Sonera SmartTrust Ltd, P.O. Box 425, FIN-00051 SONERA, Elimäenkatu 17-19, Helsinki, Finland
T +358 (0) 2040 63031 F +358 (0) 2040 62730 E info@smarttrust.com www.smarttrust.com

WHITE PAPER
Validation of Electronic Signatures

By Hans Nilsson, SmartTrust & Denis Pinkas, Bull,

Sonera SmartTrust Ltd, P.O. Box 425, FIN-00051 SONERA, Elimäenkatu 17-19, Helsinki, Finland
T +358 (0) 2040 63031 F +358 (0) 2040 62730 E info@smarttrust.com www.smarttrust.com

Table of Contents

Sonera SmartTrust Ltd, P.O. Box 425, FIN-00051 SONERA, Elimäenkatu 17-19, Helsinki, Finland
T +358 (0) 2040 63031 F +358 (0) 2040 62730 E info@smarttrust.com www.smarttrust.com

Purpose of this paper

The purpose of this paper is to serve as a tutorial, and as input, for the discussion of a common validation model for electronic signatures.

The paper describes the technology used in this process, and the records required to support signature validation following the creation of a signature. Particular emphasis is given to the scenario in which the recipient of a document requires evidence that it was signed by an individual who later denies having done so. This security service is known as "non-repudiation".

Within this paper, the term **signer** is used to denote the person who creates and signs an electronic document. The **recipient** denotes the individual for whom the document is intended, to validate, and to act upon.

Validate or verify?

The PKI community has a tendency towards the use of inconsistent terminology. When describing the procedure required to ensure that a digital certificate or signature can be trusted, we usually say "validate the certificate" but "verify the signature». Too often these terms are used interchangeably. We recommend, therefore, a rule to make usage consistent and also to align it both with generally accepted PKI community practice and the dictionary.

Validate should be used when referring to a process intended to establish the soundness or correctness of a data structure against a policy, like a public key certificate or a certification path. **Verify** should be used when referring to a process intended to test or prove the correctness of a value.

In other words, we verify values or items, but we validate against a policy data structures that are composed of or depend on verified items and other validated data structures against the same policy.

Sonera SmartTrust Ltd, P.O. Box 425, FIN-00051 SONERA, Elimäenkatu 17-19, Helsinki, Finland
T +358 (0) 2040 63031 F +358 (0) 2040 62730 E info@smarttrust.com www.smarttrust.com

Validation of signatures

The retrospective validation of digital signatures can be broken down into a number of distinct problems, or regimes, depending upon the use of the digital signature mechanism and the time that has elapsed since the signatures were created.

A digital signature may be used either for peer entity or data authentication purposes, or as an electronic signature for non-repudiation purposes. Usage in the context of peer entity or data authentication is not discussed further in this paper.

Usage of digital signatures in the context of creating electronic signatures for non-repudiation can be seen in several time frames:

Near term: the validation is performed soon after the generation of the signature and while all the certificates and CRLs required to validate the various signatures are current and generally available. At that time the recipient must ensure that they have obtained all the data they will need at a later date for long term validation. If the signer has not supplied all the information needed, the receiver will need to collect it at the time of validation.

Long term: the validation is performed after expiration of the certificate that was used at the time of generation of the signature, and either:

- After one change of the certification key originally used to issue that certificate
- After several changes of the certification keys from the chain of certificates to be used to validate the signature
- After several changes of one of the self-signed certificates used to validate the certification path.

Archival: validation is performed after the time when the cryptography used is no longer secure. At this time it may be possible to derive private keys from public keys, or to generate hash collisions.

This paper describes the technology used, and records needed, to support signature validation in each of the regimes, in the context of non-repudiation.

Sonera SmartTrust Ltd, P.O. Box 425, FIN-00051 SONERA, Elimäenkatu 17-19, Helsinki, Finland
T +358 (0) 2040 63031 F +358 (0) 2040 62730 E info@smarttrust.com www.smarttrust.com

Near term validation of electronic signatures

The need to identify the certificate

It is the responsibility of the CA to make available in repositories all the information needed to validate any signature from any unexpired certificate it has issued. This includes all unexpired certificates and all CRLs on which any current certificate might have appeared.

Some signers will be able to obtain multiple and different certificates containing the same public key from different CAs or even from the same CA.

In order to validate an electronic signature, the recipient must obtain the **certificate indicated by the signer to be used**, and must also ensure that the user alone was able to use the private key associated with that certificate.

The prime benefit of this is that a single private key can be used with all of them, which is an advantage when a smart card is used to protect the private key, as the storage capacity of a smart card is always limited. When several CAs are involved, each certificate may contain a different identity, for example as a private individual or as a company employee. When the same CA is involved, additional attributes such as roles may be added to the identifier in order to highlight explicitly a role adopted by the user. Thus, it is necessary for the recipient to find out which of the signer's certificates that was intended to be used for validation of the signature.

In order to identify unambiguously the certificate which is to be used for the validation of the signature, an **identifier of the certificate from the signer** must be incorporated as part of the signed data. Many current schemes simply add the certificate after the signed data, and thus are vulnerable to substitution attacks.

A further advantage of including the identifier of the certificate is to counter the threat posed by a "false" CA, which could issue a certificate to someone using the public key of another person. If the signer certificate was simply appended to the signature, and thus not protected by the signature, it would be possible to substitute one certificate with another, giving the impression that message had been signed by someone else.

There is another technique available to counter this threat, although it is not as reliable as the technique of including the identifier of the certificate in the signed data. This technique mandates all CAs to perform a Proof Of Possession (POP) of the private key at the time of registration. The problem with this technique, however, is that it does not provide any guarantee at the time of validation, and proof "after the event" is obtainable only if the CA has kept the POP in an audit trail.

It should be noted that the technique of including the identifier of the certificate in the signed data also applies to a situation where the CA key is actually revoked, or even compromised. Otherwise, in the event of a CA certificate revocation, all signers might attempt immediately to repudiate their signatures, using the following line of argument:

> The recipient himself created and signed the documents at an earlier date with his own private key, and then sent them for time-stamping, in the hope that the CA key might be compromised later.
>
> When this eventuality did indeed occur, compromising the CA key, the recipient used the compromised key to create a false certificate in the name of the signer, but with a key pair known to the recipient.
>
> The recipient can oppose this line of argument by showing that the signer's certificate, as indicated in the signed data, actually existed and was not revoked at the time of signing.

Note: There are other ways to support roles, such as the use of Attribute Certificates. However, to keep this paper reasonably short and focused, they are not discussed further here.

The need for the signing policy

The recipient may extract the identifier (name) of the signer from the certificate. In order to validate that certificate he must also be in possession of an appropriate self-signed certificate from a CA, obtained beforehand in a trusted manner. That self-signed certificate will contain, in particular, the name of a CA trusted to issue certificates containing given forms of identifiers (names), a certification key, and a validity period. Since multiple self-signed certificates from the same CA, used for different certificate policies and/or for different naming constraints, may exist, it will be necessary to know unambiguously which one was intended to be selected by the signer, and thus to be used by the recipient.

In order to identify unambiguously the non-repudiation policy to be used to validate the signature, an **identifier of the signing policy** from the signer must be part of the signed data. That signing policy, among other information, indicates to the recipient the self-signed certificates to be used.

The need for time stamping

One important property in the context of non-repudiation is that a signature, having been found once to be valid, shall continue to be so, for the same data, months or years later. .

In order to perform the validation, the certificate used by the signer at the time of the signature must be obtained, and its validity at the time of the signature proven. It could be the case that a certificate was valid at the time of the signature but revoked some time later. In this event, evidence must be provided that the document was signed before the signing key was revoked.

Time-stamping by a Time Stamping Authority (TSA) can provide such evidence. A time stamp is obtained by sending the hash value of the given data to the TSA. The returned «time-stamp» is a signed document that contains the hash value, the identity of the TSA, and the time of stamping. This proves that the given data existed before the time of stamping.

If the hash of a digital signature is sent to a TSA and is time-stamped before the revocation of the private key used to generate that signature, evidence will be provided that the digital signature was formed before the revocation of the public key certificate.

If a recipient wants to hold a valid electronic signature he will have to ensure that he has obtained a valid time stamp for it, before that key (and any key involved in the validation) is revoked. The sooner after the signature time, the better.

It is important to note that signatures may be generated "off-line" and time-stamped at a later time by anyone, for example by the signer or any recipient interested in the value of the signature. The time stamp can thus be provided by the signer together with the signed document, or obtained by the recipient following receipt of the signed document.

If the recipient is unable to show a time stamp of the signed document, the signer may try to repudiate his signature, and thus withdraw the signed document. This can be done in the following way:

The signer revokes his certificate as soon as he changes his mind. The certificate will then be entered into the CRL.

The signer then claims that someone else has created the signature after the certificate was revoked, and that the recipient has not checked the CRL. Alternatively, the signer can claim that the recipient had access to the private key and created the signature himself.

In order for a signed message to be valid under a signing policy for non-repudiation purposes, and in case the certificate used for signing is revoked, the recipient needs to obtain a **time stamp from a TSA** before the date of revocation of the signer's certificate.

Summary of evidence required for non-repudiation

If the signer later repudiates (denies) a signature, the recipient will need to produce the following records as evidence in court:

- The description of the signing policy.
- The signed document including the appropriate data (signing policy, type of event or action, signing time, identifier of the signer's certificate and attribute certificates, if any).
- The certificate containing the signer's public key.
- An attribute certificate, if any is necessary according to the signing policy.
- A time stamp from an appropriate TSA covering the signed document.
- A valid chain of unrevoked CA certificates (i.e. cross-certificates) at the time of the signature, up to a trust point defined in the signing policy.

And one of the following:

- A CRL from the time of signing, where the certificate is not included.
- A response carried by OCSP (Online Certificate Status Protocol) which shows that the certificate was not revoked at the time of signing.

Thus, all this information needs to be collected, saved and stored by the recipient the first time he receives and validates the signature in case there is a need to prove the validity of the electronic signature in a court a few years later. However, as will be explained later, this procedure will not suffice where one or more of the certification keys needed to validate the certification path might have been compromised at the time of validation.

Sonera SmartTrust Ltd, P.O. Box 425, FIN-00051 SONERA, Elimäenkatu 17-19, Helsinki, Finland
T +358 (0) 2040 63031 F +358 (0) 2040 62730 E info@smarttrust.com www.smarttrust.com

Long term validation of electronic signatures

Long term validation begins when the certificate needed to validate a signature has expired at the time of validation, and when either:

- The certification key from the CA from the signer has changed.
- The certification key from a root CA has expired at the time of validation.
- One or more of the cross-certificates has expired.

It is possible that changes to the certification key originally used to issue certificates may have occurred. Some CAs could have ceased activities, transferring them to another CA. Some CA keys may have become compromised.
In order to perform the validation, the recipient must use a certification path valid at the time of the signature.

All certificates, cross-certificates, CRLs or OCSP responses must be time-stamped, in order to protect against the event of a certification key becoming compromised at a later date.

At the time of the first validation of the signature, the recipient will need to gather the various cross-certificates together with either current corresponding CRLs, where the cross-certificates are not included, or OCSP responses, which show that each cross-certificate has not been revoked, to perform the validation.

The public key certificate from the signer and the attribute certificate, if any is used, as well as each component from the certification path, should be time-stamped. Each piece of information could be individually time-stamped. Since, in any case, a time stamp must be provided either by the signer or by a recipient, the time stamp can include the whole certificate from the signer, and the whole attribute certificate from the signer, if any is used.

The whole certification path may be time-stamped separately so that the same data can be used for validating several electronic signatures.

This process leads to the following situation:

The recipient needs to obtain and save, close enough to the signing time, the public key certificate from the signer and the attribute certificate (if any is used by the signer). This will make the signed message valid under a **long term** signing policy, and offer protection in the event of the certification key of the certification authority and the attribute authority of the signer becoming compromised.

The recipient needs to obtain and save, close enough to the signing time, a time stamp of a valid certification chain, together with either current CRLs, where each cross-certificate is not included, or OCSP responses which shows that each cross-certificate has not been revoked. This will make the signed message valid under a **long term** signing policy, and protect it from the effects of any certification key from the certification chain becoming compromised.

Archived electronic signatures

Archival storage is indefinite, and may extend long past the date where the cryptography used to sign documents is still secure. Future advances in computing and cryptography are likely to make possible the generation of private keys from public keys where this is unfeasible today.

Current cryptography soon broken

In the event of it appearing possible to break a given cryptographic algorithm, but where the availability of new algorithms (or old ones with greater key lengths) is anticipated, a sequence of time stamps will protect against forgery. Each time stamp will need to be affixed to the whole evidence before the signing key is compromised or the algorithm is broken. TSAs should have long keys (for example 2048 bits).

It should be noted that mandatory time stamping, which must be obtained from either the signer or the recipient shortly after the signature, protects not only in the event of the signing key becoming compromised, but also against the breaking of the signature algorithm used by the signer. In this way, the signer's keys not do need to be long and their resistance must only exceed the validity period of the certificate.

If the certificate is valid for two years, its resistance should be more than two years but does not need to be twenty years, even if the signature must be validated twenty years later.

Current hash function soon broken

In event of it becoming possible to create hash collisions for a given hash algorithm, the whole signed document, certification path and CRLs or OCSP responses will need to be time-stamped using a stronger hash algorithm before the breaking of the hash function.

In summary: if it was impossible to forge the signature at the time the signed document was time-stamped by a trusted TSA, the signature is valid and can be relied upon.

However, if the recipient re-signs his archived material purely by **himself**, without using a trusted TSA, the material may be used only for authentication purposes, and not as evidence in court for non-repudiation.

Appendix One

Frequently Asked Question (FAQ)

Question: Since a time stamp from a TSA can be trusted, why can't a time stamp covering the signed data, excluding the signature from the signer, replace the time indicated by the signer?

Answer: In such a case, it would be possible to mount the following planned attack: an attacker identifies his victim in advance and prepares a message that would apply to the victim, requesting a time stamp covering the prepared message. A few weeks or months later, the attacker manages to get the private key of the victim and then signs the time-stamped message. The signed message from the attacker would thus appear to have been signed during the validity period of the certificate and would be declared valid.

The signing time as indicated by the signer

As described above, the time stamp serves as evidence for the recipient that a signature was applied before the date of a possible revocation or expiration of the signer's certificate. It does not indicate the signing time.

For certain applications, however, to show when the event or action was recognized by the signer as being valid, the signing time as indicated by the signer may be important. In order to achieve this, with more or less accuracy, two different approaches may be used, as described below.

Let us assume that the signer includes in his signature a signing time (T1). As we have discussed, a good insurance for the recipient against the possible revocation of the signing key is to have the signature countersigned by a signing time (T2), obtained from a TSA as soon as possible after the signature, and to establish the validity of the signing, i.e. that it is neither revoked nor expired at that time. Let us also assume that the recipient accepts this insurance.

Now let us take the position of a recipient who receives a signature containing (T1), counter-signed by a TSA at (T2). It is possible to make a case for (T1) as a trusted or untrusted time, and for (T1) being greater or less than (T2). Let's look first at the less interesting case: (T1 > T2).

T1 > T2. If (T1) is trusted (for example obtained from a TSA), this situation simply cannot occur. If (T1) is untrusted, this means that the signer may have post-dated his signature.
If such a signature is verified at a time T < T1, the signature will be declared as invalid. This is similar to a check effectively signed on January 30th but with the date of February 20th, and which cannot be paid before February 20th. If a recipient accepts such a signature, he runs the risk of not being paid if the signer's certificate is revoked before that date.

If the signature is verified at a time T > T1, and it can be proven that the signing key was still valid at (T1), then the signature may now be declared as valid. Hence a new time stamp with a time greater than (T1) must be obtained by the recipient, from a valid TSA, to form a valid signature.

T1 < T2. If (T1) is untrusted, the best that can be said is that the signature was made before (T2). If (T1) is trusted (i.e. obtained from a TSA), it may be stated that the signature was made after (T1) and before (T2). [However, the signer may capture (T1) well in advance, so (T1) is not an accurate minor margin for the signing time.] Additionally, the recipient may remove the time stamp containing (T2) and replace it with another one obtained at a later date. In order to prevent the replacement of the TSA time stamp, another signature made by the signer to replace the earlier signature, together

with the time stamp, would be necessary. Using this procedure, the signing policy would include a requirement for a double signature from the signer and a real-time interaction with the TSA. This would make off-line signatures impossible.

The alternative, and recommendable approach which follows is described in annex D of ISO 10181-4 (Non Repudiation Framework). It has the advantage of allowing off-line signatures and mandating only one asynchronous interaction with a TSA. Such an interaction may be made at will by the signer or any recipient. Only two signatures are needed: one from the signer and one from the TSA. The rationale of this approach is to introduce an additional condition in the signing policy, according to which the electronic signature will be valid only if (T2 - T1) is smaller than a maximum indicated in the signing policy.

It is now possible to say that the signature was made after (T2 - maximum) and before (T2). Hence the accuracy of the signing time is equal to the maximum indicated in the signing policy. In practice (T1) and (T2) should be «close enough», for example a few minutes, hours or even days apart, depending on the nature or the value of the transaction.

Under the terms of this signing policy, a recipient holding a valid electronic signature would be unable to change the time stamp at will, as he would then get an invalid electronic signature.

The signing time then serves as a reminder to the recipient of the latest date at which a time stamp should be obtained, according to the signing policy. It also provides the recipient with an assurance that he will not encounter problems in the event of a later revocation of the signer's key.

In order for the signing time to be valid, as indicated by the signer, the **signing time** and the **TSA time** as indicated in the time-stamp from a valid TSA must be "**close enough**", according to a limit indicated in the signing policy.

Question: How can the signer's own indication of time be of any value in the signed data, since it is not a trusted time?

Sonera SmartTrust Ltd, P.O. Box 425, FIN-00051 SONERA, Elimäenkatu 17-19, Helsinki, Finland
T +358 (0) 2040 63031 F +358 (0) 2040 62730 E info@smarttrust.com www.smarttrust.com

Answer: The signer's own time is neither accurate nor trusted, but its insecurity is limited by the signing policy to the time period allowed to get the time stamp. Pre-dating at too great a distance from the current time is impossible; however, post-dating is possible.

Question: Why not use a trusted time in addition to the signing time (T1)?
Answer: It can already be said that the signature was made after (T1) and before (T2) with T1 -T2 < maximum, according to the signing policy. There is no additional value in adding a trusted time (T1), since accuracy is not improved. The requirement for a trusted time would also make off-line signatures impossible.