SKIP extension for Perfect Forward Secrecy (PFS)
<draft-ietf-ipsec-skip-pfs-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 "lid-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

This document describes an optional extension specifying how to use an ephemeral Diffie-Hellman exchange in conjunction with the SKIP protocol in order to provide perfect forward secrecy for situations where forward secrecy is necessary.
1. Introduction

This document describes how an ephemeral Diffie-Hellman key exchange can be used in conjunction with the SKIP key distributions protocol [1] to provide Perfect Forward Secrecy (PFS) for situations where PFS is required.

The certificate discovery protocol [2] is used to exchange ephemeral Diffie-Hellman values by defining a new certificate type for ephemeral DH certificates. This ephemeral certificate is then used to compute an ephemeral master key, which is used in place of the master keys Kijn used in the base SKIP protocol.

In addition a new type of Master Key-ID (MKID) type is defined here, to indicate the use of ephemeral master keys. In addition to perfect forward secrecy, principal anonymity is also supported in the context of the ephemeral certificate exchange.

No new protocol family is introduced in order to provide PFS with SKIP. Rather, existing mechanisms such as the certificate discovery protocol, and the extensible MKID types are used to optionally provide PFS over the base SKIP protocol.

Using an ephemeral Diffie-Hellman exchange introduces greater bilateral state and overhead than is present in the base SKIP protocol. When using ephemeral certificates, certain features of the base SKIP protocol that rely on statelessness (e.g. quick failover of intermediate nodes) become unavailable.

Optional use of both the stateless and stateful modes of operation (with the associated lack and presence of PFS) is specified in the context of the SKIP protocol to provide greater flexibility than is possible with protocols that provide only one or the other modes of operation.

2. Cryptographic Description of Ephemeral Certificate Exchange

Cryptographic Notation used for describing Ephemeral Certificates:

Note: All exponentiations (e.g. g^x) are mod p. The mod p reduction is assumed, and is omitted for the sake of brevity.
The ephemeral certificate exchange is described using the notation above as follows:

I->J: \{ g^x, g, p, [Cert_I]g^xj, EMKID_J_I\}Kij
J->I: \{ g^y, g, p, [Cert_J]g^xj, EMKID_J_I, EMKID_I_J\}Kij

The ephemeral master key (denoted as EKijn) is computed as

\[ \text{EKijn} = \text{MD5}(Kij \mid g^xy \mid n \mid 01) \mid \text{MD5}(Kij \mid g^xy \mid n \mid 00) \]

where \( n \) is the counter from the SKIP header. This master key computation is very similar to the master key computation specified in the base SKIP protocol, with the exception of the inclusion of the ephemeral Diffie-Hellman shared secret \( g^xy \) in the master key hash computation. As in the base SKIP protocol, "00" and "01" refer to one byte values containing the values 0 and 1 respectively, and "\mid" refers to concatenation.

The values EMKID_I_J and EMKID_J_I refer to the ephemeral Master Key-ID to be used in SKIP packets sent from I to J and J to I, respectively. I picks the ephemeral MKID to be used in packets sent from J to I, and J picks the ephemeral MKID to be used in packets sent from I to J. In either case, both ephemeral MKIDs identify the same EKijn computed as specified in Section 2 above. This EKijn is used to encrypt the packet key \( Kp \) present in the SKIP header.

The encryption of each principal’s certificate using \( g^xj \) is optional. It is used to provide anonymity of the parties involved in the ephemeral exchange. In case anonymity is not desired or necessary (e.g. node to node communications) the encryption using \( g^xj \) may
be omitted.

3. Ephemeral Certificate Format

An ephemeral certificate contains essentially an ephemeral randomly generated Diffie-Hellman public value, authenticated using the long-lived certified Diffie-Hellman values used by the base SKIP protocol. The certificate is authenticated using $Kij$ from the base SKIP protocol as a key to compute a MAC over the certificate contents.

Each principal involved in an ephemeral certificate exchange computes an ephemeral master key by combining ephemeral Diffie-Hellman shared secret values with the long-lived Diffie-Hellman shared secret values as specified above. This ephemeral master key is then used to encrypt the traffic keys $Kp$ communicated in the SKIP header.

In addition to the ephemeral Diffie-Hellman public values, an ephemeral certificate contains the identity and certified Diffie-Hellman public values of the exchange initiator. This identity MAY be encrypted in order to provide anonymity.

The following is the format of an ephemeral certificate:
Ephemeral Diffie-Hellman Certificate Format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Ver   | Rsvd  |  Protocol     |  Port         | Cert MAC Alg |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Validity Interval (Seconds) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| DH Public Value Length | DH Public Value (g^x or g^y) ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Generator Length | Generator (g) ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Modulus Length | Modulus (p) ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Ephemeral Master Key-ID EMKID_I_J |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Ephemeral Master Key-ID EMKID_J_I |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Cert Enc. Alg | Cert Type | Encrypted Cert. Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Encrypted Long-Lived Certificate ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Ephemeral Certificate MAC ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
~
```

The version field specifies the version number of the ephemeral certificate. The version field MUST be 1 for this version of the certificate format.

The protocol and port # together identify the responder for which the certificate exchange is intended. An example of a responder could be a telnet or ftp process, in which case the protocol field would specify TCP, and the port # would identify the corresponding listener port for the telnet or ftp daemon process. If the protocol field is zero, then
the node, as opposed to a process on the node, is the principal with
which the ephemeral certificate exchange takes place. In the certificate
sent from J to I (responder to initiator) these fields MUST be zero
filled.

"Cert MAC Alg" identifies the MAC algorithm which is used to compute a
MAC over the certificate contents. The scope of the MAC computation is
the entire certificate, with the MAC field treated as zero filled for
the purposes of the MAC computation.

The "Validity Interval" specifies how long the ephemeral master key
derived from this exchange should be used for. This value is in seconds.
A responder MAY choose a different value for this field than the
initiator, in which case the actual validity interval for this master
key is the minimum of the two values in the exchange. At the end of the
validity interval, the ephemeral master key and the all associated
secret information is destroyed by both the responder and the initiator.
A new exchange may be initiated either subsequent or prior to the expiry
of the ephemeral master key, in case there is still encrypted traffic
that needs to be sent in PFS mode.

DH Public Value Length specifies the length of the DH public value
field. DH Public Value contains the ephemeral DH public value (g^x or
g^y), specified as a string of octets with the most significant octet
first. Similarly, Generator Length, Generator and Modulus Length,
Modulus specify the lengths and values of the Generator (g) and the
Modulus (p) used for the DH computation. A responder MUST use the same
values for the generator and modulus as the initiator.

The field EMKID_I_J specifies what the ephemeral MKID should be for
packets sent from I to J. Since J picks the value of this field, this
field MUST be zero filled in the ephemeral certificate sent from I to J.
The value of this field is specified in the ephemeral certificate sent
from J to I.

The field EMKID_J_I specifies what the ephemeral MKID field contains for
packets sent from J to I. I picks the value of this field, and J MUST
fill in the same value in this field as was present in the ephemeral
certificate that J received from I.

Each of EMKID_I_J and EMKID_J_I is used only as the Destination MKID in
a SKIP header. When used in ephemeral master key mode, the Source MKID
MUST be absent, and indicated by a zero filled Source NSID field in the
SKIP header.
The combination of an ephemeral Destination MKID and the destination IP address uniquely identifies an ephemeral master key.

"Cert Enc. Alg" specifies the encryption algorithm used to encrypt I and J’s long-lived certificate. This is the same DH certificate as used in the base SKIP protocol. The type of this certificate is indicated using the "Cert Type" field. This value MUST NOT refer to an ephemeral certificate type.

The certificate is encrypted using the low-order key-size bits of g^xj as the encryption key. If the encryption algorithm requires per message variables (e.g. an IV) then this is derived using the high order variable size bits of g^xj. Since only I and J can properly compute g^xj, the encryption of I and J’s certificate provides principal anonymity for situations where anonymity is desired. The anonymity protection provided is secure against both active and passive forms of attack.

If the "Cert Enc. Alg" field is zero, then the long-lived certificate is in the clear. In this case the field "Encrypted Long-Lived Certificate" contains the long-lived DH certificate in the clear.

When J receives an encrypted long-lived certificate, it first computes g^xj in order to decrypt the long-lived DH certificate. Having obtained (and verified) the long-lived certificate (which contains the value g^i) J computes g^ij, and thereby Kij which it uses to verify the MAC field "Certificate MAC" of the ephemeral certificate. If the MAC field is incorrect, the ephemeral certificate MUST be discarded. If the MAC field is correct, J computes EKijn as specified above and responds with its own ephemeral certificate, containing g^y.

When I receives an ephemeral certificate, it uses the value EMKID_J_I to locate the request for which this is the corresponding response. A non-zero EMKID_I_J field indicates that this a response to an ephemeral certificate request initiated by I, as opposed to a new certificate exchange initiated by J.

4. Informational

The following shows an example of how the ephemeral certificate exchange is used in conjunction with the SKIP header.
Assume that the value EMKID_J_I is 1001, and the value EMKID_I_J is 2007 after a successful ephemeral certificate exchange. EKijjn is computed as described in Section 2 above.

SKIP Header in packet sent from I to J

```plaintext
0                   1                   2                   3
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Clear IP Header protocol = SKIP... (typically 20-bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver | Rsvd.| Src NSID=0 | Dst NSID=EMKID|NEXT HEADER |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Counter n                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Kij alg.   |   Crypt alg   |  MAC Alg.     |  Comp Alg     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Kp encrypted in EKijjn... (typically 8-16 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Destination MKID  (contains the value 2007) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SKIP Header in packet from J to I

```plaintext
0                   1                   2                   3
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Clear IP Header protocol = SKIP... (typically 20-bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver | Rsvd.| Src NSID=0 | Dst NSID=EMKID|NEXT HEADER |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Counter n                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Kij alg.   |   Crypt alg.  |  MAC Alg.     |  Comp Alg     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Kp encrypted in EKijjn... (typically 8-16 bytes) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Destination MKID  (contains the value 1001) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
5. Certificate Type and Name Space Assignments

5.1 Certificate Type Assignment

The ephemeral Diffie-Hellman certificate type as defined in this document is pending assignment by IANA.

5.2 NSID Assignment

The NSID for ephemeral MKIDs EMKID is pending assignment by IANA.

6. Generalization for Key-Agreement techniques other than classic DH

Although the ephemeral certificate exchange scheme specified above uses the constructions of classic Diffie-Hellman (exponentiation over finite fields) the scheme is fully generalizable to other key-agreement techniques, such as Elliptic Curve (EC) variants of Diffie-Hellman. In order to use these other DH variants, a new ephemeral certificate type may be defined that contains parameters specific to these other DH variant schemes.

7. Security Considerations

The topic of this memo is security.

References


Author Information:

Ashar Aziz  
Sun Microsystems, Inc.  
MS PAL1-550,  
2550 Garcia Ave.  
Mountain View, CA 94043  

e-mail: ashar.aziz@Eng.Sun.COM