International Standard

ISO/IEC 20008-3

First edition
2024-12

Information security — Anonymous digital signatures —

Part 3: Mechanisms using multiple public keys

Sécurité de l'information — Signatures numériques anonymes —

Partie 3: Mécanismes utilisant plusieurs clés publiques

Reference number
ISO/IEC 20008-3:2024(en)

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

Contents

Page

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 on 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 Joint 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 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 and www.iec.ch/national-committees.

Introduction

Anonymous digital signature mechanisms are a special type of digital signature mechanism in which, where a digital signature is present, an unauthorized entity cannot discover the signer's identifier yet can verify that a legitimate signer has generated a valid signature.

The ISO/IEC 20008 series specifies anonymous digital signature mechanisms. ISO/IEC 20008-1 specifies principles and requirements for two categories of anonymous digital signatures mechanisms:

1) signature mechanisms using a group public key;

2) signature mechanisms using multiple public keys.

This document specifies a number of anonymous signature mechanisms in the second category.

Anonymous signature mechanisms in the second category allow a signer to form a group spontaneously by combining the public keys of the relevant users with the signer's own public key. The verifier can confirm that a signature is generated by one of the users within this group but cannot find out who the actual signer is. Unlike the first category described in ISO/IEC 20008 series, mechanisms in the second category do not require a group manager, and the private key and public key are generated individually by each user.

Some mechanisms described in this document are unlinkable, where no one can determine whether two signatures are generated by the same signer or not. Some mechanisms have a linking capability, making it possible to be determined whether two signatures were generated by the same signer under certain conditions. Some mechanisms have a tracing capability, where, given two (message, signature) pairs generated by the same signer within the same ring setting, the true signer can be identified. Some mechanisms have a threshold setting, where the verifier can confirm a signature is generated by subsets (of specified size) of group users, but cannot determine which users were involved.

Information security — Anonymous digital signatures —

Part 3:
Mechanisms using multiple public keys

1 Scope

This document specifies anonymous digital signature mechanisms in which a verifier uses multiple public keys to verify a digital signature.

This document provides:

— a general description of an anonymous digital signature mechanism using multiple public keys;

— a variety of mechanisms that provide such anonymous digital signatures.

For each mechanism, this document specifies the process for:

— generating the private key and public key of each user;

— producing signatures;

— verifying signatures;

— linking signatures (if the mechanism supports linking);

— tracing signatures (if the mechanism supports tracing);

— producing signatures with threshold capability (if the mechanism supports a threshold capability);

— verifying signatures with threshold capability (if the mechanism supports a threshold capability).

This document does not define the implementation of a public key infrastructure (PKI) and the means for distinct entities to exchange, extract and verify their respective public key certificates.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10116, *Information technology — Security techniques — Modes of operation for an n-bit block cipher*

ISO/IEC 10118-1, *Information technology — Security techniques — Hash-functions — Part 1: General*

ISO/IEC 10118-2, *Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an n-bit block cipher*

ISO/IEC 10118-3, *IT Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

ISO/IEC 10118-4, *Information technology — Security techniques — Hash-functions — Part 4: Hash-functions using modular arithmetic*

ISO/IEC 18033-2:2006, *Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers*

ISO/IEC 18033-3, *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*

ISO/IEC 18033-4, *Information technology — Security techniques — Encryption algorithms — Part 4: Stream ciphers*

ISO/IEC 18033-5:2015, *Information technology — Security techniques — Encryption algorithms — Part 5: Identity-based ciphers*

ISO/IEC 29192-2, *Information security — Lightweight cryptography — Part 2: Block ciphers*

ISO/IEC 29192-3, *Information technology — Security techniques — Lightweight cryptography — Part 3: Stream ciphers*

ISO/IEC 18031, *Information technology — Security techniques — Random bit generation*

ISO/IEC 18032, *Information security — Prime number generation*

ISO/IEC 20008-1, *Information technology — Security techniques — Anonymous digital signatures — Part 1: General*

ISO/IEC 20008-2:2013, *Information technology — Security techniques — Anonymous digital signatures — Part 2: Mechanisms using a group public key*

RFC 9380, *Hashing to Elliptic Curves, Internet Research Task Force (IRTF)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 20008-1, ISO/IEC 18033-4 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
symmetric encryption system
encryption system based on symmetric cryptographic techniques

[SOURCE: ISO/IEC 18033-1:2021, 3.29, modified — the admitted terms “symmetric encipherment system” and “symmetric cipher” were removed.]

4 Symbols and abbreviated terms

For the purposes of this document, the following symbols and abbreviated terms apply.

Bsn	Linking base, either a special symbol ⊥ or an arbitrary string.
$\oplus$	Bitwise exclusive OR (XOR) operation.
$\setminus$	Set subtraction operation.
$\mathrm{Enc}_k(m)$	A symmetric-key encryption function taking a secret key k and a variable-length message m as input and giving a ciphertext c as output, e.g. the encryption function for one of the symmetric encryption systems specified in ISO/IEC 18033-3, ISO/IEC 18033-4, ISO/IEC 10116, ISO/IEC 29192-2 or ISO/IEC 29192-3.
$\mathrm{Dec}_k(C)$	A symmetric-key decryption function taking a secret key k and a ciphertext C as input and giving a message m or a decryption failure symbol ⊥ as output.
C	A ciphertext generated from a symmetric-key encryption function.

N	The size of the ring, i.e. the number of entities in the set consisting of the true signer and the potential signer (or signers).
π	Index of the true signer, where $1 \le \pi \le N$. If $\pi = N$, $\pi + 1$ is set to 1. Similarly, if $\pi = 1$, $\pi - 1$ is set to N.
$\mathrm{lb}(x)$	The base 2 logarithm of x.
$\max(X)$	The largest integer within the integer set X.
gcd(*a*,*b*)	The greatest common divisor of the integers *a* and *b*.
H	A hash-function taking an arbitrary string as input and giving a fixed-length bit string as output, e.g. one of the dedicated hash-functions specified in ISO/IEC 10118-3.
||	*X* || *Y* denotes the result of the concatenation of octet strings *X* and *Y* in the order specified. In cases where the result of concatenating two or more octet strings is input to a cryptographic function as part of one of the mechanisms specified in this document, this result shall be composed so that it can be uniquely resolved into its constituent octet strings, i.e. so that there is no possibility of ambiguity in interpretation. This latter property can be achieved in a variety of different ways, depending on the application. For example, it can be guaranteed by fixing the length of each of the octet strings throughout the domain of use of the mechanism, or by encoding the sequence of concatenated octet strings using a method that guarantees unique decoding, e.g. using the distinguished encoding rules defined in ISO/IEC 8825-1[9].

5 General model and requirements

5.1 General

This clause specifies the general model and requirements for the anonymous digital signature mechanisms specified in this document. Some of the contents of this clause are taken from ISO/IEC 20008-1. In addition, specific requirements applying to mechanisms using multiple public keys are addressed.

5.2 Model

An anonymous digital signature mechanism using multiple public keys is also known as a ring signature mechanism. A ring signature mechanism involves a set of possible signers. Each possible signer has a signature key pair in the same form as in a conventional signature mechanism. These possible signers are independent of each other in the sense that it is not necessary for them to agree on being involved in the same signature process. To create a ring signature, one of them is the true signer and the others are the potential signers. The true signer chooses one or more potential signers and forms a ring of N possible signers. The true signer may select only one potential signer and, in this case, $N = 2$.

Some ring signature mechanisms have special capabilities. These mechanisms make it possible to link two signatures, trace a signature back to its true signer or apply thresholds to the signing process.

A ring signature mechanism is defined by the specification of the following processes:

— key generation process;

— signature process;

— verification process;

— linking process (if supported);

— tracing process (if supported);

— threshold signature process (if supported);

— threshold verification process (if supported).

The security properties of a general ring signature include unforgeability and anonymity. Other specific properties such as linkability, traceability and threshold are associated with ring signatures with special capabilities.

The anonymous digital signature mechanisms using multiple public keys specified in this document involve a range of types of entity. Some of these entities exist in every mechanism whereas others exist only in some mechanisms. The following list defines the role of each entity:

— True signer: an entity generating a digital signature.

— Potential signer: an entity whose public key is used in creating a digital signature, which means the public key is used in both the signature process and signature verification process, although the potential signer is not involved in these two processes.

— Verifier: an entity verifying a digital signature.

— Linker: an entity that checks whether two signatures have been generated by the same signer with a linking base. This entity exists in some of the mechanisms.

— Tracer: an entity that identifies the true signer when given as input two (message, signature) pairs which are both generated by the same signer within the same ring setting. This entity exists in some of the mechanisms.

5.3 Requirements

In order to use any of the mechanisms specified in this document, the following requirements shall be met.

— Each verifier shall have access to an authentic copy of the public key of each entity in the ring.

— A collision-resistant hash function shall be as specified in accordance with the ISO/IEC 10118 series.

— A robust random bit generator shall be as specified in accordance with ISO/IEC 18031.

— A robust prime number generator shall be as specified in accordance with ISO/IEC 18032.

— Symmetric-key encryption and decryption functions shall be as specified in accordance with ISO/IEC 18033-3, ISO/IEC 18033-4, ISO/IEC 10116, ISO/IEC 29192-2 or ISO/IEC 29192-3.

— Object identifiers described in Annex A shall be used.

— Conversion functions described in Annex B shall be used.

Numerical examples of mechanisms in this document are provide in Annex C.

6 Mechanisms without special capability

6.1 General

This clause specifies three digital signature mechanisms without any special capabilities such as linking, tracing or threshold.

NOTE 1 The mechanism given in 6.2 is based on the scheme originally specified in Reference [7].

NOTE 2 The mechanism given in 6.3 is based on the scheme originally specified in Reference [1].

NOTE 3 The mechanism given in 6.4 is based on the scheme originally specified in Reference [1].

6.2 Mechanism 1

6.2.1 Symbols

The following symbols apply in the specification of this mechanism.

— b: integer

— d_i: integer in the range $[0,(p_i-1)(q_i-1)-1]$

— k: l_k-bit string

— l_b: integer equal to or greater than 160

— l_k: integer equal to or greater than 256

— l_n: integer equal to or greater than 2048

— n: l_n-bit string

— $p_i, q_i, p_i', q_i', e_i$: prime numbers

— v, x_i, y_i: b-bit integers

— H: a hash function that outputs an l_k-bit hash-code as defined in Annex B

— Enc_k: symmetric-key encryption function that takes a secret key k and a variable-length message as input and outputs an integer in the range $\left[0,2^b-1\right]$

— Dec_k: symmetric-key decryption function taking a secret key k and a ciphertext as input and outputs an integer in the range $\left[0,2^b-1\right]$

6.2.2 Key generation process

The key generation process involves each user $i (1 \le i \le N)$ independently performing the following steps.

a) Choose prime numbers p_i', q_i' with $l_n/2$ bits such that $p_i = 2p_i'+1, q_i = 2q_i'+1$ are prime numbers. Compute modulus $n_i = q_i p_i$.

b) Choose a number $e_i \in [0,(p_i-1)(q_i-1)-1]$ such that $\gcd(e_i,(p_i-1)(q_i-1))=1$.

c) Compute $d_i = e_i^{-1} \bmod (p_i-1)(q_i-1)$.

d) Output the following:

— Public key = $vk_i = (n_i, e_i)$;

— Private key = $sk_i = d_i$.

6.2.3 Ring signature process

In the ring signature process, a true signer first chooses a set of $N-1$ potential signers. The true signer also chooses a random integer π satisfying $1 \le \pi \le N$, which denotes the index of the true signer. The true signer then signs a message using its private key and the public keys of the potential signers, without requiring their approval or assistance. In this process the potential signers are arranged in random order.

When inputting a private key (n_π, d_π), a randomly ordered set of public keys $L=\{(n_1,e_1),\ldots,(n_N,e_N)\}$ [including (n_π,e_π)], and a message $m\in\{0,1\}^*$ to be signed, the ring signature process involves the following steps.

a) Compute $b=\lceil \mathrm{lb}(\max(\{n_1,\ldots,n_N\})+1)\rceil+l_b$.

b) Compute $k=H(m,n_1,e_1,\ldots,n_N,e_N)$.

c) Choose a random integer $v\in\left[0,2^b-1\right]$.

d) Choose random integers $x_i\in\left[0,2^b-1\right]$ for all $i\in[1,N]\setminus\pi$.

e) Compute $y_i=g_i(x_i)$ for all $i\in[1,N]\setminus\pi$, where $g_i(x)$ is defined as $g_i(x)=\lfloor x/n_i\rfloor n_i+\left(x^{e_i} \bmod n_i\right)$ if $(\lfloor x/n_i\rfloor+1)n_i\le 2^b$, and $g_i(x)=x$ otherwise.

f) Compute

$y_\pi=\mathrm{Dec}_k(y_{\pi+1}\oplus\mathrm{Dec}_k(\ldots(\ldots\oplus\mathrm{Dec}_k(y_N\oplus\mathrm{Dec}_k(v))\ldots)))\oplus\mathrm{Enc}_k(y_{\pi-1}\oplus\mathrm{Enc}_k(\ldots(\ldots\oplus\mathrm{Enc}_k(y_1\oplus v)\ldots)))$ when $\pi\in[2,N-1]$, or

$y_\pi=\mathrm{Dec}_k(v)\oplus\mathrm{Enc}_k(y_{\pi-1}\oplus\mathrm{Enc}_k(\ldots(\ldots\oplus\mathrm{Enc}_k(y_1\oplus v)\ldots)))$ when $\pi=N$, or

$y_\pi=\mathrm{Dec}_k(y_{\pi+1}\oplus\mathrm{Dec}_k(\ldots(\ldots\oplus\mathrm{Dec}_k(y_N\oplus\mathrm{Dec}_k(v))\ldots)))\oplus v$ when $\pi=1$.

i) Compute $x_\pi=g_\pi^{-1}(y_\pi)$, where $g_\pi^{-1}(y)$ is defined as $g_\pi^{-1}(y)=\lfloor y/n_\pi\rfloor n_\pi+\left(y^{d_\pi} \bmod n_\pi\right)$ if $(\lfloor y/n_\pi\rfloor+1)n_\pi\le 2^b$, and $g_\pi^{-1}(y)=y$ otherwise.

j) Output the signature $\sigma=(v,x_1,\ldots,x_N)$ with the ordered set of public keys L.

6.2.4 Ring signature verification process

In the verification process, a verifier uses the public keys of the true signer and potential signers involved in the signature process to verify the signature. The verifier checks whether the signature was signed by one of the signers in the set without learning who the true signer is.

When inputting a message m, a signature $(v,x_1,\ldots,x_N)$, and an ordered set of public keys $L=\{(n_1,e_1),\ldots,(n_N,e_N)\}$, the verification process takes the following steps.

a) Compute $b=\mathrm{lb}(\max(\{n_1,\ldots,n_N\})+1)+l_b$.

b) Compute $y_i=g_i(x_i)$ for all $i\in[1,N]$ where $g_i(x)$ is defined as $g_i(x)=\lfloor x/n_i\rfloor n_i+\left(x^{e_i} \bmod n_i\right)$ if $(\lfloor x/n_i\rfloor+1)n_i\le 2^b$, and $g_i(x)=x$ otherwise.

c) Compute $k=H(m,n_1,e_1,\ldots,n_N,e_N)$.

d) If $v=\mathrm{Enc}_k(y_N\oplus\mathrm{Enc}_k(y_{N-1}\oplus\mathrm{Enc}_k(\ldots\oplus\mathrm{Enc}_k(y_1\oplus v)\ldots)))$ then return 1 (valid), else return 0 (invalid).

6.3 Mechanism 2

6.3.1 Symbols

The following symbols apply in the specification of this mechanism.

— g_i : a generator in group G_i

— G_i : a cyclic group with order q_i

— q_i : prime number

— x_i, c_i, s_i, α : integers in the range $[0, q_i - 1]$

— y_i, e_i : group elements in G_i

— H_i : a hash function that outputs an integer in the range $[0, q_i - 1]$ as defined in Annex B

6.3.2 Key generation process

The key generation process involves each user $i\,(1 \le i \le N)$ independently performing the following steps.

a) Choose a random $x_i \in [0, q_i - 1]$, and compute $y_i = g_i{}^{x_i} \in G_i$.

b) Output the following:

— Public key = $y_i = g_i{}^{x_i} \in G_i$;

— Private key = $x_i \in [0, q_i - 1]$.

6.3.3 Ring signature process

In the ring signature process, a true signer first chooses a set of $N-1$ potential signers. The true signer also chooses a random integer π satisfying $1 \le \pi \le N$, which denotes the index of the true signer. The true signer then signs a message using its private key and the public keys of the potential signers, without requiring their approval or assistance. In this process the potential signers are arranged in random order.

When inputting a private key x_π, a randomly ordered set of public keys $L = \{y_1, ..., y_N\}$ (including y_π), and a message $m \in \{0,1\}^*$ to be signed, the ring signature process involves the following steps.

a) Choose a random $\alpha \in [0, q_\pi - 1]$, and compute $e_\pi = g_\pi{}^{\alpha} \in G_\pi$ and $c_{\pi+1} = H_{\pi+1}(L, m, e_\pi)$.

b) Choose a random $s_i \in [0, q_i - 1]$, and compute $e_i = g_i{}^{s_i} y_i{}^{c_i} \in G_i$ and $c_{i+1} = H_{i+1}(L, m, e_i)$ for $i = \pi+1, \ldots, N, 1, \ldots, \pi-1$, where index $N+1$ is equivalent to 1.

c) Compute $s_\pi = \alpha - c_\pi x_\pi \bmod q_\pi$.

d) Output the signature $\sigma = (c_1, s_1, ..., s_N)$.

6.3.4 Ring signature verification process

In the verification process, a verifier uses the public keys of the true signer and potential signers involved in the signature process to verify the signature. The verifier checks whether the signature was signed by one of the signers in the set without learning who the true signer is.

When inputting a message m, a signature $\sigma = (c_1, s_1, ..., s_N)$ and a set of public keys $L = \{y_1, ..., y_N\}$, the verification process takes the following steps.

a) Compute $e_i = g_i{}^{s_i} y_i{}^{c_i} \in G_i$ and $c_{i+1} = H_{i+1}(L, m, e_i)$ for $i = 1, \ldots, N-1$.

b) Compute $e_N = g_N{}^{s_N} y_N{}^{c_N} \in G_N$.

c) If $c_1 = H_1(L, m, e_N)$ return 1 (valid), else return 0 (invalid).

6.4 Mechanism 3

6.4.1 Symbols

The following symbols apply in the specification of this mechanism.

— d_i , f_i : integers in the range $[0,(p_i-1)(q_i-1)-1]$

— l_n : integer equal to or greater than 2048

— n_i : l_n -bit integer

— p_i , q_i , p_i' , q_i' : prime numbers

— H_i : a hash function that outputs an integer in the range $[0,n_i-1]$ as defined in Annex B

6.4.2 Key generation process

The key generation process involves each user $i\,(1\le i\le N)$ independently performing the following steps.

a) Choose prime numbers p_i',q_i' with $l_n/2$ bits such that $p_i=2p_i'+1, q_i=2q_i'+1$ are prime numbers. Compute modulus $n_i=q_ip_i$.

b) Choose a number $f_i\in[0,(p_i-1)(q_i-1)-1]$ such that $\gcd(f_i,(p_i-1)(q_i-1))=1$.

c) Compute $d_i=f_i^{-1} \bmod (p_i-1)(q_i-1)$.

d) Output the following:

— Public key = $vk_i=(n_i,f_i)$;

— Private key = $sk_i=d_i$.

6.4.3 Ring signature process

In the ring signature process, a true signer first chooses a set of $N-1$ potential signers. The true signer also chooses a random integer π satisfying $1\le\pi\le N$, which denotes the index of the true signer. The true signer then signs a message using its private key and the public keys of the potential signers, without requiring their approval or assistance. In this process the potential signers are arranged in random order.

When inputting a private key sk_π, a randomly ordered set of public keys $L=\{vk_1,...,vk_N\}$ (including vk_π), and a message $m\in\{0,1\}^*$ to be signed, the ring signature process involves the following steps.

a) Choose a random $e_\pi\in[0,n_\pi-1]$, and compute $c_{\pi+1}=H_{\pi+1}(L,m,e_\pi)$.

b) Choose a random $s_i\in[0,n_i-1]$, and compute $e_i=c_i+s_i^{f_i} \bmod n_i$ and $c_{i+1}=H_{i+1}(L,m,e_i)$ for $i=\pi+1,...,N,1,...,\pi-1$, where index $i=N+1=1$.

c) Compute $s_\pi=(e_\pi-c_\pi)^{d_\pi} \bmod n_\pi$.

d) Output the signature $\sigma=(c_1,s_1,...,s_N)$.

6.4.4 Ring signature verification process

In the verification process, a verifier uses the public keys of the true signer and potential signers involved in the signature process to verify the signature. The verifier checks whether the signature was signed by one of the signers in the set without learning who the true signer is.

When inputting a message m, a signature $\sigma=(c_1,s_1,...,s_N)$, and a set of public keys $L=\{vk_1,...,vk_N\}$, the verification process takes the following steps.

a) Compute $e_i=c_i+s_i{}^{f_i} \bmod n_i$ and $c_{i+1}=H_{i+1}(L,m,e_i)$ for $i=1,\ldots,N-1$.

b) Compute $e_N=c_N+s_N{}^{f_N} \bmod n_N$.

c) If $c_1=H_1(L,m,e_N)$ return 1 (valid), else return 0 (invalid).

7 Mechanisms with linking capability

7.1 General

This clause specifies one digital signature mechanism with linking capability (as known as "linkable ring signature").

Linking capability allows any verifier to know if two signatures are generated by the same signer. However, it does not allow the verifier to know the identity of the actual signer. Linkability can be group-linkable type (the signer can be linked if the signer has generated two signatures and the group is the same for these two signatures) or event-linkable type (the signer can be linked if the signer has generated two signatures and the event is the same for these two signatures). For the latter, signers partaking in the same event shall agree on consistent event tag or string that is used in signing.

NOTE 1 The mechanism given in 7.2 is based on the scheme originally specified in Reference [6].

NOTE 2 Security properties of a linkable ring signature include unforgeability, anonymity, linkability and non-slanderability. Unforgeability and anonymity are the same as in the general ring signature described in Clause 5. Linkability ensures any signatures produced by one signer to be linkable. Any public verifier knows if any two signatures are generated by the same signer. Non-slanderability prevents any malicious party from slandering an honest user for generating two linked signatures. While the original paper in Reference [6] does not specify the non-slanderability property of the mechanism (it was first defined in Reference [5], a year after the publication of Reference [6]), non-slanderability is considered an important property of linkable ring signature if it is deployed in many applications such as blockchain. The non-slanderability property of Reference [6] is later proven in Reference [4].

7.2 Mechanism 1

7.2.1 Symbols

The following symbols apply in the specification of this mechanism.

— c_i, s_i, u, x_i : integers in the range $[1, q-1]$

— g : a generator in group G

— G : a cyclic group with order q

— q : prime number

— $y_i, h, \tilde{y}$: group elements in G

— H_1 : a hash function that outputs an integer in the range $[0, q-1]$ as defined in Annex B

— H_2 : a hash function that outputs an element in G as defined in Annex B. It shall be statistically independent to H_1

7.2.2 Key generation process

The key generation process involves each user $i\,(1 \le i \le N)$ independently performing the following steps.

a) Choose a random $x_i \in [0, q-1]$, and compute $y_i = g^{x_i} \in G$.

b) Output the following:

— Public key = $y_i = g^{x_i} \in G$;

— Private key = $x_i \in [0, q-1]$.

7.2.3 Ring signature process

In the ring signature process, a true signer first chooses a set of $N-1$ potential signers. The true signer also chooses a random integer π satisfying $1 \le \pi \le N$, which denotes the index of the true signer. The true signer then signs a message using its private key and the public keys of the potential signers, without requiring their approval or assistance. In this process the potential signers are arranged in random order.

When inputting a private key x_π, a randomly ordered set of public keys $L = \{y_1, \ldots, y_N\}$ (including y_π), and a message $m \in \{0,1\}^*$ to be signed, the ring signature process involves the following steps.

a) Compute the linking base h. h can be generated as $h = H_2(L)$ if it is group-linkable type. (Refer to 7.2.6 for event-linkable type).

b) Compute the linking tag $\tilde{y} = h^{x_\pi}$.

c) Choose a random integer u in $[0, q-1]$.

d) Compute $c_{\pi+1} = H_1\left(L, \tilde{y}, m, g^u, h^u\right)$.

e) Choose random integers $s_i \in [0, q-1]$ where $i \in [1, N] \setminus \pi$.

f) Compute $c_{i+1} = H_1\left(L, \tilde{y}, m, g^{s_i} y_i^{c_i}, h^{s_i} \tilde{y}^{c_i}\right)$ for $i = \pi+1, \ldots, N, 1, \ldots, \pi-1$.

g) Compute $s_\pi = u - x_\pi c_\pi \bmod q$.

h) Output the signature $\sigma = (c_1, s_1, \ldots, s_N, \tilde{y})$ with the ordered set of public keys L.

7.2.4 Ring signature verification process

In the verification process, a verifier uses the public keys of the true signer and potential signers involved in the signature process to verify the signature. The verifier checks whether the signature was signed by one of the signers in the set without learning who the true signer is.

When inputting a message m, a signature $(c_1, s_1, \ldots, s_N, \tilde{y})$, and a list of public keys $L = \{y_1, \ldots, y_N\}$, the verification process takes the following steps:

a) Compute the linking base h as in the signing process.

b) Compute $z_i' = g^{s_i} y_i^{c_i}$, $z_i'' = h^{s_i} \tilde{y}^{c_i}$ and $c_{i+1} = H_1\left(L, \tilde{y}, m, z_i', z_i''\right)$ for $i = 1, \ldots, N-1$.

c) Compute $z_N' = g^{s_N} y_N^{c_N}$ and $z_N'' = h^{s_N} \tilde{y}^{c_N}$.

d) If $c_1 = H_1\left(L, \tilde{y}, m, z_N', z_N''\right)$ return 1 (valid), else return 0 (invalid).

7.2.5 Ring signature linking process

Given two valid signatures $\sigma=(c_1,s_1,\ldots,s_N,\tilde{y})$ and $\sigma'=\left(c_1',s_1',\ldots,s_N',\tilde{y}'\right)$, the linking process takes the following step:

a) If $\tilde{y}=\tilde{y}'$, output 1 (linked), else output 0 (not linked).

7.2.6 Event-linkable type

In case of event-linkable type (the signer can be linked if the signer has generated two signatures and the event is the same for these two signatures), some changes shall be made. In the signing process 7.2.3, step a), the computation of h shall be changed to $h=H_2(event)$ and the description event shall be put inside the hash in step d) and step f). In the verification process 7.2.4, the description event shall be put inside the hash in step b) and step d).

8 Mechanisms with tracing capability

8.1 General

This clause specifies one digital signature mechanism with tracing capability (as known as “traceable ring signature”).

Tracing capability allows any verifier to know who the actual signer is, if two signatures are generated by the same signer. Tracing capability is issue-traceable (the signer can be traced if the signer has generated two signatures and the issue is the same for these two signatures). Tracing algorithm outputs ‘traced’ if issues are same and messages are different, ‘linked’ if issues are same and messages are same, and ‘independent’ if issues are different and messages are different or same.

NOTE The mechanism given in 8.2 is based on the scheme originally specified in Reference [3].

8.2 Mechanism 1

8.2.1 Symbols

The following symbols apply in the specification of this mechanism.

— c_i,s_i,α,x_i: integers in the range $[0,q-1]$

— g: a generator in group G

— G: a cyclic group with order q

— q: prime number

— y_i,h,σ_i: group elements in G

— H,H': hash functions that output an element in G as defined in Annex B. These two hash functions shall be statistically independent.

— H'': a hash function that outputs an integer in the range $[0,q-1]$ as defined in Annex B. The hash function shall be statistically independent to H,H'.

8.2.2 Key generation process

The key generation process involves each user $i\,(1\le i\le N)$ independently performing the following steps.

a) Choose a random $x_i\in[0,q-1]$, and compute $y_i=g^{x_i}\in G$.

b) Output the following:

— Public key = $y_i = g^{x_i} \in G$;

— Private key = $x_i \in [0, q-1]$.

8.2.3 Ring signature process

In the ring signature process, a true signer first chooses a set of $N-1$ potential signers. The true signer also chooses a random integer π satisfying $1 \le \pi \le N$, which denotes the index of the true signer. The true signer then signs a message using its private key and the public keys of the potential signers, without requiring their approval or assistance. In this process the potential signers are arranged in random order.

When inputting a private key x_π, a randomly ordered set of public keys $L = \{issue, y_1, \ldots, y_N\}$ (including y_π), and a message $m, issue \in \{0,1\}^*$ to be signed, the ring signature process involves the following steps.

NOTE The computation of $a_i = g^{s_i} y_i^{c_i}$ in the following step b) is differ from the original description of Reference [3]. The description in the following step b) is correct and the original description is not correct.

a) Compute $h = H(L)$ and $\sigma_\pi = h^{x_\pi}$. Compute $A_0 = H'(L, m)$ and $A_1 = (\sigma_\pi / A_0)^{1/\pi}$. Compute $\sigma_j = A_0 A_1^j$ for all $j \ne \pi$.

b) Choose a random $\alpha \in [0, q-1]$.

c) Compute $a_\pi = g^\alpha$, $b_\pi = h^\alpha$.

d) Choose random $s_i, c_i \in [0, q-1]$ where $i \in [1, N] \setminus \pi$. Compute $a_i = g^{s_i} y_i^{c_i}$, $b_i = h^{s_i} \sigma_i^{c_i}$ where $i \in [1, N] \setminus \pi$.

e) Compute $c = H''(L, m, A_0, A_1, a_1, \ldots, a_N, b_1, \ldots, b_N)$ and $c_\pi = c - \sum_{i \ne \pi} c_i \bmod q$.

f) Compute $s_\pi = \alpha - c_\pi x_\pi \bmod q$.

g) Output the signature $\sigma = (A_1, c_1, \ldots, c_N, s_1, \ldots, s_N)$.

8.2.4 Ring signature verification process

In the verification process, a verifier uses the public keys of the true signer and potential signers involved in the signature process to verify the signature. The verifier checks whether the signature was signed by one of the signers in the set without learning who the true signer is.

When inputting a message m, a signature $\sigma = (A_1, c_1, \ldots, c_N, s_1, \ldots, s_N)$, and a list of public keys $L = \{issue, y_1, \ldots, y_N\}$, the verification process takes the following steps:

a) Check $c_i, z_i \in [0, q-1]$, and check $g, A_1, y_i \in G$.

b) Compute $h = H(L)$, $A_0 = H'(L, m)$, and $\sigma_i = A_0 A_1^i$ for $i = 1, \ldots, N$.

c) Compute $a_i = g^{s_i} y_i^{c_i}$, $b_i = h^{s_i} \sigma_i^{c_i}$ for $i = 1, \ldots, N$.

d) Compute $c = H''(L, m, A_0, A_1, a_1, \ldots, a_N, b_1, \ldots, b_N)$. If $c = \sum_i c_i \bmod q$ return 1 (valid). Else return 0 (invalid).

8.2.5 Ring signature tracing process

Given two pairs of valid signature and message $(\sigma=(A_1,\ldots),m)$ and $(\sigma'=(A_1',\ldots),m')$ with the same list of public keys $L=\{issue, y_1,\ldots,y_N\}$, the tracing process takes the following step.

a) Compute $h=H(L)$, $A_0=H'(L,m)$, and $\sigma_i=A_0A_1^i$, $\sigma_i'=A_0A_1'^i$ for $i=1,\ldots,N$.

b) Set TList as empty list.

c) If $\sigma_i=\sigma_i'$ then append y_i to TList for $i=1,\ldots,N$.

d) If TList $=\{y_i\}$, output y_i (traced). Else if TList $=\{y_1,\ldots,y_N\}$, output 1 (linked). Otherwise output 0 (independent).

9 Mechanisms with threshold capability

9.1 General

This clause specifies two digital signature mechanisms with threshold capability.

NOTE The mechanisms given in 9.2 and 9.3 are based on the schemes originally specified in Reference [2].

9.2 Mechanism 1

9.2.1 Symbols

The following symbols apply in the specification of this mechanism.

— c_i, s_i, α_i : integers in the range $[0, q_i-1]$

— F_t : a finite field of order t, where t is prime number of the same bit length as q_i

— g_i : a generator in group G_i

— G_i : a cyclic group with order q_i

— q_i : prime number

— y_i, e_i : group elements in G_i

— H_0 : a hash function that outputs an element in finite field F_t as defined in Annex B

— H_i : a hash function that outputs an integer in the range $[0, q_i-1]$ as defined in Annex B

9.2.2 Key generation process

The key generation process involves each user $i\,(1\le i\le N)$ independently performing the following steps.

a) Choose a random $x_i\in[0,q_i-1]$, and compute $y_i=g_i^{x_i}\in G_i$.

b) Output the following:

— Public key = $y_i=g_i^{x_i}\in G_i$;

— Private key = $x_i\in[0,q_i-1]$.

9.2.3 Ring signature process

In the k-out-of-N threshold ring signature process, k true signers choose a set of $N-k$ potential signers. $S=\{i_1,\ldots,i_k\}\subset\{1,\ldots,N\}$ denotes the indices of the true signers and signs a message by using the private

keys of true signers and the public keys of the potential signers without requiring their approval or assistance.

When inputting a set of private keys $\{x_{i_1},\ldots,x_{i_k}\}$, a randomly ordered set of public keys $L=\{y_1,\ldots,y_N\}$ (including $y_{i_1},\ldots,y_{i_k}$), and a message $m\in\{0,1\}^*$ to be signed, the k-out-of-N threshold ring signature process involves the following steps.

a) For $i\in S$, choose a random $\alpha_i\in[0,q_i-1]$, and compute $e_i=g_i{}^{\alpha_i}\in G_i$.

b) For $i\in[1,N]\setminus S$, choose random $z_i\in F_t$, $s_i\in[0,q_i-1]$, compute $c_i=H_i(z_i)$, and $e_i=g_i{}^{s_i}\,y_i{}^{c_i}\in G_i$.

c) Compute $z_0=H_0(L,m,k,e_1,e_2,\ldots,e_N)$ and find a polynomial P over F_t of degree $N-k$ such that $P(i)=z_i$ for $i\in[0,N]\setminus S$ by Lagrange interpolation[8].

d) For $i\in S$, compute $c_i=H_i(P(i))$, $s_i=\alpha_i-c_i x_i \bmod q_i$.

e) Output the signature $\sigma=(P,s_1,\ldots,s_N)$.

9.2.4 Ring signature verification process

In the verification process, a verifier uses the public keys of the true signers and potential signers involved in the signature process to verify the signature. The verifier checks whether the signature was signed by the signers in the set without learning who the true signers are.

When inputting a message m, a signature $\sigma=(P,s_1,\ldots,s_N)$, and a set of public keys $L=\{y_1,\ldots,y_N\}$, the verification process takes the following steps.

a) For $i=1,\ldots,N$, compute $c_i=H_i(P(i))$, and $e_i=g_i{}^{s_i}\,y_i{}^{c_i}\in G_i$.

b) If $P(0)=H_0(L,m,k,e_1,e_2,\ldots,e_N)$ and P is F_t coefficient degree $N-k$ polynomial, then return 1 (valid). Else return 0 (invalid).

9.3 Mechanism 2

9.3.1 Symbols

The following symbols apply in the specification of this mechanism.

— f_i, d_i: integer in the range $[0,(p_i-1)(q_i-1)-1]$

— F_t: a finite field of order t, where t is prime number of the same bit length as q_i

— l_n: integer equal to or greater than 2048

— n_i: l_n-bit integer

— p_i, q_i, p_i', q_i': prime numbers

— H_0: a hash function that outputs an element in finite field F_t as defined in Annex B

— H_i: a hash function that outputs an integer in the range $[0,n_i-1]$ as defined in Annex B

9.3.2 Key generation process

The key generation process involves each user $i\,(1\le i\le N)$ independently performing the following steps.

a) Choose prime numbers p_i', q_i' with $l_n/2$ bits such that $p_i=2p_i'+1, q_i=2q_i'+1$ are prime numbers. Compute modulus $n_i=q_i p_i$.

b) Choose a number $f_i \in [0,(p_i-1)(q_i-1)-1]$ such that $\gcd(f_i,(p_i-1)(q_i-1))=1$.

c) Compute $d_i = f_i^{-1} \bmod (p_i-1)(q_i-1)$.

d) Output the following:

— Public key = $vk_i = (n_i, f_i)$;

— Private key = $sk_i = d_i$.

9.3.3 Ring signature process

In the k-out-of-N threshold ring signature process, k true signers choose a set of $N-k$ potential signers. $S=\{i_1,\ldots,i_k\}\subset\{1,\ldots,N\}$ denotes the indices of the true signers and signs a message by using the private keys of true signers and the public keys of the potential signers without requiring their approval or assistance.

When inputting a set of private keys $\{sk_{i_1},\ldots,sk_{i_k}\}$, a randomly ordered set of public keys $L=\{vk_1,\ldots,vk_N\}$ (including $y_{i_1},\ldots,y_{i_k}$), and a message $m\in\{0,1\}^*$ to be signed, the k-out-of-N threshold ring signature process takes the following steps.

a) For $i\in S$, choose a random $e_i \in [0,n_i-1]$.

b) For $i\in[1,N]\setminus S$, choose random $z_i \in F_t, s_i \in [0,n_i-1]$, compute $c_i = H_i(z_i)$, and $e_i = c_i + s_i^{f_i} \bmod n_i$.

c) Compute $z_0 = H_0(L,m,k,e_1,e_2,\ldots,e_N)$ and find a polynomial P over F_t of degree $N-k$ such that $P(i)=z_i$ for $i=0, i\in[1,N]\setminus S$ by Lagrange interpolation[8].

d) For $i\in S$, compute $c_i = H_i(P(i))$, $s_i = (e_i - c_i)^{d_i} \bmod n_i$.

e) Output the signature $\sigma = (P, s_1, \ldots, s_N)$.

9.3.4 Ring signature verification process

In the verification process, a verifier uses the public keys of the true signers and potential signers involved in the signature process to verify the signature. The verifier checks whether the signature was signed by the signers in the set without learning who the true signers are.

When inputting a message m, a signature $\sigma = (P, s_1, \ldots, s_N)$, and a set of public keys $L=\{vk_1,\ldots,vk_N\}$, the verification process takes the following steps.

a) For $i=1,\ldots,N$, compute $c_i = H_i(P(i))$, and $e_i = c_i + s_i^{f_i} \bmod n_i$.

b) If $P(0) = H_0(L,m,k,e_1,e_2,\ldots,e_N)$ and P is F_t-coefficient degree $N-k$ polynomial, then return 1 (valid). Else return 0 (invalid).

Annex A
(normative)

Object identifiers

This annex lists the object identifiers assigned to the mechanisms specified in this document.

```
AnonymousSignatureUsingMultiplePK { iso(1) standard(0) anonymous-digital-signatures(20008)
part3(3) asn1-module(1) mechanisms-using-multiple-public-keys(0) }

DEFINITIONS EXPLICIT TAGS ::= BEGIN

-- EXPORTS All; --

-- IMPORTS None; --

OID ::= OBJECT IDENTIFIER -- alias

-- Synonyms –

id-as-mpk OID ::= { iso(1) standard(0) anonymous-digital-signatures(20008) part3(3)
algorithm(0) }

-- Assignments --
id-as-mpk-rs-m-1 OID ::= { id-as-mpk rs-mechanism1(1) }
id-as-mpk-rs-m-2 OID ::= { id-as-mpk rs-mechanism2(2) }
id-as-mpk-rs-m-3 OID ::= { id-as-mpk rs-mechanism3(3) }
id-as-mpk-lrs-m-1 OID ::= { id-as-mpk lrs-mechanism1(4) }
id-as-mpk-trs-m-1 OID ::= { id-as-mpk trs-mechanism1(5) }
id-as-mpk-thrs-m-1 OID ::= { id-as-mpk thrs-mechanism1(6) }
id-as-mpk-thrs-m-2 OID ::= { id-as-mpk thrs-mechanism2(7) }
END
```

Annex B
(normative)

Conversion functions

B.1 General

This annex specifies the conversion functions and hash-functions that shall be used with this standard where specified.

B.2 Conversion functions

The following conversion functions described in ISO/IEC 18033-2:2006 shall be used with this standard where specified.

— I2OSP: conversion function from integers to octet strings.

— FE2OSP: conversion function from extended filed to octet strings.

— EC2OSP: conversion function from points on elliptic curve to octet strings.

— OS2IP: conversion function from octet strings to integers.

— OS2FEP: conversion function from octet strings to extended filed.

— OS2ECP: conversion function from octet strings to points on elliptic curve.

B.3 Hash functions

The following hash functions described in ISO/IEC 18033-5:2015 shall be used with this standard where specified.

— IHF1: hash function from bit strings to integers.

— PHF1: hash function from bit strings to points on supersingular elliptic curve.

The following hash functions described in ISO/IEC 20008-2:2013 shall be used with this standard where specified.

— HBS2PF: hash function from bit strings to prime field.

— HBS2ECP: hash function from bit strings to points on elliptic curve.

The following hash functions described in RFC 9380 shall be used with this standard where specified.

— Hash functions from bit strings to finite field.

— Hash functions from bit strings to points on elliptic curve.

Hash functions used in this standard are assumed to be random oracles to satisfy the security proofs for the mechanisms.

Annex C
(informative)

Numerical examples of mechanisms in this document

C.1 Recommended parameters

Recommended parameters for the mechanisms specified in 6.2, 6.4, and 9.3 are as follows.

128-bit security RSA, e.g. 3072-bit RSA.

128-bit security hash function, e.g. SHA-256.

Recommended parameters for the mechanisms specified in 6.3, 7.2, 8.2, and 9.2 are as follows.

128-bit security elliptic curve, e.g. secp256r1 or secp256k1.

128-bit security hash function, e.g. SHA-256.

C.2 Numerical examples

C.2.1 General

Numerical examples using hash function to elliptic curves in RFC 9380 are provided in C.2.2 to C.2.6.

C.2.2 Parameters

Parameters for elliptic curve.

secp256k1.

Parameters for expand_message defined in RFC 9380.

expand_message_xmd is used.

dst = QUUX-V01-CS02-with-secp256k1_XMD:SHA-256_SSWU_RO_

Parameters for hash_to_field defined in RFC 9380.

L = ceil([ceil($\log_2(p)$) + k] / 8), where p is characteristic of the finite field and k=128 is security parameter.

count=1: the number of outputs

Parameters for hash_to_curve defined in RFC 9380.

suit = secp256k1_XMD:SHA-256_SSWU_RO_

Conversion function from EC point to octet string.

EC2OSP((x,y)) = I2OSP(4,1)||I2OSP(x,32)||I2OSP(y,32)