IP Security Maintenance and Extensions (ipsecme)
Internet-Draft
Updates: RFC 5996 (if approved)
Intended status: Standards Track
Expires: June 7, 2013

T. Kivinen
INSIDE Secure

December 4, 2012

# Signature Authentication in IKEv2
# draft-kivinen-ipsecme-signature-auth-00.txt

## Abstract

The Internet Key Exchange Version 2 (IKEv2) protocol has limited support for the Elliptic Curve Digital Signature Algorithm (ECDSA). The current support only includes support for three Elliptic Curve groups, and there is fixed hash algorithm tied to each curve. This document generalizes the IKEv2 signature support so it can support any signature method supported by the PKIX and also adds signature hash algorithm negotiation. This generic mechanism is not limited to ECDSA, but can also be used with other signature algorithms.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on June 7, 2013.

## Copyright Notice

Table of Contents

## 1. Introduction

This document adds support for new IKEv2 ([RFC5996]) authentication method to support all kinds of signature methods. The current signature based authentication methods in the IKEv2 are per algorithm, i.e. there is one for RSA Digital signatures, one for DSS Digital Signatures (using SHA-1) and three for different ECDSA curves each tied to exactly one hash algorithm. This design starts to be cumbersome when more ECDSA groups are added, as each of them would require new authentication method and as with ECDSA there is no way to extract the hash algorithm from the signature, each ECDSA algorithm would need to come with fixed hash algorithm tied to it.

With the SHA-3 definitions coming out, it is seen that it might be possible that in the future the signature methods are used with SHA-3 also, not only SHA-2. This means new mechanism for negotiating the hash algorithm for the signature algorithms is needed.

The RSA Digital Signatures format in the IKEv2 is specified to use RSASSA-PKCS1-v1_5, but there has been some discussions that newer padding methods should also be possible (See section 5 of [RFC4055]). The DSS Digital Signatures format in the IKEv2 is specified to always use SHA-1, which limits the security of that, meaning there is no point of using long keys with it.

This documents specifies two things, one is one new authentication method, which includes the enough information inside the Authentication payload data that the signature hash algorithm can be extracted from there (see Section 3). The another thing is to add indication of supported signature hash algorithms by the peer (see Section 4). This allows peer to know which hash algorithms are supported by the other end and use one of them (provided one is allowed by policy). There is no need to actually negotiate one common hash algorithm, as different hash algorithms can be used in different directions if needed.

The new digital signature method needs to be flexible enough to include all current signature methods (ECDSA, ECGDSA, RSASSA-PSS, ElGamal, etc), and also allow adding new things in the future. For this the signature algorithm is specified by and OID which specifies both the signature and hash algorithms (i.e. sha1WithRSAEncryption, dsa-with-sha1, dsa-with-sha256, ecdsa-with-SHA1, ecdsa-with-SHA256 etc), meaning any signature and hash algorithm specified by an OID can be used.

## 2. Terminology

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 [RFC2119].

## 3. Authentication Payload

This document specifies new "Digital Signature" authentication method. This method can be used with any types of signatures. As the authentication methods are not negotiated in the IKEv2, the peer is only allowed to use this authentication method if the SIGNATURE_HASH_ALGORITHMS Notify Payload has been sent and received.

In this newly defined authentication method, the Authentication Data field inside the Authentication Payload does not include only the signature value, but instead the signature value is prefixed with the algorithm identification OID. This OID identifies both the signature algorithm and the hash used when calculating the signature. To make implementations easier, the OID is prefixed by the 8-bit length field. This length field allows simple implementations to be able to know the length of the OID, so they can use it as binary blob which is compared against the known OIDs, i.e. they do not need to be able to parse or generate ASN.1 DER OIDs (Note, that the 2nd byte of the ASN.1 DER OID, also includes the length, but adding it outside makes things bit easier for implementors).

The OIDs used here are the same OIDs which are used inside the AlgorithmIdentifier of the PKIX (Section 4.1.1.2 of [RFC5280]), but only the algorithm OID is included, no parameters etc. The EC curve is always known by the peer because it needs to have the certificate or the public key of the other end before it can do signature verification and public key specifies the curve.

> XXX While reading RFC4055, it seemed that the OID is not enough to specify the hash function used for the RSASSA-PSS, i.e. it seems that we would need to include full AlgorithmIdentifier ASN object, as it includes also the parameters, and the hash function is specified in the parameters. Is my reading of RFC4055 correct? XXX

The Authentication payload is defined in IKEv2 as follows:

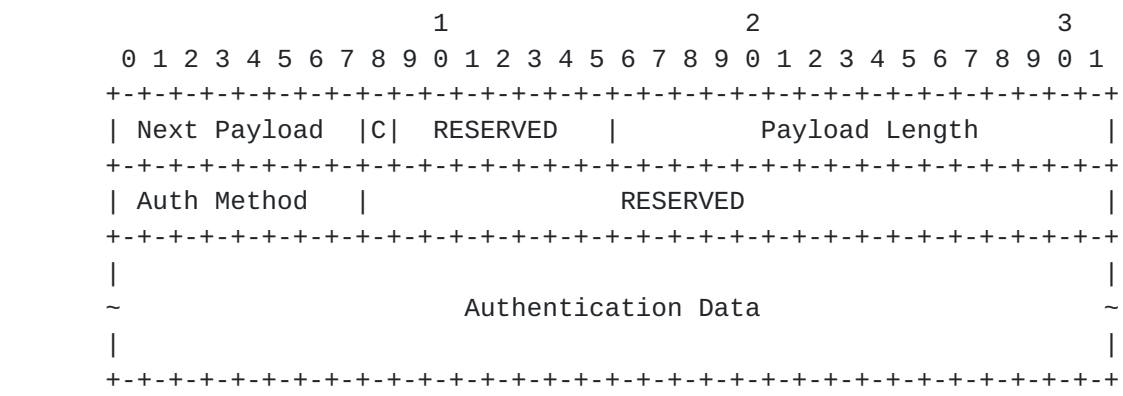


Figure 1: Authentication Payload Format.

o Auth Method (1 octet) - Specifies the method of authentication used.

| Mechanism | Value |
|---|---|
| Digital Signature | <TBD> |

Computed as specified in Section 2.15 of RFC5996 using a private key associated with the public key sent in certificate payload, and using one of the hash algorithms sent by the other end in the SIGNATURE_HASH_ALGORITHMS notify payload. If both ends send and receive SIGNATURE_HASH_ALGORITHMS and signature authentication is to be used, then this method MUST be used. The Authentication Data field has bit different format than in other Authentication methods (see below).

o Authentication Data (variable length) - see Section 2.15 of RFC5996. For "Digital Signature" format the Authentication data contains special format as follows:

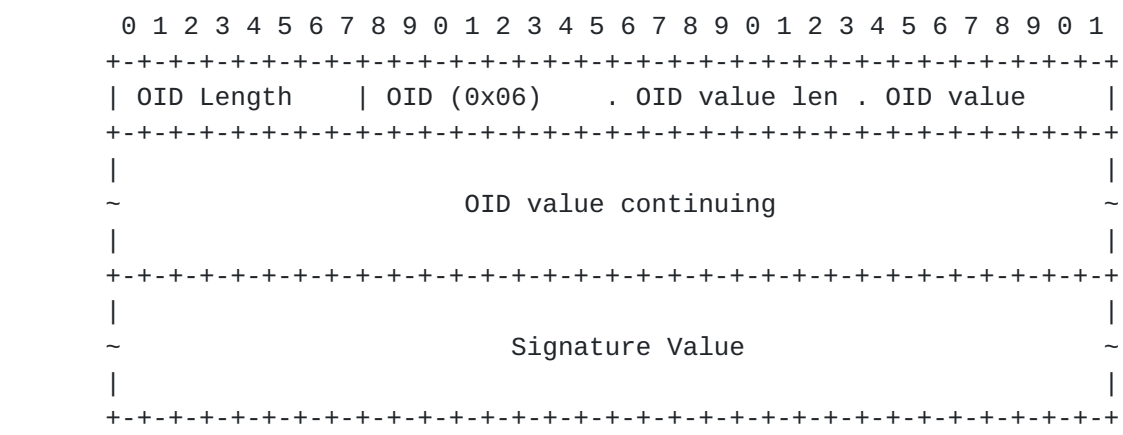


Figure 2: Authentication Data Format.

Where the OID Length is the length of the ASN.1 encoded OID value, and after that is the actual Signature Algorithm OID followed by the actual signature value. There is no padding between OID and signature value. ASN.1 encoded OIDs always start with byte of 0x06 followed by the length of the actual OID value (which is shown in the figure above). For the hash truncation the method of X9.62, SEC1 and IO 14888-3 MUST be used. XXX Need reference for X9.62/SEC1 etchere XXX.

## 4. Hash Algorithm Notification

The supported hash algorithms that can be used for the signature algorithms are now indicated with new SIGNATURE_HASH_ALGORITHMS Notification Payload sent inside the IKE_SA_INIT exchange. This notification also indicates the support of the new signature algorithm method, i.e sending this notification tells that new "Digital Signature" authentication method is supported and that following hash functions are supported by sending peer. Both ends sends their list of supported hash-algorithms and when calculating signature a peer MUST pick one algorithm sent by the other peer. Note, that different algorithms can be used in different directions. The algorithm OID matching selected hash algorithm (and signature algorithm) used when calculating the signature is sent inside the Authentication Data field of the Authentication Payload.

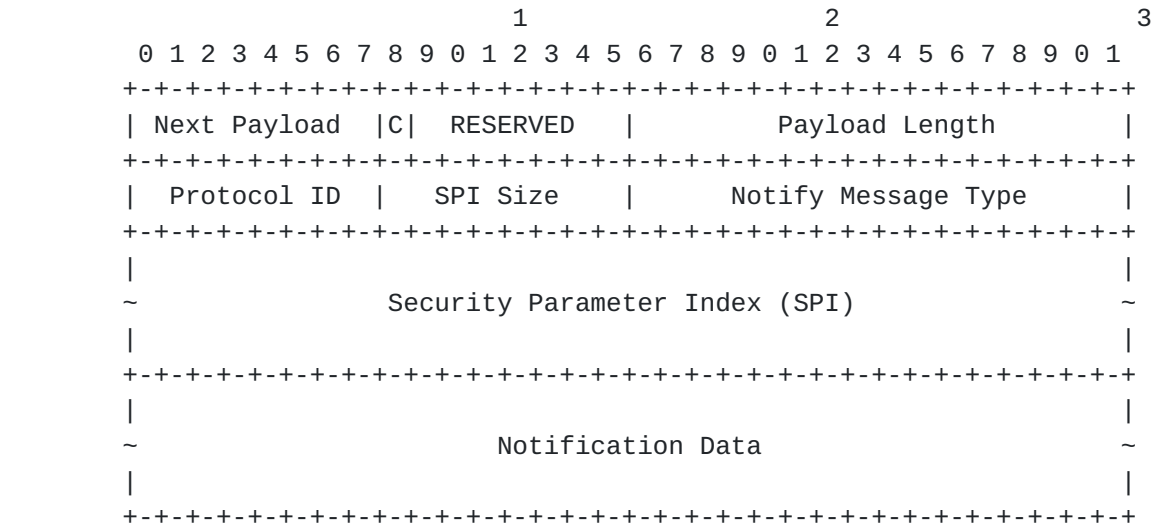


Figure 3: Notify Payload Format.

Protocol ID is 0, SPI Size 0, and Notify Message Type <TBD from status types>. The Notification Data value contains list of 16-bit hash algorithm identifiers from the newly created Hash Algorithm Identifiers for the IKEv2 IANA registry.

## 5. Security Considerations

XXX The text about the guidance how to select suitable hash functions is missing here. XXX

This new digital signature method does not tie the EC curve to the specific hash function, which was done in the old IKEv2 ECDSA methods. This means it is possible to use 512-bit EC curve with SHA1, i.e. this allows mixing different security levels. This means that the security of the authentication method is the security of the weakest of components (signature algorithm, hash algorithm, curve). This might make the security analysis of the system bit more complex. Note, that this kind of mixing of the security can be disallowed by the policy.

The hash algorithm registry does not include MD5 as supported hash algorithm, as it is not considered safe enough for signature use.

XXX Need reference for MD5 considered unsafe. XXX

The current IKEv2 uses RSASSA-PKCS1-v1_5, and does not allow using newer padding methods like RSASSA-PSS. This new method allows using other padding methods.

XXX Need reference for RSASSA-PSS vs RSASSA-PKCS1-v1_5 security. XXX

The current IKEv2 only allows using normal DSA with SHA-1, which means the security of the regular DSA is limited to the security of SHA-1. This new methods allows using longer keys and longer hashes with DSA.

## 6. IANA Considerations

This document creates new IANA registry for IKEv2 Hash Algorithms. Changes and additions to this registry is by expert review.

The initial values of this registry is:

| Hash Algorithm | Value |
|---|---|
| RESERVED | 0 |
| SHA1 | 1 |
| SHA2-256 | 2 |
| SHA2-384 | 3 |
| SHA2-512 | 4 |

MD5 is not included to the hash algorithm list as it is not considered safe enough for signature hash uses.

Values 5-1023 are reserved to IANA. Values 1024-65535 are for private use among mutually consenting parties.

## 7. Acknowledgements

Most of this work was based on the work done in the IPsecME design team for the ECDSA. The design team members were: Dan Harking, Johannes Merkle, Tero Kivinen, David McGrew, and Yoav Nir.

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

### 8.2. Informative References

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, June 2005.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, March 2009.

## Appendix A. Examples

Author's Address

Tero Kivinen
INSIDE Secure
Eerikinkatu 28
HELSINKI FI-00180
FI

Email: kivinen@iki.fi