Encoding of an Unsigned Diffie-Hellman Public Value

<draft-ietf-ipsec-skip-udh-00.txt>

Status of this Memo

This document is a submission to the IETF Internet Protocol Security (IPSEC) Working Group. Comments are solicited and should be addressed to the working group mailing list (ipsec@ans.net) or to the authors.

This document is an Internet-Draft. 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 draft documents are 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."

To learn the current status of any Internet-Draft, please check the "id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this memo is unlimited.
Abstract

It is useful to be able to communicate public keys in the absence of a certificate hierarchy and a signature infrastructure. This document describes a method by which certificates which communicate Diffie-Hellman public values and parameters may be encoded and securely named.
CONTENTS

Status of this Memo.................................. 1
Abstract............................................ 2
1. Unsigned Public Keys............................... 3
2. Encoding of an Unsigned DH public value......... 3
3. Verification of the Unsigned Public Value....... 5
4. Security Considerations............................ 5
Acknowledgements.................................... 5
References........................................... 5
Author’s Address(es)................................ 6

1. Unsigned Public Keys

In public key cryptography, certificates provide a binding between an entity’s name and their public key. The signature on the certificate
provides this binding. However, certificates tend to be difficult to implement and usually require infrastructure to verify signatures. This infrastructure and certificates, in general, are not in wide use on the Internet. Instead of explicitly binding a name to a public value using a signature, the name may be derived directly from the public key. This can be done by defining the name of the certificate to be the message digest of the public key.

Although the public value is distributed in an unsigned manner, there is still a strong binding between a name and the public value, given the collision resistance properties of a message digest. The entity’s names need to be securely distributed out of band.

This distribution of keys has a number of advantages over conventional signed certificates: no infrastructure is required to use Unsigned Public Keys. No signature algorithm needs to be supported. No complex encoding of certificates is required.

A disadvantage of this method is that the name must be securely (but not secretly) communicated to anyone using the key. Since the name is the hash value of the public key, it is a cryptic string of hexadecimal digits which is not user-friendly.

The encoding does not specify the hash algorithm used to generate the name. The hash algorithm must be transferred out of band. This may be done by creating a "certificate type" that includes this information. One valid certificate type is "MD5 of Hashed DH Public Key".

2. Encoding of an Unsigned DH public value

This encoding scheme is used to authenticate/distribute a DH public value, for cases where the entity’s name is the message digest of the public value.
The following is how the public value is encoded for purposes of message digest computation and distribution in the network. All values are in network order. All variable-length fields must begin with a non-zero byte.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+------------------------------------------+
<table>
<thead>
<tr>
<th>Not Valid Before</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Valid After</td>
</tr>
<tr>
<td>------------------------------------------</td>
</tr>
<tr>
<td>PrimeLen</td>
</tr>
<tr>
<td>------------------------------------------</td>
</tr>
<tr>
<td>~ Prime (p) (variable length)</td>
</tr>
<tr>
<td>~ Generator (g) (variable length)</td>
</tr>
</tbody>
</table>
|                                ~ PublicValueLen | Public Value (variable length)~
|                                ~ Public Value (g^i mod p) (variable length) |
```

"Not Valid Before" is the time at which the public value becomes valid. It is in NTP time format [3] (the Integer portion). "Not Valid After" is the time at which the public value expires. It is in NTP [3] time format (the Integer portion).

PrimeLen is Length of the DH Prime (p) in bytes. Prime contains the binary representation of the DH prime with most significant byte first. GenLen is the length of the Generator (g) in bytes. Generator is the binary representation of generator with most significant byte first. PublicValueLen is the Length of the Public Value (g^i mod p) in bytes. PublicValue is the binary representation of the DH public value with most significant byte first.

The Name associated with the public key and parameters is the cryptographic hash of the above encoding.
3. Verification of the Unsigned Public Value

Verification of the Encoding in this instance means verifying that the message digest of the entire encoding (as specified above) is the same as the (securely known) name of the entity. When using this instead of signed certificates, certificate verification MUST be done by performing the message digest computation.

4. Security Considerations

The unsigned DH public value can ONLY be used when entities are named using the message digest of their DH public value, AND these names are securely communicated. Unsigned DH public values MUST NOT be used instead of signed DH certificates when entities are named using something other than the message digest of their public value, since this opens up the possibility of an intruder-in-the-middle attack described in [1]. In order to use other naming schemes, signed certificates such as X.509, Secure DNS, PGP, etc. should be used.

Acknowledgements

We would like to thank all of the people who helped make this draft possible.

Jeff Schiller originally suggested using the hash of the public key as the Entity’s name.

Bill Danielson, Marc Dye, Colin Plumb, Rich Skrenta and Ben Stoltz for reviewing this draft and providing constructive suggestions.

References


Author’s Address(es)

Ashar Aziz
Sun Microsystems, Inc.
M/S PAL1-550
2550 Garcia Avenue
Mountain View, CA 94043

Email: ashar.aziz@eng.sun.com
Alternate email address: ashar@incog.com

Tom Markson
Sun Microsystems, Inc.
M/S PAL1-550
2550 Garcia Avenue
Mountain View, CA 94043

Email: markson@incog.com
Alternate email address: markson@eng.sun.com

Hemma Prafullchandra
Sun Microsystems, Inc.
M/S PAL1-550
2550 Garcia Avenue
Mountain View, CA 94043

Email: hemma@eng.sun.com
Alternate email address: hemma@incog.com