DRAFT AMENDMENT

ISO/IEC 14888-3:2016 DAM 1

ISO/IEC JTC **1**/SC **27** Secretariat: **DIN**

Voting begins on: **2017-07-20**

Voting terminates on: **2017-10-11**

Information technology — Security techniques — Digital signatures with appendix —

Part 3: Discrete logarithm based mechanisms

AMENDMENT 1: SM2 digital signature mechanism

Technologies de l'information — Techniques de sécurité — Signatures numériques avec appendice —

Partie 3: Mécanismes basés sur un logarithme discret

AMENDEMENT 1: .

ICS: 35.030

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC 14888-3:2016/DAM 1:2017(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

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.

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).

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

For an explanation on 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 the following URL www.iso.org/iso/foreword.html.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC XXX

This second/third/... edition cancels and replaces the first/second/... edition (), [clause(s) / subclause(s) / table(s) / figure(s) / annex(es)] of which [has / have] been technically revised.

ISO XXXX consists of the following parts. [Add information as necessary.]

Information technology - Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms

— Amendment 1: SM2 + Chinese IBS + KR Defect report

AA: Page 3, Clause 4:

Change the following paragraph:

ID a data string containing an identifier of the signer, used in Mechanisms IBS-1 and IBS-2

to:

ID a data string containing an identifier of the signer, used in Mechanisms SM2, IBS-1, IBS-2 and Chinese IBS

BB: Page 4, Clause 4:

Change the following paragraph:

P a generator of G_1 which is used in Mechanisms IBS-1 and IBS-2

to:

P a generator of G_1 which is used in Mechanisms IBS-1, IBS-2 and Chinese IBS.

CC: Page 7, 5.2.1:

Change the following sentence of paragraph 2:

given that (*A*, *B*, *C*) is a permutation of (*S*, T_1, T_2),

to:

given that (*A*, *B*, *C*) is a permutation of (*S*, T_1, T_2) or (*S*, T_1, $S + T_2$),.

DD: Page 7, 5.2.1:

Change the following sentence of paragraph 4:

Given that (A, B, C) is a permutation of (S, T_1, T_2), U is the master private key and D is a parameter depending on the particular mechanism.

to:

Given that (A, B, C) is a permutation of (S, T_1, T_2) or (S, T_1, $[Y^{-1}]S + T_2$), U is the master private key, Y is the public verification key and D is a parameter depending on the particular mechanism.

EE: Page 8, 5.2.4:

Change the following sentence:

In the process of preparing the message, one of M_1 and M_2 becomes message M, the other becomes empty.

to:

In the process of preparing the message, one of M_1 and M_2 becomes message M (with a prefix, optionally), the other becomes empty.

FF: Page 12, 6.1:

Change the first sentence of paragraph 1:

Clause 6 specifies ten certificate-based mechanisms.

to:

Clause 6 specifies eleven certificate-based mechanisms.

GG: Page 12, 6.1:

Change the last sentence of paragraph 1:

The mechanisms using arithmetic in the additive group of elliptic curve points are the Elliptic Curve DSA (EC-DSA), the Elliptic Curve KCDSA (EC-KCDSA), the Elliptic Curve German Digital Signature Algorithm (EC-GDSA), The Elliptic Curve Russian Digital Signature Algorithm (EC-RDSA), the Elliptic Curve Schnorr Digital Signature Algorithm (EC-SDSA), and the Elliptic Curve Full Schnorr Digital Signature Algorithm (EC-FDSA).

to:

The mechanisms using arithmetic in the additive group of elliptic curve points are the Elliptic Curve DSA (EC-DSA), the Elliptic Curve KCDSA (EC-KCDSA), the Elliptic Curve German Digital Signature Algorithm (EC-GDSA), the Elliptic Curve Russian Digital Signature Algorithm (EC-RDSA), the Elliptic Curve Schnorr Digital Signature Algorithm (EC-SDSA), the Elliptic Curve Full Schnorr Digital Signature Algorithm (EC-FSDSA) and the SM2 algorithm.

HH: Page 12, 6.1:

Change the last sentence of paragraph 2:

Elliptic curves for EC-DSA, EC-KCDSA, EC-GDSA, EC-RDSA, EC-SDSA and EC-FSDSA are restricted to non-singular and non-supersingular curves.

to:

Elliptic curves for EC-DSA, EC-KCDSA, EC-GDSA, EC-RDSA, EC-SDSA, EC-FSDSA and SM2 are restricted to non-singular and non-supersingular curves.

II: Page 39, After 6.11.5.7:

Add the following new 6.12 thru 6.12.5.7 after 6.11.5.7:

6.12 SM2

6.12.1 Introduction

The SM2 algorithm is a signature mechanism based on elliptic curves with verification key $Y=[X]G$; that is, the parameter D is equal to 1. The message is prepared such that M_1 is the concatenation of the hash-code Z and the message M to be signed, where Z is the hash-code of a message that is the concatenation of the length of ID (the identifier of the signing entity), the value of ID, a_1, a_2, the x-coordinate of G, the y-coordinate of G, the x-coordinate of Y and the y-coordinate of Y, i.e., $M_1 = Z||M$, and M_2 is empty. The witness function is defined by the formula

$$R = BS2I(\gamma, h(M_1)) + FE2I(r, \Pi_X) \bmod q.$$

The conversion rules, *BS2I* and *FE2I*, are given in Annex B.

The assignment function is defined by the formula

$$(T_1, T_2) = (-1, R).$$

The coefficients (A, B, C) of the SM2 signature equation are set as follows:

$$(A, B, C) = (T_1, S+T_2, S).$$

Thus the signature equation becomes

$-K+(R+S)X+S \equiv 0 \pmod{q}$.

NOTE The SM2 signature mechanism is taken from [42]. The notation here has been changed from [42] to conform with the notation used in ISO/IEC 14888-3.

6.12.2 Parameters

F a finite field

$\boldsymbol{E}$ an elliptic curve group over field ***F***

$\#\boldsymbol{E}$ the cardinality of $\boldsymbol{E}$

q a prime divisor of $\#\boldsymbol{E}$

G a point on the elliptic curve of order q

Hash-function identifier or OID with specified hash-function

All these parameters can be public and can be common to a group of users.

NOTE It is recommended that all users check the proper generation of the public parameters.

6.12.3 Generation of signature key and verification key

The signature key of a signing entity is a secretly generated random or pseudo-random integer X such that $0<X<q-1$. The parameter D is 1. The corresponding public verification key Y is

$Y = [X]G$.

A user's secret signature key X and public verification key Y are normally fixed for a period of time. The signature key X shall be kept secret.

6.12.4 Signature process

6.12.4.1 Producing the randomizer

The signing entity generates a random or pseudo-random integer K such that $0 < K < q$.

6.12.4.2 Producing the pre-signature

The input to this stage is the randomizer K and the signing entity computes $\Pi = [K]G$.

6.12.4.3 Preparing message for signing

Let *entlen* be the bit-length of a distinguishing identifier *ID* of the signing entity. Let *ENTL* be two bytes string transformed from the integer *entlen*, i.e., $ENTL=I2BS(16,entlen)$. Then Z can be computed as follows

$Z=h(ENTL \,||\, ID \,||\, FE2BS(r,a_1) \,||\, FE2BS(r,a_2) \,||\, FE2BS(r,G_X) \,||\, FE2BS(r,G_Y) \,||\, FE2BS(r,Y_X) \,||\, FE2BS(r,Y_Y))$.

The message is prepared such that M_1 is the concatenation of the hash-code Z and the message M to be signed, i.e., $M_1 = Z||M$ and M_2 is empty.

The conversion rules, *I2BS* and *FE2BS*, are given in Annex B.

6.12.4.4 Computing the witness

The signing entity computes $H = h(M_1)$, and then computes $R = (BS2I(\gamma, H) + FE2I(r, \prod_X)) \bmod q$.

6.12.4.5 Computing the assignment

The signing entity computes the assignment $(T_1, T_2)=(-1, R)$.

6.12.4.6 Computing the second part of the signature

The signature is (R, S) where R is computed in 6.12.4.4, and

$$S = ((1 + X)^{-1}(K - RX)) \bmod q.$$

It is required to check if $R = 0$, $R+K=q$, or $S = 0$. If one of $R = 0$, $R+K=q$, or $S = 0$ holds, a new value of K should be generated and the signature should be recalculated.

NOTE 1 It is extremely unlikely that $R = 0$, $R+K=q$, or $S = 0$ if signatures are generated properly.

NOTE 2 It is easy to see that $R+S=(1+X)^{-1}(R+K) \bmod q$. In view of $0<R+K<2q$, we have $R+S=0 \pmod q$ iff $R+K=q$.

6.12.4.7 Constructing the appendix

The appendix will be the concatenation of (R, S) and an optional text field, *text*, $((R, S), text)$.

6.12.4.8 Constructing the signed message

A signed message is the concatenation of the message, M, and the appendix.

$M||((R, S), text)$

6.12.5 Verification process

6.12.5.1 General

The verifying entity acquires the necessary data items required for the verification process.

6.12.5.2 Retrieving the witness

The verifier retrieves the witness R and the second part of the signature S from the appendix. The verifier then first checks to see that $0 < R < q$ and $0 < S < q$; if either condition is violated, the signature shall be rejected.

6.12.5.3 Preparing message for verification

The verifier retrieves M from the signed message and divides the message into two parts M_1 and M_2. $M_1= Z||M$, where $Z=h(ENTL||ID||FE2BS(r,a_1)||FE2BS(r,a_2)||FE2BS(r,G_X)||FE2BS(r,G_Y)||FE2BS(r,Y_X)||FE2BS(r,Y_Y))$, and M_2 is empty.

6.12.5.4 Retrieving the assignment

The input to the assignment function consists of the witness R from 6.12.5.2. The assignment $T = (T_1, T_2) =(-1, R)$.

6.12.5.5 Recomputing the pre-signature

The inputs to this stage are system parameters, verification key Y, assignment $T = (T_1, T_2)$ from 6.12.5.4 and the second part of the signature S from 6.12.5.2. The verifier computes $W= (T_2+S) \bmod q$, and checks if $W=0$. If the equation $W=0$ holds, the signature shall be rejected.

The verifier then obtains a recomputed value Π' of the pre-signature by computing it using the formula

$$\Pi' = [S]G + [W]Y.$$

6.12.5.6 Recomputing the witness

The computations at this stage are the same as in 6.12.4.4. The verifier executes the witness function. The inputs are Π' from 6.12.5.5 and M_1 from 6.12.5.3. The output is the recomputed witness R'.

6.12.5.7 Verifying the witness

The verifier compares the recomputed witness, R' from 6.12.5.6 to the retrieved version of R from 6.12.5.2. If $R' = R$, then the signature is verified.

JJ: Page 40, 7.1:

Change the first sentence of paragraph 1:

Clause 7 specifies two identity-based mechanisms that are based on pairings.

to:

Clause 7 specifies three identity-based mechanisms that are based on pairings.

KK: Page 45, After 7.3.5.7:

Add the following new 7.4 thru 7.4.5.7 after 7.3.5.7:

7.4 Chinese IBS

7.4.1 Introduction

The Chinese IBS algorithm is an identity-based signature scheme on an additive group of elliptic curve points. It takes

$$(A, B, C) = (T_1, S, [Y^{-1}]S+T_2),$$

where $T_1 = [-Y^{-1}]P$, $T_2 =[Y^{-1}R]P$, $D = -1$. Thus the signature equation becomes

$$[-KY^{-1}]P +[U^{-1}]S + [Y^{-1}]S +[Y^{-1}R]P \equiv 0_E \text{ (in } G_1).$$

NOTE The Chinese IBS signature mechanism is taken from [44]. The notation here has been changed from [44] to conform with the notation used in ISO/IEC 14888-3.

7.4.2 Parameters

The signature mechanism takes place in an environment where the entities involved share the following parameters, which have been defined in Clause 4: G_1, G_2, P, Q, q, < >, and h.

Given a hash function h with output bit length v, a non-negative integer n with bit-length b_n, and a bit string Z, the function $h_i(Z, n)$ for i =1,2 is defined as follows:

1. Set a 32-bit counter ct_1= 0x00000001, and let $hlen$=8⌈(5 b_n)/32⌉, where ⌈z⌉ denotes the smallest integer no less than z.
2. For j = 1 to ⌈$hlen$ /v⌉, let $Ha_j = h(0x0i \,||Z||ct_j)$ and then set $ct_{j+1}=ct_j+1$.
3. Set Ha as the first $hlen$ bits of $Ha_1 \,||\, Ha_2 \,||\, \cdots \,||\, Ha_{\lceil hlen\ /v \rceil}$.
4. Output $(BS2I\ (Ha)$ mod $(n\text{-}1))+1$.

NOTE It is recommended that all users check the proper generation of the public parameters.

7.4.3 Generation of master key and signature/verification key

A master key pair of a KGC is (U, V), where U is the master private key generated by choosing an integer such that 0 < U < q at random, and V is the master public key generated by computing $V = [U]Q$. The KGC publishes V and keeps U secret.

A signature and verification key pair of a signer is (X, Y), where Y is the public verification key generated from an identity string ID, an identifier of the private key generation function hid, and the function h_1, i.e., $Y = h_1(ID||hid, q)$, and X is the private signature key generated by computing $X = [U(U+Y)^{-1}]P$, which is done by the KGC and given to the signer. If $U+Y$ mod q = 0, KGC generates another master key pair, publishes the master public key and updates private signature keys.

7.4.4 Signature process

7.4.4.1 Producing the randomizer

The signing entity generates a random or pseudo-random integer K such that 0 < K < q. The signer keeps the value K secret.

7.4.4.2 Producing the pre-signature

The signer takes K, P and V as input to produce the pre-signature result

$$\Pi = \langle P, V \rangle^K.$$

NOTE The pairing <P, V> can be pre-computed.

7.4.4.3 Preparing message for signing

The signer prepares the message such that M_2 is empty and M_1 is the signed message M, i.e., $M_1 = M$.

7.4.4.4 Computing the witness

Let $\Pi = (\Pi_a, \Pi_b)$. The signer applies the function h_2 to the concatenation of M_1, $FE2BS(r,\Pi_a)$ and $FE2BS(r,\Pi_b)$ to obtain the witness

$$R = h_2(M_1 \,||\, FE2BS(r,\Pi_a) \,||\, FE2BS(r,\Pi_b), q).$$

If K-R mod q = 0, a new value of K should be generated and the signature should be recalculated.

For fields of higher extension degree, more terms will appear in the value to be hashed. For example, for extension degree 3, $\Pi = (\Pi_a, \Pi_b, \Pi_c)$ and the input to h_2 would be

$$M_1 \,||\, FE2BS(r,\Pi_a) \,||\, FE2BS(r,\Pi_b) \,||\, FE2BS(r,\Pi_c).$$