TECHNICAL REPORT

ISO/IEC TR 29149

First edition
2012-03-15

Information technology — Security techniques — Best practices for the provision and use of time-stamping services

Technologies de l'information — Techniques de sécurité — Meilleures pratiques pour la fourniture et l'utilisation de services d'horodotage

Reference number
ISO/IEC TR 29149:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web 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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

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.

ISO/IEC TR 29149 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

Introduction

This Technical Report explains how to provide and use time-stamping services so that time-stamp tokens are effective when used to provide

- timeliness and data integrity services, or
- non-repudiation services (in conjunction with other mechanisms).

ISO/IEC 18014 specifies time-stamping services, explaining how to generate, renew, and verify time-stamp tokens. The goal of a non-repudiation service is to treat evidence concerning a claimed event or action in order to resolve disputes about the occurrence or non-occurrence of the event or action. Depending on the non-repudiation service which is required, the non-repudiation policy in effect for a specific application, and the legal environment within which the application operates, time-stamp tokens from time-stamping authorities may be required as components of non-repudiation information.

Information technology — Security techniques — Best practices for the provision and use of time-stamping services

1 Scope

This Technical Report explains how to provide and use time-stamping services so that time-stamp tokens are effective when used to provide timeliness, data integrity, and non-repudiation services in conjunction with other mechanisms. It defines:

- how time-stamp requesters should use time-stamp token generation services;
- how TSAs (time-stamping authorities) should provide a service of guaranteed quality;
- how TSAs should deserve trust based on good practices;
- which algorithms and parameters should be used in TST (time-stamp token) generation and TST renewal, so that TSTs resist during the time period during which the TSTs can be verified as being valid;
- how time-stamp verifiers should use the time-stamp token verification services, both when validating individual TSTs, and when validating sequences of renewal TSTs.

2 Terms and definitions

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

2.1
certification authority
CA
authority trusted by one or more users to create and assign public-key certificates

NOTE Optionally, the certification authority may create the users' keys.

[ISO/IEC 9594-8:2005]

2.2
digital signature
data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, e.g. by the recipient

[ISO 7498-2:1989]

2.3
evidence
information which is used, either by itself or in conjunction with other information, to establish proof about an event or action

NOTE Evidence does not necessarily prove the truth or existence of something, but can contribute to the establishment of such a proof.

[ISO/IEC 13888-1:2009]

2.4
evidence user
entity that uses non-repudiation evidence

[ISO/IEC 13888-1:2009]

2.5
hash-function
function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:

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

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

NOTE Computational feasibility depends on the specific security requirements and environment.

[ISO/IEC 10118-1:2000]

2.6
hash-value
string of bits which is the output of a hash-function

[ISO/IEC 10118-1:2000, modified — The term "hash-code" is used to represent this concept in ISO/IEC 10118-1:2000.]

2.7
imprint
string of bits, either the hash-value of a data string or the data string itself

[ISO/IEC 13888-1:2009]

2.8
message authentication code
MAC
string of bits which is the output of a MAC algorithm

NOTE A MAC is sometimes called a cryptographic check value (see for example ISO 7498-2).

[ISO/IEC 9797-1:2011]

2.9
non-repudiation
ability to prove an action or event has taken place, so that this event or action cannot be repudiated later

[ISO 7498-2:1989]

2.10
non-repudiation token
special type of security token as defined in ISO/IEC 10181-1, consisting of evidence, and, optionally, of additional data

[ISO/IEC 13888-1:2009]

2.11
object identifier
OID
globally unique value associated with an object to unambiguously identify it

[ISO/IEC 8824-1:2002 | ITU X.680:2002]

2.12
private key
that key of an entity's asymmetric key pair which should only be used by that entity

[ISO/IEC 9798-1:1997]

2.13
public key
that key of an entity's asymmetric key pair which can be made public

NOTE In the case of an asymmetric signature scheme, the public key defines the verification transformation. In the case of an asymmetric encipherment system, the public key defines the encipherment transformation. A key that is 'publicly known' is not necessarily globally available. The key may only be available to all members of a pre-specified group.

[ISO/IEC 11770-3:2008]

2.14
public key certificate
public key information of an entity signed by the certification authority and thereby rendered unforgeable

[ISO/IEC 11770-3:2008]

2.15
signer
entity generating a digital signature

[ISO/IEC 13888-1:2009]

2.16
time stamp
data item which denotes a point in time with respect to a common time reference

[ISO/IEC 11770-1:2010]

2.17
time-stamp token renewal
process of issuing a new time stamp token to extend the validity period of an earlier time-stamp token

[ISO/IEC 18014-1:2008, adapted]

2.18
time-stamp requester
entity which possesses data it wants to be time-stamped

NOTE A requester can also be a trusted third party including a time-stamping authority.

[ISO/IEC 18014-1:2008]

2.19
time-stamp token
TST
data structure containing a verifiable binding between a data items' representation and a time-value

NOTE A time-stamp token can also include additional data items in the binding.

[ISO/IEC 18014-1:2008]

2.20
time-stamp verifier
entity which possesses data and wants to verify that it has a valid time-stamp bound to it

NOTE The verification process may be performed by the verifier itself or by a trusted third party.

[ISO/IEC 18014-1:2008]

2.21
time-stamping authority
TSA
trusted third party trusted to provide a time-stamping service

[ISO/IEC 18014-1:2008]

2.22
time-stamping policy
named set of rules that indicates the applicability of a time-stamp token to a particular community or class of application with common security requirements

[ISO/IEC 18014-1:2008]

2.23
time-stamping service
TSS
service providing evidence that a data item existed before a certain point in time

[ISO/IEC 18014-1:2008]

2.24
trusted third party
TTP
security authority, or its agent, trusted by other entities with respect to security related activities

[ISO/IEC 10181-1:1996]

3 Symbols and abbreviated terms

In the remainder of this document the following notation will be used:

HMAC	Hash Message Authentication Code
H(D)	The hash-value of data D, using hash-function H
MAC	Message Authentication Code
OID	Object Identifier
PKI	Public Key Infrastructure
$S_X(y)$	The signature computed on data y using a signature algorithm and the private key of entity X
TSA	Time-Stamping Authority
TSP	Time-Stamp Packet: the combination of the TST and the data upon which the TST is generated

TSS	Time-Stamping Service
TST	Time-Stamp Token
TST(D, *t*)	time-stamp token on data D, at point in time *t*

4 Time-stamping services

Time-stamping services include generation, renewal, and verification of time-stamp tokens, as defined in ISO/IEC 18014-1.

Time-stamp tokens are associations between data and points in time, and are created in a way that aims to provide evidence that the data existed before the associated date and time. This evidence may be used by non-repudiation services.

Time-stamping services involve the following entities (from ISO/IEC 18014-1):

- the time-stamp requester, that has some data (e.g. a document) to time-stamp;
- the Time-Stamping Authority (TSA), that generates time-stamp tokens (TST);
- the time-stamp verifier, that verifies time-stamps bound to data.

Time-stamping services (TSS) provide three specific services:

- time-stamp token generation, where the requester submits data items, and receives a time-stamp token; this service is provided by the TSA;
- time-stamp token renewal, a special case of time-stamp token generation, where the requester submits an existing first time-stamp token and related data items, and receives a new time-stamp token, such that the validity period of the first time-stamp token is extended by the new time-stamp token; this service is provided by the TSA;
- time-stamp token verification, when the verifier validates the time-stamp token; this service may also involve the TSA or other trusted third parties.

Users of the time-stamping services handle time-stamp packets (TSP), encompassing the data plus the time-stamp token (TST).

5 Use cases for non-repudiation

5.1 Introduction

Time-stamping services provide tokens that may be used, in combination with an adequate non-repudiation policy, to support non-repudiation claims.

Non-repudiation services provide a user B with protection against another user A later denying that an action or event has taken place. While these services do not prevent A from trying to repudiate B’s claim, they provide evidence to support the resolution of such disagreement. In general, the evidence needs to be convincing to a third party arbitrator C.

The following clauses present some use cases.

5.2 Use case #1

Non-repudiation services

Data D existed before time t.

Integrity of data D is guaranteed after time t.

Evidence generation

1. User A has some data D.
2. A gets a time-stamp token on D at t: TST(D, t).

Evidence verification

1. User B receives a time-stamp packet TSP(D, TST(D, t)).
2. B checks that the TST corresponds to the data D, and verifies TST(D, t).

5.3 Use case #2

Non-repudiation services

Data D existed before time t.

User A signed D before time t.

Integrity of data D is guaranteed after time t.

Evidence generation

1. User A has some data D.
2. A signs D: S_A(D).
3. A gets a time-stamp token on S_A(D) at t: TST(S_A(D), t).

Evidence verification

1. User B receives the data D and a time-stamp packet TSP(S_A(D), TST(S_A(D), t)).
2. B checks that the TST corresponds to S_A(D), and verifies the TST.
3. B verifies the signature, using verification data at t.

See "7.5 signature verification" below.

5.4 Use case #3

Non-repudiation services

Data D existed before time t2.

User A signed D after time t1 and before time t2.

Integrity of data D is guaranteed after time t2.

Evidence generation

1. User A has some data D.
2. A requests a time-stamp token on anything (null included) at t1: TST(any, t1).
3. A prepares a message M, containing <D, TSP(any, TST(any, t1))>.
4. A signs the message M: S_A(M).
5. A requests a time-stamp on the signature S_A(M) at t2: TST(S_A(M), t2).

NIST SP 800-102 [36] introduces a special kind of time-stamp token that does not refer to any user's data[1]. Here, using "any" as data for the TST is equivalent to those 'time marks'.

Evidence verification

1. User B receives the message M, and the time-stamp packet TSP(S_A(M), TST(S_A(M), t2)).
2. B checks that the second TST at t2 corresponds to the signature S_A(M), and verifies the second TST at t2.
3. B verifies A's signature on message M at t2.
4. B verifies the first TST on any at t1.

See "7.5 signature verification" below.

6 Potential issues

6.1 Security requirements for custody of evidences

A time-stamp token is an evidence to be used in the future if a dispute arises. As a general rule, the user of the evidence should look after the evidence in coordination with the TSA.

If the evidence user needs to prove that he has access to the data at the current time, the evidence user requests a time-stamp token on the data.

The interest of the evidence user is that the token is available for verification. The evidence user is expected to take the needed measures to guarantee the availability of the time-stamp token, and the verification means, either by herself, or using some third party to save copies. The copies have to guarantee the integrity and the availability of the time-stamp packet, and of the verification means[2].

For some time-stamping mechanisms, the TSA is required for verification of the time-stamp token. The evidence user may require guarantees that those means are available when needed.

[1] These tokens have no message imprint. These tokens just bind the TSA to a point in time. The TSA guarantees that the token is not available before the stated time t. Therefore, nobody may have such a token before t, and any operation involving this token is guaranteed to be carried out after t.

[2] There may be confidentiality requirements on the data D, but that is out of the scope of this technical report. All the mechanisms for time-stamping that are described in ISO/IEC 18014 avoid confidentiality requirements on the time-stamp token because only the hash-value of D is part of the token. When data D is constrained to a small number of options (for instance in polls), the requester of the time-stamp may add random data to D in order to hide the actual information in H(D + random).

If the time-stamp token may be used long after it is issued, and there is a real risk that the protecting cryptography might become weak or broken, the evidence user may renew the TST. That implies that new time-stamp tokens are requested with different hash-functions and/or to different TSA. These new tokens are aggregated to the information whose integrity and timeliness is to be preserved.

6.2 Weak cryptography: hash-functions

6.2.1 Hash-function properties

Hash-functions need to satisfy the following properties:

1) pre-image resistance—for essentially all pre-specified outputs, it is computationally infeasible to find any input which hashes to that output, i.e., to find any pre-image x_0 such that $h(x_0) = y$ when given any y for which a corresponding input is not known.

2) 2nd pre-image resistance—it is computationally infeasible to find any second input which has the same output as any specified input, i.e., given x, to find a 2nd pre-image $x_0 \neq x$ such that $h(x) = h(x_0)$. Also known as "weak collision resistance".

3) collision resistance—it is computationally infeasible to find any two distinct inputs x, x_0 which hash to the same output, i.e., such that $h(x) = h(x_0)$. (Note that here there is free choice of both inputs.) Also known as "strong collision resistance".

6.2.2 Attacks on time-stamped data

Time-stamp tokens that only record the hash-value of the data D are subject to attacks if the hash-function fails to meet any of the conditions listed above.

Theoretically, the weakest property is strong collision resistance. The source of the information may prepare two documents, whose hash-values collide, and elect to use one or the other in the future.

No pre-image resistance (noPR):

Some data D2 may replace original data D at any moment after producing H(D), having access to H(D).

No weak collision resistance (noWCR):

Some data D2 may replace original data D at any moment after producing H(D), having access to D and H(D).

No strong collision resistance (noSCR):

Some data D2 may replace original data D. The attacker needs to prepare D and D2 before producing H(D). Later on, the attacker may argue that the time-stamp corresponds to D2.

These attacks are more difficult if the data, on which the hash-value is calculated, is structured, since the replacement needs to meet the structure. For instance, when elaborating a false signature, the fake data have to look like a valid signature. For this countermeasure to be effective, the structure needs to disallow the injection of arbitrary data without notice.

Table 1 — Consequences for TST that only record H(D)

Attacker	No pre-image resistance	No weak collision resistance	No strong collision resistance
originator	may replace D2 for D, at any moment		may replace D2 for D, before the generation of the TST
evidence user	may replace D2 for D, after reception of the TST	may replace D2 for D, after reception of D and the TST	not applicable

The following preventive countermeasures protect against these attacks:

- usage of two hash-functions; either (a) requesting two TST using different hash-functions, or (b) by submitting a time-stamp request that includes multiple hash-values over the same document using different hash-functions, as described in ISO/IEC 18014-1, or (c) applying renewal operations with a different hash-function

- require a structure on the data subject to the hash-function; this countermeasure assumes that finding collisions on structured data is harder than finding collisions on raw data where it is easier to insert bits as needed

Renewal operations protect the evidence beyond the period covered by the previous TST, and may be used as a reaction to early announcements of potential weaknesses. Renewal is to be performed before weaknesses are real.

If the hash-function is unexpectedly broken, due to a cryptographic breakthrough, there may be little or no time left to renew previous time-stamp tokens. Notice that it is extremely unlikely that two hash-functions are broken on the same date[3]. Using a second function "buys time for renewal".

6.2.3 Attacks on TSTInfo

Time-stamp tokens that use hash-functions to protect the TSTInfo[4], are subject to the same consequences when weak hash-functions are employed in the process of generating the TST. See Table 2.

Table 2 — Consequences TSTs that use hash-functions to protect the TSTInfo

Attacker	No pre-image resistance	No weak collision resistance	No strong collision resistance
originator or evidence user	may replace Info2 for Info, after reception of the TST where Info and Info2 are instances of TSTInfo (see 18014-1, Annex I)		not applicable

There is a straightforward preventive countermeasure protect against these attacks: usage of more than one hash-function. For linking mechanisms, it is foreseen in ISO/IEC 18014-3 to use more than one hash-function when it generates the TST. For independent tokens as in ISO/IEC 18014-4 using the digital signature mechanism, this requirement may be met by requesting TSTs from providers using different hash-functions on the respective TSTInfo.

[3] Attention should be paid to families of algorithms; that is, to algorithms that are based on the same theoretical concepts, since a whole family may be broken simultaneously.

[4] Hash-functions are used, at least, by the digital signature mechanism in ISO/IEC 18014-2, and in linking mechanisms in ISO/IEC 18014-3.

6.3 Weak cryptography: digital signatures

For time-stamp tokens generated using digital signature techniques, the following additional issues apply.

Digital signatures are produced by means of a signature operation using a private key and a signature algorithm. If a hash-function is involved in the signature algorithm, the weaknesses of hash-functions covered in Clause 7.2 apply. Additionally, the following attacks are possible:

— the disclosure of the private key permits the generation of fake signatures, incorrectly binding the signer to false data.

  o short private keys open an opportunity to discover them by brute force

  o poor implementations may be subject to time attacks, power attacks, or even fault attacks that make it easier to discover the key, and will eventually reveal its value

  o weak public key models may allow the discovery of the private part out of the knowledge of the public part

As soon as any of these components becomes weak, the signatures become weak, and the time-stamp tokens cannot be trusted any longer.

Renewal deals with signatures that become weaker as a consequence of time. But renewal is to be performed before trust is lost[5].

If the loss of trust is abrupt, it is too late to renew, and the only countermeasure is to retain more than one TST using different algorithms. Using a second TST "buys time for renewal".

6.4 Weak cryptography: message authentication codes

MACs are produced by means of an operation using a secret key and a MAC algorithm. If a hash-function is involved in the MAC algorithm, the weaknesses of hash-functions covered in Clause 7.2 apply. Additionally,

— the secret key needs to remain secret, and

— the secret key needs to be long enough to resist brute-force attacks.

As soon as any of these components becomes weak, the time-stamp tokens cannot be trusted any longer.

Renewal deals with MAC protections than become weaker as a consequence of time. But renewal is to be performed before trust is lost.

If the loss of trust is abrupt, it is too late to renew, and the only countermeasure is to retain more than one TST using different algorithms. Using a second TST "buys time for renewal".

6.5 Signature verification

6.5.1 General

Signature verification requires the public key of the signer, and one or more certificates.

The problems arise

— when the private key is compromised,

— when certificates are not available,

[5] Formats as the one described in rfc 5698 may be useful to automate the discovery of broken algorithms both for hash-functions and for signing functions.

- when certificates expire, and
- when certificates are revoked or suspended.

Evidence users may face the situation where a signature is sound now, but becomes invalid when one or more of the previous conditions occur. Certificates are valid within their usability period, and while they are neither revoked, nor suspended. Digital signatures are definitely invalid after the signing key is compromised; but the acceptance of invalid certificates may be subject to different treatment in the non-repudiation policy, taking into account the reason for the revocation or suspension. The following clauses cover the potential scenarios.

6.5.2 Verification of signatures protected by a time-stamp token

Time-stamp tokens may be used to freeze the point in time where a user's signature is verified, and isolate the value of the evidence from the evolution of private keys and certificates.

When the time-stamp token is requested at time t, the requester should take care that all the information needed to verify a signature at time t is retained:

- all the certificates, the one of the signer, and those of the certification authorities
- the revocation information of all the certificates; either certificate revocation lists (CRL), or online responses (e.g. OCSP).

See Clause 8.4.

6.5.3 Verification of time-stamp tokens protected by a digital signature

Time-stamp tokens may use the digital signature mechanism to protect the TSTInfo, and are therefore subject to the previous concerns, and need to be protected accordingly.

Single TSTs generated at time t1, may be verified at any later time t2. The verification of the protecting signature is performed with the information valid at t2.

6.6 Time-stamp token renewal

Sequences of time-stamp tokens are created during renewal operations. A time-stamp token that is renewed while it is verifiable, extends its validity from the original point in time where the first time-stamp was generated to the end of the validity period of the renewed time-stamp. The renewal operation freezes a point in time where the time-stamp token is verified, and extends its validity beyond the validity of the original certificates and algorithms.

Let TST1 be a time-stamp token generated at time t1. Let TST2 be a time-stamp token generated at time t2, renewing TST1. The evidence is verified at time t3, being t1 < t2 < t3. Then:

- TST2 is validated at t3; if this validation fails, the evidence is void.
- If TST2 is valid at t3, TST1 is validated at t2; if this validation fails, the evidence is void.
- If TST2 is valid at t3, and TST1 is valid at t2, the evidence is valid at t1.

As a consequence of the previous statements, renewal tokens should include all the information needed to validate the previous token at the point in time when the renewal token is generated. See Clause 8.4 for the case of tokens protected by the digital signature mechanism.

6.7 Time-stamping service availability

TSAs are expected to provide a quality of service to be stated in their practice statements. This is important for token generation, token renewal and token verification. Unavailability implies delays until the system becomes available again, and may disrupt business activity.

Service availability should be addressed by conducting a Business Impact Analysis whose result is a Business Continuity Plan.

Unavailability is countered by redundancy of means:

Access redundancy

There should be redundant means of access, including communication lines, and highly available authentication service.

Redundancy of equipment for time-stamp token generation and verification

There should be redundant means to generate and validate time-stamp tokens. This implies:

- ❑ cryptographic devices
- ❑ archival means, either for generation of tokens, and for accountability
- ❑ installations to host the equipment

Whenever there is redundancy, every component is expected to fulfil the same security requirements, and use equivalent cryptographic mechanisms. When pieces of equipment serve a single time-stamp service provider, each piece needs its own private cryptographic material, and methods are needed to validate each of them. For instance, when TSTs are digitally signed, each signing device has its own certificate, and users of the service should be able to validate any of them.

6.8 Time-stamping service continuity

The aim of a TST is to be verified in the future, from a few minutes after its generation, to years after. Time-stamp service providers should take measures to guarantee the availability of verification information for a period of time that should be stated in the practice statements.

Requesters should take into consideration the announcements for termination of service and renew the time-stamps before the time-stamp service provider terminates. The information submitted as data for time-stamp token renewal should enclose the original TST, and any other information needed to validate the original TST at the time of renewal.

Unplanned termination of activity by a time-stamp service provider should be countered by back-up copies of token verification material that should be available to verifiers until the planned termination of service.

7 Recommendations

7.1 Recommendations for requesters of time-stamp tokens

The entity requesting time-stamps should carry out some activities when electing a time-stamp service provider, and some activities when requesting time-stamps.

When electing a time-stamp service provider, the requester should:

- assess the Time-Stamping Policy: time-stamps should match the intended use foreseen by the requester, in particular, assess that the policy fulfil the requirements for non-repudiation services
- assess the mechanisms, and approve those that are acceptable
  - current cryptographic strength of the hash-functions
  - current cryptographic strength of the signing method (either digital signature or message authentication codes), if applicable
  - foreseen cryptographic strength of the algorithms used in the TST is adequate for the expected life of the TST (the period it may need to be used)
  - secret keys management, if applicable: creation, storage, usage, custody of copies (if needed at all), and destruction
- collect and assess further information such as [third party] reviews of the time-stamp service provider.

When requesting a time-stamp, the requester should verify the TST on reception:

- verify the correctness of the TST;
- verify that the TSA is the expected one;
- verify that the Time-Stamping Policy is the expected one (it is enough to assess the OID);
- verify that the mechanism is one of the approved ones; and
- verify the current validity of certificates in the certification chain of the signing key (if using the digital signature mechanism).

7.2 Recommendations for verifiers of time-stamp tokens

The entity verifying time-stamps should follow the procedures described in the standard for the selected time-stamping mechanism, and be ready to provide evidence of the steps carried on in order to

- assess current validity of signing certificate, and the chain of certificates, if applicable
- assess the TSA identity
- assess service satisfaction according to published time-stamping policy
- [third party] audits
- assess the mechanism
  - current cryptographic strength of the hash-function(s)
  - current cryptographic strength of the signing mechanism (either digital signature or message authentication codes), if applicable
  - secret keys management, if applicable: creation, storage, usage, custody of copies (if needed at all), and destruction.

In the case of renewal, the strength of the cryptography used in the previous time-stamp token should be checked with respect to the date in which the next time-stamp token is generated.

7.3 Recommendations for time-stamp service providers

7.3.1 Overview

The TSA should build trust on its services by taking appropriate technical and organizational measures, and being subject to independent audits.

The TSA should produce, and make available to its users, requesters and verifiers, a Time-Stamping Policy describing the general rules that should be followed by the time-stamping service, and a Time-Stamping Practice Statement describing how processes and procedures implement the rules laid down in the policy.

The TSA should have a risk assessment carried out in order to evaluate business assets and threats to those assets in order to determine the necessary security controls and operational procedures to meet the commitments in the Time-Stamping Policy. This risk assessment should be available to auditors.

In preparation of an external audit carried by an independent third-party, the TSA should carry out an internal audit to verify that it meets the processes and procedures described in the Time-Stamping Practice Statement.

7.3.2 Time-Stamping Policy

The Time-Stamping Policy is a document, produced by the TSA, and available to the users of time-stamping services, describing the general rules that should be followed by the service provider.

- should provide the unique identification of the time-stamping authority
- should state the accuracy of time in the generated time-stamp tokens
- should describe the functional conditions of service
  - accepted hash-functions (it may reject other requests)
  - implemented mechanisms (it should reject others)
  - if using digital signatures: algorithms, length of keys, and certificates
  - if using message authentication codes: algorithms, and length of keys
- should describe the non functional conditions of service
  - availability of the services (service level)
  - support for dispute resolution
- should describe the procedures to change cryptographic keys
- should describe the procedures to terminate service

When different services are provided, each one should be uniquely identified, and described.

7.3.3 Time-Stamping Practice Statement

The Time-Stamping Practice Statement is a document, produced by the TSA, and available to the users of time-stamping services, describing the processes and procedures implement by the service provider to meet the Time-Stamping Policy.

The following paragraphs present items that may be covered in the Time-Stamping Policy:

- should describe the legal measures taken to guarantee service
  - the legal entity of the provider
  - the legislation to which it responds
- should describe the established agreements with providers
  - time services
  - certification services
  - communication services

- should describe the organizational measures taken to guarantee service
  - personnel screening
  - personnel contracts and disciplinary provisions
  - personnel roles and segregation of tasks
  - procedures to generate, preserve, recover, and destroy keys
  - third party audits
  - certificates of excellence: processes, security, etc.
- should describe the technical measures taken to guarantee service
  - sources of time, either primary and back-up sources
  - usage of cryptographic devices for cryptographic operations
  - storage of secret keys
  - storage of activity logs for long-time inspection
  - protection of systems: access control, hardware and software maintenance, ...
  - time acquisition: maximum accepted deviation
  - business continuity provisions
  - service level agreements
- should describe the commercial measures taken to guarantee service
  - insurances
  - payment of penalties for failures
  - additional services: legal, technical, etc.

The subject is also considered in [5] and [29].

7.3.4 Protection of audit logs

A TSA may audit its operations and thus log information in audits trails. This is to support verification of issued tokens and auditing of its operations. Activity logs need to be secured from different points of view:

- [I] integrity of the information
- [T] timeliness of the information
- [Auth] authenticity of the information
- [A] availability of the information
- [C] confidentiality of the information

See [33] for additional guidance.

There are several options to provide these guarantees, as shown in Table 3 — Options to protect logs.

Table 3 — Options to protect logs

Mechanism	[I]	[T]	[Auth]	[A]	[C]	See …
digital signatures	√		√			See clause 7.3 ISO/IEC 9796-2:2010 Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms ISO/IEC 9796-3:2006 Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms ISO/IEC 14888 (all parts) Digital signatures with appendix
time-stamping	√	√				ISO/IEC 18014 (all parts) Time-stamping services
access control + privilege management					√	ISO/IEC 10181-3:1996 Access control framework ISO/IEC 10181-4:1997 Non-repudiation framework
encryption					√	ISO/IEC 18033 (all parts) Encryption algorithms
linked items	√	√				ISO/IEC 18014-3:2009 Time-stamping services -- Part 3: Mechanisms producing linked tokens

For the short-term audit logs should be readily available. For the long-term, audit logs need to be archived.

7.4 Recommendations for signature verification

When the procedure for non-repudiation requires the validation of a time-stamped digital signature, those signatures are to be validated at the time t when the time-stamp token was generated (see Clause 7.5). That validation implies access to certificates and revocation information at time t. That access may be simplified by using enriched signatures that include the validating information as in CAdES-XL or XAdES-XL formats.

CAdES-XL

> The acronym stands for CMS Advanced Electronic Signatures. It is defined in [2] and [32]. XL is one of the defined sets of information, containing all the information needed for signature verification.

XAdES-XL

> The acronym stands for XML Advanced Electronic Signatures. It is defined in [4]. XL is one of the defined sets of information, containing all the information needed for signature verification.

7.5 Non-repudiation policy

Adapted from ISO/IEC 10181-4:1997 [19].

A non-repudiation policy may include the following:

- Rules for the generation of evidence e.g. specifications of the classes of activity for which non-repudiation evidence should be generated; specifications of the TSSs to be used to generate evidence; the roles in which those TSSs may act; the procedures that entities are to follow when generating evidence.
- Rules for the verification of evidence e.g. specifications of the TSSs whose evidence is acceptable; for each TSS, the forms of evidence that will be accepted from that TSS.
- Rules for the storage of evidence e.g. the means to be used to ensure the integrity of stored evidence.
- Rules for the use of evidence e.g. specification of the purposes for which evidence may be used.

  NOTE When using time-stamp tokens as evidences, it may be difficult to prevent unauthorized use of evidence.

- Rules for adjudication e.g. specification of the agreed adjudicator(s) that may settle a dispute.

A different authority may define each of these sets of rules. For example, the owner of a system could define the rules for generation of evidence, while the law of the country in which the system exists could define the rules for adjudication.

If different parts of the policy are inconsistent, then the non-repudiation service may fail to operate correctly e.g. by allowing an event which did in fact occur to be successfully denied during the dispute resolution phase.

The adjudicator when resolving a dispute may use the non-repudiation policy itself. For example, the adjudicator might refer to the non-repudiation policy to determine whether the rules for generation of evidence have been complied with.

Security policies can be explicitly stated, or implicitly defined by implementations. An explicit statement of the non-repudiation policy (e.g. a natural language document) can help detect conflicts between different parts of the policy and can also aid the adjudicator.

Non-repudiation policies also deal with incidents related to the cryptographic material used to generate evidences; namely, key compromise and key revocation.

Non-repudiation policies for interactions between security domains may result from agreements between independent security domains or may be imposed by a super-domain.

8 Algorithms

8.1 Overview

The following algorithms are state of the art when this technical report is published. They should be revised in the future to factor in advances in cryptography.

8.2 Hash functions

Functions which map strings of bits to fixed-length strings of bits, satisfying the following two properties:

- it is computationally infeasible to find for a given output, an input that maps to this output
- it is computationally infeasible to find for a given input, a second input that maps to the same output.

See further detail on hash-function properties in Clause 7.2.

Table 4 — Algorithms that satisfy the properties of hash functions

Hash-function	Length in bits	Reference	Expected lifetime
SHA-1	160	ISO/IEC 10118-3 [16] FIPS 180-2 [7]	3 years – unknown 6 years – unusable
RIPEMD-160	160	ISO/IEC 10118-3 [16]	6 years – unusable
SHA-224	224	FIPS 180-2 [7]	10 years – unknown
SHA-256	256	ISO/IEC 10118-3 [16] FIPS 180-2 [7]	10 years – unknown
GOST R 34.11-94	256	GOST R 34.11-94 [10]	unknown
SHA-384	384	ISO/IEC 10118-3 [16] FIPS 180-2 [7]	> 10 years
SHA-512	512	ISO/IEC 10118-3 [16] FIPS 180-2 [7]	> 10 years ...
WHIRLPOOL	512	ISO/IEC 10118-3 [16]	> 10 years ...

8.3 Keyed message authentication algorithms

Algorithms for computing a function which maps strings of bits and a secret key to fixed-length strings of bits, satisfying the following two properties:

- for any key and any input string the function can be computed efficiently;
- for any fixed key, and given no prior knowledge of the key, it is computationally infeasible to compute the function value on any new input string, even given knowledge of the set of input strings and corresponding function values, where the value of the ith input string may have been chosen after observing the value of the first I-1 function values.

8.4 Signature algorithms

Recommended minimum key sizes for encryption operation. When used, hash-functions should provide at least twice length in bits.

Table 5 — Signature algorithms

Signing algorithm	1 year	3 years	6 years	10 years
rsa	1.024	1.536	2.048	2.048
dsa	1.024	not recommended		
ecdsa	163	224	224	224

Bibliography

[1] [ANSI X9.95] Trusted time stamp management and security, ANSI 2005

[2] [ETSI 101 733] CMS Advanced Electronic Signatures – CAdES, ETSI TS 101 733 V1.7.4 (2008-07)

[3] [ETSI 101 861] Time stamping profile, ETSI TS 101 861 V1.2.1 (2002-03)

[4] [ETSI 101 903] XML Advanced Electronic Signatures (XAdES), ETSI TS 101 903 V1.4.1 (2009-06)

[5] [ETSI 102 023] Electronic Signatures and Infrastructures (ESI); Policy requirements for time-stamping authorities, ETSI TS 102 023 V1.2.1 (2003-01)

[6] [ETSI 102 176-1] Algorithms and Parameters for Secure Electronic Signatures - Part 1: Hash functions and asymmetric algorithms, ETSI TS 102 176-1 V2.0.0 (2007-11)

[7] [FIPS 180-2] Secure Hash Standard (SHS), 2002

[8] [FIPS 186-3] Digital Signature Standard (DSS), 2009

[9] [FIPS 198] The Keyed-Hash Message Authentication Code (HMAC), 2002

[10] [GOST R 34.11-94] *Information Technology — Cryptographic Information Security — Hash Function*

[11] ISO 7498-2:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture*

[12] ISO/IEC 9796-2:2010, *Information technology — Security techniques — Digital signature schemes giving message recovery — Part 2: Integer factorization based mechanisms*

[13] ISO/IEC 9796-3:2006, *Information technology — Security techniques — Digital signature schemes giving message recovery — Part 3: Discrete logarithm based mechanisms*

[14] ISO/IEC 9797-1:2011, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher*

[15] ISO/IEC 9797-2:2011, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function*

[16] ISO/IEC 10118-3:2004, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

[17] ISO/IEC 10118-3:2004/Amd.1:2006, *Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions — Amendment 1: Dedicated Hash-Function 8 (SHA-224)*

[18] ISO/IEC 10181-3:1996, *Information technology — Open Systems Interconnection — Security frameworks for open systems: Access control framework*

[19] ISO/IEC 10181-4:1997, *Information technology — Open Systems Interconnection — Security frameworks for open systems: Non-repudiation framework*

[20] ISO/IEC 13888 (all parts), *Information technology — Security techniques — Non-repudiation*

[21] ISO/IEC 14888-1:1998, *Information technology — Security techniques — Digital signatures with appendix — Part 1: General*

[22] ISO/IEC 14888-3:1998, *Information technology — Security techniques — Digital signatures with appendix — Part 3: Certificate-based mechanisms*

[23] ISO/IEC 18014-1:2008, *Information technology — Security techniques — Time-stamping services — Part 1: Framework*

[24] ISO/IEC 18014-2:2009, *Information technology — Security techniques — Time-stamping services — Part 2: Mechanisms producing independent tokens*

[25] ISO/IEC 18014-3:2009, *Information technology — Security techniques — Time-stamping services — Part 3: Mechanisms producing linked tokens*

[26] [RFC 2104] HMAC: Keyed-Hashing for Message Authentication, 1997

[27] [RFC 3161] Internet X.509 Public Key Infrastructure - Time-Stamp Protocol (TSP), August 2001

[28] [RFC 3275] XML-Signature Syntax and Processing, March 2002

[29] [RFC 3628] Policy Requirements for Time-Stamping Authorities (TSAs), November 2003

[30] [RFC 4490] Using the GOST 28147-89, GOST R 34.11-94, GOST R 34.10-94, and GOST R 34.10-2001 Algorithms with Cryptographic Message Syntax (CMS), May 2006

[31] [RFC 4491] Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile, May 2006

[32] [RFC 5126] CMS Advanced Electronic Signatures (CAdES), February 2008

[33] [RFC 5280] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008

[34] [RFC 5652] Cryptographic Message Syntax (CMS), September 2009

[35] [SP 800-92] Guide to Computer Security Log Management, NIST – National Institute of Standards and Technology, Special Publication 800-62, September 2006

[36] [SP 800-102] Recommendation for Digital Signature Timeliness, NIST – National Institute of Standards and Technology, Special Publication 800-102, draft Nov 12, 2008

[37] [XML DSIG] XML-Signature Syntax and Processing, W3C Recommendation,
http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/,
http://www.w3.org/TR/xmldsig-core/, 12 February 2002

[38] [XML DSS] Digital Signature Service Core Protocols, Elements, and Bindings, Versión 1.0.
oasis-dss-core-spec-v1.0-os, 11 April, 2007
http://docs.oasis-open.org/dss/v1.0/

[39] [XML TSP] XML Timestamping Profile of the OASIS Digital Signature Services, Version 1.0.
oasis-dss-profiles-timestamping-spec-v1.0-os, 11 April, 2007
http://docs.oasis-open.org/dss/v1.0/

[40] ISO/IEC 9594-8:2005, *Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks*

[41] ISO/IEC 10118-1:2000, *Information technology — Security techniques — Hash-functions — Part 1: General*

[42] ISO/IEC 8824-1:2002 | ITU-T Rec. X.680(2002), *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

[43] ISO/IEC 9798-1:1997, *Information technology — Security techniques — Entity authentication — Part 1: General*

[44] ISO/IEC 11770-1:2010, *Information technology — Security techniques — Key management — Part 1: Framework*

[45] ISO/IEC 11770-3:2008, *Information technology — Security techniques — Key management — Part 3: Mechanisms using asymmetric techniques*

ICS 35.040

Price based on 21 pages