DRAFT

NIST Special Publication 800-102

Recommendation for Digital Signature Timeliness

Elaine Barker

Computer Security Division
Information Technology Laboratory

C O M P U T E R S E C U R I T Y

November 2008

U.S. Department of Commerce
Carlos M. Gutierrez, Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Director

NIST National Institute of Standards and Technology • Technology Administration • U.S. Department of Commerce

Abstract

Establishing the time when a digital signature was generated is often a critical consideration. A signed message that includes the (purported) signing time provides no assurance that the private key was used to sign the message at that time unless the accuracy of the time can be trusted. With the appropriate use of timestamps from a Trusted Timestamp Authority (TTA) and/or verifier-supplied data, the signatory can provide some level of assurance about the time that the message was signed.

KEY WORDS: digital signatures, timeliness, timestamp, Trusted Timestamp Authority

Acknowledgements

The National Institute of Standards and Technology (NIST) gratefully acknowledges and appreciates contributions by Rich Davis from the National Security Agency concerning the many security issues associated with this Recommendation. NIST also thanks the many contributions by the public and private sectors whose thoughtful and constructive comments improved the quality and usefulness of this publication.

Table of Contents

Recommendation for Digital Signature Timeliness

1 Introduction

A digital signature is an electronic analogue of a written signature; the digital signature can be used to provide assurance that the claimed signatory signed the information. In addition, a digital signature may be used to detect whether or not the information was modified after it was signed (i.e., to detect the integrity of the signed data).

Establishing the time when a digital signature was generated is often a critical consideration. A signed message that includes the (purported) signing time provides no assurance that the private key was used to sign the message at that time unless the accuracy of the time can be trusted. With the appropriate use of timestamps from a Trusted Timestamp Authority (TTA) and/or verifier-supplied data, the signatory can provide some level of assurance about the time that the message was signed.

2 Authority

This document has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This recommendation is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), Securing Agency Information Systems, as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This Recommendation has been prepared for use by Federal agencies. It may be used by non-governmental organizations on a voluntary basis and is not subject to copyright (attribution would be appreciated by NIST.)

Nothing in this Recommendation should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should this Recommendation be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official.

Conformance testing for implementations of this Recommendation will be conducted within the framework of the Cryptographic Module Validation Program (CMVP), a joint effort of NIST and the Communications Security Establishment of the Government of Canada. The requirements of this Recommendation are indicated by the word "**shall**."

3 Definitions, Acronyms and Symbols

3.1 Definitions

Approved	FIPS-approved and/or NIST-recommended. An algorithm or technique that is either 1) specified in a FIPS or NIST Recommendation, or 2) adopted in a FIPS or NIST Recommendation or 3) specified in a list of NIST-approved security functions.
Digital signature	The result of a cryptographic transformation of data that, when properly implemented, provides origin authentication, assurance of data integrity and signatory non-repudiation.
Entity	An individual (person), organization, device or process. Used interchangeably with "party".
Hash value	The result of applying a hash function to data.
Key	A parameter used in conjunction with a cryptographic algorithm that determines its operation. Examples applicable to this Recommendation include: 1. The computation of a digital signature from data, and 2. The verification of a digital signature.
Message	The data that is signed.
Nonce	A time-varying value that has at most a negligible chance of repeating, for example, a random value that is generated anew for each use, a timestamp, a sequence number, or some combination of these.
Party	An individual (person), organization, device or process. Used interchangeably with "entity".
Private key	A cryptographic key that is used with an asymmetric (public key) cryptographic algorithm. The private key is uniquely associated with the owner and is not made public. It is used to compute a digital signature that may be verified using the corresponding public key.

Public key	A cryptographic key that is used with an asymmetric (public key) cryptographic algorithm and is associated with a private key. The public key is associated with an owner and may be made public. In the case of digital signatures, the public key is used to verify a digital signature that was signed using the corresponding private key.
Relying party	A party that depends on the validity of the digital signature process.
Security strength	A number associated with the amount of work (that is, the number of operations) that is required to break a cryptographic algorithm or system.
Shall	Used to indicate a requirement of this Recommendation.
Signatory	The entity that generates a digital signature on data using a private key.
Signature generation	The process of using a digital signature algorithm and a private key to generate a digital signature on data.
Signature verification	The process of using a digital signature algorithm and a public key to verify a digital signature on data.
Timestamp	A token or packet of information that is used to provide assurance of timeliness; contains timestamped data, including a time, and a signature generated by a Trusted Timestamp Authority (TTA).
timestamp	Contains the time and, possibly, other information.
timestamped_data	The data on which a digital signature is generated by a TTA.
timestamp_packet	A unit of information that is transmitted by a TTA that contains *timestamped_data* and a *timestamp_signature*.
timestamp_signature	A digital signature that is generated using a TTA's private signature key.
timestamp_time	The time provided in a timestamp.

Trusted timestamp	A timestamp that has been signed by a Trusted Timestamp Authority.
Trusted Timestamp Authority	An entity that is trusted to provide accurate time information.
TTA_supplied_info	Additional information that is included in the *timestamped_data* by a TTA during the generation of a *timestamp_signature*.
user_supplied_info	Additional information that is provided by an entity when requesting a timestamp from a TTA.
Verifier	The entity that verifies the authenticity of a digital signature using the public key.

3.2 Acronyms

FIPS	Federal Information Processing Standard.
FISMA	Federal Information Security Management Act.
NIST	National Institute of Standards and Technology.
OMB	Office of Management and Budget.
TSP	Timestamp Packet.
TTA	Trusted Timestamp Authority.

3.3 Symbols

D	The signed and timestamped message.
H(*M*)	A hash value that is generated on *M*.
M	Message
$SIG_E(\ldots)$	A digital signature generated by the entity E using an **approved** hash function and an **approved** digital signature algorithm. The data that is signed is the information contained within the parentheses.

4 Using Timestamps from a Trusted Timestamp Authority

One method of obtaining assurance of the time of digital signature generation is by the use of a trusted timestamp authority (TTA) that is trusted by both the signatory and the verifier. The discussions in this section are intended to assist the reader in determining exactly what assurances are obtained using different timestamp schemes.

4.1 Notation

A *trusted timestamp authority* (TTA) is an entity that is trusted to produce timestamp packets. A *timestamp packet* (TSP) is transmitted by a TTA and contains:

- A digital signature (the *timestamp_signature*) that is generated using the TTA's private key, and
- The *timestamped_data* upon which the digital signature is generated.

The *timestamped_*data includes a *timestamp* — an accurate, unambiguous representation of the time of generation of the accompanying timestamp signature.

The following notation will be used in this discussion. Note that the commas are used to indicate a list of data, not to indicate the order or format of that data.

1. TSP = *timestamp_packet = timestamped_data, timestamp_signature*

   where the TSP contains both *timestamped_data* and a *timestamp_signature*, although the exact format of the TSP is not specified herein.

2. *timestamp_signature* = SIG_{TTA} (*timestamped_data*)

   where "SIG_{TTA}" is a digital signature operation on the *timestamped_data* using the TTA's private digital signature key. This private key **shall** only be used for the generation of digital signatures on *timestamped_data*.

3. *timestamped_data = user_supplied_info, TTA_supplied_info, timestamp,*

   where

   a. *user_supplied_info* is information that is provided by an entity when requesting a timestamp from the TTA; the *user_supplied_info* may, in fact, be *Null*. If provided, the information is used by the TTA during *timestamp_signature* generation, but need not be transmitted in the timestamp packet returned to the requestor. However, this information must be available to entities that will verify the *timestamp_signature*.

   b. *TTA_supplied_info* is additional information that is used by the TTA during *timestamp_signature* generation; the *TTA_supplied_info* may, in fact, be *Null*. All or part of this information may be omitted from the timestamp packet transmitted by the TTA if the omitted portion(s) can be recreated and used by the entities that will verify the *timestamp_signature*.

   c. *timestamp* contains the time and possibly other information.

Therefore, the generic TSP produced by a TTA has the form:

TSP = user_supplied_info, TTA_supplied_info, timestamp,
SIG_{TTA}(*user_supplied_info, TTA_supplied_info, timestamp*)

where *user_supplied_info* and *TTA_supplied_info* may be *Null*.

4.2 Timestamp Provision by a TTA

A TTA may either broadcast a timestamp packet (TSP) or provide a TSP in response to a request from a requesting entity. The requesting entity and any other party that needs to verify the TTA's digital signature (i.e., a relying party) must be aware of the security strength provided by the TTA's digital signature. If an *X*-bit security strength is required by the requesting entity's or relying party's application, then the TTA's digital signature **shall** provide at least *X*-bits of security strength in order to fulfill that requirement (see SP 800-57).

When a TTA broadcasts a TSP, the *user_supplied_info* in the TSP is *Null*. The *timestamp_signature* is generated on the *TTA-supplied_info* (which may be *Null*) and the *timestamp*. The TSP is then assembled and broadcast. Portions of *the TTA_supplied_info* that are known by all intended recipients of the TSP may be omitted from the data transmitted in the *timestamped_data* field of the TSP, even though the entire *TTA_supplied_info* is included in the data used during the generation and verification of the *timestamp_signature*.

When an entity requests a timestamp, the requesting entity provides *user_supplied_info* (which may be *Null*) to the TTA in the request. A *timestamp_signature* is generated on the *user_supplied_info*, the *TTA-supplied_info* (which may be *Null*), and a *timestamp*. A TSP is assembled from the *timestamp_signature* and *timestamped_data*, and then sent to the requesting entity. Portions of the *TTA-supplied_info* that are known by all intended recipients of the TSP, and the *user_supplied_info* that is otherwise made known to the verifying entity may be omitted from the data transmitted in the *timestamped_data* field of the returned TSP, even though they must be included in the (complete) version of the *timestamped_data* that is used during the generation and verification of the *timestamp_signature*.

4.3 Signatory Provision of a Timestamp with a Signed Message

There are several useful schemes in which an entity A obtains a timestamp packet (TSP) from a trusted timestamp authority (TTA) and then combines the TSP with a message (*M*) and a signature into a payload of data (*D*) that is sent to recipient entity B.

In the following schemes, signatures are generated by entity A or a TTA using an **approved** digital signature algorithm. Let $SIG_A(\ )$ be a signature generated by entity A, and let $SIG_{TTA}(\ )$ be a signature generated by a TTA. $SIG_A(\ )$ is verified using entity A's public signature verification key, and $SIG_{TTA}(\ )$ is verified using the TTA's public signature verification key. The following discussions assume that entity B successfully verifies all received signatures.

4.3.1 Optional (or No) User Information Provided to the TTA by Entity A

As shown in Figure 1, entity A may request a *timestamp* from a TTA, or entity A may use a *timestamp* that is broadcast from a TTA (i.e., entity A does not explicitly request a timestamp from the TTA).

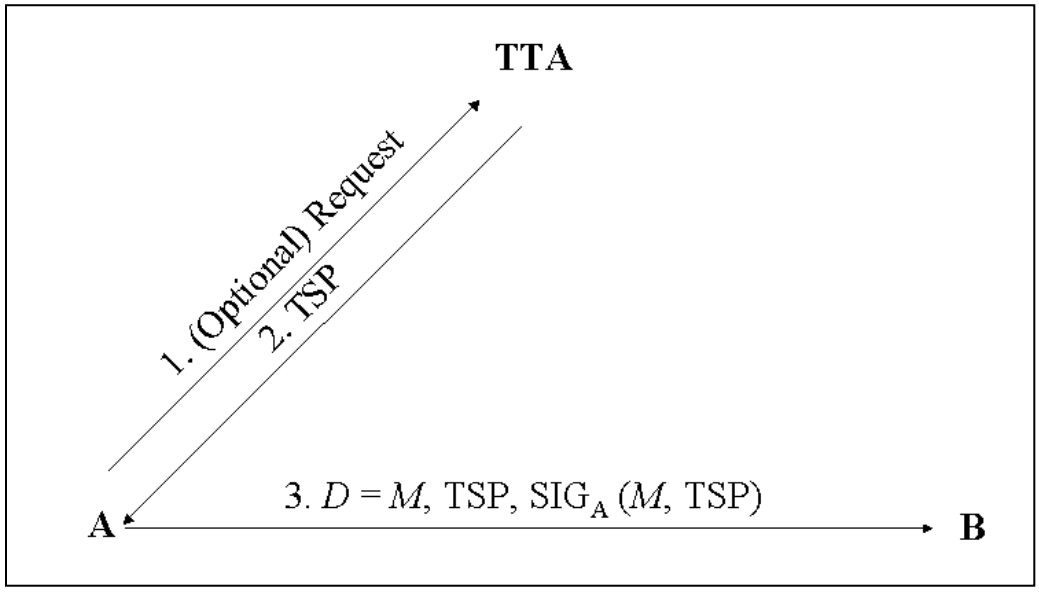

Figure 1: No User Info Provided to the TTA

1. The Request message in Figure 1 is sent only if entity A explicitly requests a timestamp from the TTA. If the request is sent, the Request message contains any desired *user_supplied_info* (see Section 4.1).
2. The TTA sends a TSP to entity A (or broadcasts a TSP that is obtained by A), where:

   TSP = *timestamped_data*, *timestamp_signature*$_{TTA}$.

   If a Request message was sent (in step 1):

   timestamped_data = user_supplied_info, TTA-supplied_info, timestamp.

   timestamp _signature$_{TTA}$ = SIG$_{TTA}$(*user_supplied_info, TTA_supplied_info, timestamp*).

   If a Request message was not sent (in step 1):

   timestamped_data = TTA_supplied_info, timestamp.

   timestamp_signature$_{TTA}$ = SIG$_{TTA}$(*TTA_supplied_info, timestamp*),

   i.e., *user_supplied_info* is *Null* in the case of a broadcast TSP.

   If there is a mutual agreement between entity A and the TTA, the following information may be omitted from the TSP data that is transmitted by the TTA:

   - Any portion of the *user_supplied_info* may be omitted, since it is known

by entity A.

- Any portion of the *TTA_supplied_info* may be omitted when this information is known by entity A.

However, any such information omitted from the transmitted data **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA} = SIG_{TTA}(timestamped\_data)$.

Upon receiving the TSP from the TTA, entity A **should** verify $timestamp\_signature_{TTA}$ using the TTA's public signature verification key.

3. Entity A signs (*M*, TSP), assembles *D* and sends it to entity B:

   $D = M$, TSP, $SIG_A(M, TSP)$,

   where the TSP is as specified in step 2.

   If any portion of the *user_supplied_info* was omitted from the TSP received from the TTA, then the entire *user_supplied_info* **shall** be inserted into the TSP used in the assembly of *D* unless there is a mutual agreement between entity A and entity B, in which case, any portion of the *user_supplied_info* may be omitted from the TSP data transmitted by entity A if it is known by entity B. However, the entire *user_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA}$ and $SIG_A(M, TSP)$.

   If any portion of the *TTA_supplied_info* was omitted from the TSP received from the TTA, then the entire *TTA_supplied_info* **shall** be inserted into the TSP used in the assembly of *D* unless there is a mutual agreement between entity A and entity B, in which case, any portion of the *TTA_supplied_info* may be omitted from the TSP data transmitted by entity A if it is known by entity B. However, the entire *TTA_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA}$ and $SIG_A(M, TSP)$..

4. Upon receiving *D*, entity B does the following:

   - Verifies $timestamp\_signature_{TTA}$ using the TTA's public signature verification key, and
   - Verifies $SIG_A(M, TSP)$ using entity A's public signature verification key.

   Note that it is irrelevant which of these steps is performed first; it is only important that both verifications are successful.

Entity B knows the following:

a. *M* could have been assembled either before or after the TSP was received.

b. $SIG_A(M, TSP)$ was generated at some point after the time indicated by the *timestamp* in the TSP.

c. *D* was assembled after the time indicated by the *timestamp* in the TSP.

If a more precise time is required for the generation of $SIG_A(M, TSP)$, a second trusted timestamp may be acquired (as specified in Section 4.4) that will provide assurance that

(*M*, TSP) had existed and been signed by (at least) the time indicated in the second timestamp.

4.3.2 A Hash of *M* is Provided to the TTA by Entity A

Entity A may provide the hash value of *M* to the TTA when requesting a timestamp as shown in Figure 2. Let *H* denote a hash value that is generated on *M* using an **approved** hash function (i.e., *H* = Hash(*M*)).

1. Entity A sends *H* to the TTA in a timestamp request, i.e., the *user_supplied_info* contains *H* (*user_supplied_info* = *H*, *other_info*). Note that *other_info* may be *Null*.

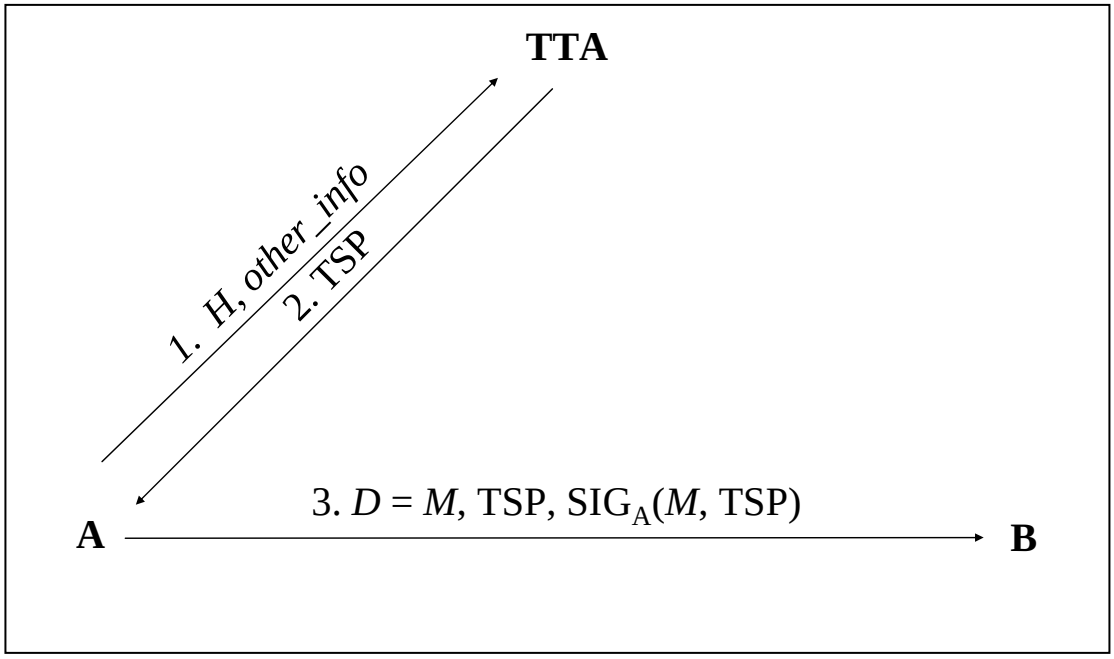

Figure 2: Entity A Provides a Hash Value to the TTA

2. The TTA returns a TSP to entity A:

   TSP = *timestamped_data*, $timestamp\_signature_{TTA}$

   where:

   timestamped_data = user_supplied_info, TTA-supplied_info, timestamp.

   $timestamp\_signature_{TTA}$ = SIG_{TTA}(*user_supplied_info, TTA_supplied_info, timestamp*).

   If there is a mutual agreement between entity A and the TTA, the following information may be omitted from the TSP data that is transmitted by the TTA:

   - Any portion of the *user_supplied_info* may be omitted, since it is known by entity A.

- Any portion of the *TTA_supplied_info* may be omitted if this information is already known by entity A.

However, even though they may be omitted from the transmitted data, the entire *user_supplied_info* and the entire *TTA_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA} = SIG_{TTA}(timestamped\_data)$.

Upon receiving the TSP from the TTA, entity A **should** verify $timestamp\_signature_{TTA}$ using the TTA's public signature verification key.

3. Entity A signs (*M*, TSP), assembles *D* and sends it to entity B:

   $D = M, TSP, SIG_A(M, TSP)$

   where the TSP is as defined in step 2.

   If any portion of the *user_supplied_info* was omitted from the TSP received from the TTA, then the entire *user_supplied_info* **shall** be inserted into the TSP used in the assembly of *D* unless there is a mutual agreement between entity A and entity B, in which case, the following information may be omitted from the TSP data transmitted in *D* by entity A:

   - *H* may be omitted, since it can be (re)calculated by entity B.
   - Any portion of the *other_info* in the *user_supplied_info* may be omitted if this information is already known by entity B.

   However, the entire *user_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA}$ and $SIG_A(M, TSP)$..

   If any portion of the *TTA_supplied_info* was omitted from the TSP received from the TTA, then the entire *TTA_supplied_info* **shall** be inserted into the TSP used in the assembly of *D* unless there is a mutual agreement between entity A and entity B, in which case, any portion of the *TTA_supplied_info* may be omitted from the TSP data transmitted by entity A if the information is already known by entity B. However, the entire *TTA_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation/ verification of $timestamp\_signature_{TTA}$ and $SIG_A(M, TSP)$.

4. Upon receiving *D*, entity B does the following:

   - Computes $H' = Hash(M)$. If *H* was received in *D* (see step 3), then entity B also verifies that $H' = H$.
   - Verifies $timestamp\_signature_{TTA}$ using the TTA's public signature verification key, and
   - Verifies $SIG_A(M, TSP)$ using entity A's public signature verification key.

   Note that it is irrelevant which signature verification is performed first; it is only important that both verifications are successful.

Entity B knows the following:

1. *M* was assembled, and *H* was generated prior to the time indicated by the timestamp in the TSP obtained from the TTA; in particular, *H* was included in the *timestamped_data* that was signed by the TTA.
2. *M* has remained unchanged since the time indicated by the *timestamp* in the TSP.
3. $SIG_A(M, TSP)$ was generated at some point after the time indicated by the *timestamp* in the TSP.
4. *D* was assembled after the time indicated by the *timestamp* in the TSP.

If a more precise time is required for the generation of $SIG_A(M, TSP)$, a second trusted timestamp may be acquired (as specified in Section 4.4) that will provide assurance that (*M*, TSP) had been signed by (at least) the time indicated in the second timestamp.

4.3.3 A Digital Signature on *M* is Provided to the TTA by Entity A

Entity A may provide the digital signature of *M* to the TTA when requesting a timestamp as shown in Figure 3.

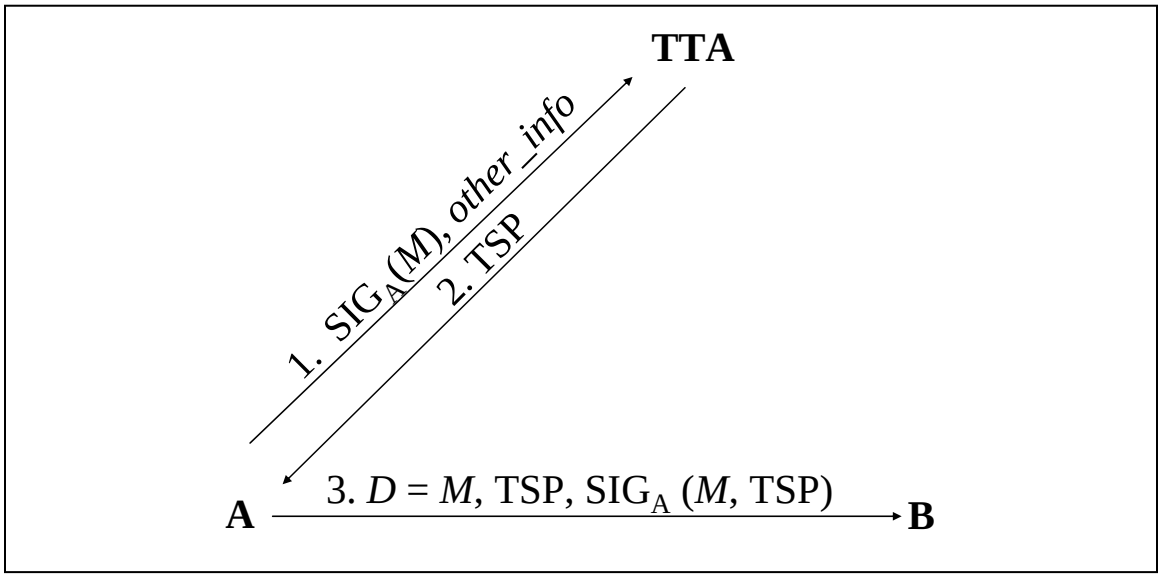

Figure 3: Entity A Provides a Signature to the TTA

1. Entity A sends $SIG_A(M)$ to the TTA in a timestamp request, i.e., *user_supplied_info* contains $SIG_A(M)$ (*user_supplied_info* = $SIG_A(M)$, *other_info*). Note that *other_info* may be *Null*.
2. The TTA returns a TSP to entity A:

   TSP = *timestamped_data*, $timestamp\_signature_{TTA}$

   where:

   timestamped_data = user_supplied_info, TTA_supplied_info, timestamp.

   $timestamp\_signature_{TTA}$ = SIG_{TTA}(*user_supplied_info, TTA_supplied_info,*

timestamp).

If there is a mutual agreement between entity A and the TTA, the following information may be omitted from the TSP transmitted by entity A:

- Any portion of the *user_supplied_info* may be omitted, since it is known by entity A. However, if $SIG_A(M)$ is omitted from the TSP, it **shall** be inserted back into the TSP when sent to entity B (see step 3).
- Any portion of the *TTA_supplied_info* may be omitted when this information is known by entity A.

However, the entire *user_supplied_info* and the entire *TTA_supplied_info* **shall** be included in the generation / verification of $timestamp\_signature_{TTA}$.

Upon receiving the TSP from the TTA, entity A **should** verify $timestamp\_signature_{TTA}$ using the TTA's public signature verification key.

3. Entity A signs (*M*, TSP), assembles *D* and sends it to entity B:

   $D = M, TSP, SIG_A(M, TSP)$

   where TSP is as defined in step 2.

   If any portion of the *user_supplied_info* was omitted from the TSP received from the TTA, then the entire *user_supplied_info* **shall** be inserted into the TSP used in the assembly of *D* unless there is a mutual agreement between entity A and entity B, in which case, any portion of the *user_supplied_info* may be omitted from the TSP transmitted to entity B if it is known by entity B. However, the entire *user_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA}$ and $SIG_A(M, TSP)$..

   If any portion of the *TTA_supplied_info* was omitted from the TSP received from the TTA, then the entire *TTA_supplied_info* **shall** be inserted into the TSP used in the assembly of *D* unless there is a mutual agreement between entity A and entity B, in which case, any portion of the *TTA_supplied_info* may be omitted from the TSP data transmitted by entity A. However, the entire *TTA_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA}$ and $SIG_A(M, TSP)$.

4. Upon receiving *D*, entity B does the following:

   - Verifies $SIG_A(M)$ using entity A's public signature verification key,
   - Verifies $timestamp\_signature_{TTA}$ using the TTA's public signature verification key, and
   - Verifies $SIG_A(M, TSP)$ using entity A's public signature verification key.

   Note that the order of performing these steps is irrelevant; it is only important that all verifications are successful.

Entity B knows the following:

1. *M* and $SIG_A(M)$ were generated before the time indicated by the TSP's

timestamp, and $SIG_A(M)$ was included in the *timestamped_data* that was signed by the TTA.

2. *M* has remained unchanged since the time indicated by the *timestamp* in the TSP.
3. $SIG_A(M,$ TSP) was generated at some time after the time indicated by the *timestamp* in the TSP.
4. *D* was assembled after the time indicated by the *timestamp* in the TSP.

If a more precise time is required for the generation of $SIG_A(M,$ TSP), a second trusted timestamp may be acquired (as specified in Section 4.4) that will provide assurance that $SIG_A(M,$ TSP) was generated by (at least) the time indicated in the second timestamp.

4.3.4 A Digital Signature on *M* is Provided to the TTA by Entity B

Entity B may provide a digital signature to the TTA that was received from entity A as shown in Figure 4.

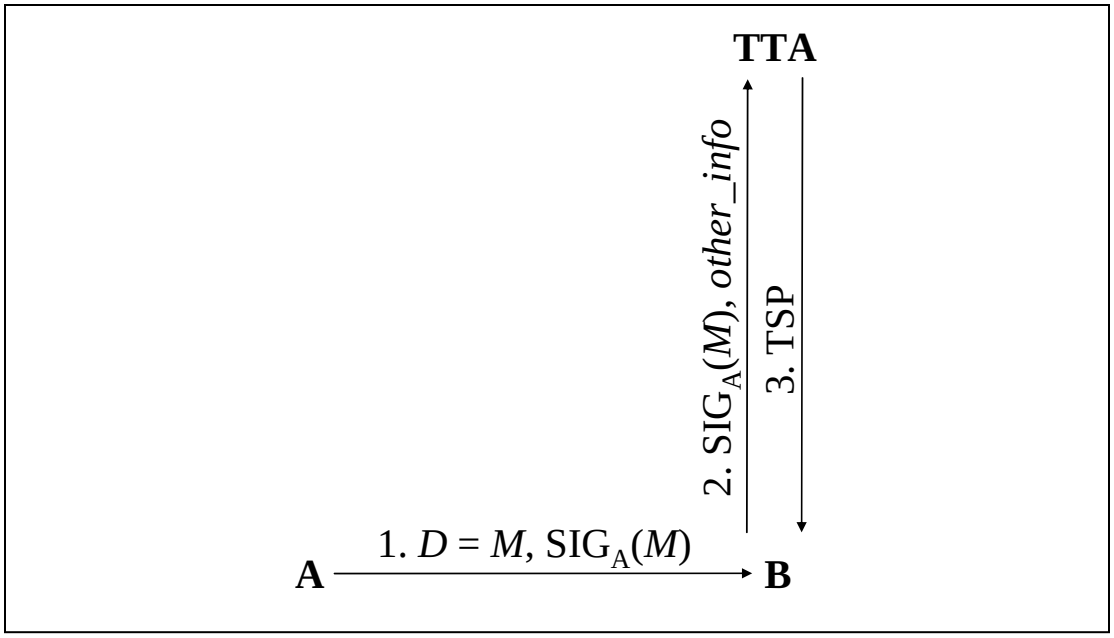

Figure 4: Entity B Provides a Signature to the TTA

1. Entity A signs *M*, assembles *D* and sends it to Entity B:

   $$D = M, SIG_A(M)$$

   Entity B may verify $SIG_A(M)$ upon receipt using entity A's public signature verification key, and may record the (approximate) time that the signature was received from Entity A for his own purposes.

2. Entity B sends $SIG_A(M)$ to the TTA in a timestamp request, i.e., *user_supplied_info* contains $SIG_A(M)$ (*user_supplied_info* = $SIG_A(M)$, *other_info*). Note that the *other_info* may be *Null*.

3. The TTA returns a TSP to entity B:

   TSP = *timestamped_data*, $timestamp\_signature_{TTA}$

   where:

   timestamped_data = user_supplied_info, TTA_supplied_info, timestamp.

   $timestamp\_signature_{TTA} = SIG_{TTA}$(*user_supplied_info, TTA_supplied_info, timestamp*).

   If there is a mutual agreement between entity B and the TTA, the following information may be omitted from the TSP transmitted by entity B:

   - Any portion of the *user_supplied_info* may be omitted, since it is known by entity B.
   - Any portion of the *TTA_supplied_info* may be omitted when this information is known by entity B.

   However, the entire *user_supplied_info* and the entire *TTA_supplied_info* **shall** be included in the generation / verification of $timestamp\_signature_{TTA}$.

4. Upon receiving the TSP, entity B does the following:

   - Verifies $timestamp\_signature_{TTA}$ using the TTA's public signature verification key, and
   - Verifies $SIG_A(M)$ using entity A's public signature verification key if it was not verified prior to sending it to the TTA.

   Note that the order of performing these steps is irrelevant; it is only important that all verifications are successful.

Entity B has evidence that *M* and $SIG_A(M)$ were generated before the time indicated by the TSP's *timestamp*, and $SIG_A(M)$ was included in the *timestamped_data* that was signed by the TTA. This evidence can be presented to any third party who also trusts the TTA.

4.4 Using an Additional Timestamp

A refinement of the signature generation time may be obtained if a second timestamp is requested from a TTA. Any entity could make the request, although requests by entity A and entity B are discussed below. This procedure has the most value if the timestamp request is made as close as possible to the generation of entity A's signature on the first timestamp packet; thus, a minimal time interval is established during which the signature was generated. The discussions assume that all digital signatures are successfully verified.

In the following schemes, the initial steps of a scheme specified in Sections 4.3.1 - 4.3.3 are executed first, with the following changes:

- *user_supplied_info* becomes $user\_supplied\_info_1$,
- the *other_info* within $user\_supplied\_info_1$ becomes $other\_info_1$.

- *TTA_supplied_info* becomes $TTA\_supplied\_info_1$,
- *timestamp* becomes $timestamp_1$,
- TTA becomes TTA_1 (alternatively represented as TTA1 in a subscript), and
- TSP becomes TSP_1.

The schemes indicate the use of two TTAs: TTA_1 and TTA_2. TTA_2 may, in fact, be the same TTA as TTA_1.

4.4.1 Entity A Requests the Second Timestamp

Both timestamps may be obtained by entity A. The TTA(s) providing the timestamps must be trusted by both entity A and entity B, and also by any third party that needs to be convinced of the signature generation time.

Figure 5 shows that the first timestamp is obtained as specified in Section 4.3.1, 4.3.2, or 4.3.3, after which entity A requests a second timestamp. The first two steps are the same as those specified for the schemes in 4.3.1 - 4.3.3.

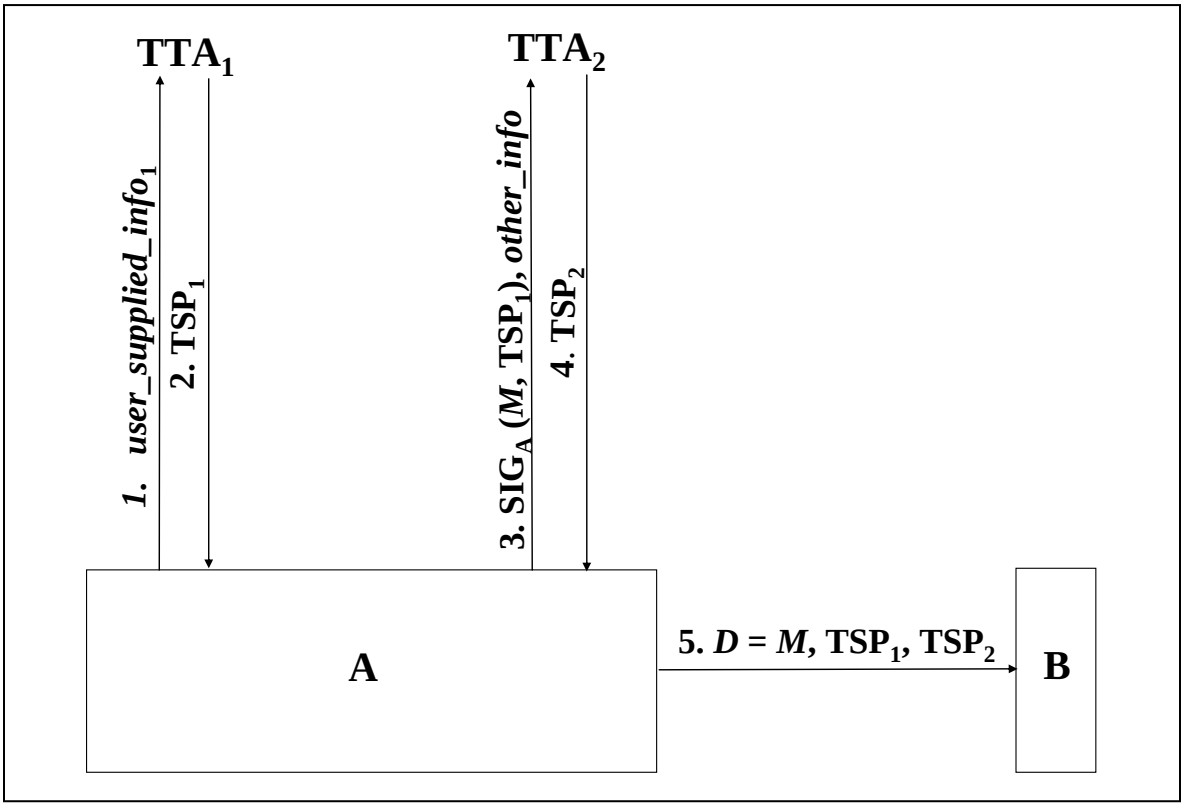

Figure 5: After Using a Scheme from 4.3, Entity A Requests A Second Timestamp

The process continues as follows, where *H* denotes a hash value that is generated on *M* using an **approved** hash function (i.e., *H* = Hash(*M*)):

3. Entity A generates $SIG_A(M, TSP_1)$ and sends it to TTA_2 in a second timestamp request, i.e., $user\_supplied\_info_2$ contains $SIG_A(M, TSP_1)$ ($user\_supplied\_info_2$ =

$SIG_A(M,TSP_1)$, *other_info*$_2$). Note that *other_info*$_2$ may be *Null*.

TSP_1 is defined to be the TSP used for the scheme in 4.3.1, 4.3.2 or 4.3.3.

4. TTA_2 returns TSP_2 to entity A:

   TSP_2 = *timestamped_data*$_2$, *timestamp_signature*$_{TTA2}$

   where:

   timestamped_data$_2$ = $SIG_A(M, TSP_1)$, *TTA_supplied_info*$_2$, *timestamp*$_2$.

   timestamp_signature$_{TTA2}$ = $SIG_{TTA2}(SIG_A(M, TSP_1)$, *TTA_supplied_info*$_2$, *timestamp*$_2$).

   If there is a mutual agreement between entity A and TTA_2, the following information may be omitted from the TSP_2 data transmitted by TTA_2:

   - Any portion of the *user_supplied_info* may be omitted, since it is known by entity A. However, if $SIG_A(M, TSP_1)$ is omitted from the TSP, it must be inserted back into the TSP when sent to entity B (see step 5).
   - Any portion of the *TTA_supplied_info*$_2$ may be omitted when the information is known by entity A.

   However, even if they are omitted from the transmitted TSP_2 data, both the entire $SIG_A(M, TSP_1)$ and the entire *TTA_supplied_info*$_2$ **shall** be included in the *timestamped_data*$_2$ that is used in the generation / verification of *timestamp_signature*$_{TTA2}$ = SIG_{TTA2} (*timestamped_data*$_2$).

   Upon receiving TSP_2 from TTA_2, entity A **should** verify *timestamp_signature*$_{TTA2}$ using the TTA_2's public signature verification key.

5. Entity A assembles *D* and sends it to entity B:

   D = *M*, TSP_1, TSP_2

   where TSP_1 is defined in the appropriate subsection of Section 4.3, and TSP_2 is as defined in step 4. Note that if $SIG_A(M, TSP_1)$ was not transmitted in the *timestamped_data* of the TSP sent from the TTA, it **shall** be inserted into TSP_1 used in the assembly of *D*.

   If omitted from the TSPs received from the TTAs, the entire *user_supplied_info*$_1$ and the entire *TTA_supplied_info*$_1$ **shall** be inserted into TSP_1, and the entire *user_supplied_info*$_2$ and the entire *TTA_supplied_info*$_2$ **shall** be inserted into TSP_2 unless there is a mutual agreement between entity A and entity B, in which case the following may be omitted from the transmitted TSP_1 and TSP_2 data in *D*:

   - Any portion of *user_supplied_info*$_1$ that is present in TSP_1 may be omitted if it is known or can be determined by entity B.
   - Any portion of *user_supplied_info*$_2$ that is present in TSP_2 may be omitted if it is known or can be determined by entity B.
   - Any portion of *TTA_supplied_info*$_1$ that is present in TSP_1 may be omitted

if it is already known by entity B.

- Any portion of *TTA_supplied_info*$_2$ may be omitted if it is already known by entity B.

However, any such information omitted from the transmitted data **shall** be included in the appropriate *timestamped_data* field (*timestamped_data*$_1$ and/or *timestamped_data*$_2$) used in the generation / verification of the quantities:

timestamp_signature$_{TTA1}$ = SIG$_{TTA1}$ (*timestamped_data*$_1$),

SIG$_A$(*M*, TSP$_1$), and

timestamp_signature$_{TTA2}$ = SIG$_{TTA2}$ (*timestamped_data*$_2$).

6. Upon receiving *D*, entity B does the following:

   - If the scheme in Section 4.3.2 was used, entity B computes $H' = \text{Hash}(M)$. If *H* was received by entity B in *D*, entity B also verifies that $H' = H$.
   - Verifies *timestamp_signature*$_{TTA1}$ using TTA$_1$'s public signature verification key,
   - Verifies SIG$_A$(*M*, TSP$_1$) using entity A's public signature verification key, and
   - Verifies *timestamp_signature*$_{TTA2}$ using TTA$_2$'s public signature verification key.

   Note that the order of performing these verifications is irrelevant; it is only important that all verifications are successful.

Entity B knows the following in addition to what is known in the appropriate scheme from Section 4.3:

a. SIG$_A$(*M*, TSP$_1$) was generated between the times indicated in *timestamp*$_1$ and *timestamp*$_2$, and was included in the *timestamped_data* signed by TTA$_2$.

b. *D* was assembled after the time indicated by *timestamp*$_2$.

4.4.2 Entity B Requests the Second Timestamp

Entity B may request a second timestamp after receiving and verifying the message *D* from entity A (see step 4 of the schemes in Sections 4.3.1 - 4.3.3). The TTA that provides the first timestamp must be trusted by both entity A and entity B, but the TTA that provides the second timestamp (TTA$_2$) may need to be trusted only by entity B. In general, any party that relies on the accuracy of the bounds on the generation time of SIG$_A$(*M*, TSP$_1$) must trust both TTAs.

Figure 6 shows entity B requesting a second timestamp after receiving the message *D* from entity A. The first four steps of this scheme are the same as those specified in one of the schemes in Sections 4.3.1 - 4.3.3.

After verifying the signatures in *D* (in step 4), the scheme proceeds as follows:

5. Entity B sends entity A's digital signature from *D* in *user_supplied_info*$_2$ to TTA$_2$

in a timestamp request (shown as message 4 in Figure 6), where *user_supplied_info*$_2$ = SIG$_A$ (*M*, TSP$_1$), *other_info*$_2$. Note that *other_info*$_2$ may be *Null*.

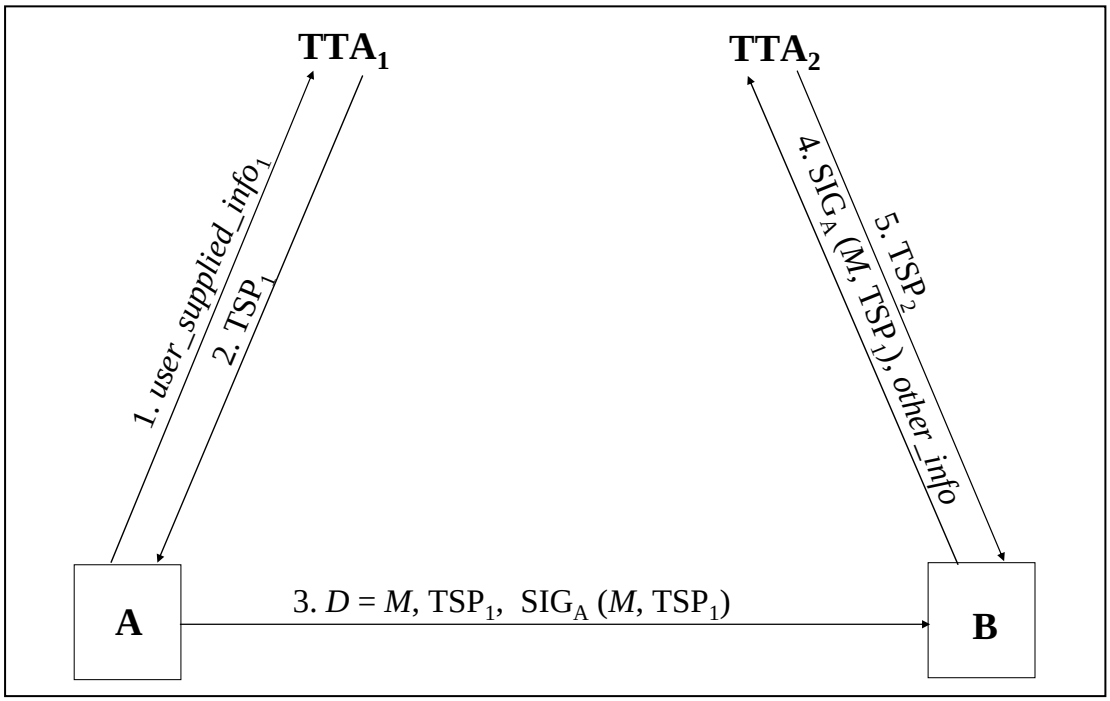

Figure 6: Entity A Used a Scheme from 4.3; Entity B Requests the Second Timestamp

6. TTA$_2$ returns TSP$_2$ to entity B (shown as message 5 in Figure 6):

   TSP$_2$ = *timestamped_data*$_2$, *timestamp_signature*$_{TTA2}$

   where:

   timestamped_data$_2$ = SIG$_A$ (*M*, TSP$_1$), *TTA_supplied_info*$_2$, *timestamp*$_2$

   timestamp_signature$_{TTA2}$ = SIG$_{TTA2}$(SIG$_A$ (*M*, TSP$_1$), *TTA_supplied_info*$_2$, *timestamp*$_2$).

   If there is a mutual agreement between entity B and TTA$_2$, the following information may be omitted from the TSP$_2$ data transmitted by TTA$_2$:

   - Any portion of the *user_supplied_info* may be omitted, since it is known by entity B.
   - Any portion of *TTA_supplied_info*$_2$ may be omitted if the information is already known by entity B.

   However, even though they may be omitted from the transmitted TSP$_2$ data, the entire *user_supplied_info* and the entire *TTA_supplied_info*$_2$ **shall** be included in the *timestamped_data*$_2$ that is used in the generation / verification of *timestamp_signature*$_{TTA2}$ = SIG$_{TTA2}$ (*timestamped_data*$_2$).

7. Entity B then verifies $timestamp\_signature_{TTA2}$ using TTA_2's public signature verification key.

In addition to what is known from each scheme, entity B now has evidence that the signature $SIG_A(M, TSP_1)$ was generated between the times indicated in $timestamp_1$ and $timestamp_2$. This evidence can be presented to any third party who also trusts the TTA.

5 Evidence of Timeliness Using Verifier-Supplied Data

Independent of the use of a trusted timestamping service by entity A, entity A can provide evidence to the verifier (entity B) of the timeliness of its signature by:

- Combining fresh data that was supplied by the intended verifier with any other data to be included in the message, and
- Generating a digital signature on the combination.

In the following schemes, a nonce is used to assist in establishing timeliness. A nonce is a time-varying value, represented as a byte string that has (at most) a negligible chance of repeating. For example, a nonce may be composed of one (or more) of the following components:

1. A random value that is generated anew for each nonce, using an **approved** random bit generator. The security strength of the RBG used to obtain each random value **shall** be equal to or greater than the security strength associated with the digital signature process. The appropriate security strength **shall** have been determined prior to the generation of this random value. The length of the RBG output **shall** be at least the security strength associated with the digital signature process (e.g., if the security strength of the digital signature process is 112 bits, then the length of the RBG output **shall** be at least 112 bits). A nonce containing a component of this type is called a *random nonce*.
2. A timestamp of sufficient resolution (detail) so that it is different each time it is used.
3. A monotonically increasing sequence number.

If a combination of a timestamp and a monotonically increasing sequence number is used without a random nonce, the sequence number **shall** be reset only when the timestamp changes. (For example, a timestamp may show the date but not the time of day, so a sequence number is appended that will not repeat during a particular day.)

When using a nonce, a random nonce **should** be used.

In the following schemes, let *Nonce* be the verifier-supplied data (i.e., supplied by entity B). Let *D* be the signed message to be sent to entity B, and let $SIG_A( )$ be the digital signature that is computed by entity A using an **approved** hash function and an **approved** signature algorithm; $SIG_A( )$ is verified using entity A's public signature verification key. The discussions assume that entity B successfully verifies all digital signatures.

5.1 The Basic Scheme

As shown in Figure 7, the scheme proceeds as follows:

1. Entity B sends a newly

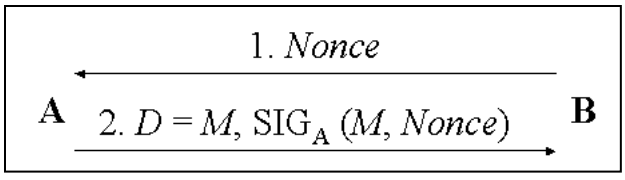

Figure 7: Using Verifier Supplied Data

generated *Nonce* to entity A.

2. Entity A signs (*M, Nonce*), assembles *D*, and sends it to entity B, where:

   $D = M, \text{SIG}_A(M, Nonce)$.

3. Upon receiving *D*, entity B then verifies $\text{SIG}_A(M, Nonce)$ using entity A's public signature verification key.

Entity B knows the following:

a. *M* could have been assembled either before or after the *Nonce* was received from entity B.

b. $\text{SIG}_A(M, Nonce)$ was generated at some time after the *Nonce* was received by entity A.

c. *D* was assembled by entity A after the *Nonce* was received.

If a more precise time is required for the generation of $\text{SIG}_A(M, Nonce)$, a trusted timestamp may be acquired as specified in Section 5.2; the timestamp will provide assurance that $\text{SIG}_A(M, Nonce)$ was generated by (at least) the time indicated in the timestamp.

5.2 Using a Timestamp to Obtain More Precision

When verifier-supplied data is used, a more precise time for the generation of a digital signature can be provided by requesting a timestamp. Any entity could make the request, although requests by entity A and entity B are discussed below. This procedure has the most value if the timestamp request is made as close to the generation of entity A's signature as possible, since it may be used to establish a minimal time interval during which entity A's signature was generated. The discussions assume that entity B successfully verifies all digital signatures.

5.2.1 Entity A Requests the Timestamp

Entity A may request a timestamp before sending a message to entity B. The TTA that provides the timestamp must be trusted by both parties.

Figure 8 depicts the case where entity A provides more precision as to when $\text{SIG}_A(M, Nonce)$ was generated. Note that the *Nonce* supplied by entity B could include a timestamp, in which case, this (additional) timestamp could establish an interval during which $\text{SIG}_A(M, Nonce)$ was generated.

The first step is the same as that specified in Section 5.1. The process continues as follows:

2. Entity A generates $\text{SIG}_A(M, Nonce)$ and sends it to the TTA in a timestamp request, where *user_supplied_info* = $\text{SIG}_A(M, Nonce)$, *other_info*. Note that *other_info* may be Null.

3. The TTA returns a TSP to entity A:

   TSP = $timestamped\_data, timestamp\_signature_{TTA}$

where:

timestamped_data = SIG_A(*M*, *Nonce*), *TTA_supplied_info*, *timestamp.*

timestamp_signature$_{TTA}$ = SIG_{TTA}(SIG_A(*M*, *Nonce*), *TTA_supplied_info*, *timestamp*).

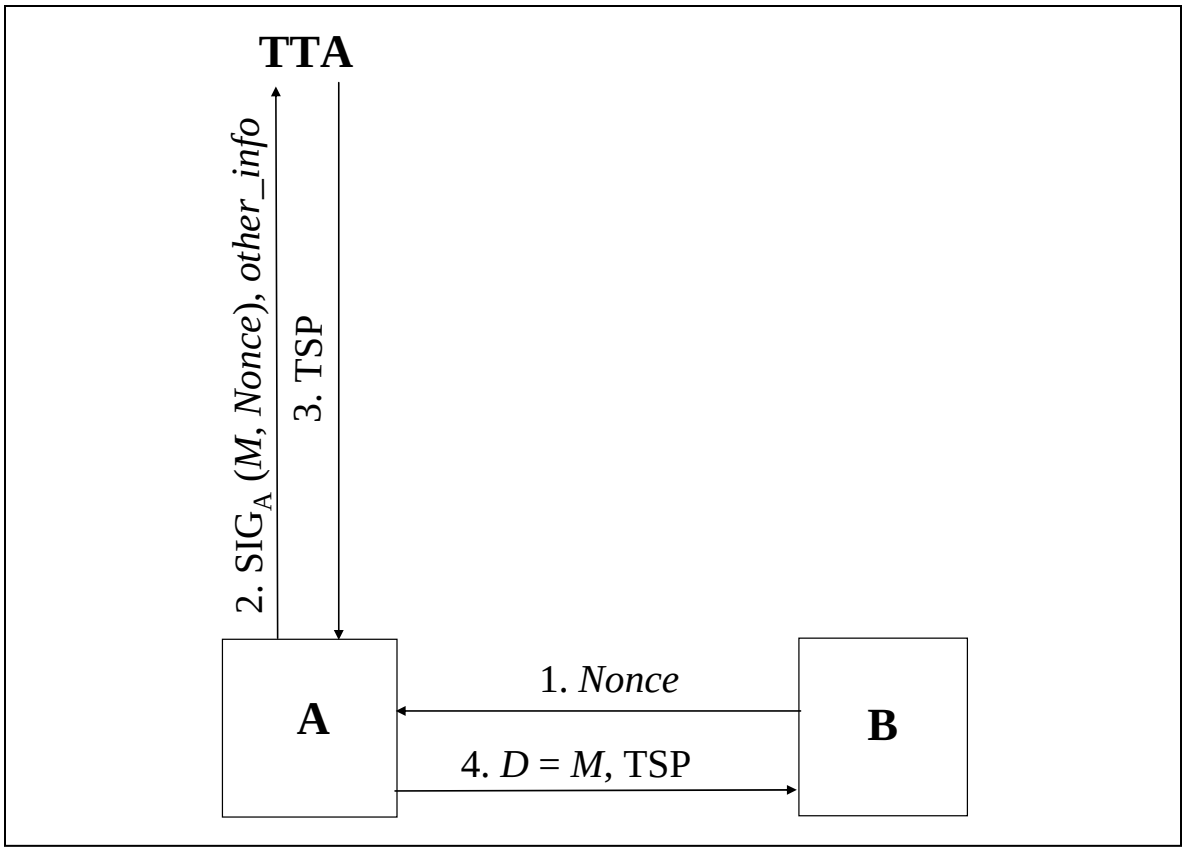

Figure 8: Entity A Requests a Timestamp

If there is a mutual agreement between entity A and the TTA, the following information may be omitted from the TSP data transmitted by the TTA:

- Any portion of the *user_supplied_info* may be omitted, since it is known by entity A. However, if SIG_A(*M*, *Nonce*) is omitted from the TSP, it must be inserted back into the TSP when sent to entity B (see step 4).
- Any portion of the *TTA_supplied_info* may be omitted if this information is already known by entity A.

However, even though they may be omitted from the transmitted TSP data, both the entire *user_supplied_info* and the entire *TTA_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of *timestamp_signature*$_{TTA}$ = SIG_{TTA}(*timestamped_data*).

Upon receiving the TSP from the TTA, entity A **should** verify *timestamp_signature*$_{TTA}$ using the TTA's public signature verification key.

4. Entity A assembles *D* and sends it to entity B:

   D = *M*, TSP

   where TSP is as defined in step 3.

   If any portion of the *user_supplied_info* was omitted from the TSP received from the TTA, then the entire *user_supplied_info* **shall** be inserted into the TSP used in the assembly of *D* unless there is a mutual agreement between entity A and entity B, in which case, any portion of the *user_supplied_info* may be omitted from the TSP data transmitted to entity B if it is known by entity B. However, the entire *user_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA}$ and $SIG_A(M, Nonce)$.

   If any portion of the *TTA_supplied_info* was omitted from the TSP received from the TTA, then the entire *TTA_supplied_info* **shall** be inserted into the TSP used in the assembly of *D* unless there is a mutual agreement between entity A and entity B, in which case, any portion of the *TTA_supplied_info* may be omitted from the TSP data transmitted by entity A if it is known by entity B. However, the entire *TTA_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA}$ and $SIG_A(M, Nonce)$.

5. Upon receiving *D*, entity B does the following:

   - Verifies $SIG_A(M, Nonce)$ using entity A's verification public key, and
   - Verifies $timestamp\_signature_{TTA}$ using the TTA's public signature verification key.

   Note that the order of performing these verifications is irrelevant; it is only important that both verifications are successful.

In addition to what has been determined by entity B in Section 5.1, entity B now has determined that $SIG_A(M, Nonce)$ was generated between the time that the *Nonce* was sent to entity A and the time indicated in the timestamp

5.2.2 Entity B Requests the Timestamp

Entity B may request a timestamp after receiving the message *D* from entity A. The TTA that provides the timestamp may need to be trusted only by entity B unless another party needs to be convinced of the signature generation time.

Figure 9 depicts the case where entity A provides evidence of timeliness using data supplied by entity B, and entity B determines (more precisely) when the digital signature generated by entity A was received. This scheme is similar to the scheme in Section 4.3.4, with the only difference being the nonce sent from entity B to entity A. Note that the *Nonce* supplied by entity B could include a timestamp (trusted by both entity A and entity B), in which case, this (additional) timestamp would establish an interval during which the signature was generated.

The first three steps are the same as those specified in Section 5.1. After verifying the signature in *D*, the scheme proceeds as follows:

4. Entity B sends $SIG_A(M, Nonce)$ as the *user_supplied_info* to the TTA in a timestamp request (shown as message 3 in Figure 9); therefore,*user_supplied_info* contains $SIG_A(M, Nonce)$, i.e., *user_supplied_info* = $SIG_A(M, Nonce)$, *other_info*. Note that other_info may be Null.

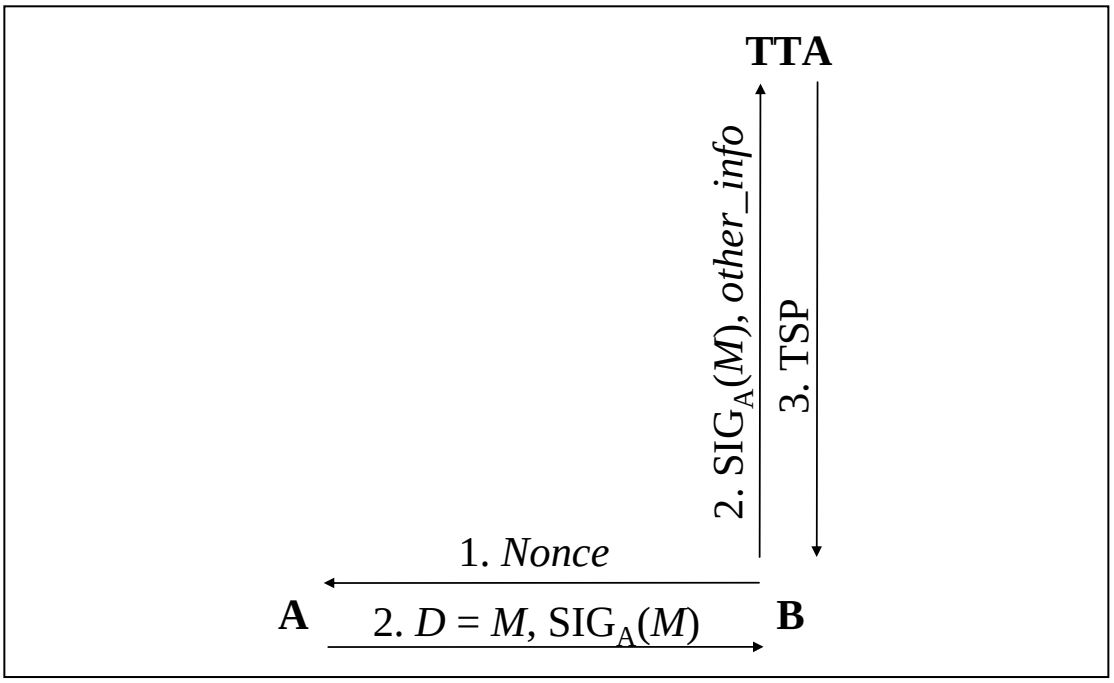

Figure 9: Entity B Requests a Timestamp

5. The TTA returns a TSP to entity B (shown as message 4 in Figure 9):

   TSP = *timestamped_data*, $timestamp\_signature_{TTA}$

   where:

   timestamped_data = $SIG_A(M, Nonce)$, *TTA_supplied_info*, *timestamp*

   $timestamp\_signature_{TTA} = SIG_{TTA}(SIG_A(M, Nonce), TTA\_supplied\_info, timestamp)$.

   If there is a mutual agreement between entity B and the TTA, the following information may be omitted from the TSP data transmitted by the TTA:

   - Any portion of the *user_supplied_info* may be omitted, since it is known by entity B.
   - Any portion of the*TTA_supplied_info* may be omitted when this information is known by entity B.

   However, even though they may be omittted from the transmitted TSP data, both the entire $SIG_A(M, Nonce)$ and the entire *TTA_supplied_info* **shall** be included in the *timestamped_data* that is used in the generation / verification of $timestamp\_signature_{TTA} = SIG_{TTA}(timestamped\_data)$.

6. Entity B then verifies $timestamp\_signature_{TTA}$ using the TTA's public signature verification key.

In addition to what has been determined by Entity B in Section 5.1, entity B has proof that $timestamp\_signature_{TTA}$ was generated before the time indicated in the *timestamp*.

Appendix A: References

[1] Federal Information Processing Standard (FIPS) 186-3, The Digital Signature Standard, dated XXX.

[2] NIST Special Publication (SP) 800-57, Recommendation for Key Management – Part 1: General, Revised June 12, 2006.

[3] Request for Comment 3161, Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP), IETF Standards Track, August 2001.

[4] ISO/IEC 18014, *Information Technology – Security Techniques – Time-stamping Services*, 2002.

[5] ANS X9.95, *Trusted Timestamp Management and Security*, 2005.