INTERNATIONAL STANDARD

ISO/IEC 19784-1

First edition
2006-05-01

AMENDMENT 3
2010-08-01

Information technology — Biometric application programming interface —

Part 1: BioAPI specification

AMENDMENT 3: Support for interchange of certificates and security assertions, and other security aspects

Technologies de l'information — Interface de programmation d'applications biométriques —
Partie 1: Spécifications BioAPI

AMENDEMENT 3: Support pour interéchange de certificats et de déclarations de sécurité, et autres aspects de sécurité

Reference number
ISO/IEC 19784-1:2006/Amd.3:2010(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

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.

Amendment 3 to ISO/IEC 19784-1:2006 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 37, *Biometrics*.

This amendment to ISO/IEC 19784-1 defines a new version 2.2 of BioAPI which adds support for biometric fusion and security assertions to ISO/IEC 19784-1. It extends the API and the SPI of BioAPI by specifying new functions and new values for existing data types.

ISO/IEC 19784-1:2006 provides no direct support for biometric fusion. In addition, the use of FARs in the representation of matching scores is not suitable, in general, for performing score-level fusion (although it does allow some limited forms of fusion). This amendment adds support of biometric fusion to ISO/IEC 19784-1.

Information technology — Biometric application programming interface —

Part 1:
BioAPI specification

AMENDMENT 3: Support for interchange of certificates and security assertions, and other security aspects

1) General amendment items

1-1) Add the following at the end of the first paragraph of the Foreword:

", BioAPI 2.1, and BioAPI 2.2"

1-2) Replace the last paragraph of the Foreword with the following:

This is the first ISO/IEC standard on BioAPI. Previous versions were published by ANSI and the BioAPI Consortium. As the last official non-ISO release was designated Version 1.1, the version specified in this part of ISO/IEC 19784-1 is designated Version 2.0 onwards. This is to distinguish the versions of BioAPI products in the marketplace.

1-3) Replace the first paragraph of the Introduction with the following:

This part of ISO/IEC 19784 provides a high-level generic biometric authentication model suited to most forms of biometric technology. Support for multimodal biometrics and security assertions is also provided.

2) Amendment items for interchange of certificates and security assertions, and other security aspects

2-1) Replace the last paragraph of the Scope with the following:

This part of ISO/IEC 19784 specifies a version of the BioAPI specification that is defined to have a version number described as Major 2, Minor 0, or version 2.0. It also specifies a version number described as Major 2, Minor 1, or version 2.1 that provides an enhanced Graphical User Interface. It also specifies a version number described as Major 2, Minor 2, or version 2.2 that provides features supporting fusion and security. Some clauses and sub-clauses apply only to one of these versions, some to two or more. This is identified at the head of the relevant clauses and sub-clauses.

2-2) Remove the amended paragraph in Amd.2 after the last paragraph of the Scope and add the following paragraph after the last amended NOTEs of the Scope:

Conformance requirements are specified in Clause 2.

2-3) Add the following documents to Clause 3:

ISO/IEC 19785-4:2010, *Information technology — Common Biometric Exchange Formats Framework — Part 4: Security block format specifications*

ISO/IEC 24761:2009, *Information technology — Security techniques — Authentication context for biometrics*

2-4) Add the following term and definition before 4.1:

4.0
ACBio instance
report generated by a biometric processing unit (BPU) compliant to ISO/IEC 24761 to show the validity of the result of one or more subprocesses executed in the BPU

[ISO/IEC 24761]

2-5) Add the following term and definition after 4.3:

4.3bis
authentication context for biometrics
ACBio
International Standard that specifies the form and encoding of ACBio instances

[ISO/IEC 24761]

2-6) Add the following terms and definitions after 4.5:

4.5bis
biographic data (BioAPI 2.2)
non-biometric data that potentially affects a biometric operation

4.5ter
biographic BIR (BioAPI 2.2)
non-biometric BIR that potentially affects a biometric operation

2-7) Add the following term and definition after 4.10:

4.10bis
biometric processing unit
BPU
entity that executes one or more subprocesses that perform a biometric verification at a uniform level of security

[ISO/IEC 24761]

NOTE A sensor, a smart card, and a comparison device are examples of BPUs.

2-8) Add the following term and definition after 4.14:

4.14bis
BPU IO Index
integer assigned to each biometric data stream between BPUs by the subject, such as software, which utilizes the function of the BPU so that the validator can reconstruct the data flow among BPUs

[ISO/IEC 24761]

2-9) Add the following term and definition after 4.16:

4.16bis
decision BIR (BioAPI 2.2)
BIR which contains decision in the BDB

2-10) Replace the terms of 4.22 with the following terms:

4.22
score
score data
scoring

2-11) Add the following term and definition after 4.22:

4.22bis
score BIR (BioAPI 2.2)
BIR which contains score in the BDB

2-12) Add the following term and definition after 4.22bis:

4.22ter
secure BioAPI (BioAPI 2.2)
BioAPI API and SPI interfaces that include the security features defined for version 2.2 of BioAPI

2-13) Replace Clause 5 with the following:

ACBio – Authentication Context for Biometrics

API – Application Programming Interface

BDB – Biometric Data Block

BFP – BioAPI Function Provider

BIR – Biometric Information Record

BPU – Biometric Processing Unit

BSP – Biometric Service Provider

CBEFF – Common Biometric Exchange Formats Framework

FMR – False Match Rate

FPI – Function Provider Interface

GUI – Graphical User Interface

ID – Identity/Identification/Identifier

IRI – Internationalized Resource Identifier (see RFC 3987)

MAC – Message Authentication Code

MOC – Match on Card

PID – Product ID

SB – Security Block

SBH – Standard Biometric Header

NOTE This term and abbreviation is imported from ISO/IEC 19785-1.

SPI – Service Provider Interface

UUID – Universally Unique Identifier

2-14) Replace 6.6.6 as follows:

6.6.6 On BioAPI_BSPAttach/BioAPI_BSPAttachSecure (BioAPI 2.2 or greater), the application is required to select at most one BioAPI Unit of each category that is currently in an "inserted" state (or to select BioAPI_DONT_CARE) and is managed by that BSP or by an associated BFP. The BSP then either accesses that BioAPI Unit (for directly managed BioAPI Units), or else interacts with the associated BFP in order to access that BioAPI Unit.

2-15) Add the following text after 7.1:

7.1bis BioAPI_ACBio_PARAMETERS (BioAPI 2.2)

7.1bis.1 Structure giving information which is used to generate ACBio instances

```
typedef struct bioapi_acbio_parameters {

    uint32_t Challenge[4];

    uint32_t *InitialBPUIOIndexOutput;

    uint32_t *SupremumBPUIOIndexOutput;

} BioAPI_ACBio_PARAMETERS;
```


7.1bis.2 Definitions

Challenge – Challenge from the validator of a biometric verification when ACBio is used. This value shall be set to the field `controlValue` of type `ACBioContentInformation` in ACBio instances.

InitialBPUIOIndexOutput – The initial value of BPU IO index which is to be assigned to the output from the BioAPI Unit, BFP, or BSP when the ACBio instances are generated. The range between *InitialBPUIOIndexOutput* and *SupremumBPUIOIndexOutput* shall be divided into the number of BSP Units and BFPs which are attached to the BSP, and assigned to the BSP Units and BSPs.

SupremumBPUIOIndexOutput – The supremum of BPU IO indexes which are to be assigned to the output from the BioAPI Unit, BFP, or BSP when the ACBio instances are generated.

7.1ter BioAPI_ASN1_BIR (BioAPI 2.2)

A container for biometric data, (or non-biometric data) that may affect a biometric operation, encoded in ASN.1 DER. The BioAPI_ASN1_BIR structure is used when a BioAPI API with security functionality is used. The BioAPI_ASN1_BIR structure associates a length, in bytes, with the address of an arbitrary block of contiguous memory which contains an ASN.1 encoded BIR of type `BioAPI-BIR`, specified in Annex E. The ASN.1 encoded BIR consists of an SBH of type `BioAPIBIRHeader`, a BDB of type `BiometricData`, and an SB of type `CBEFFSecurityBlock`. The BDB may contain raw sample data, partially processed (intermediate) data, completely processed data, score data (resulting from a matching or fusion operation), decision data (resulting from a decision or fusion operation), or biographic data which can be provided as input to a biometric operation to modify its behavior. The BioAPI_ASN1_BIR may be used to enroll a user (thus being stored persistently), or may be used to verify or identify a user (thus being used transiently). It may also be stored and used later to provide feedback to subsequent biometric operations. Type BioAPI_ASN1_BIR is an alias of type BioAPI_DATA.

NOTE The ASN.1 type `BioAPI-BIR` corresponds to the C structured type BioAPI_BIR element by element.


```
typedef BioAPI_DATA BioAPI_ASN1_BIR;
```

7.1quater BioAPI_ASN1_ENCODED (BioAPI 2.2)

A container for ASN.1 DER encoded data. The BioAPI_ASN1_ENCODED structure is used to express information about cryptographic keys. The BioAPI _ASN1_ ENCODED structure associates a length, in bytes, with the address of an arbitrary block of contiguous memory which contains an ASN.1 encoded data. Type BioAPI_ASN1_ENCODED is an alias of type BioAPI_DATA.

```
typedef BioAPI_DATA BioAPI_ASN1_ENCODED;
```

2-16) In 7.4.1, modify the paragraph as follows:

A container for biometric data, or non-biometric data that may affect a biometric operation. A BioAPI_BIR consists of a BioAPI_BIR_HEADER, a BDB, and an optional SB. The BDB may contain raw sample data, partially processed (intermediate) data, completely processed data, score data (resulting from a matching or fusion operation), decision data (resulting from a decision or fusion operation), or biographic data which can be provided as input to a biometric operation to modify its behavior. The BioAPI_BIR may be used to enroll a user (thus being stored persistently), or may be used to verify or identify a user (thus being used transiently). It may also be stored and used later to provide feedback to subsequent biometric operations.

2-17) In 7.9.1, modify a) with the following:

a) it identifies the type of biometric sample (raw, intermediate, processed, score data, decision data or biographic data) that is contained in the BDB;

2-18) Replace 7.9.4 as follows:

7.9.4 The 'index' flag shall be set if an index is present in the BIR header and not set if no index is present in the BIR header.

```
typedef uint8_t BioAPI_BIR_DATA_TYPE;

   #define BioAPI_BIR_DATA_TYPE_RAW             (0x01)
   #define BioAPI_BIR_DATA_TYPE_INTERMEDIATE    (0x02)
   #define BioAPI_BIR_DATA_TYPE_PROCESSED       (0x04)
   #define BioAPI_BIR_DATA_TYPE_SCORE           (0x08)
   #define BioAPI_BIR_DATA_TYPE_DECISION        (0x09)
   #define BioAPI_BIR_DATA_TYPE_BIOGRAPHIC      (0x0A)
   #define BioAPI_BIR_DATA_TYPE_ENCRYPTED       (0x10)
   #define BioAPI_BIR_DATA_TYPE_SIGNED          (0x20)
```


NOTE 1 The BioAPI BIR Data Type corresponds to a combination of the "CBEFF_BDB_processed_level", "CBEFF_BDB_encryption_options", and "CBEFF_BIR_integrity_options" in ISO/IEC 19785-1.

NOTE 2 BioAPI_BIR_DATA_TYPE_DECISION (BioAPI 2.2 or greater) and BioAPI_BIR_DATA_TYPE_BIOGRAPHIC (BioAPI 2.2 or greater) have two bits on while others have only one bit on.

NOTE 3 BioAPI_BIR_DATA_TYPE_SCORE is used in BioAPI 2.2 or greater.

2-19) Replace the text of 7.10 with the following:

A handle to refer to a BioAPI BIR or an ASN.1 encoded BIR that exists within a BSP.

2-20) Replace 7.12.1 with the following:

7.12.1 A value which defines the purpose(s) or use(s) for which the BioAPI BIR is intended (when used as an input to a BioAPI function) or suitable (when used as an output from a BioAPI function or within the BIR header).

```
typedef uint8 BioAPI_BIR_PURPOSE;

   #define BioAPI_PURPOSE_VERIFY (1)
   #define BioAPI_PURPOSE_IDENTIFY (2)
   #define BioAPI_PURPOSE_ENROLL (3)
   #define BioAPI_PURPOSE_ENROLL_FOR_VERIFICATION_ONLY (4)
   #define BioAPI_PURPOSE_ENROLL_FOR_IDENTIFICATION_ONLY (5)
   #define BioAPI_PURPOSE_AUDIT (6)
```



```
#define BioAPI_PURPOSE_DECIDE (7)

#define BioAPI_NO_PURPOSE_AVAILABLE (0)
```

NOTE 1 The condition NO VALUE AVAILABLE is indicated by setting the value to zero. This value is used only for BIRs that are not originally generated by a BioAPI BSP, but originate from another source and have been transformed into a BioAPI BIR. BSPs shall not use this value.

NOTE 2 BioAPI_PURPOSE_DECIDE is used in BioAPI 2.2 or greater.

2-21) In 7.12.3, replace e) and f) as follows:

e) The BioAPI_Process, BioAPI_CreateTemplate, BioAPI_ProcessWithAuxBIR (BioAPI 2.1 or less), BioAPI_ProcessUsingAuxBIRs (BioAPI 2.2 or greater), BioAPI_Decide (BioAPI 2.2 or greater), and BioAPI_Fuse (BioAPI 2.2 or greater) functions do not have Purpose as an input parameter, but read the Purpose field from the BIR header of the input BIR.

f) The BioAPI_Process and BioAPI_ProcessUsingAuxBIRs functions may accept as input any intermediate BIR with a Purpose of BioAPI_PURPOSE_VERIFY or BioAPI_PURPOSE_IDENTIFY, and shall output only BIRs with the same purpose as the input BIR.

2-22) In 7.12.3, add i) as follows:

i) All score BIRs and decision BIRs must have the Decide purpose. Biographic BIRs may have any purpose. No other types of BIRs may have the Decide purpose.

2-23) Add the following text after 7.25:

7.25bis BioAPI_ENCRYPTION_ALG (BioAPI 2.2)

Identifies encryption algorithm supported by a BioAPI Unit. This BioAPI type shall be the XML value notation of ASN.1 identifier (see ISO/IEC 8824-1) assigned to encryption algorithms. Example to specify AES with 128 bit keys in CBC mode, the char would be “2.16.840.1.101.3.4.1.2” which is the XML value notation for.

```
typedef char *BioAPI_ENCRYPTION_ALG;

   #define BioAPI_ENCRYPTION_ALG_NOT_SUPPORTED NULL
```

7.25ter BioAPI_ENCRYPTION_INFO (BioAPI 2.2)

7.25ter.1 Structure giving information of the cryptographic algorithm and key(s) of a BioAPI Unit or a biometric application which is used to encrypt/decrypt biometric data

```
typedef struct bioapi_encryption_info {

    BioAPI_ENCRYPTION_ALG ENCAlg;

    BioAPI_KEY_INFO *ENCKeyInfo;

} BioAPI_ENCRYPTION_INFO;
```

7.25ter.2 Definitions

ENCAlg – Encryption algorithm.

ENCKeyInfo – An array of key information for encryption.

2-24) In 7.27, replace NOTE as follows:

NOTE: It may be impossible to mask an INSERT event coming from an attach session of a BSP, because the event may occur just after a ***BioAPI_BSPLoad*** call, before any ***BioAPI_EnableEvents*** call has had any chance to be processed. This is because ***BioAPI_EnableEvents*** requires a handle which is provided by ***BioAPI_BSPAttach***/***BioAPI_BSPAttachSecure*** (BioAPI 2.2 or greater), and ***BioAPI_BSPAttach***/***BioAPI_BSPAttachSecure*** (BioAPI 2.2 or greater) itself shall follow ***BioAPI_BSPLoad***. An INSERT event will be raised by the BSP on the ***BioAPI_BSPLoad*** call if a BioAPI Unit is already "inserted", and this event will reach the application before the application can call ***BioAPI_EnableEvents***.

2-25) Replace 7.38 with the following

A unique identifier, returned on ***BioAPI_BSPAttach***/***BioAPI_BSPAttachSecure*** (BioAPI 2.2 or greater), that identifies an attached BioAPI BSP session.

```
typedef uint32_t BioAPI_HANDLE;
```


2-26) Add the following text after 7.38:

7.38bis BioAPI_HASH_ALG (BioAPI 2.2)

Identifies the hash algorithm supported by a BioAPI Unit, which is used in the generation of ACBio instance. This BioAPI type shall be the XML value notation of ASN.1 identifier (see ISO/IEC 8824-1) assigned to hash algorithms. Example to specify the SHA-1 hash algorithm, the char would be “1.3.14.3.2.26” which is the XML value notation for.

```
typedef char *BioAPI_HASH_ALG;

   #define BioAPI_HASH_ALG_NOT_SUPPORTED NULL
```

2-27) Add the following text after 7.45:

7.45bis BioAPI_KEY_INFO (BioAPI 2.2)

7.45bis.1 Union giving information of cryptographic key of a BioAPI Unit or a biometric application which is used to encrypt/decrypt biometric data or to generate/validate MAC of BIR.

```
typedef union bioapi_key_info {

    BioAPI_KEY_TRANSPORT KTInfo;

    BioAPI_ASN1_ENCODED KEKInfo;

} BioAPI_KEY_INFO;
```


7.45bis.2 Definitions

KTInfo – Key information when key management technique of key transport is used

KEKInfo – Key information when key management technique of previously distributed symmetric key-encryption keys is used

NOTE: For details of key management, see RFC 3852.

7.45ter BioAPI_KEY_TRANSPORT (BioAPI 2.2)

7.45ter.1 Structure giving information of cryptographic key of a BioAPI Unit or a biometric application, which is used to encrypt/decrypt biometric data or to generate/.validate MAC of BIR, when key management technique of key transport is used.

```
typedef struct bioapi_key_transport {

    BioAPI_ASN1_ENCODED IssuerAndSerialNumber;

    BioAPI_ASN1_ENCODED Certificate;

} BioAPI_KEY_TRANSPROT;
```

7.45ter.2 Definitions

IssuerAndSerialNumber – ASN.1 encoded data of ASN.1 type `IssuerAndSerialNumber` which contains the information of issuer and serial number of the X.509 certificate for the public key.

Certificate – ASN.1 encoded data of ASN.1 type `Certificate` which contains X.509 certificate of the public key.

NOTE: For details of the definitions of the types **`IssuerAndSerialNumber`** and **`Certificate`**, see RFC 3852.

7.45quater BioAPI_MAC_ALG (BioAPI 2.2)

Identifies the MAC algorithm supported by a BioAPI Unit. Example to specify the HMAC algorithm with SHA-1, the char would be “1.3.6.1.5.5.8.1.2” which is the XML value notation for.

```
typedef char *BioAPI_MAC_ALG;

  #define BioAPI_MAC_ALG_NOT_SUPPORTED NULL
```

7.45quinquies BioAPI_MAC_INFO (BioAPI 2.2)

7.45quinquies.1 Structure giving information of the MAC algorithm and key(s) of a BioAPI Unit or a biometric application which is used to generate/validate MAC of BIR

```
typedef struct bioapi_mac_info {

    BioAPI_MAC_ALG MACAlg;

    BioAPI_KEY_INFO *MACKeyInfo;

} BioAPI_MAC_INFO;
```

7.45quinquies.2 Definitions

MACAlg – MAC algorithm.

MACKeyInfo – An array of key information for message authentication code.

2-28) In 7.46, add the following masks:

```
#define BioAPI_PROCESSUSINGAUXBIRS        (0x01000000)
#define BioAPI_VERIFYMATCHUSINGAUXBIRS    (0x02000000)
#define BioAPI_DECIDE                     (0x04000000)
#define BioAPI_FUSE                       (0x08000000)
#define BioAPI_SECURITY                   (0x10000000)
```

2-29) In 7.47, modify the description about BioAPI_PAYLOAD as follows:

If set, the BSP supports payload carry (accepts a payload during ***BioAPI_Enroll*** or ***BioAPI_CreateTemplate*** and returns the payload upon successful ***BioAPI_Verify***, ***BioAPI_VerifyMatch***, or ***VerifyMatchUsingAuxBIRs*** (BioAPI 2.2 or greater)).

2-30) In 7.47, modify the description about BioAPI_ADAPTATION as follows:

If set, the BSP supports BIR adaptation in the return parameters of a ***Verify***, ***VerifyMatch***, or ***VerifyMatchUsingAuxBIRs*** (BioAPI 2.2 or greater) operation.

2-31) Add the following text after 7.50:

7.50bis BioAPI_SECURITY_OPTIONS_MASK (BioAPI 2.2)

A mask that indicates what security options are supported by the BioAPI Unit.

```
typedef uint32_t BioAPI_SECURITY_OPTIONS_MASK;
```

```
#define BioAPI_ENCRYPTION (0x00000001)
```

If set, indicates that the BioAPI Unit supports encryption.

```
#define BioAPI_MAC (0x00000002)
```

If set, indicates that the BioAPI Unit supports MAC generation.

```
#define BioAPI_DIGITAL_SIGNATURE (0x00000004)
```

If set, indicates that the BioAPI Unit supports digital signature.

```
#define BioAPI_ACBio_GENERATION_WITH_MAC (0x00000010)
```

If set, indicates that the BioAPI Unit supports ACBio generation using MAC.

```
#define BioAPI_ACBio_GENERATION_WITH_DIGITAL_SIGNATURE (0x00000020)
```

If set, indicates that the BioAPI Unit supports ACBio generation using digital signature.

7.50ter BioAPI_SECURITY_PROFILE (BioAPI 2.2)

7.50ter.1 Structure giving information of the cryptographic algorithms and keys of a BioAPI Unit or a biometric application which is used to encrypt/decrypt biometric data or to generate/validate the MAC or digital signature of the BIR and also giving the information of the hash algorithm, the information about the MAC generation, and the digital signature used in ACBio generation. When this structure is used in the structure of BioAPI_UNIT_SCHEMA (BioAPI 2.2) as the output parameter of ***BioAPI_QueryUnits***, the parameters in this structure indicate the information supported in the BioAPI Unit. On the other hand, when this structure is used as the parameter of ***BioAPI_BSPAttachSecure***, the parameters in this structure indicates the information which is to be used in the execution of security operations.

```
typedef struct bioapi_security_profile {

    BioAPI_SECURITY_OPTIONS_MASK SupportedSecOption;

    BioAPI_ENCRYPTION_INFO **ENCInfo;

    BioAPI_MAC_INFO **MACInfo;

    BioAPI_DIGITAL_SIGNATURE_ALG *SIGNAlg;

    BioAPI_OPERATIONS_MASK ACBioOption;

    BioAPI_HASH_ALG **HASHAlgForACBio;

    BioAPI_MAC_INFO **MACInfoForACBio;

    BioAPI_DIGITAL_SIGNATURE_ALG *SIGNAlgForACBio;

} BioAPI_SECURITY_PROFILE;
```


7.50ter.2 Definitions

SupportedSecOption – A mask which indicates which security options are supported or to be executed by the BSP Unit.

ENCInfo – Encryption information used in the encryption of the BDB.

MACInfo – MAC information used to keep the integrity of the BIR.

SIGNAlg – Digital signature algorithm used to keep the integrity of the BIR.

ACBioOption – A mask which indicates which security options of MAC or digital signature are supported or to be executed by the BSP Unit.

HASHAlgForACBio – Hash algorithm used to generate ACBio instances.

MACInfoForACBio – MAC information used to generate ACBio instances.

SIGNAlgForACBio – Digital signature algorithm used to generate ACBio instances.

7.50quater BioAPI_DIGITAL_SIGNATURE_ALG (BioAPI 2.2)

Identifies the digital signature algorithm supported by a BioAPI Unit. This BioAPI type shall be the XML value notation of ASN.1 identifier (see ISO/IEC 8824-1) assigned to digital signature algorithms. Example to specify the digital signature algorithm using SHA1 with RSA according to ISO/IEC 9796-2, the char would be “1.3.36.3.4.3.2.1” which is the XML value notation for.