



INTERNATIONAL STANDARD ISO/IEC 24761:2009  
TECHNICAL CORRIGENDUM 1

Published 2013-03-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION  
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

## Information technology — Security techniques — Authentication context for biometrics

### TECHNICAL CORRIGENDUM 1

*Technologies de l'information — Techniques de sécurité — Contexte d'authentification biométrique*

*RECTIFICATIF TECHNIQUE 1*

Technical Corrigendum 1 to ISO/IEC 24761:2009 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

---

#### *Page 1, Clause 2*

Add the following at the end:

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

#### *Page 11, Clause 6*

Replace the definition of `EncapsulatedContentInfoACBio` with:

```
EncapsulatedContentInfoACBio ::= SEQUENCE {  
    eContentType CONTENT-TYPE.&id({ContentTypeACBioContentInfo}),  
    eContent [0] EXPLICIT OCTET STRING  
    ( CONTAINING CONTENT-TYPE.&Type  
      ({ContentTypeACBioContentInfo}{@eContentType})) }
```

Page 11, Clause 6

Replace the definition of *ACBioContentInformation* and *Version* with:

```
ACBioContentInformation ::= SEQUENCE {
    version Version DEFAULT v1,
    bpuInformation BPUInformation,
    controlValue OCTET STRING (SIZE(16)),
    biometricProcess BiometricProcess,
    brtCertificateInformation BRTCertificateInformation OPTIONAL}

Version ::= INTEGER { v1(1) } ( v1, ... )
```

Page 15, 6.1

Replace the definition of *BPURReportInformation* with:

```
BPURReportInformation ::= CHOICE {
    bpuReport [0] EXPLICIT BPURReport,
    bpuReportReferrer URI}
```

Page 15, 6.2

Replace the definitions of *BiometricProcess* and *SubprocessIndexList* with:

```
BiometricProcess ::= SEQUENCE {
    subprocessIndexList SubprocessIndexList,
    bpuInputExecutionInformationList BPUIOExecutionInformationList OPTIONAL,
    bpuOutputExecutionInformationList BPUIOExecutionInformationList }

SubprocessIndexList ::= SEQUENCE SIZE(1..MAX) OF SubprocessIndex
```

Page 16, 6.3

Replace the definitions of *BRTCertificateList* and *BRTCertificateReferrerList* with:

```
BRTCertificateList ::= SEQUENCE SIZE(1..MAX) OF BRTCertificate

BRTCertificateReferrerList ::= SEQUENCE SIZE(1..MAX) OF URI
```

Page 17, 7.2

Replace the definitions of *BPURReportInformation*, *ContentTypeBPURReport*, *EncapsulatedContentInfoBPURReport*, and *contentBPURReport* respectively with:

```
BPURReportInformation ::= CHOICE {
    bpuReport [0] EXPLICIT BPURReport,
    bpuReportReferrer URI}

ContentTypeBPURReport CONTENT-TYPE ::= {contentBPURReport }

EncapsulatedContentInfoBPURReport ::= SEQUENCE {
    eContentType CONTENT-TYPE.&id({ContentTypeBPURReportContentInfo}),
    eContent [0] EXPLICIT OCTET STRING
        ( CONTAINING CONTENT-TYPE.&Type
          ({ContentTypeBPURReportContentInfo}{@eContentType}))}

contentBPURReport CONTENT-TYPE ::= {
    ContentBPURReport
    IDENTIFIED BY id-contnetBPURReport}
```

## Page 17, 7.2

Replace the second paragraph with:

**BPUR** is defined in a similar way to **ACBioInstance**. A BPU Report consists of two fields; the first field of fixed value of **id-contnetBPUR** and the second of type **ContentBPUR**, which is a data of parameterized **SIGNEDDATA** with encapsulated content of type **BPURContentInformation**, which consists of two components, **bpurFunctionReport** and **bpurSecurityReport**. The signature shall be generated using the private key of the product vendor of the BPU.

NOTE The functions of and data flow in a BPU in enrolment may be different from those in biometric verification. In such a case, two **BPUR**s may be prepared, one for enrolment, another for biometric verification. Otherwise one **BPUR** may be prepared for both enrolment and biometric verification. The latter case is noted in 7.2.1.

In ASN.1 notation, **BPUR** is described as follows:

## Page 18, 7.2.1

Replace the definitions of **BPURFunctionReport**, **BPURSubprocessInformationList**, and **BPURIOStaticInformationList** respectively with:

```
BPURFunctionReport ::= SEQUENCE {
    bpurSubprocessInformationList BPURSubprocessInformationList,
    bpurInputStaticInformationList BPURIOStaticInformationList OPTIONAL,
    bpurOutputStaticInformationList BPURIOStaticInformationList }

BPURSubprocessInformationList ::= SEQUENCE SIZE(1..MAX) OF BPURSubprocessInformation

BPURIOStaticInformationList ::= SEQUENCE SIZE(1..MAX) OF BPURIOStaticInformation
```

## Page 18, 7.2.1

Replace the last paragraph with:

**bpurInputStaticInformationList** is a list of elements of type **BPURIOStaticInformation** as many as the number of the input data to the BPU. **bpurOutputStaticInformationList** is a list of elements of type **BPURIOStaticInformation** as many as the number of the output data from the BPU. The type **BPURIOStaticInformation** is defined in 7.2.1.2.

In enrolment, storage subprocess shall output the hash value of the input of biometric sample which is to be stored as the biometric reference template, and the hash value is to be set in the BRT certificate. Therefore **bpurOutputStaticInformationList** shall have such a member if it is an expression for a BPU with storage subprocess in enrolment.

NOTE When the function of and data flow in a BPU in enrolment are different from those in biometric verification, the number of the elements in **bpurSubprocessInformationList** may not be equal to the number of the subprocesses in the BPU. It may be the sum of the number of the subprocesses in enrolment and that in biometric verification. In this case, **bpurSubprocessInformationList** is divided into two groups, one for enrolment and another for biometric verification. **subprocessName** of **functionDefinition** in a member of a group of **bpurSubprocessInformationList** may have the same value as the value of **subprocessName** of **functionDefinition** in a member in the other group but the value of the field **subprocessIndex** shall be different from that of the corresponding member of the list. If the **bpurSubprocessInformationList** is expressed as above, so are **bpurInputStaticInformationList** and **bpurOutputStaticInformationList** expressed in a similar way: there may be two members in the list where the value of **subprocessIOIndex** of one member is different from that of the other while the values of **dataType** are the same.

## Page 19, 7.2.1.1.1

Replace the definition of **FunctionDefinition** with:

```
FunctionDefinition ::= SEQUENCE {
```

```
subprocessName SubprocessName,
subprocessIndex SubprocessIndex,
biometricType BiometricType OPTIONAL,
biometricSubtype BiometricSubtype OPTIONAL,
inputIndex1 IOIndex OPTIONAL,
inputIndex2 IOIndex OPTIONAL,
outputIndex IOIndex,
functionDescription OCTET STRING (SIZE(1..MAX)) OPTIONAL}
```

Page 19, 7.2.1.1.1

Add the following description after the third paragraph:

A pair of components `biometricType` and `biometricSubtype` indicates the modality of biometric data processed in the subprocess. The types `BiometricType` and `BiometricSubType` are defined in ISO/IEC 19785-3. `biometricType` is mandatory if `subprocessName` does not take the value `comparison` or `decision`.

Page 20, 7.2.1.2

Replace the first paragraph, definition of `BPUIOStaticInformation`, and second paragraph with:

`BPUIOStaticInformation` is a data type which gives information about input/output to/from the BPU, and consists of two components; `dataType` and `subprocessIOIndex`.

```
BPUIOStaticInformation ::= SEQUENCE {
    dataType DataType,
    subprocessIOIndex IOIndex}
```

Page 20, 7.2.1.2

Replace the fourth paragraph with:

There shall be the component `purpose` if the first component `processedLevel` takes the value from `raw-data` to `processed-data`. There shall not be the component `purpose` if the `processedLevel` takes the value `comparison-score`, `comparison-decision`, or `hashed-data`.

Page 20, 7.2.1.2

Replace the definition of `ProcessedLevel` with:

```
ProcessedLevel ::= ENUMERATED {
    raw-data(1),
    intermediate-data(2),
    processed-data(3),
    comparison-score(4),
    comparison-result(5),
    hashed-data(6),
    ...}
```