OCSP Algorithm Agility

draft-hallambaker-ocspagility-01

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on January 3, 2009.

Abstract

The behavior of an OCSP server is specified for cases in which the OCSP server is capable of supporting more than one signature algorithm.
Table of Contents

1. Introduction ............................................. 3
   1.1. Requirements Language .............................. 3
2. OCSP Algorithm Agility Requirements ..................... 3
3. Client Indication of Preferred Signature Algorithms .... 4
4. Responder Signature Algorithm Selection ................ 4
5. Acknowledgements ......................................... 5
6. Security Considerations ................................... 5
   6.1. Use of insecure algorithms ......................... 5
   6.2. Man in the Middle Downgrade Attack ................. 6
7. Normative References ...................................... 6
Author’s Address ............................................... 6
Intellectual Property and Copyright Statements ............. 7
1. Introduction

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. OCSP Algorithm Agility Requirements

OCSP RFC 2560 [RFC2560] defines a protocol for obtaining certificate status information from an online service. A particular OCSP protocol may or may not be provided by the CA that issued the certificate whose status is being queried and may or may provide a realtime indication of the certificate status or a time delayed status indication.

RFC 2560 [RFC2560] requires the implementation but not the use of a set of mandatory cryptographic algorithms. The OCSP protocol provides a means for an OCSP responder to indicate the signature and digest algorithms used in a response but not how that algorithm is to be chosen or how the client might influence the choice of signature algorithm.

In practice this approach has proved insufficient to ensure interoperability of implementations, in particular when a transition from the use of one cryptographic algorithm to another is in progress. The mandatory cryptographic algorithms are the only ones that a responder can expect a client to implement according to the specification but this is a clearly unacceptable choice if a different signing algorithm has been chosen for the certificate whose status is being queried precisely because the mandatory algorithm is not acceptably secure.

Although security concerns may lead to the use of a different signing algorithm in a certificate there is little security advantage to be used from using a stronger signing algorithm in an OCSP response than in the certificate whose status is being reported. A possible exception being in the case where a certificate issuer determines that a certificate signing algorithm has become regarded as unacceptably secure during the period of validity for a certificate and thus requires a means of revoking the certificates that does not depend on the compromised algorithm.

While use of a non-mandated signing algorithm is often motivated by security concerns, this is not always the case: performance, key and signature value sizes are also possible motivations.
In most cases an OCSP reponder may assume that the client is capable of verifying any signature algorithm that is used in either the certificate whose status is being verified or the certificate of the OCSP responder.

There are however circumstances in which it may be desirable to use a different form of signature or demonstration of authenticity for the OCSP response. In a wireless application, the use of a signature algorithm that provides a compact signature value such as DSA or ECC might be desirable.

The introduction of delegated certificate path discovery and validation (e.g. XKMS, SCVP) gives rise to further exceptions. A certificate path discovery service may be required to establish a certificate path for a public key algorithm that the discovery service itself does not support. In many embedded applications a relying party application may use an OCSP service to verify the current status of a certificate that has previously (but not necessarily contemporaneously) been subject to path validation by another component in the system.

3. Client Indication of Preferred Signature Algorithms

A client MAY declare a preferred set of algorithms in a request using the preferred signature algorithm extension.

\[
\text{id-pkix-ocsp-preferred-signature-algorithms OBJECT IDENTIFIER ::= { id-pkix-ocsp x \}} PreferredSignatureAlgorithms ::= SEQUENCE \{
\}
\]

If a set of preferred signature algorithms is declared the client MUST support each of the specified algorithms.

If a set of preferred algorithms is declared the OCSP responder SHOULD use one of the specified signing algorithms.

4. Responder Signature Algorithm Selection

RFC 2560 [RFC2560] does not specify a mechanism for deciding the signature algorithm to be used in an OCSP response. As previously noted this does not provide a sufficient degree of certainty as to the algorithm selected to guarantee interoperability.

A responder MAY maximize the potential for ensuring interoperability by selecting the OCSP signature algorithm using the following order of precedence where the first method has the highest precedence:
1. Using an algorithm specified as a preferred signing algorithm in the client request.

2. Using the signing algorithm used to sign the CertID specified in the query.

3. Using the signing algorithm used to sign a CRL issued by the certificate issuer providing status information for the certificate specified by CertID.

4. Using a signature algorithm that has been advertised as being the default signature algorithm for the signing service using an out of band mechanism

5. Using a mandatory signing algorithm specified for the version of the OCSP protocol in use.

A responder SHOULD always apply the lowest numbered selection mechanism that is known, supported and meets the responder’s criteria for cryptographic algorithm strength.

5. Acknowledgements

The author acknowledges the helpful comments made on earlier drafts of this work by Santosh Chokhani and Stefan Santesson

6. Security Considerations

The mechanism used to choose the response signing algorithm MUST be considered to be sufficiently secure against cryptanalytic attack for the intended application.

In most applications it is sufficient for the signing algorithm to be at least as secure as the signing algorithm used to sign the original certificate whose status is being queried. This criteria may not hold in long term archival applications however in which the status of a certificate is being queried for a date in the distant past, long after the signing algorithm has ceased being considered trustworthy.

6.1. Use of insecure algorithms

The security of the signing algorithm used by the responder MUST take precedence over all other considerations. A responder MUST NOT generate a signature for a signing mechanism that is considered unacceptably insecure regardless of the other circumstances.
In archival applications it is quite possible that an OCSP responder might be asked to report the validity of a certificate on a date in the distant past. Such a certificate might employ a signing method that is no longer considered acceptably secure. In such circumstances the responder MUST NOT generate a signature for a signing mechanism that is considered unacceptably insecure.

A client MUST accept any signing algorithm in a response that it specified as a preferred signing algorithm in the request. It follows therefore that a client MUST NOT specify as a preferred signing algorithm any signing algorithm that is either not supported or not considered acceptably secure.

6.2. Man in the Middle Downgrade Attack

The mechanism to support client indication of preferred signature algorithms is not protected against a man in the middle downgrade attack. This constraint is not considered to be a significant security concern as the client MUST NOT accept any signing algorithm that does not meet its own criteria for acceptable cryptographic security no matter what mechanism is used to determine the signing algorithm of the response.

7. Normative References


Author’s Address

Phillip Hallam-Baker
VeriSign Inc

Email: pbaker@verisign.com