MSEC WG
Internet-Draft
Expires: April 24, 2005

L. Dondeti
J. Xiang
Nortel Networks
October 24, 2004

GKDP: Group Key Distribution Protocol
draft-ietf-msec-gdoiv2-01

Status of this Memo

Copyright Notice

Abstract

This document specifies a group key distribution protocol (GKDP) based on IKEv2 [2]; the new protocol is similar to IKEv2 in message and payload formats, and message semantics to a large extent. The protocol in conformance with MSEC key management architecture contains two components: member registration and group rekeying, and downloads a group security association from the GCKS to a member. This protocol is independent of IKEv2 except in its likeness.

Conventions Used In This Document

This document recommends, as policy, what specifications for Internet protocols -- and, in particular, IETF standards track protocol documents -- should include as normative language within them. The capitalized keywords "SHOULD", "MUST", "REQUIRED", etc. are used in the sense of how they would be used within other documents with the meanings as specified in BCP 14, RFC 2119 [1].

Table of Contents

1. Revision History

1. **The protocol has been renamed GKDP for Group Key Distribution** Protocol as per discussions at the MSEC meeting at IETF-60 and mailing list discussions. The name GDOIv2 will be used for a revision of GDOI which may retain the DOI concept and build upon RFC 3547.
2. After the IETF-61 meeting, we will resubmit as draft-ietf-msec-gkdp-00.

2. Introduction and Overview

Security encapsulation protocols such as IPsec and SRTP provide confidentiality, message integrity, replay protection, and in some instances access control, and data origin authentication. These security services require state establishment, maintenance, and teardown for correct operation. While these security associations can be managed manually, automatic key management protocols are essential for efficient and scalable operation. In case of point-to-point security associations, IKE and its successor IKEv2 are widely used for IPsec SAs, and MIKEY for SRTP associations. For multi-point SAs or group SAs (GSA), GDOI, GSAKMP, and MIKEY have been specified by the MSEC WG. GKDP is designed to be a counterpart - for GSA distribution and maintenance - to IKEv2 so we can reuse the work put in to its design and analysis, and of course implementation.

2.1 Why do we need another GSA management protocol?

Given the collection of key management protocols mentioned above, there is a question on the need for yet another group key management protocol. First a look back at history: So far, we have two experimental RFCs, viz., RFC 1949 [3] and RFC 2093 [4], and a standards track RFC, RFC 3547 [5] specifying or describing group key management protocols. Furthermore there is GSAKMP, currently a standards track MSEC I-D, which borrows quite a few concepts from IKEv2, but not quite similar to IKEv2. The protocol we propose is mainly to reuse as much as the IKEv2 codebase, similar to GDOI reusing payload and message formats of IKE [7] and ISAKMP [6] . Consequently, GKDP requires fewer messages compared to GDOI, specifically 4 in most cases, compared to 10 in main mode and 7 in aggressive mode of GDOI. We discuss the advantages of GKDP, the shortcomings and remedies to address those shortcomings.

2.2 GKDP usage scenarios

GKDP is a key download protocol. Key download as opposed to key negotiation has several interesting use cases.

- The first application is multicast security. As with GDOI, the current version of the GKDP spec limits the scope to single sender

multicast applications.
- The second intended application is point to point data security associations facilitated by a centralized group key server.
- Others to be listed!

3. GKDP protocol

3.1 Member registration and secure channel establishment

The first of two components in GSA establishment and maintenance is member registration.

3.1.1 Initial exchange:GSA_INIT_EXCH

The first step in the registration protocol is to establish a secure channel with the group controller and key server (GCKS). This exchange is similar to IKE_SA_INIT exchange of IKEv2. The registering member proposes various combinations of algorithms in SAi1 to constitute the secure channel, along with a nonce, Ni, and a DH exponent, KEi. The GCKS has several options:
- In the first, it honors the member's request for registration and sends the necessary information to complete the DH exchange: it selects and specifies the parameters of the secure channel, and includes a nonce Nr, and a public DH value of its own, KEr.
- The second option is for the GCKS to consider if the request for secure channel establishment is spurious. It has no way to tell except to throttle such requests by making the initiator do some work before it invests any computing resources. This is known as the DoS protection mode in IKEv2 and is explained in detail in Section 3.1.1.1 .
- Finally, if none of the proposals are acceptable to the GCKS, it may reject the initial exchange itself.

GSA_INIT_EXCH message is as follows:

```
Member->GCKS: M1:    HDR, SAi1, KEi, Ni
GCKS->Member: M2:    HDR, SAr1, KEr, Nr, [CERTREQ]
```

Figure 1: Secure channel establishment

3.1.1.1 DoS protection mode

DoS protection exchange is as follows:

```
Member->GCKS: IM:  HDR(A,0), SAi1, KEi, Ni
GCKS->Member: CM:   HDR(A,0), N(COOKIE)

Member->GCKS: M1:   HDR(A,0), N(COOKIE), SAi1, KEi, Ni
GCKS->Member: M2:  HDR(A,B), SAr1, KEr, Nr, [CERTREQ]

IM: Initial Message from the Member
CM: Challenge Message from the GCKS
```

Figure 2: DoS protection mode of GSA_INIT_EXCH

3.1.2 Authenticated exchange:GSA_AUTH_EXCH

GSA_AUTH_EXCH message is as follows:

```
Member->GCKS: M3: HDR, SK{ G-ID, IDi, [ID_CERT,] [ID_CERTREQ,] AUTH,
               [IDr,] [GM_CERT,] [GM_CERTREQ,] [POP_I] }
GCKS->Member: M4: HDR, SK{ IDr, [ID_CERT,] AUTH, GSA, KD [,SEQ]
               [GCKS_CERT,] [,POP_R]}
```

Figure 3: Authenticated Exchange

The various payloads in the GSA_AUTH_EXCH messages have the following purposes:

- G-ID: The group identity payload constructed using the IKEv2 Identification Payload specifies the secure group that M3 wants to join.
- ID_CERT: The optional ID_CERT payload contains a certificate(s) asserting the GCKS's or a member's claimed identity as in IDi or IDr payloads.
- GM_CERT: The optional GM_CERT payload contains a certificate asserting the group member's authorization to join the group G-ID as member.
- GCKS_CERT: The optional GCKS_CERT payload contains a certificate asserting the GCKS's authorization to serve the role of a group controller and key server for the group G-ID.
- AUTH: The AUTH payload constitues the "authenticated" portion of the 4 or 6 message AKE. In other words, the member in M3 and the GCKS in M4 prove that they are indeed the entities that sent M1

and M2 respectively. A pre-established shared secret or a certificate (optionally specified in the CERT payload) may be used for entity authentication.

- o POP: Similar to the AUTH payload's use in providing host/entity authentication, the POP payload is for member/GCKS authorization to assume their claimed roles. The GM_CERT or GCKS_CERT is used to sign a block of data, specified below, to constitute the POP payload.
- o GSA: The GSA payload contains the rekey and data security SA payloads. Note that this SA is not negotiated; the GCKS simply sends this SA.
- o KD: The KD payload contains the secret keys corresponding the rekey and the data security SAs included in the GSA payload.
- o SEQ: The optional SEQ payload MUST be included if the GSA payload contains a rekey SA. The SEQ payload contains a SEQ number for replay protection of the rekey messages.

3.1.2.1 Key material computation

The key material computation and the AUTH payload are identical to that described in the IKEv2 specification.

Key material and registration SA keys are computed as follows:

```
SKEYSEED = prf(Ni | Nr, g^ir)

  {SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr }
              = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr ), where
```

prf+ is defined as follows:

```
   prf+ (K,S) = T1 | T2 | T3 | T4 | ...

   where:
   T1 = prf (K, S | 0x01)
   T2 = prf (K, T1 | S | 0x02)
   T3 = prf (K, T2 | S | 0x03)
   T4 = prf (K, T3 | S | 0x04)
```

Figure 4: Registration SA key material computation

3.1.2.2 Member and GCKS authentication and authorization

GKDP requires mutual authentication between each member and a GCKS,

as well as mutual authorization. First the member and the GCKS authenticate to each other using pre-shared keys or certificates prior to establishing a secure channel. M3 and M4 contain AUTH payloads that essentially protect against man-in-the-middle attacks against the DH exchange in M1 and M2. The member and the GCKS construct AUTH payloads by computing an HMAC over or signing a block of data containing the message M1 or M2 they sent earlier, the other party's nonce payload, and a prf over own identity. More formally, the block of data for HMAC or signature is as follows:

Auth payload computation:

Auth payload in M3 is computed over:

auth-block-M3: M1 || Nr-Payload || prf(SK_pi, IDi-Payload)

Auth payload in M4 is computed over:

auth-block-M4: M2 || Ni-Payload || prf(SK_pr, IDr-Payload)

For shared secret based host authentication AUTH payload is computed as follows:

```
AUTH = prf(prf(Shared Secret,"KeyPad:GKDP-AUTH-MX"),
               <auth-block-MX>)
```

Figure 5: Auth payload computation

3.1.2.2.1 Use of asymmetric authentication methods

GKDP also allows the member and the GCKS to use different authentication methods, similar to TLS and IKEv2. More specifically, the GCKS uses a cert to authenticate itself and establish a secure channel, and the member uses EAP to send its authentication information via the secure channel.

Members may also use EAP to prove their authorization to join a secure group. For instance, consider a use case where a member may use a SIM card for authentication, or a pre-paid SIM card to pay for content distributed to a secure group. In these cases, the authentication or authorization information can be sent via EAP.

3.1.2.2.2 Proof of possession

Proof of possession payload (POP) provides a mechanism so that

members and/or GCKS can prove to the other party that they are indeed authorized to be a member or the GCKS, respectively. For POP payload derivation in GKDP, the member or the GCKS first constructs a message block, POP-HASH, containing the two nonces exchanged in GSA_INIT_EXCH and the prf over the ID payload as defined in the AUTH payload construction. Next, the member or the GCKS signs the POP-HASH value.

POP-HASH construction is as follows:

POP payload :

POP payload in M3 is constructed over the following message block:

```
POP-HASH-M3: "KeyPad:GKDP-POP-M3" ||
              Ni-Payload || Nr-Payload || prf(SK_pi, IDi-Payload)
```

POP payload in M4 is computed over:

```
POP-HASH-M4: "KeyPad:GKDP-POP-M4" ||
             Ni-Payload || Nr-Payload || prf(SK_pr, IDr-Payload)
```

Figure 6: POP payload computation block

4. GSA maintenance channel

4.1 GSA rekey protocol

GSA rekey protocol is optional to implement, but it plays a crucial role for large and dynamic groups.

The GCKS is responsible for rekeying of the secure group as per the group policy. The GCKS uses multicast or multi-unicast to transport the rekey message. When multi-unicast is used, it may be appropriate in some scenarios to have a reply message from the member(s) to the GCKS. The reply message is optional.

Rekey message is as follows:

```
Multicast:
GCKS->Member:    HDR, SK {[N], SEQ, GSA, KD, [GCKS_CERT,] SIG}

Unicast:
GCKS->Member:    HDR, SK {N, SEQ, GSA, KD, [GCKS_CERT,] SIG}
[Member->GCKS]:    [HDR, SK {N, SEQ, AUTH}]
```

Figure 7: Rekey message

5. GKDP protocol details

6. Header and payload formats

To be copied from IKEv2 and GDOI specifications. We do anticipate some minor changes however.

7. Security considerations

8. Acknowledgments

GKDP is based on IKEv2 and GDOI. Several sections of this document are quite identical to IKEv2 and GDOI specifications. We included the text for completeness of this specification. We appreciate the efforts of the contributors and editors of those protocols.

9. References

9.1 Normative References

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

[2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", Internet-Draft: draft-ietf-ipsec-ikev2-14.txt (Work in progress), May 2004.

9.2 Informative References

[3] Ballardie, T., "Scalable Multicast Key Distribution", RFC 1949, May 1996.

[4] Harney, H. and C. Muckenhirn, "Group Key Management Protocol (GKMP) Specification", RFC 2093, July 1997.

[5] Baugher, M., Weis, B., Hardjono, T. and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[6] Maughan, D., Schneider, M. and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[7] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

Authors' Addresses

Lakshminath Dondeti
Nortel Networks
600 Technology Park drive
Billerica, MA 01821
US

Phone: +1 978 288 6406
EMail: ldondeti@nortelnetworks.com

Jing Xiang
Nortel Networks
600 Technology Park drive
Billerica, MA 01821
US

Phone: +1 978 288 8985
EMail: jxiang@nortelnetworks.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.