INTERNATIONAL STANDARD

ISO/IEC 18033-5

First edition
2015-12-01

AMENDMENT 1
2021-02

Information technology — Security techniques — Encryption algorithms —

Part 5: Identity-based ciphers

AMENDMENT 1: SM9 mechanism

Technologies de l'information — Techniques de sécurité — Algorithmes de chiffrement —

Partie 5: Chiffrements identitaires

AMENDEMENT 1: Mécanisme SM9

Reference number
ISO/IEC 18033-5:2015/Amd.1:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: 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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC1, *Information technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection*.

A list of all parts in the ISO/IEC 18033 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Information technology — Security techniques — Encryption algorithms —

Part 5: Identity-based ciphers

AMENDMENT 1: SM9 mechanism

Introduction

Replace the second sentence of the fourth paragraph with the following:

> The specified mechanisms are the BF identity-based encryption mechanism, the SK identity-based key encapsulation mechanism, the BB1 identity-based key encapsulation mechanism and the SM9 identity-based key encapsulation mechanism and encryption mechanisms.

Insert the following sentence between the sixth and seventh paragraphs.

> The content of 9.4 follows Reference [8].

4.1

Insert the following line at the end of the table:

> $\lceil x \rceil$ smallest integer greater than or equal to the real number x.

5.1

Replace the first sentence with the following:

> The schemes specified in this document make use of four cryptographic transformations, *IHF1*, *SHF1*, *PHF1* and *IHF2* as specified below.

5.1

Add the following to the end:

> Annex A lists the object identifiers which shall be used to identify the algorithms specified in this document.
>
> Annex B describes security considerations for each specified mechanism.
>
> Annex C provides numerical examples.
>
> Annex D introduces techniques which can be used to remove the decryption capability of the PKG, and thereby reduce the level of trust required in this entity.

5.5

Add new subclause 5.5 as follows:

5.5 The function *IHF2*

IHF2 is based on the key derivation function *KDF2* defined in ISO/IEC 18033-2. *KDF2*(x, l) parameterized by a cryptographic hash function takes an octet string x and a non-negative integer l as input, and outputs an octet string of length l. *KDF2-a*(x, b) outputs the first b bits from *KDF2*(x, $\lceil b/8 \rceil$). *IHF2* take three items as input and outputs an integer in a specified range.

Input:

— A bit string $str \in \{0,1\}^*$

— A security parameter $\kappa \in \{128\}$

— A non-negative integer n with bit-length b_n

Output:

— An integer x, $0<x<n$.

Operation: Perform the following steps.

a) If $\kappa = 128$, *KDF2* uses SM3 as the hash function.

b) Let $hlen = 8\lceil(5\ b_n)/32\rceil$.

c) Compute $Ha = KDF2\text{-}a(str, hlen)$.

d) Output (*BS2IP* (*Ha*) mod (n-1)) + 1.

7.3.1

Replace the fifth paragraph with the following:

The allowable data encapsulation mechanisms are those described in ISO/IEC 18033-2.

7.4.1

Insert new NOTE 4 at the end as follows:

NOTE 4 The third mechanism defined in 9.4 will work to encrypt messages with either DEM2 or DEM3, which are specified in ISO/IEC 18033-2. In these DEMs, the required hash function is SM3, specified in ISO/IEC 10118-3, and the required block cipher is described in ISO/IEC 18033-3. The required message authentication code is generated by the evaluation function $MA.eval(K'', MS) = SM3(MS \,||\, K'')$, where K'' is a secret key which is part of the session key K, and MS is the octet string to be authenticated as specified in DEM2 and DEM3. The label input to both DEMs is empty.

9.1

Replace the first sentence with the following:

In this clause, three identity-based key encapsulation mechanisms are specified. These mechanisms use the following primitives.

Replace list item b) with the following:

b) Four hash functions:

Add new fourth list item as follows:

— H_4: $\{0,1\}^* \rightarrow Z_p^*$ where $H_4(s) = IHF2(0x01 \,||\, s \,||\, 0x03, p, \kappa)$

9.4

Add new Subclause 9.4 as follows:

9.4 The SM9 key encapsulation mechanism

9.4.1 Set up

The setup operation creates public system parameters and a master-secret key. This operation shall be completed by the private key issuer, an entity which shall be trusted by its subscribers.

The steps to create public system parameters and a master-secret key are:

a) Establish the set of base groups G_1, G_2, G_3, and a pairing e: $G_1 \times G_2 \rightarrow G_3$. The order of each group is p.

b) Select a random generator Q_1 in G_1 and a random generator Q_2 in G_2.

c) Generate a random master secret s in Z_p^*. Calculate the corresponding R as sQ_1.

d) Pre-calculate the pairing value $J = e(R, Q_2)$.

e) Make the system parameters and the master-public key set $params = <J, Q_1, Q_2, G_1, G_2, G_3, e, p>$ and $mpk = R$ available. Secure the master-secret key $msk = s$.

9.4.2 Private key extraction

The extract operation takes an arbitrary identity string ID_b in $\{0,1\}^*$ and calculates the corresponding private key sk_{ID} in G_2. The algorithm to compute the private key sk_{ID} corresponding to an identity string ID_b is as follows:

Input:

— The system parameters $params = <J, Q_1, Q_2, G_1, G_2, G_3, e, p>$

— The master-public key $mpk = R$

— The master-secret key $msk = s$

— An identity string ID_b

Output:

— The derived private key sk_{ID}, an element of G_2.

Operation: Use the following steps to compute sk_{ID}.

a) Compute $M = H_4(ID_b)$.

b) If $M + s = 0 \bmod p$, output "error" and stop.

c) Compute $t = (M + s)^{-1} s \bmod p$.

d) Compute $sk_{ID} = tQ_2$.

e) Output sk_{ID}.

The correctness of the value sk_{ID} can be verified by using the following algorithm:

Input:

— The system parameters *params* = <J, Q_1, Q_2, G_1, G_2, G_3, e, p>

— The master-public key $mpk = R$

— An identity string ID_b

— The corresponding private key sk_{ID}

Output:

— The value "valid" if sk_{ID} is consistent with *params*, *msk* and ID_b, and "invalid" otherwise.

Operation: Use the following steps.

a) Compute $M = H_4(ID_b)$.

b) Compute $T = e(MQ_1 + R, sk_{ID})$.

c) If $T = J$, then output the value "valid", otherwise output the value "invalid".

9.4.3 Session key encapsulation

The encapsulate operation (*KEM.Enc*) takes an arbitrary identity string ID_b in $\{0,1\}^*$ and the master-public key $mpk = R$ with the system parameters *parms*, and outputs the pair <K, CT_{KEM}> where K is a session key to be used to encrypt a message, and CT_{KEM} is the encapsulation of K to be transmitted to the receiver.

The steps to compute the encapsulation values are:

a) Select a random integer r in Z_p^*.

b) Compute $M = H_4(ID_b)$.

c) Compute $E = r(MQ_1 + R)$.

d) Compute $B = J^r$.

e) Compute $K = KDF2\text{-}a(EC2OSP(E) \,||\, FE2OSP(B) \,||\, ID_b, klen)$, where *klen* is the bit-length of the required session key.

f) Set $CT_{KEM} = EC2OSP(E)$.

g) Output <K, CT_{KEM}>.

9.4.4 Session key de-encapsulation

The de-encapsulate operation (*KEM.Dec*) takes an encapsulated value CT_{KEM} computed for identity ID_b and the private sk_{ID} that corresponds to ID_b, and computes the key value K that can be used to decrypt the message that was encrypted by the sender.

The steps to compute the de-encapsulation key are:

a) Parse CT_{KEM} as an element $E = OS2ECP(CT_{KEM})$.

b) Check whether E is in G_1; if not, output "error".

c) Compute $B = e(E, sk_{ID})$.

d) Compute $K = KDF2\text{-}a(EC2OSP(E) \,||\, FE2OSP(B) \,||\, ID_b, klen)$, where *klen* is the bit-length of the required session key.

e) Output K.

Annex A

Insert the following lines after ib-enc-mechanism-bf:

```
ib-enc-mechanism-sm9a OID ::= { ib-enc sm9a(2) }    -- sm9 kem with DEM2 as in 7.4.1
ib-enc-mechanism-sm9b OID ::= { ib-enc sm9b(3) }    -- sm9 kem with DEM3 as in 7.4.1
```

Insert the following lines after ib-kem-mechanism-bb1:

```
ib-kem-mechanism-sm9 OID ::= { ib-kem sm9(3) }
sm9-dem-one-time-mac OID ::= { ib-kem-mechanism-sm9 one-time-mac(1) }
```

Insert the following lines after { OID ib-enc-mechanism-bf PARMS HashFunction }:

```
|{ OID ib-enc-mechanism-sm9a PARMS HashFunction }
|{ OID ib-enc-mechanism-sm9b PARMS HashFunction }
```


Insert the following line after { OID ib-kem-mechanism-bb1 PARMS HashFunction }:


```
|{ OID ib-kem-mechanism-sm9 PARMS HashFunction }
```


Annex B

Replace the last sentence with the following:

> Security analyses of the BF, SK, BB1 and SM9 mechanisms can be found in References [4], [5], [3] and [9], respectively.

Annex C

Add new Clause C.4 as follows:

C.4 SM9 ID-based key encapsulation mechanism

C.4.1 Example 1

C.4.1.1 Set up

This example makes use of the same Barreto-Naehrig elliptic curve $y^2 = x^3 + 5$ used in ISO/IEC 14888-3:2018, F.15.1. An element $\underline{A}_0$ in Fq^2 is represented as $\underline{A}_{0,1}\sigma+\underline{A}_{0,0}$, where $\underline{A}_{0,0}$ and $\underline{A}_{0,1}$ are elements of Fq and σ is an element of Fq^2 such that $\sigma^2 + 2 = 0 \bmod q$. Let ν be an element of Fq^4 such that $\nu^2 - \sigma = 0$ in Fq^2 and ω be an element of Fq^{12} such that $\omega^3 - \nu = 0$ in Fq^4, an element of Fq^{12} is represented as $\underline{A}\omega^2+\underline{B}\omega+\underline{C}$, where $\underline{A}$, $\underline{B}$, $\underline{C}$ are elements of Fq^4 which are represented as $\underline{A} = \underline{A}_1\nu + \underline{A}_0$, $\underline{B} = \underline{B}_1\nu + \underline{B}_0$, $\underline{C} = \underline{C}_1\nu + \underline{C}_0$ respectively, and $\underline{A}_0$, $\underline{A}_1$, $\underline{B}_0$, $\underline{B}_1$, $\underline{C}_0$, $\underline{C}_1$ are elements of Fq^2. In this towered fashion, an element of Fq^{12} is represented as a vector $(\underline{A}_{1,1}, \underline{A}_{1,0}, \underline{A}_{0,1}, \underline{A}_{0,0}, \underline{B}_{1,1}, \underline{B}_{1,0}, \underline{B}_{0,1}, \underline{B}_{0,0}, \underline{C}_{1,1}, \underline{C}_{1,0}, \underline{C}_{0,1}, \underline{C}_{0,0})$ with components in Fq. P is a point on the curve $y^2 = x^3 + 5$, Q is a point on the corresponding sextic twist $y^2 = x^3 + 5\sigma$. Pairing e is implemented as the optimal R-ate pairing on two input points P and $\phi(Q)$, i.e.,

$e(P,Q)=[d(\phi(Q),P,u)\cdot f(u\phi(Q),\pi_q(\phi(Q)),P)\cdot f(u\phi(Q)+\pi_q(\phi(Q)),-\pi_{q^2}(\phi(Q)),P)]^{\frac{q^{12}-1}{p}}$, where $q = 36z^4 + 36z^3 + 24z^2 + 6z + 1$, $p = 36z^4 + 36z^3 + 18z^2 + 6z + 1$, $u = 6z + 2$, z is the parameter of the BN curve, ϕ is the group homomorphism such that $\phi(x, y) = (\sigma^{-1/3}x, \sigma^{-1/2}y)$ and π is the Frobenius endomorphism such that $\pi_{q^i}(x,y)=\left(x^{q^i}, y^{q^i}\right)$ for $i = 1, 2$.

```
q   = b6400000 02a3a6f1 d603ab4f f58ec745 21f2934b 1a7aeedb e56f9b27 e351457d
p   = b6400000 02a3a6f1 d603ab4f f58ec744 49f2934b 18ea8bee e56ee19c d69ecf25
Q1x = 93de051d 62bf718f f5ed0704 487d01d6 e1e40869 09dc3280 e8c4e481 7c66dddd
Q1y = 21fe8dda 4f21e607 63106512 5c395bbc 1c1c00cb fa602435 0c464cd7 0a3ea616
Q2x = 85aef3d0 78640c98 597b6027 b441a01f f1dd2c19 0f5e93c4 54806c11 d8806141,
      37227552 92130b08 d2aab97f d34ec120 ee265948 d19c17ab f9b7213b af82d65b
Q2y = 17509b09 2e845c12 66ba0d26 2cbee6ed 0736a96f a347c8bd 856dc76b 84ebeb96,
      a7cf28d5 19be3da6 5f317015 3d278ff2 47efba98 a71a0811 6215bba5 c999a7c7
s   = 0001edee 3778f441 f8dea3d9 fa0acc4e 07ee36c9 3f9a0861 8af4ad85 cede1c22
Rx  = 787ed7b8 a51f3ab8 4e0a6600 3f32da5c 720b17ec a7137d39 abc66e3c 80a892ff
Ry  = 769de617 91e5adc4 b9ff85a3 1354900b 20287127 9a8c49dc 3f220f64 4c57a7b1
J   = 9746fc5b 231cedf3 6f835c47 893d63c6 ff652bcb 92375ce3 c2ab256d 1fd56413,
      232a2f80 cfbae061 f196bb99 213d5030 6648ac33 cdc78e8f 8a1563ff bf3bd3eb,
      68e8a16c 0ac905f6 92904abc c004b1ac f12106bd 0a15b6e7 08d76e72 b9288ef2,
      9436a60c 403f4f8b ac4dd3e3 93e25419 e634fc2b 3daf247f 6092a802 f60d5c58,
      a140eaef 3893d574 cb83c01d 951a53f5 1975760b e57f3bbd 89817498 d2158352,
      95a2bcce 25359d03 3fc654bd 6a9e462e 5bd0686f f6ddd745 5f71fff1 5affd3f0,
      b0432019 0b1e90ce df6ac570 147a23ae 6f0eae45 034e6c62 124dd6e8 978f78ad,
      a504e3b4 3c1dd367 94217fa1 b05ac046 c4131854 c3d3e3a5 b5967a64 a861f0a2,
      897f7b35 d1c0e21d 84d75cff ac08c73e 744a16a4 7ee76e28 a0b03849 888d10ff,
      24443bb4 24b12c41 eaf6d34d 92520590 1f5cba59 cfeba352 24660db3 848b0bf5,
      0825403f b3f681ab 2b036dbb a25483d5 cb98bd56 f3df95f0 a7a705a2 f6fd804b,
      9ce7bc68 062182cf 5d9f4a98 c5a4ed1f 3b4ce4ea 817d19ed 7ef2ce98 e6f5864d
```


C.4.1.2 Private key extraction


```
ID    = Bob
IDb   = 426f62
M     = 9cb1f628 8ce0e510 43ce7234 4582ffc3 01e0a812 a7f5f200 4b85547a 24b82716
t     = 864e4d83 91948b37 535ecfa4 4c3f8d4e 545ada50 2ff8229c 7c32f529 af406e06
skIDx 94736acd 2c8c8796 cc4785e9 38301a13 9a059d35 37b64141 40b2d31e ecf41683,
        115bae85 f5d8bc6c 3dbd9e53 42979acc cf3c2f4f 28420b1c b4f8c0b5 9a19b158
skIDy = 7aa5e475 70da7600 cd760a0c f7beaf71 c447f384 4753fe74 fa7ba92c a7d3b55f,
        27538a62 e7f7bfb5 1dce0870 4796d94c 9d56734f 119ea447 32b50e31 cdeb75c1
T     = 9746fc5b 231cedf3 6f835c47 893d63c6 ff652bcb 92375ce3 c2ab256d 1fd56413,
        232a2f80 cfbae061 f196bb99 213d5030 6648ac33 cdc78e8f 8a1563ff bf3bd3eb,
        68e8a16c 0ac905f6 92904abc c004b1ac f12106bd 0a15b6e7 08d76e72 b9288ef2,
        9436a60c 403f4f8b ac4dd3e3 93e25419 e634fc2b 3daf247f 6092a802 f60d5c58,
        a140eaef 3893d574 cb83c01d 951a53f5 1975760b e57f3bbd 89817498 d2158352,
        95a2bcce 25359d03 3fc654bd 6a9e462e 5bd0686f f6ddd745 5f71fff1 5affd3f0,
        b0432019 0b1e90ce df6ac570 147a23ae 6f0eae45 034e6c62 124dd6e8 978f78ad,
        a504e3b4 3c1dd367 94217fa1 b05ac046 c4131854 c3d3e3a5 b5967a64 a861f0a2,
        897f7b35 d1c0e21d 84d75cff ac08c73e 744a16a4 7ee76e28 a0b03849 888d10ff,
        24443bb4 24b12c41 eaf6d34d 92520590 1f5cba59 cfeba352 24660db3 848b0bf5,
        0825403f b3f681ab 2b036dbb a25483d5 cb98bd56 f3df95f0 a7a705a2 f6fd804b,
        9ce7bc68 062182cf 5d9f4a98 c5a4ed1f 3b4ce4ea 817d19ed 7ef2ce98 e6f5864d
Private key skID is valid.
```

C.4.1.3 Session key encapsulation

```
r   = 0000aac0 541779c8 fc45e3e2 cb25c12b 5d2576b2 129ae8bb 5ee2cbe5 ec9e785c
IDb = 426f62
M   = 9cb1f628 8ce0e510 43ce7234 4582ffc3 01e0a812 a7f5f200 4b85547a 24b82716
Ex  = 24454711 64490618 e1ee2052 8ff1d545 b0f14c8b caa44544 f03dab5d ac07d8ff
Ey  = 42ffca97 d57cddc0 5ea405f2 e586feb3 a6930715 532b8000 759f1305 9ed59ac0
B   = 63253798 b7535975 a90f2025 61fc5457 0fee88bf 69e3b7a5 12697069 e59e1f5d,
      42d54b98 4af01d71 0ba0030c 18738f6b 14e4df47 2acaf893 99228d85 af117904,
      b426dff0 40c49f9a 43bcd7fd 7d757b7d 1d8d7311 c08fc3b5 7616c5ee 137785a3,
      28d19396 dbdfac50 eee62b1c 7f994bb6 f9bd9efb 2221a1be 1b6eb3e8 f71485b4,
      a3eef46e 1b99f614 d7bd7f57 574ba7eb b502af0b daba0787 c5c4dbc5 6a344a25,
      a06790b6 05cea0bb af34776d 6b1fc019 8a02d05b baac6f64 a555ab2c a576f0da,
      b405cbbf 22197b94 fd18d27d a0b0e52c 8754ee94 27963469 1fea6e13 ffd0584e,
```