BS ISO/IEC 23264-2:2024

BSI Standards Publication

Information security — Redaction of authentic data

Part 2: Redactable signature schemes based on asymmetric mechanisms

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 23264-2:2024

National foreword

This British Standard is the UK implementation of ISO/IEC 23264-2:2024.

The UK participation in its preparation was entrusted to Technical Committee IST/33/2, Cryptography and Security Mechanisms.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2024
Published by BSI Standards Limited 2024

ISBN 978 0 539 04796 7

ICS 35.030

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 August 2024.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 23264-2:2024

International Standard

ISO/IEC 23264-2

First edition
2024-08

Information security — Redaction of authentic data —

Part 2: Redactable signature schemes based on asymmetric mechanisms

Sécurité de l'information — Rédaction de données authentifiées —

Partie 2: Schémas de signature éditable basés sur des mécanismes asymétriques

Reference number
ISO/IEC 23264-2:2024(en)

© ISO/IEC 2024

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 23264-2:2024

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 23264-2:2024

Contents

Page

Foreword ... **v**
Introduction ... **vi**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms and definitions ... **1**
4 Symbols and conventions ... **3**
4.1 Symbols ... 3
4.2 Conventions ... 4
5 General ... **5**
6 Generic construction from signature schemes and hash-functions ... **5**
6.1 Parameters ... 5
6.2 Construction ... 6
6.2.1 Key generation process ... 6
6.2.2 Redactable attestation process ... 6
6.2.3 Redaction process ... 7
6.2.4 Verification process ... 8
7 Scheme SBZ02-MERSAProd ... **8**
7.1 Parameters ... 8
7.2 Construction ... 9
7.2.1 Key generation process ... 9
7.2.2 Redactable attestation process ... 9
7.2.3 Redaction process ... 10
7.2.4 Verification process ... 11
8 Scheme BBDFFKMOPPS10 ... **12**
8.1 Parameters ... 12
8.2 Construction ... 12
8.2.1 Key generation process ... 12
8.2.2 Redactable attestation process ... 12
8.2.3 Redaction process ... 14
8.2.4 Verification process ... 15
9 Scheme DPSS15 ... **17**
9.1 Parameters ... 17
9.2 Subroutine: RSA Accumulators ... 17
9.3 Construction ... 18
9.3.1 Key generation process ... 18
9.3.2 Redactable attestation process ... 19
9.3.3 Redaction process ... 20
9.3.4 Verification Process ... 20
10 Scheme MHI06 ... **21**
10.1 Parameters ... 21
10.2 Construction ... 22
10.2.1 Key generation process ... 22
10.2.2 Redactable attestation process ... 22
10.2.3 Redaction process ... 23
10.2.4 Verification Process ... 24
11 Scheme MIMSYTI05 ... **25**
11.1 Parameters ... 25
11.2 Construction ... 25
11.2.1 Key generation process ... 25
11.2.2 Redactable attestation process ... 25

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

11.2.3 Redaction process ... 26
11.2.4 Verification Process ... 27

Annex A (normative) **Object identifiers** ... **29**

Annex B (informative) **Overview of properties of redactable signature schemes based on asymmetric mechanisms** ... **30**

Annex C (informative) **Criteria for inclusion of schemes in this document** ... **33**

Annex D (informative) **Numerical examples** ... **34**

Bibliography ... **57**

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

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 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 23264 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 and www.iec.ch/national-committees.

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

Introduction

This document specifies cryptographic mechanisms to redact authentic data where the redactable attestation scheme is based on asymmetric mechanisms.

Attestation schemes, in particular digital signature schemes or message authentication codes, can be used to provide data integrity and data origin authentication. Redactable attestation can be used to blank out parts, herein called fields, of an attested message without invalidating the attestation on the remaining contents of the message. This redaction process requires a redaction key. The redaction key computationally does not reveal the attestation key, by which schemes can allow for public redactions. Any other modification of the document (e.g. redaction of other message parts, or insertion/modification of any parts) will invalidate the attestation. Schemes can have specific additional security properties, which are described in ISO/IEC 23264-1. The achievable properties for each scheme are stated in this document.

Redactable attestation schemes are a basic building block in many privacy-preserving applications, such as privacy-preserving data sharing or authentication, where a party may decide to forward only necessary information to a receiver, while the latter is still assured that the received information was previously attested, for example, by a public authority.

The objective of the ISO/IEC 23264 series is to remedy existing incompatibilities or inconsistently defined properties found in academic literature, and to ease the real-world adoption of this technology. Specifically, the goal of this document is to focus on algorithms that enable the authenticity-preserving redaction of general data structures like sets or ordered lists based on asymmetric cryptography. It adheres to the common terminology and description of cryptographic properties for redactable attestation schemes given in ISO/IEC 23264-1.

The ISO/IEC 23264 series complements ISO/IEC 27038, which specifies the redaction of digital documents without considering the authenticity of the data.

This document contains the following algorithms based on asymmetric cryptography:

— generic construction from signature schemes and hash-functions

— scheme SBZ02-MERSAProd

— scheme BBDFFKMOPPS10

— scheme DPSS15

— scheme MHI06

— scheme MIMSYTI05

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

Information security — Redaction of authentic data —

Part 2: Redactable signature schemes based on asymmetric mechanisms

1 Scope

This document specifies cryptographic mechanisms to redact authentic data. The mechanisms described in this document offer different combinations of the security properties defined and described in ISO/IEC 23264-1. For all mechanisms, this document describes the processes for key generation, generating the redactable attestation, carrying out redactions and verifying redactable attestations.

This document contains mechanisms that are based on asymmetric cryptography using three related transformations:

— a public transformation defined by a verification key (verification process for verifying a redactable attestation),

— a private transformation defined by a private attestation key (redactable attestation process for generating a redactable attestation), and

— a third transformation defined by the redaction key (redaction process) allowing to redact authentic information within the constraints set forth during generation of the attestation such that redacted information cannot be reconstructed.

This document contains mechanisms which, after a successful redaction, allow the attestation to remain verifiable using the verification transformation and attest that non-redacted fields of the attested message are unmodified. This document further details that the three transformations have the property whereby it is computationally infeasible to derive the private attestation transformation, given the redaction and or the verification transformation and key(s).

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 23264-1, *Information security — Redaction of authentic data — Part 1: General*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23264-1 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/

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

3.1
balanced tree
height-balanced tree
tree (3.13) in which the heights of the immediate subtrees of each node differ at most by one

[SOURCE: ISO/IEC 2382:2015, 2121638, modified — notes to entry have been removed.]

3.2
binary tree
ordered *tree* (3.13) in which each node has at most two other nodes that are directly subordinate

[SOURCE: ISO/IEC 2382:2015, 2121636, modified — notes to entry have been removed.]

3.3
balanced binary tree
binary tree (3.2) which is a *balanced tree* (3.1)

3.4
collision-resistant hash-function
hash-function (3.6) satisfying the following property: it is computationally infeasible to find any two distinct inputs which map to the same output

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment (see ISO/IEC 10118-1:2016, Annex C).

[SOURCE: ISO/IEC 10118-1:2016, 3.1, modified — reference to ISO/IEC 10118-1:2016, Annex C has been added in the note.]

3.5
hash-code
string of bits which is the output of a *hash-function* (3.6)

Note 1 to entry: The literature on this subject contains a variety of terms that have the same or similar meaning as hash-code. Modification Detection Code, Manipulation Detection Code, digest, hash-result, hash-value and imprint are some examples.

[SOURCE: ISO/IEC 10118-1:2016, 3.3]

3.6
hash-function
function which maps strings of bits of variable (but usually upper bounded) length to fixed-length strings of bits, satisfying the following two properties:

— for a given output, it is computationally infeasible to find an input which maps to this output;

— for a given input, it is computationally infeasible to find a second input which maps to the same output

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment. See ISO/IEC 10118-1:2016, Annex C.

[SOURCE: ISO/IEC 10118-1:2016, 3.4, modified — reference to ISO/IEC 10118-1:2016, Annex C has been added in the note.]

3.7
height
maximum number of nodes in any path leading from the *root node* (3.10) to a *leaf node* (3.8)

[SOURCE: ISO/IEC 2382:2015, 2121637, modified — “leaf node” has replaced "terminal node" in the definition; notes to entry have been removed.]

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

3.8
leaf node
node that has no *subordinate node* (3.11)

[SOURCE: ISO/IEC 2382:2015, 2121489, modified — term "leaf node" has replaced the original terms "leaf" and "terminal node".]

3.9
parent node
node to which at least one other node is directly subordinate

[SOURCE: ISO/IEC 2382:2015, 2121488, modified — notes to entry have been removed.]

3.10
root of a tree
root node
node of a *tree* (3.13) that has only *subordinate nodes* (3.11)

3.11
subordinate node
node at the other end of an outgoing arc; a node may have zero, one, or more subordinates

[SOURCE: ISO/IEC 9804:1998, 3.6.64]

3.12
subtree
part of a *tree* (3.13) including a node and all its *subordinate nodes* (3.11)

[SOURCE: ISO/IEC 2382:2015, 2121634, modified —notes to entry have been removed.]

3.13
tree
data structure containing nodes that are linked together hierarchically by oriented arcs with at most one parent node for each node, and with only one *root node* (3.10)

[SOURCE: ISO/IEC 2382:2015, 2121633, modified – “by oriented arcs” has been added to the definition; notes to entry have been removed.]

4 Symbols and conventions

4.1 Symbols

Throughout this document, the following symbols are used.

adm	description of redacted or original admissible changes
adm'	description of redacted admissible changes
ak	attestation key
att	redactable or redacted attestation
att'	redacted attestation
m	redacted or original message
m'	redacted message
$m_1, \ldots, m_n$	individual field

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

mod	description of modification instructions
n	number of fields
pk	public verification key used internally by asymmetric signature algorithm or accumulator scheme
rk	redaction key
root	value assigned as content to the node forming the root of a tree
sk	secret signing key used internally by asymmetric signature algorithm or accumulator scheme
tag	string used to mark messages or message fields
vk	verification key
Z	set of one or more domain parameters
Σ	output of a digital signature scheme as defined in ISO/IEC 14888-1
reject, *accept*	output of a digital signature verification

4.2 Conventions

A triple of message, attestation, and admissible changes is denoted as (*m, att, adm*). In the same way a triple of redacted message, a redacted attestation and redacted admissible changes is denoted by (m', att', adm').

A specific value of a symbol *sym* is denoted as *sym**, or *sym*** in order to differentiate their specific values. An equality is stated using the equals sign =.

EXAMPLE 1 The statement that *m**=*m*** means that the contents of *m** are equal to the contents of *m***.

The symbol || denotes a concatenation of strings.

When the symbol || is applied to values which are not represented as strings, the values are required to be converted into a string before they are concatenated. See for example Reference [25] for a conversion of an integer into a string.

The symbol $\lfloor a \rfloor$ indicates the largest integer not exceeding *a*, following the definition of the floor function specified in ISO/IEC 15444-1.

EXAMPLE 2 $\lfloor -2,46 \rfloor = -3, \left\lfloor \frac{1}{3} \right\rfloor = 0$

The symbol a / b denotes the result of the division of *a* by *b*.

This document uses set notation and symbols to denote set-like operations:

\	operation *set minus*, i.e. $A \setminus B$ denotes the contents of set *A* without contents of set *B*
∪	operation *set union*, i.e. $A \cup B$ denotes all contents from set *A* together with those of set *B*
⊆	statement *sub set or equal*, i.e. $A \subseteq B$ denotes that all contents from set *A* are contained as contents in set *B*
|*A*|	denotes the size of the set *A*, i.e. the number of elements contained in *A*
∅	empty set

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

Where applicable, the output of a hash-function is interpreted as an integer.

5 General

This document adheres to the common terminology and description of cryptographic properties for redactable attestation schemes given in ISO/IEC 23264-1.

Redactable attestation schemes provide data integrity and data origin authentication. While redactable attestation can be used to allow a redactor to blank out parts, herein called fields, of an attested message without invalidating the attestation on the remaining contents of the message, any other modification of the document (e.g. redaction of other message parts, or insertion/modification of any parts) will invalidate the attestation. This redaction process requires a redaction key. The redaction key computationally does not reveal the attestation key, by which schemes can allow for redactions that are either public or by a party other than the attestor. The verification key is cryptographically linked to the attestation key and allows verification of the origin of the attested data.

The schemes can have specific additional security properties, which are described in ISO/IEC 23264-1. The achievable properties for each scheme are stated in this document.

This document describes algorithms that enable the authenticity-preserving redaction of general data structures like sets or ordered lists based on asymmetric cryptography.

This document contains the following algorithms based on asymmetric cryptography:

— Generic construction from signature schemes and hash-functions

— Scheme SBZ02-MERSAProd

— Scheme BBDFFKMOPPS10

— Scheme DPSS15

— Scheme MHI06

— Scheme MIMSYTI05

Finally, this document contains the following annexes:

— Annex A, which provides object identifiers which shall be used to identify the mechanisms defined in this document.

— Annex B, which contains an overview of the different properties as defined and described in ISO/IEC 23264-1 that are achieved by the different algorithms contained in this document.

— Annex C, which lists criteria for inclusion in this document.

— Annex D, which gives numerical examples to understand and check implementations of the algorithms contained in this document.

6 Generic construction from signature schemes and hash-functions

6.1 Parameters

The generic construction makes use of the following parameters:

— digital signature scheme as defined in ISO/IEC 14888-1;

— collision-resistant hash-function Hash as defined in ISO/IEC 10118-1;

— security parameter λ.

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

The security parameter λ shall be met by all involved algorithms, especially for the hash-function and digital signature scheme.

6.2 Construction

6.2.1 Key generation process

The key generation algorithm of the redactable attestation scheme consists of the following two procedures:

a) generate the set of domain parameters *Z*;

b) generate the attestation key *ak*, verification key *vk* and redaction key *rk* as follows:

   1) use the key generation process of the digital signature scheme, such that the digital signature scheme's signature key serves as the attestation key *ak*,

   2) use the digital signature scheme's verification key as the verification key *vk*,

   3) use the digital signature scheme's verification key also as the redaction key *rk*, such that *rk* = *vk.*

The security parameter λ is taken into account such that the digital signature scheme offers at least the indicated security strength.

6.2.2 Redactable attestation process

This process takes the following inputs:

— set of domain parameters Z

— attestation key ak

— message m consisting of n fields $m_1, \ldots, m_n$

— admissible changes $adm = \{1, \ldots, n\}$

The process Split() to split the message m into fields is generally out of scope of this document. However, the process Split() shall be defined in such a way that fields can be transformed into the message and vice versa, in a reproducible way every time this conversion is necessary, i.e. if Split(m)= $m_1, \ldots, m_s$ then Split^{-1}($m_1, \ldots, m_s$) = m. The scheme protects the content of each field and their order.

The process requires the generation of random data. Refer to ISO/IEC 18031 to generate this securely.

The process does not allow the specification of admissible changes, so by default all fields of the message ($m_1, \ldots, m_n$) are admissible and can be redacted by anyone. In order to make this clear for the user, the input for *adm* shall contain all field indices, i.e. $adm = \{1, \ldots, n\}$.

The process is as follows:

a) Generate a Merkle tree[17] as follows: Generate a binary tree which is also a balanced tree, henceforth referred to as balanced binary tree, of sufficient height such that it has *k* leaf nodes with $2n > k \geq n$.

b) Choose a uniformly random λ-bit tag_{msg} for the message; and choose *n* random λ-bit tags tag_i, one for each field of *m*. No tag_i shall contain only zeros, denoted as $tag_i \neq 0^{\lambda}$.

c) For each $i = 1, \ldots, n$, compute the hash-code $h_i = \mathrm{Hash}\left(tag_{msg} \| m_i \| tag_i\right)$ using a collision-resistant hash-function.

d) Initialize the Merkle tree using $h_1, \ldots, h_n$ as values for the *n* left-most leaf nodes of the balanced binary tree, use the empty string for all remaining $k - n$ leaf nodes.

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

e) Calculate the Merkle tree's root, denoted as *root*, using the collision-resistant hash-function Hash by computing the value for each parent node in the tree as Hash (*left-subordinate-node-value*||*right-subordinate-node-value*), where *left-subordinate-node-value* contains the value assigned to the left subordinate node and *right-subordinate-node-value* contains the value of the right subordinate node.

f) Use the digital signature scheme's signature process on inputs:

   1) message: $(root, tag_{msg}, n)$;

   2) set of domain parameters: *Z;*

   3) signature key mapped from the attestation key: *ak*.

   Receive as output the signature Σ.

The process outputs:

— redactable attestation $att = (\Sigma, n, tag_{msg}, (tag_1, \ldots, tag_n))$

6.2.3 Redaction process

This process takes the following inputs:

— set of domain parameters Z;

— message *m* composed of *n* fields denoted as $m_1, \ldots, m_n$;

— redaction key rk;

— admissible changes adm;

— modification instructions mod that are in accordance with the admissible changes adm, i.e. $mod \subseteq adm$.

The process is as follows:

a) Verify that $att = (\Sigma, n, tag_{msg}, (tag_1, \ldots, tag_n))$ is a valid attestation on *m* under the verification key $vk = rk$ and abort if this is not the case.

b) Set $m' = m$ and $att' = att$.

c) Adjust the admissible changes to no longer contain fields to be redacted, i.e. *adm'* = *adm* \ *mod*.

d) For all $i \in mod$:

   1) Compute the hash-code $h_i = \mathrm{Hash}(tag_{msg} \| m_i \| tag_i)$ using the collision-resistant hash-function.

   2) Replace the content of m_i with h_i, i.e. set $m' = (m_1, \ldots, m_{i-1}, h_i, m_{i+1}, \ldots, m_n)$.

   3) Set $tag_i = 0^\lambda$ to indicate that this field has been redacted, i.e. modify the attestation to $att' = (\Sigma, n, tag_{msg}, (tag_1, \ldots, tag_{i-1}, 0^\lambda, tag_{i+1}, \ldots, tag_n))$.

The process outputs:

— redacted message *m';*

— redacted attestation *att';*

— redacted admissible changes *adm'* = *adm* \ *mod*.

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

6.2.4 Verification process

The process takes the following inputs:

— set of domain parameters Z;

— verification key vk;

— redacted or original message $m=(m_1,\ldots,m_n)$;

— redacted or redactable attestation $att=(\Sigma,n,tag_{msg},(tag_1,\ldots,tag_n))$;

— original or redacted admissible changes adm.

The process is as follows:

a) Generate a Merkle tree[17] as follows: generate a balanced binary tree of sufficient height such that it has k leaf nodes with $2n>k\geq n$.

b) For each $i=1,\ldots,n$, compute the hash-code $h_i=\mathrm{Hash}(tag_{msg}\|m_i\|tag_i)$ if $tag_i\neq 0^{\lambda}$ using a collision-resistant hash-function; and if $tag_i=0^{\lambda}$ then set h_i to the value supplied as m_i as it has been redacted previously, which corresponds to the value of $\mathrm{Hash}(tag_{msg}\|m_i\|tag_i)$.

c) Initialize the Merkle tree using $h_1,\ldots,h_n$ as values for the n left-most leaf nodes of the balanced binary tree, use the empty string for all remaining $k-n$ leaf nodes.

d) Calculate the value for the root (denoted as $root$) of the Merkle tree.

e) Use the digital signature's verification process on inputs:

   — set of domain parameters Z;

   — verification key mapped from the verification vk

   — message: $(root,tag_{msg},n)$;

   — signature: Σ.

   Receive from the digital signature's verification process an output $o\in\{accept,reject\}$.

The process outputs:

— The final verification outcome o.

NOTE 1 The redactable attestation scheme specified in 6.2.1 to 6.2.4 satisfies the following properties: unforgeability, privacy, detectability of redactions and mergeable. See Annex B for further details.

NOTE 2 Security proofs for unforgeability and privacy are found in the Reference [12]. Detectability of redactions and mergeability are shown in Reference [15]. See Annex B for further details.

NOTE 3 This scheme was originally introduced by R. Steinfeld, L. Bull, and Y. Zheng in 2001.[12]

7 Scheme SBZ02-MERSAProd

7.1 Parameters

The SBZ02-MERSAProd scheme makes use of the following parameters:

— collision-resistant hash-function Hash as defined in ISO/IEC 10118-1;

— security parameter λ.

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

The security parameter λ shall be met by all involved algorithms for the hash-function and taken into account during the choice of the domain parameters and keys.

7.2 Construction

7.2.1 Key generation process

The key generation algorithm of the redactable attestation scheme consists of the following two procedures:

a) generate the set of domain parameters Z;

b) generate the attestation key ak and verification key vk and redaction key rk as follows:

1) The verification key vk consists of a unique RSA (Rivest, Shamir, and Adleman) modulus $N = p \cdot q$ generated according to ISO/IEC 9796-2 and a list $e_1, \ldots, e_l$ of public exponents which are pairwise co-prime and co-prime to $(p-1)\cdot(q-1)$. Here, l shall be equal to or larger than the number of fields in the message being attested.

2) The attestation key ak consists of the modulus N and the secret exponents $d_1, \ldots, d_l$ such that for all $i = 1, \ldots, l$ it holds that $d_i \equiv e_i^{-1}$ modulo $(p-1)\cdot(q-1)$.

3) The verification key also serves as redaction key, such that $rk = vk$.

The security parameter λ shall be taken into account when choosing above parameters, such that the indicated security strength is achieved.

NOTE In case the number of exponents is larger than the number of fields in the message, i.e. in the case of $n < l$, some of the l exponents are not facilitated.

7.2.2 Redactable attestation process

This process takes the following inputs:

— set of domain parameters Z;

— attestation key ak consisting of the l secret exponents $d_1, \ldots, d_l$;

— a message m consisting of n fields $m_1, \ldots, m_n$ in no particular order, such that each field includes a unique index number i with $1 \le i \le l$;

— admissible changes adm represented as two sets containing the unique index number corresponding to the fields' index numbers: the first set adm_{fix} containing the indices of fields that are not admissible to redaction and the second set adm_{red} containing indices of fields admissible to redactions.

NOTE 1 Reference [12] does not define in detail how to efficiently encode the admissible changes.

NOTE 2 This algorithm does not directly protect the order of the fields, as the index number i for each field is only required to map to a specific exponent in the verification key; the order of those exponents is not protected inside the algorithms presented in this document.

The process Split() to split the message m into fields is generally out of scope of this document. However, the process Split() shall be defined in such a way that fields can be transformed into the message and vice versa, in a reproducible way everytime this conversion is necessary, i.e. if Split(m)= $m_1, \ldots, m_s$ then Split^{-1}($m_1, \ldots, m_s$) = m. The scheme protects the contents of each field, but not their order.

The process requires the generation of randomness. Refer to ISO/IEC 18031 to generate this securely.

The process is as follows:

a) Generate a random bit string tag_{CES} of length λ.

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

b) For each $i=1,\ldots,n$, compute a hash-code h_i from the concatenation of the content of each field m_i together with the *adm* and the tag_{CES} using the collision-resistant hash-function: $h_i = \mathrm{Hash}(adm \| tag_{CES} \| n \| i \| m_i)$.

NOTE 3 How to represent numbers as strings for concatenation is outside the scope of this document.

c) Use the function trans to transform the output of the collision-resistant hash-function to numbers that can be used as input to the forthcoming RSA operation in step d) and e).

NOTE 4 Reference [12] does not define in detail how to encode the output of the hash-function for use with RSA.

NOTE 5 A compatible transform function is the optimal asymmetric encryption padding (OAEP) as standardized in PKCS#1 v2.2 and RFC 8017[29] and in ISO/IEC 18033-2.

d) For each $i=1,\ldots,n$, sign the $\mathrm{trans}(h_i)$ with a different exponent to obtain a signature per field: $s_i = (\mathrm{trans}(h_i))^{d_i}$ modulo N.

e) Compute Σ as the product of all fields' signatures s_i: $\Sigma = \prod_{i=1}^{n} s_i$ modulo N

The process outputs:

— redactable attestation $att = (\Sigma, n, tag_{CES})$.

7.2.3 Redaction process

This process takes the following inputs:

— set of domain parameters Z;

— message m composed of n fields denoted as $m_1,\ldots,m_n$ such that each field includes a unique index number i that corresponds to the indices to the exponents in the verification key;

— attestation *att* of the form $att = (\Sigma, n, tag_{\mathrm{CES}})$;

— redaction key rk containing the list $e_1,\ldots,e_l$ of public exponents;

— admissible changes adm;

— modification instructions mod, which are represented as a set of the unique indices i that corresponds to the indices of the fields that will be redacted, that are according to the admissible changes adm, i.e. $mod \subseteq adm_{red}$.

The process is as follows:

a) Verify that $att = (\Sigma, n, tag_{CES})$ is a valid attestation on m under the verification key $vk = rk$ and abort if this is not the case.

b) Check that modification instructions mod are a subset or equal to adm.

c) Set m' to those fields that shall remain, i.e. remove all m_i if $i \in mod$.

d) Let X be the set of the indices of all fields in m'.

e) Compute for all $k \in X$, using the Chinese-Remainder-Theorem (CRT), the coefficients c_k such that:

1) $c_k \equiv 1$ modulo e_k and

2) $c_k \equiv 0$ modulo e_i for those $i \in \{1,\ldots,k-1,k+1,\ldots,n\}$.

f) For each $k \in X$ compute the hash-code $h_k = \mathrm{Hash}(adm \| tag_{CES} \| n \| k \| m_k)$.

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

g) Use the same function trans as during the attestation process [7.2.2, step c)] to transform the output of the collision-resistant hash-function to numbers that can be used as input to the forthcoming RSA operation [in step h)]; compute h_i=trans(h_i) for $i=1,2,\ldots,n$.

h) For each $k \in X$ compute $s_k = \Sigma^{c_k} / \prod_{i=1,2,\ldots,n} h_i^{\lfloor c_k / e_i \rfloor}$ modulo N.

i) Using the s_k computed in h) for the remaining fields, compute the product Σ' of all signatures $\Sigma' = \prod_{k \in X} s_k$ modulo N.

The process outputs:

— redacted attestation $att' = (\Sigma', n, tag_{CES})$;

— redacted message m';

— admissible changes adm.

NOTE 1 The redaction process does not adjust the initially attested number of fields of the message, denoted as n.

NOTE 2 To calculate the coefficients in step e), any algorithm that solves the equations related to the CRT can be used.

NOTE 3 In step c), the indices corresponding to the remaining m_i do not change and in particular they will still use the same corresponding exponent e_i.

NOTE 4 In step h), the division is a division modulo N, and not over the integers, while the division in the exponent is an integer division rounded down.

NOTE 5 The redaction process does not support consecutive redactions (i.e. redacting again an already redacted attestation) due to the current described handling of indices.

7.2.4 Verification process

The process takes the following inputs:

— set of domain parameters Z;

— verification key vk containing the list $e_1, \ldots, e_l$ of public exponents;

— redacted or original message $m = (m_1, \ldots, m_n)$ such that each field includes a unique index number i that corresponds to the indices of the exponents in the verification key vk;

— redacted or redactable attestation $att = (\Sigma, n, tag_{CES})$;

— admissible changes adm consisting of adm_{fix} and adm_{red}.

The process is as follows:

a) Let X be the set of the indices of all fields of the message m.

b) Check if the indices from X are corresponding to the adm by:

   1) checking if adm_{fix} is a subset of or equal to the set X, and

   2) checking if the set X is a subset of or equal to the union of adm_{fix} and adm_{red}.

   If any check fails the verification outcome is invalid, i.e. set $o = reject$ and abort.

c) Recompute the hash-code using the concatenation of the content of each field m_i together with the adm and the tag_{CES} using the collision-resistant hash-function: $h_i = \text{Hash}(adm \| tag_{CES} \| n \| i \| m_i)$.

d) Use the same function trans as during the attestation process to transform the output of the collision-resistant hash-function to numbers capable as input to the RSA operation [in step f)]; compute h_i = trans(h_i).

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

e) Compute $e = \prod_{i \in X} e_i$.

f) Compute $r = \prod_{i \in X} h_i^{\frac{e}{e_i}}$ modulo N .

g) If $r = \Sigma^e$ then set $o = accept$ otherwise set $o = reject$.

The process outputs:

— the final verification outcome o .

NOTE 1 The redactable attestation scheme specified in 7.2.1 to 7.2.4 satisfies the following properties: unforgeability, privacy, detectability of redactions, and disclosure control. See Annex B for further details.

NOTE 2 Security proofs for unforgeability and privacy are found in the original work.[12] Detectability of redactions and mergeability are shown in.[15] See Annex B for disclosure control and for further details.

NOTE 3 This scheme was originally introduced by R. Steinfeld, L. Bull, and Y. Zheng in 2001;[12] in the original work it is named MERSAProd or MERP.

8 Scheme BBDFFKMOPPS10

8.1 Parameters

The BBDFFKMOPPS10 scheme makes use of the following parameters:

— digital signature scheme as defined in ISO/IEC 14888-1;

— security parameter λ .

The security parameter λ shall be met by all involved algorithms for the signature scheme.

8.2 Construction

8.2.1 Key generation process

The key generation algorithm of the redactable attestation scheme consists of the following two procedures:

a) generate the set of domain parameters *Z*;

b) generate the attestation key ak and verification key vk and redaction key rk as follows:

1) use the key generation process of the digital signature scheme, such that the digital signature scheme's signature key maps to the attestation key ak ,

2) the digital signature scheme's verification key maps to the verification key vk ,

3) the digital signature scheme's verification key also maps to the redaction key rk , such that $rk = vk$.

The security parameter λ is taken into account such that the digital signature scheme offers at least the indicated security strength.

8.2.2 Redactable attestation process

This process takes the following inputs:

— set of domain parameters Z ;

— attestation key ak consisting of the secret key of the digital signature algorithm;

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024

— message m encoded as a tree T with b arcs $A=\{a_1,\ldots,a_b\}$ between t nodes $V=\{v_1,\ldots,v_t\}$. The tree's arcs are directed; pointing from a node denoted as parent to a node denoted as a subordinate node. All arcs' subordinate nodes (if more than one) are ordered, and in Clause 8 the ordered sequence of q subordinate nodes of node p is denoted as $VL_p=\{v_{p,1},\ldots,v_{p,q}\}$. The ordering relation is denoted as a is-left-of relation $R_p \subseteq VL_p \times VL_p$, which yields that $(v_{p,i},v_{p,j}) \in R_p$ if and only if $i<j$. One single node shall be distinguishable from the others as it has no parent node and is denoted *root*. The ordered nodes, together with the arcs between them, form the message to become attested.

— admissible changes $adm=\{1,\ldots,t\}$.

The scheme protects the content of each field and their ordering within the tree structure, but not directly a non-tree-structured linear message. To use such a scheme for non-tree-structured message, a transform from the linear message onto a tree shall be provided. The encoding of the tree and the process Split() to split the message m into arcs and nodes are both out of scope of this document, however the process Split() shall be defined in such a way that nodes, arcs and their ordering can be transformed into the message and vice versa in a reproducible way every time this conversion is necessary, i.e. if Split(m)= $\{a_1,\ldots,a_b\},\{v_1,\ldots,v_t\},\{R_1,\ldots,R_t\}$ then $\text{Split}^{-1}(\{a_1,\ldots,a_b\},\{v_1,\ldots,v_t\},\{R,\ldots,R_t\})=m$.

The process requires the generation of random data. Refer to ISO/IEC 18031 to generate this securely.

The process does not allow admissible changes to be specified. Therefore, by default, a leaf node of the tree, i.e. a vertex that has no subordinate nodes, can always be redacted in this scheme.

In order to make this clear for the user, the input for *adm* shall contain all *t* nodes as indices, i.e. $adm=\{1,\ldots,t\}$; however, redacting a non-leaf node requires previously redacting the nodes of the subtree below it.

The process is as follows:

a) For each $v \in V$, generate a random bit string tag_v of length λ.

b) Initialize the attestation *att* as empty.

c) Traverse the tree T starting from the node that is the root of the tree using a post-order traversal algorithm. During this post-order traversal, perform the following action for each arc $a=(v,w) \in A$:

   1) Set m_a to the value of $0 \| v \| tag_v \| w \| tag_w$.

   2) Use the digital signature scheme's signature process on inputs:

      i) m_a;

      ii) set of domain parameters *Z*;

      iii) signature key mapped from the attestation key *ak*.

   Receive as output the signature Σ_a.

   3) Concatenate the resulting signature to the front of current value of the attestation *att* to form the latest value of *att*, i.e. $att=\Sigma_a \| att$.

NOTE 1 Step c) is performed with the intention to attest all arcs and the content of the nodes and to store the generated digital signatures in a serialized form by traversing the tree in a fixed manner, i.e. post-order traversal.

NOTE 2 The following is an example for a recursive algorithm that performs a post-order traversal of the complete tree *T* starting at the root by calling the following algorithm with the root node; the algorithm gets as additional input an action to perform on each node or each arc:

d) If the node given as input has subordinate nodes, then the left-most subordinate node that has not been marked as visited yet is traversed, and the post-order function is called recursively with this node as an input.

e) If the node given as input has no more subordinate nodes or all subordinate nodes are marked as visited, then the node is marked as visited and the results of performing the action given as input on the contents of

© ISO/IEC 2024 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 23264-2:2024