

General principles of digital signature verification

7th November 2002

Contents

1	Abstract	1
2	Terms and definitions	2
3	Introduction	2
4	Establishing the time of digital signature creation by means of time-stamping	2
5	Determining the time of signature creation without using the time stamp	6
6	Determining the signer	6
7	Requirements to the basic formats	7

1 Abstract

This document describes general principles of digital signature verification, independent of the means used to represent components of digital signatures.

2 Terms and definitions

- **Validity attestation** – validity confirmation or revocation list

3 Introduction

Two aspects need to be verified when verifying a digital signature:

- when was the signature given; and
- who gave it.

In general, the exact moment of digital signature creation can not be established, as the process of signing can not be carried out under the surveillance of any third parties. Nevertheless, it is possible to establish a suitable time interval, see Sections 4 and 5 for discussion.

When establishing the signer, two problems may arise: which certificate was used and whether this certificate was valid at the time of signing. Solutions to these problems are described in Section 6.

4 Establishing the time of digital signature creation by means of time-stamping

Description of the method

Time-stamping service described in the specification [TSS] supports determining a moment in time when given data already existed. Thus, taking a time stamp one may fix the upper bound of the time of signature creation. In order to fix also a lower bound, it is enough to sign such data together with the document that has a known lower bound on creation time. As the signature must be created later than the signed data, this approach allows to estimate the time of signature creation below as well. Such a data enabling to give lower estimates to the time of signing is called *freshness token* and it may contain any information not known before creation of the signed data.

Freshness tokens that are secure and easily managed can be obtained from time stamps issued by a time-stamping server using linkage-based time-stamping. These time stamps have the following properties important from the viewpoint of freshness tokens:

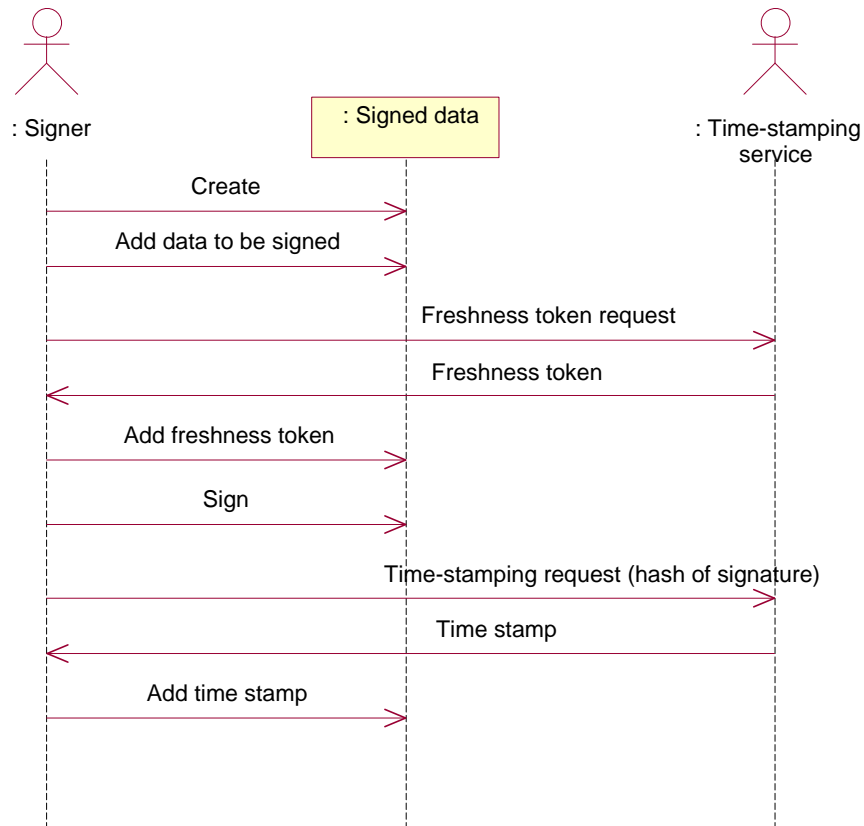


Figure 1: Messages passed during signature creation and time-stamping

- The creation time of the time stamp can be easily extracted.
- Authenticity of the time stamp can be easily established without human interference.
- Time stamps can be compared. When using direct comparison, security of the freshness tokens does not depend on unpredictability of the time stamps.

Time interval of the digital signature creation can be determined using the following method (see Figure 1):

- The signer obtains the latest time stamp of the time-stamping service provider. For that one may time stamp some random data, or the service provider

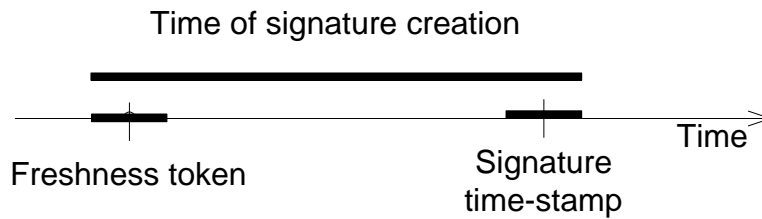


Figure 2: Determining the time of signature creation

may offer a special service of downloading the latest time stamp. This time stamp will later be used as the freshness token.

- The signer combines the content of the freshness attestation (the field `timestampInfo` of the time stamp used as the freshness token) with the data to be signed and creates a signature over the resulting data structure. The method of combining must be specified separately for each digital signature format.
- The signer takes a time stamp to the signature.
- The signer extends the freshness token to the obtained time stamp.
- The signer saves both time stamps together with signed data and the signature. The method of saving must be specified separately for each digital signature format.

Figure 2 displays the method of determining the time of signature creation. The required time moment lies between the two time stamps. Upper bound of the time of signature creation can be found as sum of the time value of the signature time stamp and possible error. Lower bound of the time of signature creation can be found as difference of the time value of the freshness token and possible error.

Algorithm for determining the time of signature creation

This subsection describes an algorithm determining the time interval of signature creation. If any of the following steps fails, determining the interval was unsuccessful and the algorithm must interrupt its work.

Inputs

The algorithm uses the following inputs:

- signed data,
- signature,
- freshness attestation,
- signature time stamp,
- data used to verify the signature (see the time-stamping protocol specification [TSS]).

Outputs

Output of the algorithm is the time interval of signature creation given by two time moments `sigmin` and `sigmax` (earliest and latest possible moments of signature creation, respectively).

The algorithm

- Check that the signature is computed over the data structure formed by combining the signed data and the `timestampInfo` field of the time stamp used as the freshness token.
- Compare the freshness token and the signature time stamp ([TSS], Section 9.2). Comparison must succeed, the value of the output `vresult` must be “earlier” and the value of the output `vtype` must be “direct”.
- Extract the time value and maximal error of the signature time stamp ([TSS], Section 9.1). The output value `sigmax` will be equal to the sum of these values.
- Extract the time value and maximal error of the freshness token ([TSS], Section 9.1), leaving out the step of time stamp verification ([TSS], Section 9.1.4). The output value `sigmin` will be equal to the difference of these values.

5 Determining the time of signature creation without using the time stamp

In case it is necessary to determine the time of signature creation using some other technical means, their usage must be regulated by a separate specification. This specification must answer the following questions:

- How the selected technical apparatus ensures determining the time of signature creation.
- How the selected technical apparatus ensures long-term preservation of signature's proof value.
- What protocols and data structures are used.

Additionally, the respective juridical framework has to be created that enables to use this solution for determining the time of signature creation.

6 Determining the signer

Determining the certificate used to create the signature

Signer of a document may have several certificates, connected to the same pair of private and public keys. As different certificates may be used in different contexts and they carry different rights and obligations, it is necessary to add the information about the certificate used to the signature.

The signer must identify the certificate used to create the signature by combining its message digest with the data structure to be signed (this data structure contains the data to be signed and other data protected by the signature, e.g. the freshness attestation).

The method of combining must be specified separately for each digital signature format.

Establishing validity of the certificate used to create the signature

Validity of the certificate can be established by the validity attestation signed by its issuer. The way of establishing the validity of the certificate during the required

time interval depends on the format of the validity attestation and the security requirements of the application.

For example, when using certificate revocation lists, one may require that the time of signature creation must belong to the period of validity of the revocation lists and that the time interval of signature creation must be less than some predefined time (e.g. one hour).

Security requirements must be specified separately for each application. By the Digital Signature Act, digital signatures are used in private communication based on the mutual agreement. This agreement must clarify which digital signatures are considered as valid, or more precisely, after which procedures and under which conditions the verifier accepts this validity. The conditions of digital signature usage in public sector must be stated in corresponding legislative acts.

In order to verify a validity attestation, its signature is verified using the issuer's public key. In order to distinguish between the attestations given during the validity period of the issuer's public key and the false attestations the attestation must be time-stamped during the validity period of the issuer's public key.

7 Requirements to the basic formats

In order to support easy processing and transferring of the signed data, the elements forming this data (as the signed data structures, freshness attestations, time stamps, validity attestations etc.) in one structured data set.

For the format of this data set it is necessary to establish:

- The way how the signed data structure is formed.
- The way how the certificates and revocation lists of format specified in the standard [X.509] are handled.
- The way how the validity attestations of type `id-pkix-ocsp-basic` specified in the standard draft [RFC 2560] are handled.
- The way how the time stamps and freshness attestations of the specification [TSS] are handled (if time stamps are used to establish the time of signature creation).

References

- [TSS] Protocols and data formats for time-stamping service
- [X.509] ITU-T recommendation X.509. Information technology – Open systems interconnection – The Directory: Public-key attribute certificate frameworks. 03/2000.
- [RFC 2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C., X.509 Internet Public Key Infrastructure: Online Certificate Status Protocol – OCSP. June 1999.