~~2022-12-22~~

~~Final text for~~ Date: 2023-02-07

ISO/IEC 20008-2~~:2013~~/Amd. 2:~~2022~~2023(E)

ISO/IEC JTC1/SC 27/WG 2

Secretariat: DIN

Style Definition: zzCopyright

Style Definition: Footer

Formatted: Different first page header

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

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

Formatted: French (Switzerland)

Formatted: French (Switzerland)

Formatted: French (Switzerland)

ISO ~~copyright office~~Copyright Office

CP 401 • ~~Ch. de Blandonnet 8~~

CH-1214 Vernier, Geneva

Phone: + 41 22 749 01 11

~~Email: copyright@iso.org~~

Email: copyright@iso.org

Website: ~~www.iso.org~~www.iso.org

Published in Switzerland.

Formatted

Formatted: Default Paragraph Font

Formatted: Indent: Left: 0 pt, Right: 0 pt, Space Before: 0 pt, No page break before, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Indent: Left: 0 pt, First line: 0 pt, Right: 0 pt, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: English (United Kingdom)

Formatted: Indent: Left: 0 pt, First line: 0 pt, Right: 0 pt, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: English (United Kingdom)

Formatted: English (United Kingdom)

Formatted: Font: 11 pt

Formatted: Space After: 0 pt, Line spacing: single

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~~www.iso.org/directives or ~~www.iec.ch/members_experts/refdocs~~www.iec.ch/members_experts/refdocs).

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~~www.iso.org/patents) or the IEC list of patent declarations received (see ~~https://patents.iec.ch~~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~~www.iso.org/iso/foreword.html. In the IEC, see ~~www.iec.ch/understanding-standards~~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*.

A list of all parts in the ISO/IEC 20008 series can be found on the ISO and IEC websites.

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~~www.iso.org/members.html and ~~www.iec.ch/national-committees~~www.iec.ch/national-committees.

Formatted: English (United Kingdom)

Formatted: Font: 11 pt

Formatted: Space After: 0 pt, Line spacing: single

Information technology — Security techniques — Anonymous digital signatures — Part 2: Mechanisms using a group public key — Amendment 2

Clause 4

Add the following symbol:

~~$F(p)$~~ $F(p)$ the finite field containing exactly ~~p~~ 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~~ τ : a security parameter.

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

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

Formatted: Right

Formatted: Different first page header

Formatted: Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers

Formatted: Font: Italic

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

- ~~$x,y,z,x',z',c_k,s_x,s_z,c'_k,s_1,u,v,w,v',r,s_2,k_r,k_x,k_z,c,z_r,z_x,z_z,c',s,l,k_s,c_m,\rho,c'_m$~~ $x,y,z,x',z',c_k,s_x,s_z,c'_k,s_1,u,v,w,v',r,s_2,k_r,k_x,k_z,c,z_r,z_x,z_z,c',s,l,k_s,c_m,\rho,c'_m$: integers in ~~$Z_p$~~ Z_p.
- ~~n_I~~ n_I : an integer of size ~~τ~~ τ -bit.
- ~~H_1~~ H_1 : a hash function that outputs elements in ~~G_1~~ G_1 .
- ~~H_2~~ H_2 , ~~H_3~~ H_3 : hash functions that output elements in ~~Z_p~~ 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~~ τ as a security parameter.

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

c) Choose two random independent generators ~~P_1~~ P_1 and ~~Q_1~~ Q_1 of ~~G_1~~ G_1 and provide additional information, denoted by ~~π_{Gen}~~ 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~~ Q_1 ~~$=$~~ $=$ ~~$[s]P_1$~~ $[s]P_1$ for an integer ~~s~~ s chosen by the group membership issuer). An example of how to verifiably select independent generators and to verify, using ~~π_{Gen}~~ π_{Gen} , the correct generation of these generators, is given in Annex G.

d) Choose a random generator ~~P_2~~ P_2 of ~~G_2~~ G_2 .

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

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

Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Formatted: Font: 11 pt
Formatted: Space After: 0 pt, Line spacing: single

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

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'~~ x' and ~~z'~~ z' in ~~Z_p~~ 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 $s_z = (z' + c_k \times z) \bmod p$.~~

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

m) Output the following:

- group public parameter = (~~G_1~~ G_1, ~~G_2~~ G_2, ~~G_T~~ G_T, ~~e~~ e, ~~P_1~~ P_1, ~~Q_1~~ Q_1, ~~P_2~~ P_2, ~~p~~ p, ~~H_1~~ H_1, ~~H_2~~ H_2, ~~H_3~~ H_3),
- group public key = (~~X_1~~ X_1, ~~Y_1~~ Y_1, ~~X_2~~ X_2, ~~Y_2, π_{Gen}, π_{Val})~~, Y_2, π_{Gen}, π_{Val}).
- group membership issuing key = (~~x, y, z~~ 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~~ P_1 and ~~Q_1~~ Q_1 were generated independently using ~~π_{Gen}.~~ π_{Gen}.

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

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

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

Formatted: Right

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Formatted: Font: 11 pt

Formatted: Space After: 0 pt, Line spacing: single

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

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

3) Verify that ~~c_k'~~ c_k' = ~~c_k~~ c_k.

c) Verify that ~~e~~ e (~~Y_1~~ Y_1, ~~P_2~~ P_2) ~~=~~ = e(~~P_1~~ P_1, ~~Y_2~~ 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~~ n_I ~~$\in$~~ $\in$ ~~$\{0,1\}^\tau$~~ $\{0,1\}^\tau$.

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

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

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

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

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

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

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~~ w ~~=~~ = (~~u~~ u ~~+~~ + ~~v~~ v ~~$\times s_1$~~ $\times s_1$) mod ~~p~~ p.

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

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

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

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

Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Formatted: Font: 11 pt
Formatted: Space After: 0 pt, Line spacing: single

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

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

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

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

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

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

r) The group membership issuer sets (~~T_1~~ T_1 , ~~T_2~~ T_2) as the member's group membership credential and sends (~~T_1~~ T_1 , ~~$T_2, s_2, c, z_r, z_x, z_z$~~ $T_2, s_2, c, z_r, z_x, z_z$) 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 $c' = H_2(P_1 \| Q_1 \| P_2 \| X_1 \| Y_1 \| X_2 \| Y_2 \| C_1 \| s_2 \| K_1' \| K_2' \| K)$.~~

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 ~~$=$~~ $=$ ~~c'~~ c'. If the verification fails, the member aborts.

v) The member computes ~~s~~ s ~~$=$~~ $=$ (~~s_1~~ s_1 ~~$+$~~ $+$ ~~s_2~~ s_2) mod ~~p~~ p .

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

NOTE 2 The group membership issuer can use the same value ~~s_2~~ s_2 (for example ~~$s_2 = 0$~~ $s_2 = 0$ mod ~~p~~ p) for several executions of the group membership issuing process. In this case, the security of Mechanism 8 relies on the

Formatted: Right

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Field Code Changed

Formatted: Font: 11 pt

Formatted: Space After: 0 pt, Line spacing: single

Pointcheval-Sanders (PS) assumption[24], instead of the q-MSDH assumption[25] if the group membership issuer uses a fresh random value ~~s_2~~ 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~~ s, ~~T_1~~ T_1, ~~T_2~~ T_2), a linking base ~~bsn~~ bsn and a message ~~m~~ m ~~$\in$~~ $\in$ ~~$\{0,1\}^*$~~ $\{0,1\}^*$ to be signed, the signature process takes the following steps. The linking base, denoted by ~~bsn~~ bsn, is either a special symbol ⊥ or an arbitrary string used for the linking capability.

Field Code Changed

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

Field Code Changed

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

Field Code Changed

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

Field Code Changed

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

Field Code Changed

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

Field Code Changed

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

Field Code Changed

g) The signer computes ~~ρ~~ ρ ~~$=$~~ $=$ (~~k_s~~ k_s ~~$+$~~ $+$ ~~$c_m \times s$~~ $c_m \times s$) mod ~~p~~ p.

Field Code Changed

h) The signer outputs the anonymous signature ~~σ~~ σ ~~$=$~~ $=$ (~~T_1'~~ T_1', ~~T_2'~~ T_2', ~~J,R,T,c_m,ρ~~ J,R,T,c_m,ρ).

Field Code Changed

6.6.4 Verification process

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

Field Code Changed

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

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

Field Code Changed

b) Verify that ~~$T_1' \neq 0_E$~~ $T_1' \neq 0_E$.

Field Code Changed

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

Formatted: Font: 11 pt

Formatted: Space After: 0 pt, Line spacing: single

Formatted: Right

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

Field Code Changed

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

Field Code Changed

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

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

Field Code Changed

g) Verify that ~~c_m'~~ c_m' ~~$=c_m$~~ $=c_m$.

Field Code Changed

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

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

Field Code Changed

i) Optionally, call the revocation checking process.

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

6.6.5 Linking process

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

Field Code Changed

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

Field Code Changed

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

Field Code Changed

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~~ s, ~~T_1~~ T_1, ~~T_2~~ T_2) is compromised, the group membership issuer puts ~~s~~ s into a revocation list RL of this type.

Field Code Changed

Formatted: Font: 11 pt

Formatted: Space After: 0 pt, Line spacing: single

— Given a valid signature ~~σ~~ σ ~~$=$~~ $=$ $($~~T_1'~~ T_1', ~~T_2'~~ T_2', ~~J,R,T,c_m,ρ~~ $J,R,T,c_m,\rho$$)$ computed using a linking base ~~bsn~~ 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 $T \neq [s']J$.~~ $s' \in$ RL, verify $T \neq [s']J$. If any of these verifications fails, output 0 (revoked), otherwise, output 1 (valid).

NOTE ~~1~~ 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~~ bsn, a verifier can build its own revocation list RL corresponding to ~~bsn.~~ bsn. If the verifier wants to blacklist the signer of a valid signature ~~σ~~ σ ~~$=$~~ $=$ $($~~T_1'~~ T_1', ~~T_2'~~ T_2', ~~J,R,T,c_m,ρ~~ $J,R,T,c_m,\rho$$)$, they put ~~$T$~~ T into a revocation list RL of this type.

— Given a signature ~~σ~~ σ ~~$=$~~ $=$ $($~~T_1'~~ T_1', ~~T_2'~~ T_2', ~~J,R,T,c_m,ρ~~ $J,R,T,c_m,\rho$$)$ computed using a linking base ~~bsn~~ 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 $T \neq \hat{T}$.~~ $\hat{T} \in$ RL, verify $T \neq \hat{T}$. If any of these verifications fails, output 0 (revoked), otherwise, output 1 (valid).

~~NOTE 2~~ 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] (~~the full~~an extended version of Reference [24]), respectively.

7.4

Add new subclause 7.4 as follows:

7.4 Mechanism 9

7.4.1 Symbols

Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Formatted: Adjust space between Latin and Asian text, Adjust space between Asian text and numbers, Tab stops: Not at 19.85 pt + 39.7 pt + 59.55 pt + 79.4 pt + 99.25 pt + 119.05 pt + 138.9 pt + 158.75 pt + 178.6 pt + 198.45 pt
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Formatted ...
Formatted: Default Paragraph Font, Font: Not Bold
Formatted: Font: 11 pt
Formatted: Space After: 0 pt, Line spacing: single

The following symbols apply in the specification of this mechanism.

- ~~P_1~~ P_1, ~~S_i~~ S_i, ~~T_1~~ T_1, ~~T_2, K, K', T_1', T_2'~~ T_2, K, K', T_1', T_2': elements of ~~G_1~~ G_1.
- ~~P_2~~ P_2 , ~~$X, Y, A, B, Y_i, C_1, C_2, C_3, C_4, K_1, K_2, K_3, K_4, K_1', K_2', K_3', K_4'$~~ $X, Y, A, B, Y_i, C_1, C_2, C_3, C_4, K_1, K_2, K_3, K_4, K_1', K_2', K_3', K_4'$: elements of ~~$G_2$~~ G_2.
- ~~W, W', R, R_i~~ W, W', R, R_i: elements of ~~G_T.~~ G_T.
- ~~$x, y, a, b, s_i, u, v, k_s, k_u, k_v, c, z_s, z_u, z_v, c', r, t, w, c_m, z, c_m'$~~ $x, y, a, b, s_i, u, v, k_s, k_u, k_v, c, z_s, z_u, z_v, c', r, t, w, c_m, z, c_m'$: elements of ~~$Z_p$~~ Z_p.
- ~~H~~ H: a hash function that outputs elements in ~~Z_p~~ 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_1, ~~G_2~~ G_2) of large prime order ~~p~~ p, such that no efficiently computable homomorphism is known between ~~G_1~~ G_1 and ~~G_2~~ G_2, in either direction, and an associated pairing function ~~e~~ e: ~~G_1~~ G_1 × ~~G_2~~ G_2 → ~~G_T~~ G_T.

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

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

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

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

e) Choose a hash function ~~H~~ H: ~~$\{0,1\}^*$~~ $\{0,1\}^*$ → ~~Z_p. An example of how to construct such~~ Z_p. Such a hash function ~~is provided~~shall be constructed as described in Annex B.

f) Output the following:

  - group public parameters: (~~G_1~~ G_1, ~~G_2~~ G_2, ~~G_T~~ G_T, ~~e~~ e, ~~p~~ p, ~~H, P_1, P_2~~ H, P_1, P_2),
  - group public key: (~~X, Y~~ X, Y),
  - group membership issuing key: (~~x, y~~ x, y).

Formatted
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Field Code Changed
Formatted
Formatted