GB

NATIONAL STANDARD OF THE

PEOPLE'S REPUBLIC OF CHINA

ICS 35.040

L 80

GB/T 36644-2018

Information security technology - Methods for obtaining security attestations for digital signature applications

信息安全技术　数字签名应用安全证明获取方法

Issued on: September 17, 2018 **Implemented on: April 01, 2019**

Issued by: State Administration for Market Regulation;
Standardization Administration of PRC.

Table of Contents

Information security technology - Methods for obtaining security attestations for digital signature applications

1 Scope

This standard specifies a set of methods for obtaining security attestations for digital signature application, to standardize the process of security attestations for digital signature application.

This standard is applicable to signature application scenarios that need to provide the security of the digital signature generation process and have clear requirements for the signature generation time.

2 Normative references

The following documents are indispensable for the application of this document. For dated reference documents, only the dated version applies to this document. For undated references, the latest version (including all amendments) applies to this document.

GB/T 20520-2006 Information security technology - Public key infrastructure - Time stamp specification

GB/T 25069-2010 Information security technology - Terminology

GB/T 32918.1-2016 Information security technology - SM2 elliptic curve public key cryptography algorithm - Part 1: General rules

GB/T 32918.2-2016 Information security technology - SM2 elliptic curve public key cryptography algorithm - Part 2: Digital signature algorithm

3 Terms and definitions

The terms and definitions as defined in GB/T 25069-2010 as well as the following terms and definitions apply to this document.

3.1

includes: the acquisition of the security attestation of the attributes of the private key, the acquisition of the security attestation of public key validity, the acquisition of the security attestation of the generation time of the digital signature.

The owner of the private key refers to the entity that is authorized to use the private key in the public-private key pair for digital signature generation. The generated digital signature can be verified by the corresponding public key. Being authorized to use the private key to generate a signature does not mean that the owner actually knows the correct private key. Therefore, before the owner performs a digital signature, it is necessary to obtain a attestation of private key possession.

According to the different generation methods of signature public-private key pairs, the ways in which the private key is known can be divided into the following five types:

a) The owner generates and maintains a public-private key pair; only the owner knows the private key;

b) The owner generates a public-private key pair with the help of TTP; however, the private key can only be known by the owner;

c) The public-private key pair is generated by TTP and provided to the owner; the owner and TTP know the private key at the same time;

d) The public-private key pair is generated by the method a); then provided to the TTP acting as the key server, so that the owner and the TTP know the private key at the same time;

e) The public-private key pair is generated by means of b); then provided to the TTP acting as the key server, so that the owner and the TTP acting as the key server know the private key at the same time.

The latter three methods need to be established on the trust that TTP will not generate a digital signature with a private key. The public-private key pair owner, the signature verifier, other signature relying parties must be able to share this trust. The methods c), d), e) have a lower level of credibility than the method a) and b).

The usage scenarios of the attestation of private key possession are as follows:

- The owner of the public-private key pair needs to obtain a security attestation of the attributes of the private key before or at the same time the signature is generated;

- Before or at the same time, the verifier needs to obtain the security

t_1 - The time when the relying party trusts the proof generation ahead of t_G;

t_2 - The time when the relying party trusts the proof generation lags of t_G;

d - The difference between t_1 and t_2;

t_A - The designated attestation time, which shall satisfy $t_1 \leq t_A \leq t_2$. For convenience, it may specify $t_A = t_1$, or $t_A = t_2$.

a, b, c, d are determined by the relying party of the signature or its organization, considering the following factors:

- The values of a, b, c are determined according to the requirements of the organization's policy on the security attestation of digital signatures; at the same time, it shall also consider the difficulty of obtaining the attestation of private key possession used;

- The value of d shall be less than half of the minimum value of a and b, that is, d < 1/2 min (a, b), meanwhile the determination of d shall also consider the error estimate of the signature acquisition time t_G. In addition, the determination of d also considers the time of secure transmission of the security attestation on the network.

According to the estimated acquisition time t_A of the attestation of private key possession, the relying party can determine the proof level at different times. As shown in Figure 2, at the time t_A - (a - d) and t_A + (b - d), the security attestation obtained has a high or medium attestation level, which depends on the process of obtaining the security attestation. After t_A + (b - d), the attestation level gradually decreases. At t_A + (b - d) + c, the security attestation level drops to a low level. After that, the security attestation level will remain low. If the policy requires a high level of security attestation, then the security attestation needs to be re-obtained.

See 5.2.4 for the determination of specific security attestation’s timeliness model parameters.

5.2.3 The process of obtaining the attestation of private key possession

5.2.3.1 General

The attestation of private key possession can be obtained by one or more of the following methods:

a) The owner of the private key uses the private key to sign a new digital signature; then uses its corresponding public key to verify;

b) Regenerate the public-private key pair; then compare it with the public-

possession, the entity obtaining the attestation will be assigned to an attribute security attestation message, referred to as the attestation message for short. Then the owner of the private key signs the message; this signature is called a attestation signature.

The attestation message shall include the following information:

a) The identity of the signer;

b) The identity of the potential verifier;

c) Time stamp token TST: the TST is generated by a trusted time stamp authority (TTSA) trusted by all relying parties. TST can be obtained from TTSA by the signer, or from TTSA by potential verifiers, then passed to the signer. All relying parties shall recognize the strength of TTSA's signature security;

d) A nonce value provided by a verifier. If a nonce is selected, the randomness of the nonce value must be equal to or exceed the randomness of the private key to be certified. If the attestation message does not contain TST, whilst the relying party requires that the attestation time be recorded when verifying the attestation signature, the nonce value needs to include a timestamp provided by the verifier, to indicate the time when the nonce value was provided to the attestation message.

It is not necessary to prove that the public key corresponding to the private key in the message is displayed. However, the inclusion of the public key display information must be determined strictly according to the following description:

- If the signer who generated the attestation signature can successfully display the public key before obtaining the attestation message, the public key information can be removed from the attestation message;

- If the attestation message contains TST, the public key display shall be before the time marked by TST. If there is no TST, the public key shall be displayed before the time included in the nonce value; if there is no TST and the nonce value does not include time, evidence that can be trusted by all parties shall be shown to prove that the public key is displayed before the attestation signature is generated;

- The public key can be displayed as an attestation with a time stamp;

- The public key can also be displayed as a signature previously issued by the signer. The private key used for signing the signature shall be the same as the private key for obtaining the attestation, that is, the same public key can be used for verification. The signature is issued earlier than the time when the attestation message is sent to the signer;

for the assignment of t_1.

The relying party selects the most credible time source from the above t_1 assignment candidates for t_1 assignment. In the case of equal credibility, the latest time shall be selected first for the assignment of t_1.

As described in 5.2.2, t_2 is a time point lagging behind or equal to the time when the attestation signature is generated. Several possible assignments of t_2 are as follows:

- If the attestation signature's generation time is included in a TST issued by a TTSA trusted by all relying parties, the time included in the TST can be used as a candidate for the assignment of t_2;
- If the verifier records the time of receipt of the attestation signature, the recorded time can be used as a candidate for the assignment of t_2;
- If the verifier records the time at which the attestation signature was verified, the recorded time can be used as a candidate for the assignment of t_2.

The relying party selects the most credible time source from the above t_2 assignment candidates for t_2 assignment. In the case of equal credibility, the earliest time shall be selected first for the assignment of t_2.

If the attestation signature has been successfully verified, meanwhile the relying party has determined the error accuracy d of the attestation time, the attestation time t_A can be estimated as follows:

- If $t_2 - t_1 \leq d$, meanwhile the credibility of t_1 is not less than t_2, t_1 shall be selected as the attestation time t_A;
- If $t_2 - t_1 \leq d$, meanwhile the credibility of t_1 is less than t_2, t_2 shall be selected as the attestation time t_A;
- If $t_2 - t_1 > d$, then it does not obtain the attestation of the private key possession, there is no need to assign the attestation time t_A.

If the attestation signature cannot be verified, then it does not obtain the attestation of the private key possession, there is no need to assign the attestation time t_A.

5.2.3.2.3 Specify the initial attestation level

After the attestation signature is verified and the attestation time is specified, the initial level of the attestation needs to be specified. The initial level of attestation is specified as follows:

same time, or after the attestation time of the attestation of private key possession of the corresponding private key.

The generation time of the ordinary message signature is represented by t_s. The t_s may have a certain value, or a range of values, or a completely uncertain value. If t_s has a value with sufficient precision, meanwhile the attestation of the private key possession corresponding to the private key has obtained, then it may follow the timeliness model in 5.1.1 and use the following method to determine the level of the security attestation of this ordinary message signature:

a) If $(t_A - (a - d)) \leq t_s \leq (t_A + (b - d))$, then the security attestation level of the ordinary message signature is equal to the initial attestation level of the acquired attestation of the private key possession;

b) If $(t_A - (a - d)) \leq t_s \leq (t_A + (b - d) + c)$, then the security attestation level of the ordinary message signature will gradually decrease from the initial state to a low level;

c) $t_s > (t_A + (b - d) + c)$, then the security attestation level of the ordinary message signature is low.

If t_s does not have a certain value, the security attestation level of the ordinary message signature is determined to be low.

5.2.4 Specific acquisition flow of attestation of private key possession

5.2.4.1 Public-private key pair owner obtaining the attestation of private key possession

The owner of a public-private key pair can use one or more of the following methods to obtain attestation of private key possession:

a) The public-private key owner adopts the attestation signature to obtain the attestation of private key possession:

The public-private key pair owner needs to complete the following:

1) Determine the appropriate value of d, for example, it can determine the value of d after determining the appropriate values of a, b, c according to the timeliness model in 5.2.2.

2) Determine a credible t_1 value;

3) Generate a new attestation message for obtaining the attestation of private key possession;

4) Use the private key to obtain the certificate to sign the certificate

t_2; the entity specifying the initial attestation level shall also know the method to determine the value of t_1 and t_2;

- The owner of the public-private key must record the attestation time and initial attestation level.

c) The owner of the public-private key pair obtains the attestation of private key possession through key regeneration:

The public-private key pair owner needs to complete the following:

1) Determine the appropriate value of d, for example, it can determine the value of d after determining the appropriate values of a, b, c according to the timeliness model in 5.2.2.

2) Determine a credible t_1 value;

3) Choose one of the following operations:

- Regenerate the key pair corresponding to the private key to be certified;
- Regenerate a key in the key pair corresponding to the private key to be certified.

4) Compare the value of the regenerated key pair (key) with the value of the key currently owned;

5) If the match is successful:

- Determine a credible value for t_2;
- If $t_1 \leq t_2 \leq t_1 + d$, specify and record the attestation time; specify the initial attestation level.

d) The owner of the public-private key pair obtains the attestation of the private key possession from the TTP through key regeneration:

1) The owner of the public-private key must determine the appropriate value of d. For example, according to the timeliness model in 5.2.2, the value of d can be determined on the basis of determining the appropriate values of a, b, c; if the TTP is responsible for specifying the attestation time, the value of d must be known to TTP;

2) The public-private key pair owner and/or TTP must determine a t_1 value, which must be trusted by the public-private key pair owner;

3) The owner of the public-private key pair shall provide the key held by it

1) Determine the appropriate value of d, for example, it can determine the value of d after determining the appropriate values of a, b, c according to the timeliness model in 5.2.2.

2) The public-private key owner generates a new attestation message for the attestation of private key possession;

3) The owner of the public-private key signs the attestation message with the private key to be certified to generate an attestation signature;

4) The public-private key owner provides the attestation message, attestation signature and other necessary data to TTP;

5) TTP determines a credible value for t_1;

6) TTP verifies the attestation signature with the public key corresponding to the private key to obtain the attestation;

7) If the verification is successful:

- TTP determines a credible value for t_2;
- If $t_1 \leq t_2 \leq t_1 + d$, TTP begins to specify the attestation time and initial attestation level;
- TTP records the attestation time and initial attestation level; meanwhile these values shall also be provided to the owner of the public-private key pair.

b) TTP obtains the attestation of private key possession from the owner of the public-private key pair through the method of key (key pair) regeneration:

1) Determine the appropriate value of d, for example, it can determine the value of d after determining the appropriate values of a, b, c according to the timeliness model in 5.2.2.

2) The public-private key pair owner provides the key information to be certified and any other necessary data to TTP;

3) TTP determines a credible value for t_1;

4) TTP needs:

- Regenerate the entire key pair of the owner of the public-private key pair;
- Or regenerate a key in the entire key pair of the owner of the public-

attestation signature, to obtain the attestation of private key possession:

1) Determine the appropriate value of d, for example, it can determine the value of d after determining the appropriate values of a, b, c according to the timeliness model in 5.2.2.

2) The owner of the public-private key must provide a new attestation message, attestation signature and other necessary data to the verifier;

3) The verifier must determine a credible value for t_1;

4) The verifier verifies the attestation signature with the public key corresponding to the private key which needs obtaining the attestation;

5) If the attestation signature verification is passed:

- The verifier must determine a credible value for t_2;
- If $t_1 \leq t_2 \leq t_1 + d$, the verifier starts to specify the attestation time and initial attestation level;
- The verifier records the attestation time and initial attestation level.

b) The verifier obtains the attestation of private key possession by cooperating with TTP:

1) Determine the appropriate value of d, for example, it can determine the value of d after determining the appropriate values of a, b, c according to the timeliness model in 5.2.2.

2) The verifier asks TTP for the attestation of private key possession;

If TTP successfully obtains the attestation of private key possession;

3) TTP shall provide the verifier with the time of obtaining the attestation, the initial attestation level, the method of obtaining attestation;

4) If required by the verifier, TTP shall also provide the upper limit of $t_2 - t_1$, the method of obtaining time t_1 and t_2, and/or the evaluation value of the TTP for the attestation at the time of signature issuance verified by the verifier;

5) Verifier:

- If the upper limit of $t_2 - t_1$ as provided by TTP is greater than d, rejects the attestation of the private key possession as provided by TTP;
- If the upper limit value of $t_2 - t_1$ as provided by TTP is less than d,

5.3.2 Obtaining the attestation of public key validity

The owner of a public-private key pair can obtain a security attestation of public key validity through the following five methods:

a) It is generated by public-private key pair owner: The owner uses the identified method to generate the public-private key pair;

b) It is generated by public-private key pair owner in cooperation with TTP: The owner, with the help of TTP, adopts the identified method to generate the public-private key pair;

c) The owner adopts a clear process to verify public key validity: The owner obtains a security attestation of public key validity by performing a clear verification process, see 5.3.4 for the specific verification process;

d) TTP uses a clear process to verify public key validity: The owner needs to receive the attestation that proves that TTP has indeed obtained a security attestation of public key validity through a clear verification process. For the specific verification process, see 5.3.4. TTP verification results must be provided to the owner;

e) The public-private key pair is generated by TTP: TTP generates a public-private key pair and provides it to the owner. If this method is adopted, a public key validity verification process shall be adopted. The verification process can be carried out by the owner according to the above method c), or TTP according to the above method d).

Among them, the combination of method a) or b) and method c) or d) can obtain more effective security attestation.

The owner or its agent needs to know which method is used to obtain the security attestation of public key validity to determine whether the obtained security attestation of public key validity meets the requirements of the owner.

5.3.3 Verifier obtains a security attestation of public key validity

The verifier of the signature obtains the security attestation of public key validity in the public-private key pair used by the issuer when signing, by using the following three methods:

a) The verifier uses a clear process to verify public key validity: the verifier obtains a security attestation of public key validity by performing a clear verification process, see 5.3.4 for the specific verification process.

b) TTP uses a clear process to verify public key validity: the verifier must receive a certificate that proves that the TTP has indeed obtained a

this way. The format of TSP is described as follows:

Among them, the comma is used to separate different data, not part of the data format.

a) TSP = timestamped_data,timestamp_signature$_{TTSA}$

TSP is composed of time stamp data and its digital signature. The digital signature is the signature of the time stamp data by the TTSA's private key.

b) timestamp_signature$_{TTSA}$ = SIG$_{TTSA}$(timestamped_data)

The digital signature algorithm SIG$_{TTSA}$ is a digital signature operation used on time stamp data. The signature private key is the digital signature private key of TTSA. The private key is only used to generate digital signatures on time stamp data.

c) timestamped_data = user_supplied_info,TTSA_supplied_info,timestamp

Among them:

1) user_supplied_info: User-supplied information, which is the information provided by an entity when requesting a timestamp from TTSA; user_supplied_info can be empty in actual applications. If this information is provided, this information will be used by TTSA when generating the time stamp signature, without the need to be returned to the requester when transmitting the time stamp packet. If this information is used, it shall be ensured that the information is visible when an entity wants to verify timestamp_signature$_{TTSA}$;

2) TTSA_supplied_info: Information provided by TTSA, which is the additional information used by TTSA when generating timestamp_signature$_{TTSA}$. TTSA_supplied_info may be empty in actual applications. As long as this part of the information can be regenerated when verifying the timestamp_signature$_{TTSA}$ signature, any part of it can be deleted from the timestamp packet;

3) The timestamp contains time and (possibly) other information.

Therefore, the general TSP format generated by TTSA is as follows:

$$TSP = user\_supplied\_info, TTSA\_supplied\_info, timestamp, SIG_{TTSA}(user\_supplied\_info, TTSA\_supplied\_info, timestamp)$$

Among them, user_supplied_info and TTSA_supplied_info may be empty.

TTSA may broadcast a TSP or respond to a TSP to the requesting entity, as

from TTSA, entity A shall:

1) Check whether the transmitted user-supplied information user_supplied_info is correct;

2) Use the public signature verification key of TTSA to verify the digital signature timestamp_signature$_{TTSA}$.

c) Entity A signs (M, TSP), assembles data D, sends it to entity B:

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

Among them, TSP is the same as specified in 5.4.2.1; the assembly data D is divided into the following two situations:

1) If any part of the user-supplied information is deleted from the TSP from TTSA, then the entire user-supplied information must be backfilled into the TSP when data D is formed, unless there is a mutual agreement between entity A and entity B, which makes the deleted part be known by entity B or regenerated. In the case of partial information deletion, the entire user-supplied information user_supplied_info shall be included in the timestamped_data used to generate/verify the digital signature timestamp_signature$_{TTSA}$ and SIG_A (M, TSP);

2) If any part of the information TTSA_supplied_info provided by TTSA is deleted from the TSP from TTSA, then the entire information provided by TTSA must be backfilled into the TSP when forming D, unless there is a mutual agreement between entity A and entity B. B can determine this information. In this case, if entity B knows or can determine that TTSA provides information, then any part of its information can be deleted from the TSP data transmitted by entity A. However, the entire information TTSA_supplied_info as provided by the TTSA shall be included in the timestamped_data used to generate/verify the digital signature and the time stamp data timestamp_signatureTTSA as sued for SIG_A (M, TSP).

d) Upon receiving D, entity B follows the following steps:

1) Use the TTSA public key to verify the digital signature timestamp_signature$_{TTSA}$;

2) Use A's public key to verify the digital signature SIG_A (M, TSP).

The order of execution of the above two steps is irrelevant, however, it is necessary to ensure that the two verifications are successful.

After completing the verification in step 4, entity B obtains the following

2) Any part of the information of TTSA_supplied_info can be deleted, as long as entity A knows the information or can be determined by entity A.

Although this information can be deleted from the TSP transmission information, the complete user_supplied_info and TTSA_supplied_info shall be included in the timestamped_data, to generate the following signature and verification:

$$timestamp\_signature_{TTSA} = SIG_{TTSA}\ (timestamped\_data).$$

Upon receiving the TSP sent from TTSA, entity A shall:

- Check whether the transmitted user information is correct;
- Use TTSA's public key to verify $timestamp\_signature_{TTSA}$.

c) Entity A signs (M, TSP), assembles D, sends it to entity B:

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

Among them, TSP is specified in step 2.

If any part of the information in user_supplied_info is deleted from the TSP from TTSA, then the entire user_supplied_info shall be backfilled into the TSP when D is formed, unless there is a mutual agreement between entity A and entity B, so that B can determine this information. In general, the following information in the TSP transmitted in D can be deleted:

1) H can be deleted because it can be (re)calculated by entity B;

2) Any other_info in user_supplied_info can be deleted as long as it can be known or determined by B;

However, the entire user_supplied_info shall be included in the timestamped_data used to generate/verify $timestamp\_signature_{TTSA}$ and SIG_A (M, TSP).

If any part of the information in TTSA_supplied_info is deleted from the TSP from TTSA, then the entire TTSA_supplied_info shall be backfilled to the TSP when D is formed, unless there is a mutual agreement between entity A and entity B, that allows B to determine this information. In this case, if entity B knows or can determine TTSA_supplied_info, then any part of its information can be deleted from the TSP data transmitted by entity A. However, the entire TTSA_supplied_info shall be included in the timestamped_data used to generate/verify $timestamp\_signature_{TTSA}$ and SIG_A (M, TSP).

TSP when D is formed, unless there is a mutual agreement between entity A and entity B; the deleted part can be known by entity B or regenerated. In the case of partial information deletion, the entire user_supplied_info shall be included in the timestamped_data used to generate/verify timestamp_signature$_{TTSA}$ and SIG$_A$ (M, TSP).

If any part of the information in TTSA_supplied_info is deleted from the TSP from TTSA, then the entire TTSA_supplied_info shall be backfilled into the TSP when D is formed, unless there is a mutual agreement between entity A and entity B, to allow B to be able to determine this information. In this case, if entity B knows or can determine TTSA_supplied_info, then any part of its information can be deleted from the TSP data transmitted by entity A. However, the entire TTSA_supplied_info shall be included in the timestamped_data used to generate/verify timestamp_signature$_{TTSA}$ and SIG$_A$ (M, TSP).

d) When entity B receives D, it needs to complete the following:

1) Use A's public key to verify SIG$_A$ (M); SIG$_A$ (M) is obtained by TSP in D;

2) Verify timestamp_signature$_{TTSA}$ with the public key of TTSA;

3) Use entity A's public key to verify SIG$_A$ (M, TSP);

The order of execution of the above steps is irrelevant, however, it is necessary to ensure that these verifications are successful.

After completing the verification in step 4, entity B obtains the following attestation:

- M and SIG$_A$ (M) are generated before the time indicated by the timestamp in the TSP, meanwhile SIG$_A$ (M) is included in the timestamped_data signed by TTSA;
- M has not been changed after the time indicated by the timestamp in TSP;
- SIG$_A$ (M, TSP) is generated after the time indicated by the timestamp in TSP;
- D is assembled after the time indicated by the timestamp in the TSP.

If a more accurate SIG$_A$ (M, TSP) generation time is required, a second credible timestamp is necessary, see 5.4.2.3.

5.4.2.2.4 Entity B provides M's digital signature to TTSA to obtain TSP

Entity B can provide TTSA with the digital signature of message M received by A and provide attestation of digital signature timeliness. The specific process is

information;

Although this information can be removed from the TSP transmission information, the complete user_supplied_info and TTSA_supplied_info shall be included in the timestamped_data used to generate/verify timestamp_signature$_{TTSA}$.

d) Upon receiving the TSP, entity B needs to complete the following:

1) Check whether the transmitted user_supplied_info is correct;

2) Verify timestamp_signature$_{TTSA}$ with the public key of TTSA;

3) If SIG$_A$ (M) is not verified before being sent to TTSA, use A's public key to verify.

The order of execution of the above steps is irrelevant, however, it is necessary to ensure that these verifications are successful.

After completing the verification in step 4, entity B obtains the following attestation:

- M and SIG$_A$ (M) are generated before the time indicated by the timestamp in the TSP;
- SIG$_A$ (M) is included in timestamped_data and signed by TTSA.

These evidences [i.e., M, SIG$_A$ (M), TSP] can be presented to any third party who trusts TTSA.

5.4.2.3 Use the second TSP to obtain a more refined attestation of signature timeliness

5.4.2.3.1 Basic scheme

If another TSP can be obtained from TTSA, a more accurate signature generation time can be obtained. The second TSP request can be issued by entity A and entity B. If the second TSP request is as close as possible to the signature generation time of the first timestamp packet of entity A, the accuracy of the attestation will be greater.

In the following scheme, the initial steps are specified by 5.4.2.2.1 to 5.4.2.2.3, meanwhile the transformation of the following information representation is completed:

a) user_supplied_info becomes user_supplied_info$_1$;

b) other_info in user_supplied_info$_1$ becomes other_info$_1$;

- Any part of user_supplied_info$_1$ in TSP$_1$, if entity B knows or can be determined by entity B;
- Any part of user_supplied_info$_2$ in TSP$_2$, if entity B knows or can be determined by entity B;
- Any part of TTSA_supplied_info$_1$ in TSP$_1$, if entity B knows or can be determined by entity B;
- Any part of TTSA_supplied_info$_2$ in TSP$_2$, if entity B knows or can be determined by entity B.

Any information deleted from the transmitted data shall be included in the appropriate timestamped_data (timestamped_dtata$_1$ and/or timestamped_data$_2$) field when generating and verifying the following equation, so that the following signature can be generated/verified:

- timestamp_signature$_{TTSA1}$ = SIG$_{TTSA1}$ (timestamped_data$_1$);
- timestamp_signature$_{TTSA2}$ = SIG$_{TTSA2}$ (timestamped_data$_2$);
- SIGA (M, TSP$_1$).

d) After receiving D, entity B needs to complete the following:

- If the scheme in 5.4.2.2.2 is used, entity B calculates H' = Hash (M). If H is received in the transmitted TSP$_1$, entity B verifies H' = H; otherwise, entity B shall explicitly set H' = H when verifying timestamp_signature$_{TTSA1}$ and SIG$_A$(M, TSP$_1$);
- Use the public key of TTSA$_1$ to verify timestamp_signature$_{TTSA1}$;
- Use A's public key to verify SIG$_A$ (M, TSP$_1$);
- Use the public key of TTSA$_2$ to verify timestamp_signature$_{TTSA2}$.

The order of execution of the above steps is irrelevant, however, it is necessary to ensure that these verifications are successful.

Through the verification of step 6, on the basis of the attestation of signature generation time obtained by the 5.4.2.2 scheme, additional attestations as follows can be obtained:

- SIG$_A$(M, TSP$_1$) is generated between timestamp$_1$ and timestamp$_2$, meanwhile is included in timestamped_data$_2$ signed by TTSA$_2$;
- Data packet D is assembled after the time indicated by timestamp$_2$.

If there is a mutual agreement between entity B and $TTSA_2$, the following information can be deleted from TSP_2 returned from $TTSA_2$ to entity B:

1) Any part of information of $user\_supplied\_info_2$, as long as the part is known by entity B;

2) Any part of $TTSA\_supplied\_info_2$, as long as the part is known by entity B or can be determined by entity B;

If the above information can be deleted from the transmitted TSP_2, however, when generating and verifying the signature $timestamp\_signature_{TTSA2} = SIG_{TTSA2}(timestamped\_data_2)$, the complete $user\_supplied\_info_2$ and $TTSA\_supplied\_info_2$ information shall still be included in $timestamped\_data_2$.

c) Entity B subsequently:

1) Verify that the transmission part of $user\_supplied\_info_2$ is correct;

2) Use the public key of $TTSA_2$ to verify $timestamp\_signature_{TTSA2}$

Through step 7, on the basis of the attestation of signature generation time as obtained by the scheme 5.4.2.2, it may also be able to obtain the following additional attestation:

- The signature SIG_A (M, TSP_1) is generated between the times marked by $timestamp_1$ and $timestamp_2$;
- Evidence [i.e., M, TSP_1, SIG_A (M, TSP_1) and TSP_2] can be provided to any third party who trusts in TTSAs.

5.4.3 Use the data provided by the verifier to obtain attestation of signature generation time

5.4.3.1 Basic scheme

In addition to using trusted timestamp services, entity A can also use the following methods, to provide attestation of signature timeliness to the verifier (entity B):

a) Combine the fresh value provided by the verifier with other data;

b) Sign the aforementioned combined message.

The following scheme uses the nonce value to help obtain the attestation of the generation time of the signature. The nonce value is a value that changes with time, which is expressed as a string that cannot ignore the time change. For example, the nonce value can consist of the following three parts:

information shall be included in the timestamped_data on which the signature timestamp_signature$_{TTSA}$ and SIG$_A$ (M, Nonce) are generated and verified.

If part of the TTSA_supplied_info information is removed from the TSP from TTSA, the complete TTSA_supplied_info information shall be added to the TSP of assembly D. Any partial information of TTSA_supplied_info can be removed from the TSP transmitted to entity B, as long as the information can be known or determined by entity B. However, the complete TTSA_supplied_info information shall be included in the timestamped_data that the signature timestamp_signature$_{TTSA}$ and SIG$_A$ (M, Nonce) rely on for generation and verification.

e) After receiving D, entity B does the following:

1) Use the public key of entity A to verify SIG$_A$ (M, Nonce);

2) Use the public key of TTSA to verify timestamp_signature$_{TTSA}$.

The order in which the above steps are performed does not matter, however, it is necessary to ensure that the two verifications are successful.

In addition to the attestation as described in 5.4.3.2, the verification of step 5 can also obtain the following attestations:

- SIG$_A$ (M, Nonce) is generated between the Nonce value and the time indicated by TSP.

5.4.3.2.2 Entity B requests a timestamp

Entity B can submit a TSP request to TTSA after receiving the message response D sent by entity A.

Figure 11 describes the program flow. This scheme is similar to the scheme in 5.4.2.2.4, the only difference is the Nonce value sent from entity B to entity A. The Nonce value provided by entity B may include a time source (both entities A and B trust it). In this case, the time marked by the Nonce value and the time marked by the TSP will establish a time interval, during which the signature is generated.

This is an excerpt of the PDF (Some pages are marked off intentionally)

------ The End ------