INTERNATIONAL STANDARD

ISO/IEC 20008-2

First edition
2013-11-15

AMENDMENT 2
2023-04

Information technology — Security techniques — Anonymous digital signatures —

Part 2:
Mechanisms using a group public key

AMENDMENT 2

Technologies de l'information — Techniques de sécurité — Signatures numériques anonymes —

Partie 2: Mécanismes utilisant une clé publique de groupe

AMENDEMENT 2

Reference number
ISO/IEC 20008-2:2013/Amd. 2:2023(E)

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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection*.

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 and www.iec.ch/national-committees.

Information technology — Security techniques — Anonymous digital signatures —

Part 2: **Mechanisms using a group public key**

AMENDMENT 2

Clause 4

Add the following symbol:

$F(p)$ the finite field containing exactly p elements.

6.1

Replace the first sentence with the following:

This clause specifies five digital signature mechanisms with linking capability.

Replace the text of NOTE 1 with the following:

In the literature, the mechanism of 6.2 is called a list signature scheme, the mechanism of 6.6 is called a pre-DAA scheme and the mechanisms of 6.3, 6.4 and 6.5 are called DAA schemes. The mechanisms given in 6.2, 6.4, 6.5 and 6.6 are based on schemes originally specified in References [9], [6], [11] and [22] respectively, in which security proofs can also be found. The mechanism in 6.3 is based on a scheme in Reference [3] which is a minor modification of the scheme in Reference [4]; the associated security analysis is given in the full version of Reference [4].

6.6

Add new subclause 6.6 as follows:

6.6 Mechanism 8

6.6.1 Symbols

The following symbols apply in the specification of this mechanism.

— τ : a security parameter.

— P_1, Q_1, X_1, Y_1, $X_1^{'}$, $\tilde{X}_1$, C_1, D, D', T_1, T_2, K_1, K_2, K, $K_1^{'}$, $K_2^{'}$, K', J, $T_1^{'}$, $T_2^{'}$, R, R', T, T', R'', T'' : elements of G_1.

— P_2, X_2, Y_2, $X_2^{'}$, $\tilde{X}_2$: elements of G_2.

— $x, y, z, x', z', c_k, s_x, s_z, c_k', s_1, u, v, w, v', r, s_2, k_r, k_x, k_z, c, z_r, z_x, z_z, c', s, l, k_s, c_m, \rho, c_m'$: integers in Z_p.

— n_I : an integer of size τ -bit.

— H_1 : a hash function that outputs elements in G_1 .

— H_2 , H_3 : hash functions that output elements in Z_p .

6.6.2 Key generation process

The key generation process has two parts: setup process and group membership issuing process. The setup process is executed by the group membership issuer to create the group public parameter, group public key, and group membership issuing key. The group membership issuing process is an interactive protocol running between the group membership issuer and a group member to create a unique group member signature key for the group member.

The setup process takes the following steps by the group membership issuer:

a) Choose τ as a security parameter.

b) Choose a bilinear group pair (G_1, G_2) of large prime order p, such that no efficiently computable homomorphism is known between G_1 and G_2, in either direction, and an associated pairing function $e: G_1 \times G_2 \rightarrow G_T$.

c) Choose two random independent generators P_1 and Q_1 of G_1 and provide additional information, denoted by π_{Gen}, that serve to demonstrate that these two generators were indeed chosen independently, that is without a potentially exploitable relationship between them (such as $Q_1 = [s]P_1$ for an integer s chosen by the group membership issuer). An example of how to verifiably select independent generators and to verify, using π_{Gen}, the correct generation of these generators, is given in Annex G.

d) Choose a random generator P_2 of G_2.

e) Choose three hash functions $H_1: \{0,1\}^* \rightarrow G_1$, $H_2: \{0,1\}^* \rightarrow Z_p$ and $H_3: \{0,1\}^* \rightarrow Z_p$. An example of how to construct such hash functions is provided in Annex B.

f) Choose three random integers x, y and z in Z_p.

g) Compute $X_1 = [z]P_1 + [x]Q_1$, $Y_1 = [y]P_1$, $X_2 = [x]P_2$ and $Y_2 = [y]P_2$.

h) Choose two random integers x' and z' in Z_p.

i) Compute $X_1^{'} = [z']P_1 + [x']Q_1$ and $X_2^{'} = [x']P_2$.

j) Compute $c_k = H_2(P_1 \| Q_1 \| P_2 \| X_1 \| Y_1 \| X_2 \| Y_2 \| X_1^{'} \| X_2^{'})$.

k) Compute $s_x = (x' + c_k \times x) \bmod p$ and $s_z = (z' + c_k \times z) \bmod p$.

l) Set $\pi_{\text{Val}} = (c_k, s_x, s_z)$ as a proof that the second component of the representation of X_1 in the base P_1 and Q_1 is equal to the discrete logarithm of X_2 in the base P_2.

m) Output the following:

   — group public parameter = (G_1, G_2, G_T, e, P_1, Q_1, P_2, p, H_1, H_2, H_3),

   — group public key = (X_1, Y_1, X_2, Y_2, π_{Gen}, π_{Val}),

   — group membership issuing key = (x, y, z).

NOTE 1 Examples of recommended parameters are provided in C.2.

Each entity involved in this anonymous signature mechanism should verify the validity of the group public key before using it. The group public key validity verification process includes the following steps:

a) Verify that P_1 and Q_1 were generated independently using π_{Gen}.

b) Verify the validity of the proof π_{Val}:

1) Compute $\tilde{X}_1 = [s_z]P_1 + [s_x]Q_1 - [c_k]X_1$ and $\tilde{X}_2 = [s_x]P_2 - [c_k]X_2$.

2) Compute $c_k^{'} = H_2\left(P_1 \| Q_1 \| P_2 \| X_1 \| Y_1 \| X_2 \| Y_2 \| \tilde{X}_1 \| \tilde{X}_2\right)$.

3) Verify that $c_k^{'} = c_k$.

c) Verify that $e(Y_1, P_2) = e(P_1, Y_2)$.

d) If any of the above verifications fails, output 0 (invalid), otherwise output 1 (valid).

The group membership issuing process requires a secure and authentic channel between the group member and the group membership issuer. How to establish such a channel is out scope of this mechanism. The group membership issuing process includes the following steps:

a) The group membership issuer chooses a nonce $n_I \in \{0,1\}^{\tau}$.

b) The group membership issuer sends n_I to the member.

c) The member chooses a random integer s_1 from Z_p.

d) The member computes $C_1 = [s_1]Y_1$.

e) The member chooses a random integer u from Z_p.

f) The member computes $D = [u]Y_1$.

g) The member computes $v = H_2(P_1 || Q_1 || P_2 || X_1 || Y_1 || X_2 || Y_2 || C_1 || D || n_I)$.

h) The member computes $w = (u + v \times s_1) \bmod p$.

i) The member sends (C_1, v, w) to the group membership issuer.

j) The group membership issuer computes $D' = [w]Y_1 - [v]C_1$.

k) The group membership issuer computes $v' = H_2(P_1 || Q_1 || P_2 || X_1 || Y_1 || X_2 || Y_2 || C_1 || D' || n_I)$.

l) The group membership issuer verifies $v = v'$. If the verification fails, abort the group membership issuing process.

m) The group membership issuer selects five random integers r, s_2, k_r, k_x and k_z from Z_p.

n) The group membership issuer computes $T_1 = [r]P_1$ and $T_2 = [x]T_1 + [r]C_1 + [r \times s_2]Y_1$.

o) The group membership issuer computes $K_1 = [k_r]P_1$, $K_2 = [k_x]T_1 + [k_r](C_1 + [s_2]Y_1)$ and $K = [k_z]P_1 + [k_x]Q_1$.

p) The group membership issuer computes $c = H_2(P_1 || Q_1 || P_2 || X_1 || Y_1 || X_2 || Y_2 || C_1 || s_2 || K_1 || K_2 || K)$.

q) The group membership issuer computes $z_r = (k_r + c \times r) \bmod p$, $z_x = (k_x + c \times x) \bmod p$ and $z_z = (k_z + c \times z) \bmod p$.

r) The group membership issuer sets (T_1, T_2) as the member's group membership credential and sends $(T_1, T_2, s_2, c, z_r, z_x, z_z)$ to the member.

s) The member computes $K_1' = [z_r]P_1 - [c]T_1$, $K_2' = [z_x]T_1 + [z_r](C_1 + [s_2]Y_1) - [c]T_2$ and $K' = [z_z]P_1 + [z_x]Q_1 - [c]X_1$.

t) The member computes $c' = H_2(P_1 \| Q_1 \| P_2 \| X_1 \| Y_1 \| X_2 \| Y_2 \| C_1 \| s_2 \| K_1' \| K_2' \| K)$.

u) The member verifies $c = c'$. If the verification fails, the member aborts.

v) The member computes $s = (s_1 + s_2) \bmod p$.

w) The group member signature key for the member is (s, T_1, T_2).

NOTE 2 The group membership issuer can use the same value s_2 (for example $s_2 = 0 \bmod p$) for several executions of the group membership issuing process. In this case, the security of Mechanism 8 relies on the Pointcheval-Sanders (PS) assumption[24], instead of the q-MSDH assumption[25] if the group membership issuer uses a fresh random value s_2 for each new session of the group membership issuing process.

6.6.3 Signature process

On input of a group member signature key (s, T_1, T_2), a linking base bsn and a message $m \in \{0,1\}^*$ to be signed, the signature process takes the following steps. The linking base, denoted by bsn, is either a special symbol ⊥ or an arbitrary string used for the linking capability.

a) If $bsn = \perp$, the signer chooses a random J from G_1, otherwise, computes $J = H_1(bsn)$.

b) The signer selects two random integers l and k_s in Z_p.

c) The signer computes $T_1' = [l]T_1$ and $T_2' = [l]T_2$.

d) The signer computes $R = [s]T_1'$ and $R' = [k_s]T_1'$.

e) The signer computes $T = [s]J$ and $T' = [k_s]J$.

f) The signer computes $c_m = H_3(T_1' \| T_2' \| J \| T \| R \| T' \| R' \| m)$.

g) The signer computes $\rho = (k_s + c_m \times s) \bmod p$.

h) The signer outputs the anonymous signature $\sigma = (T_1', T_2', J, R, T, c_m, \rho)$.

6.6.4 Verification process

On input of a message m, a linking base bsn, a signature $(T_1', T_2', J, R, T, c_m, \rho)$ and a group public key (X_1, Y_1, X_2, Y_2), the verification process takes the following steps:

a) If $bsn \neq \perp$, verify that $J = H_1(bsn)$.

b) Verify that $T_1' \neq O_E$.

c) If any of the above verifications fails, output 0 (invalid).

d) Compute $R'' = [\rho]T_1' - [c_m]R$.

e) Compute $T'' = [\rho]J - [c_m]T$.

f) Compute $c_m' = H_3(T_1' \| T_2' \| J \| T \| R \| T'' \| R'' \| m)$.

g) Verify that $c_m^{'} = c_m$.

h) Verify that $e\,(T_1^{'},\ X_2) \times e\,(R,\ Y_2) = e(T_2^{'},\ P_2)$.

i) Optionally, call the revocation checking process.

j) If any of the above verifications (steps g and h) fails, output 0 (invalid). Otherwise, output 1 (valid).

6.6.5 Linking process

Given two valid signatures $\sigma = (T_1^{'},\ T_2^{'},\ J, R, T, c_m, \rho)$ and $\hat{\sigma} = (\hat{T}_1^{'},\ \hat{T}_2^{'},\ \hat{J}, \hat{R}, \hat{T}, \hat{c}_m, \hat{\rho})$, the linking process takes the following steps:

a) If $J = \hat{J}$ and $T = \hat{T}$, output 1 (linked), otherwise, output 0 (not linked).

NOTE If the linking process outputs 0 because of J $J \neq \hat{J}$, it means that the linking process cannot determine whether two signatures were created by the same group member.

6.6.6 Revocation process

Details of the revocation process in this mechanism are surveyed in Reference [10]. There are two types of revocation (private key revocation and verifier blacklist revocation) supported in this mechanism. Private key revocation can be either global revocation or local revocation. Verifier blacklist revocation is a local revocation.

Private key revocation:

— If a group member signature key $(s,\ T_1,\ T_2)$ is compromised, the group membership issuer puts s into a revocation list RL of this type.

— Given a valid signature $\sigma = (T_1^{'},\ T_2^{'},\ J, R, T, c_m, \rho)$ computed using a linking base bsn and a revocation list RL of this type, a verifier can check revocation of this signature as follows: for each $s' \in$ RL, verify $T \neq [s']J$. If any of these verifications fails, output 0 (revoked), otherwise, output 1 (valid).

NOTE The private key revocation works only if the group membership issuer or the verifier has learned the group member signature keys of the compromised group members. This revocation process allows to identify every group signature generated using this private key. If this key can be associated with a group member (e.g. by using contextual information), then no anonymity can be retained for this group member as their signatures can therefore be traced. This is a property inherent to DAA schemes. Thus, a careful assessment of the need for revocation and the consequences for the corresponding group member will be carried out before deployment.

Verifier blacklist revocation:

— If signatures were computed using a linking base bsn, a verifier can build its own revocation list RL corresponding to bsn. If the verifier wants to blacklist the signer of a valid signature $\sigma = (T_1^{'},\ T_2^{'},\ J, R, T, c_m, \rho)$, they put T into a revocation list RL of this type.

— Given a signature $\sigma = (T_1^{'},\ T_2^{'},\ J, R, T, c_m\ \rho)$ computed using a linking base bsn and a revocation list RL of this type, a verifier can check revocation of this signature as follows: for each $\hat{T} \in$ RL, verify $T \neq \hat{T}$. If any of these verifications fails, output 0 (revoked), otherwise, output 1 (valid).

In order to use verifier blacklist revocation in this mechanism, a signer must use a specific linking base for each verifier. The value of the linking base can, for example, be chosen by the verifier or agreed in advance by the signer and verifier.

7.1

Replace the first sentence with the following:

> This clause specifies three digital signature mechanisms with opening capability.

Replace the text of NOTE with the following:

> The mechanisms and associated security proofs in 7.2, 7.3 and 7.4 are based on References [17], [14] and [23] (an extended version of Reference [24]), respectively.

7.4

Add new subclause 7.4 as follows:

7.4 Mechanism 9

7.4.1 Symbols

The following symbols apply in the specification of this mechanism.

— P_1, S_i, T_1, T_2, K, K', T_1', T_2': elements of G_1.

— P_2, X, Y, A, B, Y_i, C_1, C_2, C_3, C_4, K_1, K_2, K_3, K_4, K_1', K_2', K_3', K_4': elements of G_2.

— W, W', R, R_i: elements of G_T.

— x, y, a, b, s_i, u, v, k_s, k_u, k_v, c, z_s, z_u, z_v, c', r, t, w, c_m, z, c_m': elements of Z_p.

— H: a hash function that outputs elements in Z_p.

7.4.2 Key generation process

The group membership issuer key generation process takes the following steps:

a) Choose a bilinear group pair (G_1, G_2) of large prime order p, such that no efficiently computable homomorphism is known between G_1 and G_2, in either direction, and an associated pairing function $e : G_1 \times G_2 \rightarrow G_T$.

b) Choose a random generator P_1 of G_1 and a random generator P_2 of G_2.

c) Choose two random integers x and y in Z_p.

d) Compute $X = [x]P_2$ and $Y = [y]P_2$.

e) Choose a hash function $H : \{0,1\}^* \rightarrow Z_p$. Such a hash function shall be constructed as described in Annex B.

f) Output the following:

   — group public parameters: (G_1, G_2, G_T, e, p, H, P_1, P_2),

   — group public key: (X, Y),

   — group membership issuing key: (x, y).

The group membership opener key generation process takes the following steps:

a) Choose two random integers a and b in Z_p.

b) Compute $A = [a]P_2$ and $B = [b]P_2$.

c) Output the following:

— group membership opener public key (A, B),

— group membership opening key (a, b).

The group membership issuer manages a member-list **LIST** = (LIST[1],..., LIST[n]) where n is the number of group members who are registered so far. Each entry of the list contains information associated with each registered user. This member-list **LIST** can be published but it will only be useful to the group membership opener.

The group membership issuing process is an interactive protocol running between the group membership issuer and a user U_i to create a group member signature key for the user. It consists of the following steps:

a) U_i selects six random integers s_i, u, v, k_s, k_u and k_v in Z_p.

b) U_i computes $S_i = [s_i]P_1$ and $Y_i = [s_i]Y$.

c) U_i computes $C_1 = [u]P_2$, $C_2 = Y_i + [u]A$, $C_3 = [v]P_2$, $C_4 = Y_i + [v]B$.

d) U_i computes $K = [k_s]P_1$, $K_1 = [k_u]P_2$, $K_2 = [k_s]Y + [k_u]A$, $K_3 = [k_v]P_2$, $K_4 = [k_s]Y + [k_v]B$.

e) U_i computes $c = \mathrm{H}(P_1 \,||\, P_2 \,||\, X \,||\, Y \,||\, A \,||\, B \,||\, S_i \,||\, Y_i \,||\, C_1 \,||\, C_2 \,||\, C_3 \,||\, C_4 \,||\, K \,||\, K_1 \,||\, K_2 \,||\, K_3 \,||\, K_4)$.

f) U_i computes $z_s = (k_s + c \times s_i) \bmod p$, $z_u = (k_u + c \times u) \bmod p$ and $z_v = (k_v + c \times v) \bmod p$.

g) U_i sends S_i, C_1, C_2, C_3, C_4, c, z_s, z_u and z_v.

h) The group membership issuer computes $K' = [z_s]P_1 - [c]S_i$, $K_1' = [z_u]P_2 - [c]C_1$, $K_2' = [z_s]Y + [z_u]A - [c]C_2$, $K_3' = [z_v]P_2 - [c]C_3$ and $K_4' = [z_s]Y + [z_v]B - [c]C_4$.

i) The group membership issuer computes $c' = \mathrm{H}(P_1 \,||\, P_2 \,||\, X \,||\, Y \,||\, A \,||\, B \,||\, S_i \,||\, Y_i \,||\, C_1 \,||\, C_2 \,||\, C_3 \,||\, C_4 \,||\, K' \,||\, K_1' \,||\, K_2' \,||\, K_3' \,||\, K_4')$.

j) The group membership issuer checks if $c = c'$ and aborts if these two values are different.

k) The group membership issuer stores $(i, S_i, C_1, C_2, C_3, C_4, c, z_s, z_u, z_v)$ in LIST[i].

l) The group membership issuer selects a random integer r in Z_p.

m) The group membership issuer computes $T_1 = [r]P_1$ and $T_2 = [r \times x]P_1 + [r \times y]S_i$.

n) The group membership issuer sends T_1 and T_2 to the user U_i.

o) The signature key of the group member U_i is then (s_i, T_1, T_2).

7.4.3 Signature process

On input of a group public key (X , Y), a group member signature key (s_i, T_1, T_2) owned by the signer and a message $m \in \{0,1\}^*$ to be signed, the signature process takes the following steps.

a) The signer selects two random integers t and w in Z_p.

b) The signer computes $T_1^{'} = [t]T_1$ and $T_2^{'} = [t]T_2$.

c) The signer computes $W = e([w]T_1^{'}, Y)$.

d) The signer computes $c_m = H(T_1^{'} || T_2^{'} || W || m)$.

e) The signer computes $z = (w + c_m \times s_i) \bmod p$.

f) The signer outputs the group signature $\sigma = (T_1^{'}, T_2^{'}, c_m, z)$.

7.4.4 Verification process

On input of a message $m \in \{0,1\}^*$, a group signature $\sigma = (T_1^{'}, T_2^{'}, c_m, z)$ and a group public key (X, Y), the verification process takes the following steps:

a) Verify that $T_1^{'} \neq O_E$. If this verification fails, output 0 (invalid).

b) Compute $W' = e([z]T_1^{'}, Y) \times e\left([-c_m]T_2^{'}, P_2\right) \times e([c_m]T_1^{'}, X)$.

c) Compute $c_m^{'} = H(T_1^{'} || T_2^{'} || W' || m)$.

d) Verify that $c_m^{'} = c_m$ holds.

e) If the above verification fails, output 0 (invalid), otherwise, output 1 (valid).

7.4.5 Opening process

Given a group signature $\sigma = (T_1^{'}, T_2^{'}, c_m, z)$, the member-list **LIST**=(LIST[1],...,LIST[n]) and a group opening key (a, b), the opening process takes the following steps:

a) Compute $R = e(T_2^{'}, P_2) \times e([-1]T_1^{'}, X)$.

b) For each $i \in [1, n]$,

   1) Recover (i, S_i, C_1, C_2, C_3, C_4, c, z_s, z_u, z_v) from LIST[i].

   2) Compute $Y_i = C_2 + [-a]C_1$.

   3) Verify if $e(T_1^{'}, Y_i) = R$.

   4) If the above equation holds, output i.

7.4.6 Revocation process

The revocation process is a membership credential revocation. The group membership opener revokes a user U_i by adding some element R_i (defined below) specific to this user in a revocation list RL.

