type `DataQuality` which contains information of the quality of the biometric template. ISO/IEC 29794 should be applied to `DataQuality`.

The `acbioPartials` component of type `AuthenticationContextForBiometricsPartials` is the ACBio instance which informs of the enrolment context to show how the data capture/signal processing subprocess was done on the enrolment of the biometric template.

7.3.1 AuthenticationContextForBiometricsPartials

The type `AuthenticationContextForBiometricsPartials` has components of the type `AuthenticationContextForBiometricsPartial`, as many as the number of BPUs used on enrolment to create the biometric template. `AuthenticationContextForBiometricsPartial` is defined as `SignedData` with the object to be signed of type `ACBioContextInformationPartial` as the following. The private key used to sign shall be the BPU's private key.

```
  AuthenticationContextForBiometricsPartials ::=
   SEQUENCE OF AuthenticationContextForBiometricsPartial


  AuthenticationContextForBiometricsPartial ::=
   SignedData { EncodedACBioContextInformationPartial }


  EncodedACBioContextInformationPartial ::=
   ENCODED-VALUE-OF.&Type( EACBioContextInformationPartial )
```

In definition of `ACBioContextInformationPartial`, the types of the second and the fourth components are different from those of `ACBioContextInformationPartial`. The type `BPUInformationPartial` of the second component is defined in 7.3.1.1 and the type `BiometricProcessPartial` in 7.3.1.2.

```
  ACBioContextInformationPartial ::= SEQUENCE {
    version             Version DEFAULT v0,
    bpuInformationPartial  BPUInformationPartial,
    verifierControl     VerifierControl,
    biometricProcessPartial   BiometricProcessPartial
  }
```

7.3.1.1 BPUInformationPartial

The type `BPUInformationPartial` is defined as the following. While the type `BPUInformation` has a component of the type `BTCertificateInformation`, this type does not have the corresponding component. The first component is of type `BPUCertificateReferrerInformation`, same as in `BPUInformation`. The type of the second component is different from the type of the corresponding component of `BPUInformation`.

```
BPUInformationPartial ::= SEQUENCE {
   bpuCertificateReferrerInformation     BPUCertificateReferrerInformation OPTIONAL,
   bpuReportInformationPartial     BPUReportInformationPartial,
}

BPUReportInformationPartial ::= CHOICE {
   bpuReportPartial       BPUReportPartial,
   bpuReportPartialReferrer     URI
}
```

Definition of type `BPUReportPartial` is almost the same as that of `BPUReport`. The difference is the type `BPUFunctionReportInformationPartial` of the data signed to make the first component .

```
BPUReportPartial ::= SEQUENCE {
   bpuFunctionReportPartial     BPUFunctionReportPartial,
   bpuSecurityReport     BPUSecurityReport
}

BPUFunctionReportPartial ::= SignedData { EncodedBPUFunctionReportInformationPartial }

EncodedBPUFunctionReportInformationPartial
::= ENCODED-VALUE-OF.&Type( EBPUFunctionReportInformationPartial)
```

The type `BPUFunctionReportInformationPartial` has only one componet for the data type of input data because no BPU with two inputs is on enrolment.

```
BPUFunctionReportInformationPartial ::= CHOICE {
   bpuFunctionReportSubprocesses   BPUFunctionReportSubprocessesPartial,
   modality             Modality,
   inputType           TypeDataPartial OPTIONAL,
   outputType           TypeDataPartial
}

BPUFunctionReportSubprocessesPartial ::= SEQUENCE OF BPUFunctionReportSubprocessPartial

BPUFunctionReportSubprocessPartial ::= SEQUENCE {
   function           DefinitionFunctionPartial,
   qualityEvaluation    QualityEvaluation OPTIONAL
}

DefinitionFunctionPartial ::= SEQUENCE {
   functionNamePartial           NameFunctionPartial,
    functionDescriptionPartial   DescriptionFunctionPartial OPTIONAL
  }
```

The type **NameFunctionPartial** takes only the values **data-capture** and **signal-processing**, because only these two subprocesses are used on enrolment.

```
  NameFunctionPartial ::= ENUMERATED {
    data-capture(1),
    signal-processing(2),
  ...
  }
```

If the component **functionNamePartial** in **DefinitionFunctionPartial** takes a value **data-capture** or **signal-processing**, then the component **functionDescriptionPartial** shall have the component of type **DescriptionDataCapture** or **DescriptionSignalProcessing** respectively.

```
  DescriptionFunctiontPartial ::= CHOICE {
    descriptionDataCapture   DescriptionDataCapture,
    descriptionSignalProcessing  DescriptionSignalProcessing,
  ...
  }
```

The range of the value of type **TypeDataPartial** is restricted from **acquired-biometric-sample** to **post-processed-biometric-sample** because the BPU(s) used on enrolment is data capture or signal-processing.

```
  TypeDataPartial ::= ENUMERATED {
    acquired-biometric-sample(1),
    intermediate-biometric-sample(2),
    processed-biometric-sample(3),
    post-processed-biometric-sample(4),
  ...
  }
```

7.3.1.2 BiometricProcessPartial

Definition of type **BiometricProcessPartial** is almost the same as that of **BiometricProcess** except for the type of the component **typeData** of the type **InputOutputInformationPartial**.

```
  BiometricProcessPartial ::= SEQUENCE {
    inputInformationList      InputInformationListPartial,
    outputInformationList      OutputInformationListPartial
  }
```

```
InputInformationListPartial::= SEQUENCE OF InputOutputInformationPartial
OutputInformationListPartial::= SEQUENCE OF InputOutputInformationPartial

InputOutputInformationPartial::= SEQUENCE {
   typeData       TypeDataPartial,
   hashValue       Hash
}
```

Annex A (normative)

A.1 ASN.1 description of ACBio

```
AuthenticationContextForBiometrics {
   iso(1) standard(0) acbio(24761) module(1) acbio(2) rev(1)
}

DEFINITIONS AUTOMATIC TAGS ::= BEGIN

IMPORTS

   -- ISO/IEC 24761 Authentication context for biometrics

   ATTRIBUTE, BTCertificateInformation
      FROM BiometricTemplateCertificate {
         iso(1) standard(0) acbio(24761) module(1) btc(1) rev(1) }

   -- ISO/IEC 19785 Common Biometric Exchange Formats Framework

   BioTypeACBio, BioSubtypeACBio
      FROM TypesDataElementsACBio {
        iso(1) standard(0) acbio(24761) module(1) tde(1) rev(1) }


   -- ISO 19092-2 Biometrics - message syntax & cryptographic requirements

   DigestAlgorithmIdentifier, SignedDataType
      FROM BiometricSchema {
         iso(1) identified-organization(3) tc68(133) standard(17)
            biometrics(19092) module(0) bs(1) rev(1) } ;


AuthenticationContextForBiometrics ::= SignedData { EncodedACBioContextInformation }

EncodedACBioContextInformation ::= ENCODED-VALUE-OF.&Type( EACBioContextInformation )

-- ACBio on enrolment, used in Biometric Template Certificate

AuthenticationContextForBiometricsPartial ::=
SignedData { EncodedACBioContextInformationPartial }

EncodedACBioContextInformationPartial ::=
 ENCODED-VALUE-OF.&Type( EACBioContextInformationPartial )

ACBioContextInformation ::= SEQUENCE {
   version            Version DEFAULT v0,
   bpuInformation     BPUInformation,
   verifierControl    VerifierControl,
   biometricProcess   BiometricProcess
}

-- BPU information Block

BPUInformation ::= SEQUENCE {
   bpuCertificateReferrerInformation     BPUCertificateReferrerInformation OPTIONAL,
   bpuReportInformation     BPUReportInformation,
   btCertificateInformation      BTCertificateInformation OPTIONAL
}

-- ACBio on enrolment, used in Biometric Template Certificate

ACBioContextInformationPartial ::= SEQUENCE {
   version            Version DEFAULT v0,
   bpuInformationPartial  BPUInformationPartial,
   verifierControl    VerifierControl,
   biometricProcessPartial   BiometricProcessPartial
}

BPUInformationPartial ::= SEQUENCE {
   bpuCertificateReferrerInformation     BPUCertificateReferrerInformation OPTIONAL,
   bpuReportInformationPartial     BPUReportInformationPartial,
}



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

BPUCertificateReferrerInformation ::= SEQUENCE {
   bpuCertificateReferrer     URI,
   crlsReferrer     URI OPTIONAL
}
```

```
BPUReportInformation ::= CHOICE {
   bpuReport       BPUReport,
   bpuReportReferrer     URI
}

BPUReport ::= SEQUENCE {
   bpuFunctionReport     BPUFunctionReport,
   bpuSecurityReport     BPUSecurityReport
}

BPUFunctionReport ::= SignedData { EncodedBPUFunctionReportInformation }

EncodedBPUFunctionReportInformation ::=
 ENCODED-VALUE-OF.&Type( EBPUFunctionReportInformation )

BPUFunctionReportInformation ::= SEQUENCE {
   bpuFunctionReportSubprocesses   BPUFunctionReportSubprocesses,
   modality             Modality,
   inputType1          TypeData OPTIONAL, -- if data capture, none
   inputType2          TypeData OPTIONAL,  -- unless comparison, none
   outputType          TypeData
}

BPUFunctionReportSubprocesses ::= SEQUENCE OF BPUFunctionReportSubprocess

BPUFunctionReportSubprocess ::= SEQUENCE {
   function          DefinitionFunction,
   qualityEvaluation    QualityEvaluation OPTIONAL
}



-- For ACBioPartial, used on enrolment

BPUReportInformationPartial ::= CHOICE {
   bpuReportPartial       BPUReportPartial,
   bpuReportPartialReferrer     URI
}

BPUReportPartial ::= SEQUENCE {
   bpuFunctionReportPartial     BPUFunctionReportPartial,
   bpuSecurityReport     BPUSecurityReport
}
```

```
BPUFunctionReportPartial ::= SignedData { EncodedBPUFunctionReportInformationPartial }

EncodedBPUFunctionReportInformationPartial ::=
 ENCODED-VALUE-OF.&Type( EBPUFunctionReportInformationPartial)

-- For ACBioPartial, used on enrolment. No comparison subprocess, at most one input.

BPUFunctionReportInformationPartial ::= CHOICE {
   bpuFunctionReportSubprocesses   BPUFunctionReportSubprocessesPartial,
   modality             Modality,
   inputType          TypeDataPartial OPTIONAL, -- if data capture, none
   outputType          TypeDataPartial
}

BPUFunctionReportSubprocessesPartial ::= SEQUENCE OF BPUFunctionReportSubprocessPartial

-- For ACBioPartial, used on enrolment

BPUFunctionReportSubprocessPartial ::= SEQUENCE {
   function          DefinitionFunctionPartial,
   qualityEvaluation    QualityEvaluation OPTIONAL
}



Modality ::= SEQUENCE {
   bioTypeACBio   BioTypeACBio,
   bioSubtypeACBio    BioSubtypeACBio  OPTIONAL
}

DefinitionFunction ::= SEQUENCE {
   functionName          NameFunction,
   functionDescription   DescriptionFunction OPTIONAL
}

-- Name of function of subprocess

NameFunction ::= ENUMERATED {
   data-capture(1),
   signal-processing(2),
   storage(3),
   comparison(4),
   decision(5),
...
}

-- Description of function of subprocess

DescriptionFunction ::= CHOICE {
   descriptionDataCapture   DescriptionDataCapture,
   descriptionSignalProcessing  DescriptionSignalProcessing,
   descriptionStorage  DescriptionStorage,
   descriptionComparison  DescriptionComparison,
   descriptionDecision  DescriptionDecision,
...
}

DescriptionDataCapture ::= OCTET STRING (SIZE(1..MAX))  -- To be defined later
DescriptionSignalProcessing ::= OCTET STRING (SIZE(1..MAX))  -- To be defined later
DescriptionStorage ::= OCTET STRING (SIZE(1..MAX))  -- To be defined later
DescriptionComparison ::= OCTET STRING (SIZE(1..MAX))  -- To be defined later
DescriptionDecision ::= OCTET STRING (SIZE(1..MAX))  -- To be defined later

-- Definition of data type for input and output

TypeData ::= ENUMERATED {
   acquired-biometric-sample(1),
   intermediate-biometric-sample(2),
   processed-biometric-sample(3),
   post-processed-biometric-sample(4),
   acquired-biometric-template(5),
   intermediate-biometric-template(6),
   processed-biometric-template(7),
   post-processed-biometric-template(8),
   comparison-score(9),
   comparison-result(10),
...
}

-- For ACBioPartial, used on enrolment

DefinitionFunctionPartial ::= SEQUENCE {
   functionNamePartial          NameFunctionPartial,
   functionDescriptionPartial   DescriptionFunctionPartial OPTIONAL
```

```
}

-- Name of function of subprocess used on enrolment.
-- Only data capture and signal processing

NameFunctionPartial ::= ENUMERATED {
   data-capture(1),
   signal-processing(2),
...
}

DescriptionFunctiontPartial ::= CHOICE {
   descriptionDataCapture   DescriptionDataCapture,
   descriptionSignalProcessing  DescriptionSignalProcessing,
...
}

-- Definition of data type for input and output used on enrolment

TypeDataPartial ::= ENUMERATED {
   acquired-biometric-sample(1),
   intermediate-biometric-sample(2),
   processed-biometric-sample(3),
   post-processed-biometric-sample(4),
...
}

QualityEvaluation ::= SEQUENCE {
   qualityBiometricProcess    QualityBiometricProcess   OPTIONAL,
   qualityExtension          QualityExtension  OPTIONAL
}

QualityBiometricProcess ::= OCTET STRING (SIZE(1..MAX))  -- To be defined later in SC 37
QualityExtension ::= OCTET STRING (SIZE(1..MAX))  -- For extension

BPUSecurityReport ::= SEQUENCE {
   secLevelCryptoModule   SecLevelCryptoModule  OPTIONAL,
   secBiometricProcess    SecBiometricProcess  OPTIONAL,
   secExtension          SecExtension  OPTIONAL
}

-- To be defined later in SC27/WG 3
```

```
SecLevelCryptoModule ::= OCTET STRING (SIZE(1..MAX))

-- To be defined later in SC27/WG 3
SecBiometricProcess ::= OCTET STRING (SIZE(1..MAX))

-- For extension
SecExtension ::= OCTET STRING (SIZE(1..MAX))



BTCertificateInformation ::= CHOICE {
   btCertificate          BiometricTemplateCertificate,
   btCertificateReferrer     URI
}

-- Verifier Controlling Block

VerifierControl ::= SEQUENCE {
   controlValue  ControlValue
}

ControlValue ::= OCTET STRING

-- Biometric Process Block

BiometricProcess ::= SEQUENCE {
   inputInformationList     InputInformationList,
   outputInformationList     OutputInformationList
}

InputInformationList ::= SEQUENCE OF InputOutputInformation
OutputInformationList ::= SEQUENCE OF InputOutputInformation

InputOutputInformation ::= SEQUENCE {
   typeData       TypeData,
   hashValue       Hash
}

-- Biometric Process Block for ACBioPartial used on enrolment

BiometricProcessPartial ::= SEQUENCE {
   inputInformationList     InputInformationListPartial,
   outputInformationList     OutputInformationListPartial
}

InputInformationListPartial ::= SEQUENCE OF InputOutputInformationPartial
OutputInformationListPartial ::= SEQUENCE OF InputOutputInformationPartial

InputOutputInformationPartial ::= SEQUENCE {
   typeData       TypeDataPartial, -- type of data is restricted
   hashValue       Hash
}

Hash ::= SEQUENCE {
   DigestAlgorithm    DigestAlgorithmIdentifier,
   hashValue           OCTET STRING
}



-- Useful definitions

SignedData { ToBeSigned } ::= SignedDataType
     (CONSTRAINED BY { -- The signature is on a value of type -- ToBeSigned
                       -- and the attributes required in this standard. -- })

-- Signed data contentType and encapsulated content

id-acbioContent OID ::= {
   iso(1) standard(0) acbio(24761) contentType(2) acbioContent(2)
}

EACBioContextInformation  ::= OCTET STRING( CONTAINING ACBioContextInformation )

id-acbioContentPartial OID ::= {
   iso(1) standard(0) acbio(24761) contentType(2) acbioContentPartial(3)
}

EACBioContextInformationPartial ::=
 OCTET STRING( CONTAINING ACBioContextInformationPartial )

id-bpuFunctionReport OID ::= {
   iso(1) standard(0) acbio(24761) contentType(2) bpuFunctionReport(4)
}

EBPUFunctionReportInformation ::=
 OCTET STRING( CONTAINING BPUFunctionReportInformation )

id-bpuFunctionReportPartial OID ::= {
   iso(1) standard(0) acbio(24761) contentType(2) bpuFunctionReportPartial(5)
}

EBPUFunctionReportInformationPartial ::=
 OCTET STRING( CONTAINING BPUFunctionReportInformationPartial )

-- Authenticated attribute

acbioContextInformation ATTRIBUTE ::= {
   WITH SYNTAX  ACBioContextInformation
   ID           id-acbiocontextInformation
}

id-acbiocontextInformation OID ::= {
   iso(1) standard(0) acbio(24761) attribute(3) acbioci(2)
}

acbioContextInformationPartial ATTRIBUTE ::= {
   WITH SYNTAX  ACBioContextInformationPartial
   ID           id-acbiocontextInformationPartial
}

id-acbiocontextInformationPartial OID ::= {
   iso(1) standard(0) acbio(24761) attribute(3) acbiocipartial(3)
}

OID ::= OBJECT IDENTIFIER  -- Alias

ENCODED-VALUE-OF ::= TYPE-IDENTIFIER  -- Defined in ISO/IEC 8824-2, Annex A

Hash ::= SEQUENCE {
   digestAlgorithm    DigestAlgorithmIdentifier  OPTIONAL,
   hashValue           OCTET STRING
}

END  -- AuthenticationContextForBiometrics –
```

A.2 ASN.1 description of BT certificate

```
BiometricTemplateCertificate {
   iso(1) standard(0) acbio(24761) module(1) btc(1) rev(1)
}

DEFINITIONS AUTOMATIC TAGS ::= BEGIN

IMPORTS

   -- ISO/IEC 24761 Authentication Context for Biometrics

   ACBioPartial, ENCODED-VALUE-OF, OID, Hash
      FROM AuthenticationContextForBiometrics {
         iso(1) standard(0) acbio(24761) module(1) acbio(2) rev(1) }

   -- ISO/IEC 19785 Common Biometric Exchange Formats Framework

   IndexBTC, ValidityPeriodBTC, EnrolmentOrgBTC
      FROM TypesDataElementsACBio {
         iso(1) standard(0) acbio(24761) module(1) tde(1) rev(1) }



   -- ISO 19092 Biometric message syntax & cryptographic requirements

   SignedDataType
      FROM BiometricSchema {
         iso(1) identified-organization(3) tc68(133) standard(17)
            biometrics(19092) module(0) bs(1) rev(1) } ;



BiometricTemplateCertificate ::= SignedData { EncodedTemplateCertificateInformation }

EncodedTemplateCertificateInformation ::=
 ENCODED-VALUE-OF.&Type(ETemplateCertificateInformation)

TemplateCertificateInformation ::= SEQUENCE {
   version            Version DEFAULT v0,
   hashBTC            Hash,      -- hash value of biometric tenplate
   indexBTC           IndexBTC,  -- unique indexBTC to biometric template
   validityPeriodBTC ValidityPeriodBTC,
   enrolmentOrgBTC    EnrolmentOrgBTC,
         -- organization which created biometric template and issued the cert.
   dataQuality       DataQuality,
   acbioPartials     AuthenticationContextForBiometricsPartials  OPTIONAL
         -- ACBio on enrolment
}

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

-- To be defined later
DataQuality ::= OCTET STRING (SIZE(1..MAX))

AuthenticationContextForBiometricsPartials ::=
 SEQUENCE OF AuthenticationContextForBiometricsPartial

-- Useful definitions

SignedData { ToBeSigned } ::= SignedDataType
     (CONSTRAINED BY { -- The signature is on a value of type -- ToBeSigned
                       -- and the attributes required in this standard. -- })

-- Signed data contentType and encapsulated content

id-acbioBTCContent OID ::= {
   iso(1) standard(0) acbio(24761) contentType(2) acbioBTCContent(1)
}

ETemplateCertificateInformation ::=
 OCTET STRING( CONTAINING TemplateCertificateInformation)

-- Authenticated attribute

templateCertificateInformation ATTRIBUTE ::= {
   WITH SYNTAX  TemplateCertificateInformation
   ID           id-TemplateCertificateInformation
}

id-templateCertificateInformation OID ::= {
   iso(1) standard(0) acbio(24761) attribute(3) tci(1)
}

-- Attribute information object class
```

```
ATTRIBUTE ::= CLASS {
  &Type  OPTIONAL,
  &id    OBJECT IDENTIFIER  UNIQUE
}
  WITH SYNTAX { [WITH SYNTAX &Type] ID &id }

END  -- BiometricTemplateCertificate –
```

A.3 ASN.1 description of types of some data elements

```
TypesDataElementsACBio {
   iso(1) standard(0) acbio(24761) module(1) tde(1) rev(1)
}

DEFINITIONS AUTOMATIC TAGS ::= BEGIN

IMPORTS

   -- ISO/IEC 19785 Common Biometric Exchange Formats Framework

   BiometricType, BiometricSubtype, Index, BDBValidityPeriod, Creator
      FROM CBEFF-DATA-ELEMENTS {
         iso standard 19785 modules(0) types-for-cbeff-data-elements(1)}

   -- ISO 19092 Biometric message syntax & cryptographic requirements

 BiometricTypes, Subtypes, UUID, ValidityPeriod
      FROM BiometricSchema {
         iso(1) identified-organization(3) tc68(133) standard(17)
            biometrics(19092) module(0) bs(1) rev(1) } ;

 Enroler
      FROM BiometricEventJournal {
         iso(1) identified-organization(3) tc68(133) standard(17)
            biometrics(19092) module(0) bej(4) rev(1) }

BioTypeACBio ::= CHOICE {
   bioType37          BiometricType,   -- ISO/IEC 19785-3
   bioType68          BiometricTypes,  -- ISO 19092-2
   ..
}
```

```
BioSubtypeACBio ::= CHOICE {
   bioSubtype37        BiometricSubtype,   -- ISO/IEC 19785-3
   bioSubtype68        Subtypes,  -- ISO 19092-2
   ..
}

IndexBTC ::= CHOICE {
   index37        Index,   -- ISO/IEC 19785-3
   index68        UUID,  -- ISO 19092-2
   ..
}

ValidityPeriodBTC ::= CHOICE {
   validityPeriod37        BDBValidityPeriod,   -- ISO/IEC 19785-3
   validityPeriod68        ValidityPeriod,  -- ISO 19092-2
   ..
}

EnrolmentOrgBTC ::= CHOICE {
   enrolmentOrg37        Creator,   -- ISO/IEC 19785-3
   enrolmentOrg68        Enroler,  -- ISO 19092-2
   ..
}

END  -- TypesDataElementsACBio –
```

Annex B (informative)

B.1 Examples of the Protocol for ACBio

In this International Standard, the protocol for ACBio is not specified. Here in this annex, two examples of the protocol for ACBio are given; one for the case of STOC(STore On Card) model, the other for the case of OCM(On card matching) model.

B.1.1 An Example of the Protocol for STOC Model

We assume that the system of STOC model's biometric verification consists of four BPUs; the computer of the verifier side, the computer of the claimant side, the biometric device connected to the computer of the claimant side, and the STOC card. We also assume that the biometric device has the functions of data capture, signal processing, comparison and decision, and that STOC card has the function of storage, and that the computer of the claimant side only transmits messages among the other three. In the following example of the protocol, we do not distinguish between the BPU and the program on the BPU.

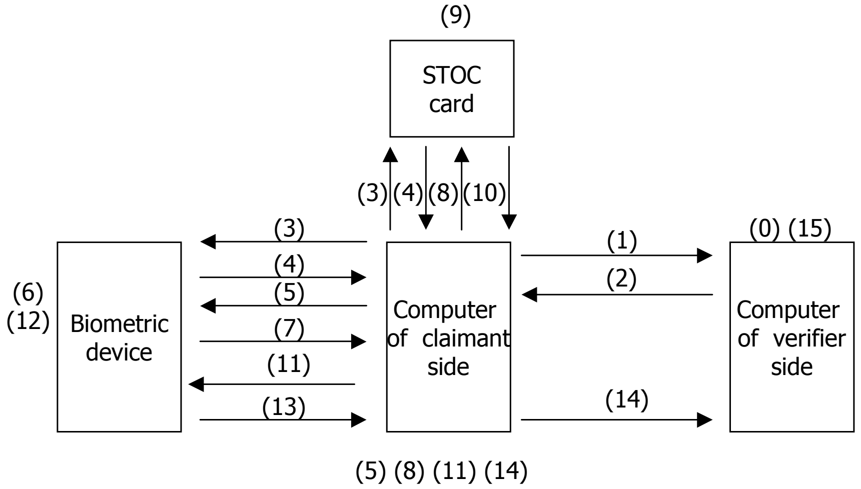

Figure 4 – An example of the protocol among the biometric process units of STOC model

Example of the protocol for this STOC model is as follows:

(0) The verifier beforehand sets up verification policy for each of the data elements of ACBio (see B.3 for an example of ACBio verification policy).

(1) The claimant claims biometric verification to the verifier via the computer of the claimant side.

(2) The computer of the verifier side requests ACBio(s) of the biometric verification process to the computer of the claimant side with the verifier's challenge and the candidate list of hash and digital signature algorithm according to the ACBio verification policy.

(3) The computer of the claimant side receives the request and hands the list of hash and digital signature algorithm to the biometric device and the STOC card.

(4) The biometric device (the STOC card) receives the list and chooses algorithms implemented on the biometric device (the STOC card respectively) from the list and sends the answer list to the computer of the claimant side.

(5) The computer of the claimant side decides the hash and the signature algorithm from the answer lists and sends a message for data capture and signal processing subprocesses to the biometric device with the verifier's challenge and the hash and the signature algorithm.

(6) The biometric device collects the biometric information from the user and generates the acquired biometric sample, and does signal processing to the acquired biometric sample to make the processed biometric sample.

(7) The biometric device sends the termination message of the signal processing subprocess to the computer of the claimant side.

(8) The computer of the claimant side receives the message from the biometric device and sends a message for the storage subprocess to the STOC card with the verifier's challenge and the hash and the signature algorithm determined in (5).

(9) The STOC card makes up the ACBio instance by the following procedures;

a) put the BPU certificate, BPU report, and the BT certificate to the BPU Information Block,

b) put the verifier's challenge, which the STOC card has received via the computer of the claimant side, into the Verifier Controlling Block,

c) put value **`precessed-biometric-template`** and the hash value of the processed biometric template into components of type **`InputOutputInformation`** of the only one element of **`outputInofrmationList`** of the Biometric Process Block, and

d) create the **`SignedData`** of the data of the type **`ACBioContentInformation`** which includes data of a) to c) above (using the digital signature algorithm determined in (5)) to make the ACBio instance for the execution on this STOC card.

(10) The STOC card sends the ACBio instance to the computer of the claimant side.

(11) The computer of the claimant side receives the ACBio instance from STOC card, stores it, and sends it to the biometric device.

(12) The biometric device receives the ACBio instance, checks the integrity of the ACBio instance, compares the processed biometric sample generated in itself with the processed biometric template from the STOC card, scores the similarity, decides the validity of the claimant user, and makes up the ACBio instance by the following procedures;

a) put the BPU certificate to the BPU Information Block,

b) put the verifier's challenge, which the biometric device receives via the computer of the claimant side, to the Verifier Controlling Block,

c) put value **`precessed-biometric-template`** and the hash value of the input processed biometric template into components of type **`InputOutputInformation`** of the only one element of **`inputInofrmationList`** of the Biometric Process Block,

d) put value **`comparison-result`** and the hash value of the comparison result into components of type **`InputOutputInformation`** of the only one element of **`outputInofrmationList`** of the Biometric Process Block, and

e) create the **SignedData** of the data of the type **ACBioContentInformation** which includes data of a) to d) above (using the digital signature algorithm determined in (5)) to make the ACBio instance for the execution on this biometric device.

(13) The biometric device sends the ACBio instance to the computer of the claimant side.

(14) The computer of the claimant side receives the ACBio instance and sends the two ACBio instances together to the computer of the verifier side.

(15) The computer of the verifier side receives the two ACBio instances and the verifier verifies by the following procedures;

a) check the integrity of ACBio instances through signature verification,

b) check the correspondence of the request given in (2) and the ACBio instances by comparing the verifier's challenge in the Verifier Controlling Block with the original value,

c) check if the information in the BPU report of the PBU Information Block satisfies the ACBio verification policy of the verifier(see B.3 for ACBio verification policy),

d) check if the series of subprocesses forms one biometric verification process, with the information on the type of input/output data and on the suprocess(es) in BPU report in BPU Information Block, by the consistency of the input and the output data communicated among BPUs of the information in the Biometric Process Block of each ACBio instance, i.e. the equality of the **outputInofrmationList** in Biometric Process Block of the ACBio instance generated by the STOC card with the **inputInofrmationList** in Biometric Process Block of the ACBio instance generated by the biometric device.

B.1.2 An Example of the Protocol for OCM Model

We assume that the system of OCM model's biometric verification consists of four BPUs; the computer of the verifier, the computer of the claimant side, the sensor device connected to the claimant side computer, and the OCM card. We also assume that the sensor device has the functions of data capture and signal processing and that OCM card has the functions of storage, comparison and decision, and that the computer of the claimant side only transmits messages among the other three. In the following example of the protocol, we do not distinguish between the BPU and the program on the BPU. A couple of steps may be performed in the same or a similar way as for the STOC model.

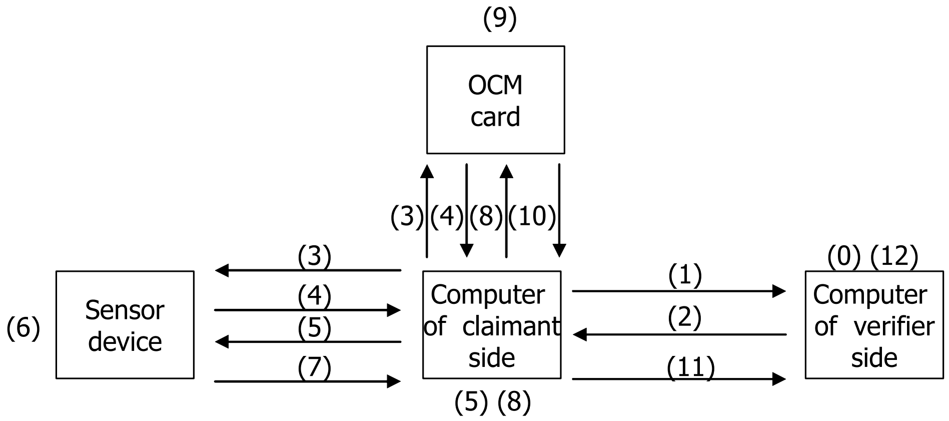

Figure 5 – An example of the protocol among the BPUs of OCM model

Example of the protocol for this OCM model is as follows:

With the sensor device in place of the biometric device and the MOC card in place of the STOC card, execute steps (0) to (5) in the same way as in the protocol described in B.1.1.

(6) The sensor device first performs the same operations as the biometric device in step (6) of the protocol described in B.1.1, and then makes up the ACBio instance by the following procedures;

a) put the BPU certificate, BPU report, and the BT certificate to the BPU Information Block,

b) put the verifier's challenge, which the sensor device receives via the computer of the claimant side, into the Verifier Controlling Block,

c) put value `precessed-biometric-sample` and the hash value of the processed biometric sample into components of type `InputOutputInformation` of the only one element of `outputInofrmationList` of the Biometric Process Block, and

d) create the `SignedData` of the data of the type `ACBioContentInformation` which includes data of a) to c) above (using the digital signature algorithm determined in (5)) to make the ACBio instance for the execution on this sensor device.

(7) The sensor device sends the ACBio instance to the computer of the claimant side together with the processed biometric sample.

(8) The computer of the claimant side receives the ACBio instance from the sensor device and sends a message for the subprocesses implemented on the OCM card with the verifier's challenge and the hash and the signature algorithm determined in (5), together with the processed biometric sample and the ACBio generated by the sensor device.

(9) The OCM card checks the integrity of the ACBio instance generated by the sensor device, compares the processed biometric sample from the sensor device with the processed biometric template stored in the OCM card, scores the similarity, decides the validity of the claimant user, and makes the ACBio instance by the following procedures;

a) put the BPU certificate, BPU report, and the BT certificate to the BPU Information Block,

b) put the verifier's challenge, which the OCM card has received via the computer of the claimant side, into the Verifier Controlling Block,

c) put value `precessed-biometric-sample` and the hash value of the input processed biometric sample into components of type `InputOutputInformation` of the only one element of `inputInofrmationList` of the Biometric Process Block,

d) put value `comparison-result` and the hash value of the comparison result into components of type `InputOutputInformation` of the only one element of `outputInofrmationList` of the Biometric Process Block, and

e) create the `SignedData` of the data of the type `ACBioContentInformation` which includes data of a) to d) above (using the digital signature algorithm determined in (5)) to make the ACBio instance for the execution on this OCM card.

(10) The OCM card sends the ACBio instance of its own to the computer of the claimant side.

(11) The computer of the claimant side receives the ACBio instance from the OCM card and sends the ACBio instance made by the sensor device and the ACBio instance made by the OCM card together to the computer of the verifier side.

(12) The computer of the verifier side receives the two ACBio instances and the verifier verifies by the following procedures;

a) check the integrity of ACBio instances with the signature verification,

b) check the correspondence of the request given in (2) and the ACBio instances by comparing the verifier's challenge in the Verifier Controlling Block and the original value,

c) check if the information in the BPU report of the PBU Information Block satisfies the ACBio verification policy of the verifier(see B.3 for ACBio verification policy),

d) check if the series of subprocesses forms one biometric verification process, with the information on the type of input/output data and on the suprocess(es) in BPU report in BPU Information Block, by the consistency of the input and the output data communicated among BPUs of the information in the Biometric Process Block of each ACBio instance, i.e. the equality of the `outputInofrmationList` in Biometric Process Block of the ACBio instance generated by the sensor device with the `inputInofrmationList` in Biometric Process Block of the ACBio instance generated by the OCM card.

B.2 Examples of ACBio

B.2.1 Examples of ACBio for STOC Model

We assume that the STOC model is as described in B.1.1. Then the ACBio generated by the biometric device and that of the STOC card are as the following:

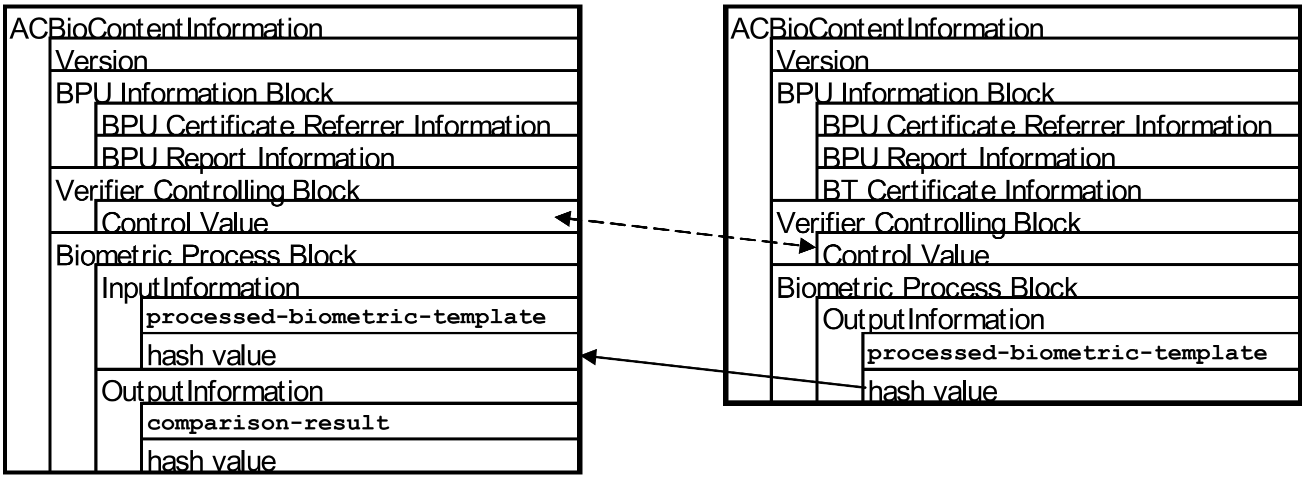

Figure 6 – generated ACBio instances of a STOC model

The left figure is the ACBio instance generated by the biometric device and the right by the STOC card. Two verifier's challenge linked by the dotted arrow have the same value. The data elements linked by the arrow corresponds to each other, i.e. the OutputInformation in the ACBio instance of the STOC card and the InputInformation in the ACBio instance of the biometric device have virtually the same values because the processed biometric template of the STOC card is the input data to the comparison subprocess on the biometric device.

B.2.2 Examples of ACBio for OCM Model

We assume that the OCM model is as described in B.1.2. Then the ACBio instance generated by the sensor device and that of the OCM card are as the following

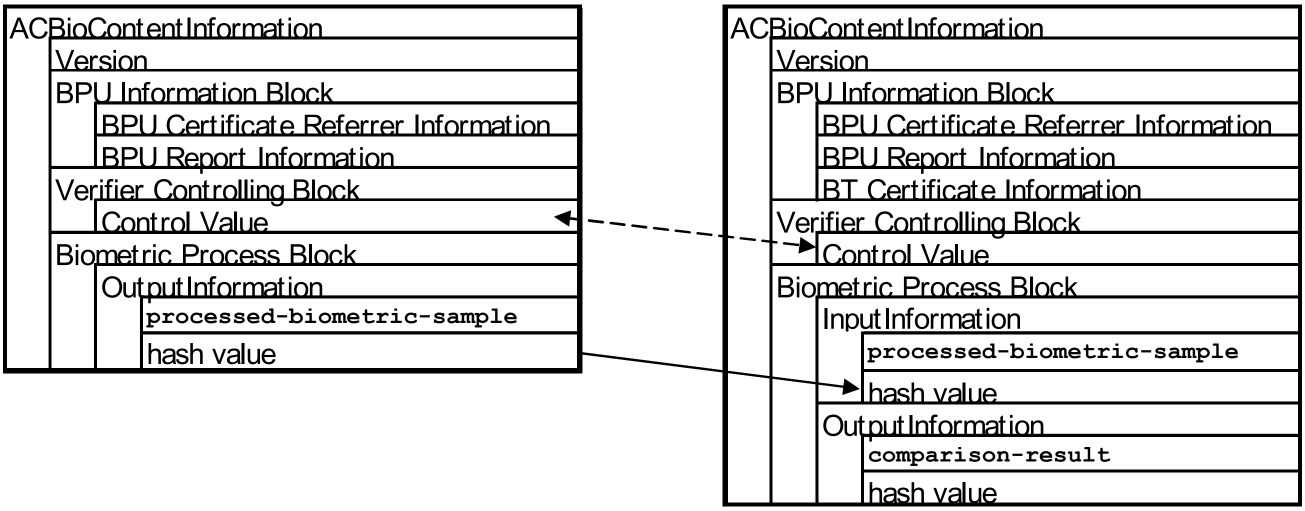

Figure 7– generated ACBio of a OCM model

The left figure is the ACBio instance generated by the sensor device and the right by the OCM card. Two verifier's challenge linked by the dotted arrow have the same value. The data elements linked by the arrow corresponds to each other, i.e. the OutputInformation in the ACBio instance of the sensor device and the InputInformation in the ACBio instance of the OCM card have virtually the same values because the processed biometric sample of the sensor device is the input data to the comparison subprocess on the OCM card.

B.3 ACBio Verification Policy

It is desirable that ACBio verification policy includes the followings:

admissible hash algorithm of BPUs

admissible digital signature algorithm of BPUs

admissible security of BPUs

admissible quality of acquired biometric sample

admissible quality of processed biometric sample

admissible quality of biometric template

admissible comparison scores

admissible comparison algorithms

admissible comparison parameters

admissible quality of comparison

admissible decision result requirements

If ACBio verification policy is sent from verifier's computer to other location, it should be signed to keep integrity..

B.4 Requirement to Evaluation from the ACBio point of view

To verify biometric verification process with ACBio, each BPU should have the evaluation result, which is described in BPU report (see 7.2) in BPU Information Block. We list our requirement to evaluation.

Criteria for evaluation on security function is specified in ISO/IEC 19790. This standard endorses ISO/IEC 19790 and applies the specification of ISO/IEC 19790 to this standard.

The covered areas of evaluation requirement in ISO/IEC 19790 are as the following:

1) Cryptographic module specification.

2) Cryptographic module ports and interfaces.

3) Roles, services, and authentication.

4) Finite state model.

5) Physical security.

6) Operational environment.

7) Cryptographic key management.

8) EMI (electromagnetic interference) /EMC (electromagnetic compatibility).

9) Self-test.

10) Design assurance.

11) Mitigation of other attacks.

Criteria for evaluation on security of biometric process is to be specified in ISO/IEC 19792. This standard endorses ISO/IEC 19792 and applies the specification of ISO/IEC 19792 to this standard.

The evaluation items in ISO/IEC 19792 are as the following:

1) Adequacy of the algorithm properties and the security assurance.

2) Adequacy of the biometric processes parameters tuning and the algorithm security assurance.

3) Correct work of quality check of the biometric functions.

4) Correct implementation of the interfaces of the biometric processes as specified.

5) No bypassing of security function of the algorithm.

6) Access control mechanism to management function.

7) Adequacy of resistance to physical attacks.

8) Sufficient quality of processed biometric sample.

9) Secure interface for communication

10) Documentation of secure operation and administration.

Criteria for evaluation on functional performance of biometric process is to be specified in ISO/IEC 19795 and ISO/IEC FCD 19784. This standard endorses these standards and applies their specification to this standard.

Examples of the evaluation items for comparison subprocess in ISO/IEC 19795 are as the following:

1) Quality independent of influential factors such as health status, age, expression, posture, illumination, background, resolution, etc.

2) Genuine attempt for FNMR (False No Match Rate).

3) Imposter attempt for FMR (False Match Rate).

4) Imposter pair testing for FMR.

ISO/IEC FCD 19784 defines a value indicating the quality of biometric data. This standard endorses ISO/IEC FDC 19784 and applies the value to this standard to indicate the functional performance of data capture and that of signal processing.

Bibliography

[1] ISO/IEC 9798 (all parts), Information Technology – Security Techniques – Entity Authentication

[2] ISO/IEC 9594-8: Information technology | ITU-T Recommendation X.509, Open Systems Interconnection -- The Directory: Authentication framework", International Organization for Standardization, Geneva, Switzerland, 2000

[3] ISO/IEC 8824 (All parts) | ITU-T Recommendations X.680-683, Information Technology - Abstract Syntax Notation One (ASN.1)ISO/IEC 8825-1 | ITU-T Recommendation X.690, Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)

[4] ISO/IEC 8825-4 | ITU-T Recommendation X.693, Information Technology - ASN.1 Encoding Rules: Specification of XML Encoding Rules (XER)

[5] ISO/IEC 7816-11:2004, Information Technology - Identification Cards - Integrated circuit cards – Part11 : Personal verification through biometric method

[6] ISO/IEC 19794 (all parts), Information Technology - Biometric Data Interchange Formats

[7] ISO CD 19092-1:2005, Financial Services – Biometrics – Part 1 : Security Framework

[8] W3C XML 1.0:2000, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, Copyright © [6 October 2000] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University), http://www.w3.org/TR/2000/REC-xml-20001006