

## Long Term Digital Signatures

### What is Long Term Signature Validation?

When a digital signature certificate is used for authentication purposes, the signature created during the authentication event is only useful for a short period of time. Therefore, it is a straightforward matter to validate the signature certificate with revocation checking, because the revocation information available from the Certificate Revocation List (CRL) distribution points or the Online Certificate Status Protocol (OCSP) responders is applicable for the certificate chain.

However, in the case of digital signatures, there is often a need for maintaining archives of signed documents long after the signing event took place. When validating the digital signatures on such documents, it is a common desire to use the date of signature when performing certificate validity and revocation checks. In such a situation, the revocation information available from the CRL distribution points or OCSP responders is not applicable to the certificate chain, because the revocation information was issued with a timestamp after the time used to validate the certificate chain.

Long Term Signature Validation approaches enable the validation of archived digital signatures through the use of trusted time sources and archived revocation information. This allows relying parties, such as auditors and investigators, to verify whether a digital signature was valid at the time it was created, rather than relying on the current state of the certificate that was used to create it.

### Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Requirements for Long Term Validation .....</b> | <b>2</b> |
| 1.1      | Trusted Source of Signature Time .....             | 2        |
| 1.2      | Revocation Information.....                        | 2        |
| <b>2</b> | <b>The Validation Process .....</b>                | <b>3</b> |
| 2.1      | Validate the Digital Signature .....               | 3        |
| 2.2      | Establishing the Signature Time .....              | 3        |
| 2.3      | Validate the Certificate Chain.....                | 4        |
| <b>3</b> | <b>The Expiration Problem .....</b>                | <b>4</b> |
| 3.1      | Signer Certificate Expiration .....                | 4        |
| 3.2      | Timestamp Server Certificate Expiration.....       | 5        |
| 3.3      | Root Certificate Authority Expiration .....        | 6        |
| 3.4      | Preserving Timestamps with Countersignatures.....  | 7        |
| <b>4</b> | <b>References .....</b>                            | <b>7</b> |
| <b>5</b> | <b>For More Information.....</b>                   | <b>8</b> |
| 5.1      | About Gemini Security Solutions.....               | 8        |

# 1 Requirements for Long Term Validation

There are two primary requirements for the long term validation of a digital signature: a trusted method of establishing the signature time, and the revocation information that was current at the time the signature was created.

## 1.1 *Trusted Source of Signature Time*

Ideally, RFC3161 timestamps should be used as the trusted source of time, as they are cryptographically verifiable by any client. Such a timestamp is basically a token containing the hash of the document and a timestamp, signed by a server that has been issued an appropriate timestamp signature certificate. However, the use of a trusted time server has the drawback of requiring a countersignature before the timestamp server's certificate expires, and not every PKI deployment has a trusted time server available.

Alternatively, the PKCS-9 signature time attribute may be used as the source of the signature time, although this is undesirable. The PKCS-9 signature time is specified by the signer during the signature event, and therefore does not carry a high level of assurance. A simple example of how this can be insecure is the event that a signature key is compromised. While the signature certificate may be revoked after the compromise, the individual who possesses the compromised key is able to create a signature with a PKCS-9 signature time that appears to be chronologically before the time of revocation. When the signature is validated using the PKCS-9 signature time, it will appear valid even though the signature was actually executed after the revocation of the certificate.

If an RFC3161 compliant timestamp server is not an option within an organization, then some other architectural solution may be devised, such as storing archived documents along with a timestamp that is created by the archival server, and enforcing the accuracy and security of the data store using policy and procedural controls. While this is more reliable than using the PKCS-9 signature time, it is still an inferior solution to RFC3161 timestamps, because users outside of the organization may not trust that the policy and procedural controls on the data store are sufficient, and external access to such a data source would be required for users outside of the organization to perform long term validation.

## 1.2 *Revocation Information*

The second requirement for long term validation is the availability of revocation information that was current at the signature time. The simplest method to achieve this is for the application that executes the signature to include the CRLs or OCSP responses within the document, whether as part of the underlying document structure, or as a signed or unsigned attribute in a PKCS-7 signature. However, this depends on the signature execution taking place in an online environment. Additionally, the application that executes the signature must be correctly configured to include the revocation information, if the application even has this capability at all. Finally, the relying party must also have software capable of interpreting the document and signature to extract the embedded revocation information.

Alternatively, an organization may opt to maintain an online archive of revocation information issued within the PKI. This archive is usually populated with Certificate Revocation Lists; OCSP responses are tailored specifically to the revocation request and signed on the fly, and therefore they are better suited for real-time revocation checking. This approach is virtually guaranteed not to be compatible out-of-the-box with any third party software, because there is no standard governing how this revocation information should be stored or made available. However, this method has the distinct advantage of allowing a developer to create a signature validation module for almost any type of document without

having to rely on the individual signing the document to take any special steps to ensure the signature will be verifiable.

## 2 The Validation Process

Long term validation is not much different from real-time validation of a digital signature. The most important difference is the establishment of the signature time, and the precautions that must be taken when checking revocation and certificate time-validity based on this signature time.

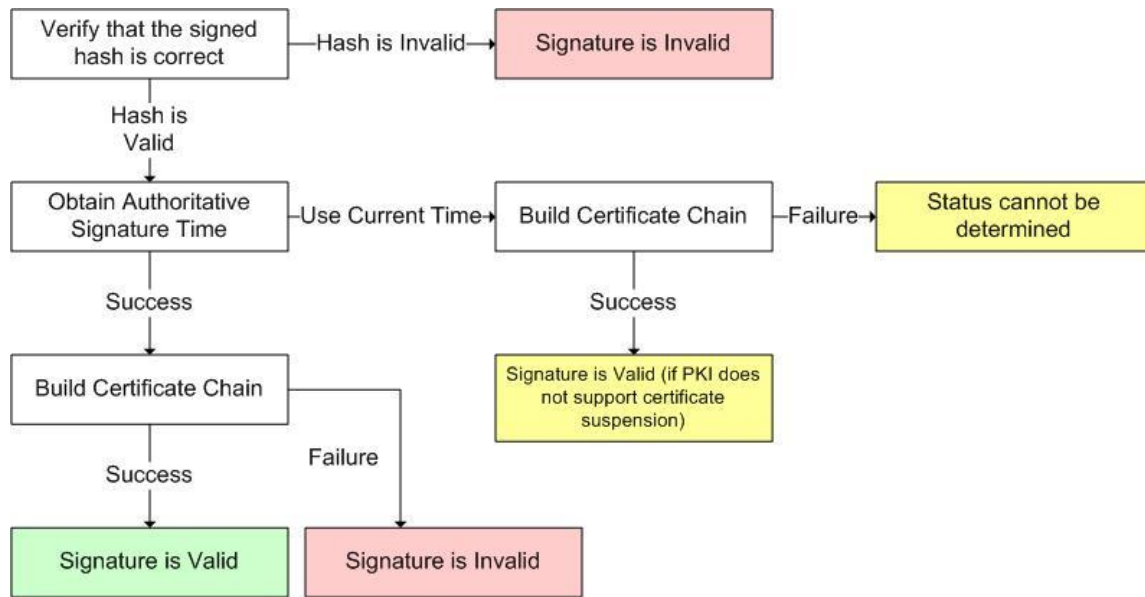


Figure – Long Term Signature Validation Process

### 2.1 Validate the Digital Signature

The first step in any signature validation is to verify that the signed hash of the document is correct. If the user's signature on the document is structurally invalid, then there is no need to perform any certificate validation.

### 2.2 Establishing the Signature Time

As discussed in section , the signature time used during the validation process depends on the method used to create the signature. If a valid RFC3161 timestamp exists on the document, then this should be the first option for establishing the signature time, as a secure timestamp may be cryptographically verified. If secure timestamps are not an option, then the source of the signature time is up to the relying party.

If it is not possible to assess the signature time from any source, then a signature may be validated using the current time as a last resort. However, the only meaningful result of this process is a result determining that the digital signature is still valid at the current time. If the certificate used to create the user signature is still time-valid and not revoked, then in most cases it can be assumed that the

certificate was valid when the signature was created<sup>1</sup>. However, if the signer's certificate is revoked at the current time, this does not necessarily indicate that it was revoked at the time of signature<sup>2</sup>, and therefore the negative result of this validation does not indicate anything about the long term validity of the signature.

### ***2.3 Validate the Certificate Chain***

The validation of the certificate chain is a slightly more complicated issue during long term validation, as compared to a real-time validation. The first step in this process is to create a certificate chain from a trusted root certificate to the signature certificate in which all certificates are time-valid at the signature time. Once this is done, any digital signatures on the CRLs or OCSP responses must be validated, and then the revocation status of the certificate chain is checked. If the certificate was time-valid, trusted and not revoked, then the certificate is determined to have been valid at the time of the signature. The complications arise when the signature ages to the point where the various signing entities involved in the signature's creation begin to expire. See section for more details about the problems caused by certificate expiration.

## **3 The Expiration Problem**

Certificates are created with a finite lifespan for several reasons, the most germane to long term validation being the fact that computers get more powerful, but the private key associated with any single certificate does not. A 1024 bit digital signature key may be difficult to crack at the present, but as hardware improves, the time required to break a private key will continue to decrease. This is true of any key length; the technology will always catch up such that the cryptographic key eventually will be susceptible to compromise.

Because of the assumption that the ability to compromise today's keys will eventually exist, certificate expiration becomes an issue that can't simply be glossed over – the expiration times of user signature certificates, root certificate authority certificates, and especially timestamp certificates can't simply be ignored because the signature time used during the validation is in the past.

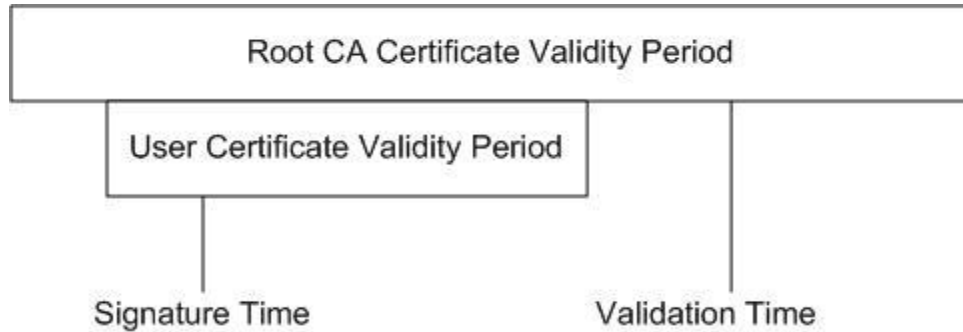
### ***3.1 Signer Certificate Expiration***

When the user's digital signature certificate expires, it can be assumed that the key is no longer trustworthy at all. This means that in the absence of an authoritative time source attached to a signature, nothing that this certificate has ever signed may be considered valid, even if the untrusted signature time is within the certificate's validity period. Once the user's certificate expires, it can be assumed that the private key is susceptible to compromise, and therefore if the signature time was specified by the user, such as in the case of a PKCS-9 signature time, it is essentially meaningless.

---

<sup>1</sup> If the PKI architecture supports suspended certificates, or any process whereby certificates may be revoked and subsequently un-revoked, then this assertion is invalid.

<sup>2</sup> While Certificate Revocation Lists and OCSP responses indicate a time of revocation, there is no assurance that this timestamp corresponds to the time the certificate was actually revoked. The only trustworthy timestamps on a CRL or OCSP response are the issue and expiration times.



**Figure – Validating a signature without an authoritative source of signature time becomes impossible once the user certificate expires.**

If there is an authoritative source of time associated with the certificate, especially if it is a cryptographically-based solution, then as long as the user’s certificate was time-valid at the authoritative time, the fact that the signer certificate is currently expired is irrelevant.



**Figure – Validating a signature when the timestamp is valid, but the signature certificate is expired. The expiration of the signature certificate does not matter, because the authoritative timestamp indicates a time when the certificate was valid.**

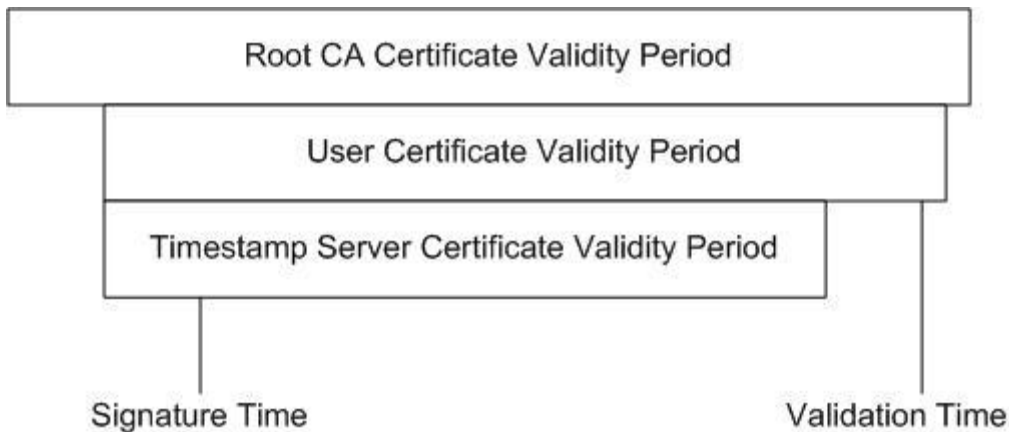
### ***3.2 Timestamp Server Certificate Expiration***

If RFC3161 timestamps are used as the authoritative time source, then the expiration of the timestamp server certificate presents a problem for long term validation. When the timestamp server’s certificate expires, then any timestamp ever issued by the timestamp server cannot be trusted by simply validating the signature on the timestamp token. There are two ways to handle the problem posed by the expiration of the timestamp server certificate.

First, the current time could be used to attempt to validate the user’s signature, rather than the timestamp’s time. However, this can only attempt to show that the signature was valid; if the signature is found to be invalid using the current time, it does not necessarily follow that the signature was invalid at the time it was created. See section for more information.



**Figure – Because the timestamp server’s certificate is expired at the current time, the timestamp cannot be trusted, and the certificate is evaluated using the current time. The signature cert is currently expired, but this does not necessarily mean that the signature was invalid at the time of signature. It is impossible to determine if this signature is valid.**



**Figure – Because the timestamp server’s certificate is expired at the current time, the timestamp cannot be trusted, and the certificate is evaluated using the current time. This example shows a signature certificate that is valid at the current time, indicating that the signature is valid. See section for more information regarding cases when this is not true.**

Alternatively, the timestamp is still interpreted as valid if the document contains a valid countersignature created before the expiration of the timestamp certificate. If the countersignature is found to be valid at the current time, then it follows that the document integrity has been preserved since the time the countersignature was applied, and therefore the timestamp was valid before the expiration of the timestamp certificate and can be trusted. For more information about countersignatures, see section .

### **3.3 Root Certificate Authority Expiration**

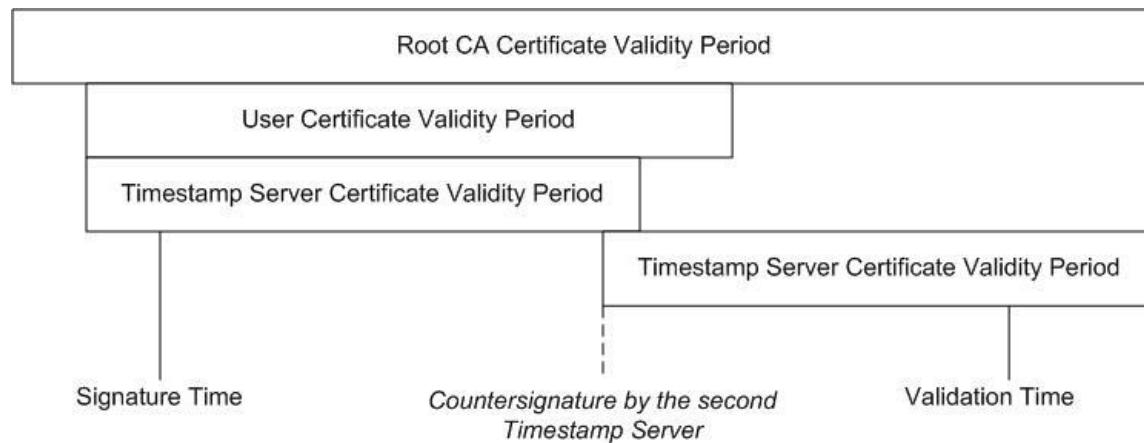
Similar to the concerns expressed in section , the expiration of the root certificate authority certificate exposes a significant number of weaknesses in the digital signature. While the signer certificate expiration may call into question the authenticity of the user signature only, the expiration of the root certificate calls into question the validity of the entire certificate chain that is built to the user’s

signature certificate, as well as the validity of the signatures on any Certificate Revocation Lists or OCSP Responses, as these are issued within the PKI that is based on that root certificate.

However, also in parallel with section , an authoritative source of time can mitigate these concerns. If an RFC3161 compliant timestamp server is used as the authoritative time source, however, the signed timestamp tokens will not be trusted if the timestamp server's certificate was issued by the expired root certificate. In this case, the signature must be countersigned before the expiration of both the root certificate and timestamp server certificate, as described in section .

### 3.4 Preserving Timestamps with Countersignatures

As the timestamp server certificate nears expiration, it becomes necessary to countersign the digital signature with an additional timestamp that extends the period over which the signature may be validated. By countersigning the signature with a timestamp issued by a server with a newer certificate, the integrity of the initial timestamp may be preserved after the expiration of the initial timestamp server certificate. This process may be repeated indefinitely as the outermost timestamp nears expiration.



**Figure – Although the validation takes place after the expiration of the original timestamp server certificate, the document was countersigned by a new timestamp server, which preserves the integrity of the initial timestamp.**

## 4 References

- Adams et al. **RFC 3161**. [Online] August, 2001. URL <http://www.ietf.org/rfc/rfc3161.txt>.
- Cooper et al. **RFC 4158**. [Online] September, 2005. URL <http://www.ietf.org/rfc/rfc4158.txt>.
- Pinkas et al. **RFC 3126**. [Online] September, 2001. URL <http://rfc.net/rfc3126.html>.

## 5 For More Information

If you have any questions about the contents of this document, or are interested in learning more about public key infrastructure (PKI), digital signatures, or long term signature validation, please contact:

Peter Hesse ([pmhesse@geminisecurity.com](mailto:pmhesse@geminisecurity.com))

Walter Turnes ([wturnes@geminisecurity.com](mailto:wturnes@geminisecurity.com))

Gemini Security Solutions, Inc.

4451 Brookfield Corporate Dr. Suite 200

Chantilly, VA 20151

703-378-5808

<http://geminisecurity.com>

### ***5.1 About Gemini Security Solutions***

Gemini Security Solutions, Inc. provides impartial information security consulting services that ensure the confidentiality, integrity, and availability of critical business information and resources. Our value is centered on our ability to deliver the right expertise and the right experience, at the right time.

Founded in 2001, and with decades of experience, our certified team of experts assists organizations to assess and deliver security solutions in compliance with business strategy, policy and regulatory requirements. Our services include risk analysis and management, regulatory compliance, identity access and management, operations security, and security software development.

Gemini Security Solutions is focused on providing big business experience with a small business approach. Working with you every step of the way, Gemini Security Solutions enables secure business operations with vigor, agility and a commitment to excellence.