Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 24, 2010

AV. Padmakumar
M. Bafna
P. Sethi
Cisco

December 21, 2009

IKEv2 Redirect based Authentication Offload and Proxy Session Resumption

draft-padmakumar-ikev2-redirect-and-auth-offload-02

Abstract

IKEv2 supports multiple authentication mechanisms like public key signatures, shared secrets and EAP. EAP based authentication requires server to maintain information about the client until EAP completes. Public key based authentication mechanisms are highly computational intensive and demands server CPU resources.

Redirect Mechanism for IKEv2 proposes a mechanism for IKEv2 that enables a VPN gateway to redirect the VPN client to another VPN gateway, for example, based on the load condition.

IKEv2 Session Resumption proposes an extension to IKEv2 that allows a client to re-establish an IKE SA with a gateway in a highly efficient manner, utilizing a previously established IKE SA.

Redirect mechanism can also be used to redirect a client to another router (trust anchor) to do mutual authentication and an optional proxy session negotiation on behalf of the server. This redirection happens during the IKE_SA_INIT and server does not maintain any information about the redirected client. After mutual authentication and optional proxy session negotiation trust anchor redirects the client back to the server with an Access Token which can be used as a dynamic pre-shared key between the server and client for password based IKE_AUTH exchange. Mechanism described here allows servers to compute the same pre-shared key dynamically, without contacting trust anchors, based on the information provided by the client during IKE_AUTH exchange. Access Token based authentication permits IDr of the responder to be different from that specified in Ticket and permits client to reuse the proxy session (negotiated between client and trust anchor) between client and server.

Such a mechanism is useful especially for low power devices like handsets. For example, a mobile node can redirect a correspondent node to its home agent. Home agent can form a proxy session with correspondent node and then redirect it back to mobile node.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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 June 24, 2010.

Copyright Notice

Table of Contents

1. Introduction

IKEv2 [RFC4306] supports multiple authentication mechanisms like public key signatures, shared secrets and EAP. EAP based authentication requires server to maintain information about the client until EAP completes. Public key based authentication mechanisms are highly computational intensive and demands server CPU resources.

Redirect Mechanism for IKEv2 [IKEv2REDIRECT]proposes a mechanism for IKEv2 that enables a VPN gateway to redirect the VPN client to another VPN gateway, for example, based on the load condition. Redirect can be done during the IKE_SA_INIT, IKE_AUTH exchange or in the middle of a session.

IKEv2 Session Resumption [IKEv2RESUMPTION] proposes an extension to IKEv2 that allows a client to re-establish an IKE SA with a gateway in a highly efficient manner, utilizing a previously established IKE SA.

Redirect mechanism can be used to redirect a client to another router (trust anchor) to do mutual authentication and an optional proxy session negotiation on behalf of the server. This redirection happens during the IKE_SA_INIT and server does not maintain any information about the redirected client. After mutual authentication and optional proxy session negotiation trust anchor redirects the client back to the server with an Access Token which can be used as a dynamic pre-shared key between the server and client for password based IKE_AUTH exchange. Mechanism described here allows servers to

compute the same pre-shared key dynamically, without contacting trust anchors, based on the information provided by the client during IKE_AUTH exchange.

IKEv2 Session Resumption assumes that the client always resumes into the same gateway that generated the ticket. IKE_AUTH exchange that follows the IKE_SESSION_RESUME exchange does not use any authentication mechanisms like pre-shared key or certificates. That means the ID values sent in the IKE_AUTH exchange can not be re-authenticated and MUST be identical to the values included in the ticket. But Authentication Offload mechanism explained in this memo allows servers to compute a pre-shared key dynamically based on the information provided by the client during IKE_AUTH exchange. Clients learn the same pre-shared key from the trust anchors during IKE_AUTH exchange. That means the IKE_AUTH exchange that follows the IKE_SESSION_RESUME exchange can independently verify the identities of both parties based on a common pre-shared key computed dynamically. This method is utilized in this memo and proxy session resumption is made possible with an IDr which is different from the

one mentioned in the ticket. Access tokens computed are linked to IDi values and MUST not be changed.

Trust anchor MUST choose the same cryptographic suit from client's offer which the server would have chosen if the client had contacted server directly.

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 RFC 2119 [RFC2119].

- Trust Anchor : A router (or set of dedicated routers) that can act as a proxy for a server to do mutual authentication and proxy session negotiation on behalf of the server.
- GP_SECRET : Gate Pass Secret (GP_SECRET) is a random number generated by server. Server shares GP_SECRET with all trust anchors in a secure way.
- GP_SECRET_INDEX : Server maintains a history of GP_SECRET and each GP_SECRET is uniquely identified by the GP_SECRET_INDEX

- o GP_KEY : Gate Pass Key (GP_KEY) is a set of secret keys, updated periodically and identified by an Index. GP_KEY is used to generate and verify GATE_PASS.
- o GATE_PASS : Identifies the transitive trust channel (server to trust anchor).
- o ACCESS_TOKEN : Passwords generated by trust anchors to be used between server and clients. Servers and anchor routers compute tokens independently but based on a common information known only to them.
- o PROXY_TICKET : An IKEv2 ticket [IKEv2RESUMPTION] is a data structure that contains all the necessary information that allows an IKEv2 responder to re-establish an IKEv2 security association. PROXY_TICKET is an IKEv2 ticket negotiated between client and trust anchor. IKEv2 Auth offload mechanism explained in this memo allows proxy session resumption with an IDr which is different from the one mentioned in the IKEv2 ticket.

3. GP_SECRET

Gate Pass Secret (GP_SECRET) is a random number generated by server. Server shares GP_SECRET with trust anchor in a secure way (distribution mechanism is explained in Section 9). Length of GP_SECRET is decided by the encryption algorithm negotiated between server and trust anchor in the context of IKE_SA.

4. GP_SECRET_INDEX

Gate Pass Secret index(GP_SECRET_INDEX) identifies the GP_SECRET. Server maintains a history of GP_SECRET and each GP_SECRET is uniquely identified by the GP_SECRET_INDEX

5. GP_KEY

GP_KEY is a set of authentication and encryption keys, updated periodically and identified by an Index. GP_KEY has the following structure.

GP_KEY = {

GP_KEY.ek

GP_KEY.ak

GP_KEY.index

}

This specification does not define the size of these fields and is left up to the server to choose based on the algorithms it use.

GP_KEY.index is carried in a GATE_PASS(Section 6).

GP_KEY.ek key is used to encrypt the {trust anchor, server specific informations } fields of the GATE_PASS.

GP_KEY.ak key is used to sign the encrypted part in the GATE_PASS.

When server receives an AUTH message authenticated by IKEv2 trust anchor method, server verifies the Signature on the GATE_PASS using the GP_KEY.ak. If this check fails, server MUST drop the AUTH message and MAY send AUTHENTICATION_FAILED message.

6. GATE_PASS

An IKEv2 GATE_PASS is a data structure generated by the server and contains all the necessary information that allows server to recompute ACCESS_TOKEN based on the client's IDi. Server encrypt and integrity protect GATE_PASS using GP_KEY and distributes to trust anchors using N(GATE_PASS) notify payload. Clients receive the same GATE_PASS from the trust anchors and include it in IKE_SA_INIT or KE_SESSION_RESUME exchange when it restarts its IKE exchange with the

server. Server computes GATE_PASS using following mechanism.

GATE_PASS = Signature | GP_KEY.Index | Secret

Length of each field is not defined by this specification and can be defined by the server depending on the algorithms it uses for computing these. Secret and Signature can be computed using following mechanism.

Secret = encrypt(GP_KEY.ek, {Trust Anchor IP | Trust Anchor ID | GP_SECRET_INDEX | server specific informations })

Signature = prf(GP_KEY.ak | GP_KEY.Index | Secret)

Where encrypt() and prf() are the encryption and pseudo random number generator algorithms that the server wants to use.

Trust anchor sends GATE_PASS to client using N(GATE_PASS) notify payload along with the N(REDIRECT) [IKEv2REDIRECT], N(ACCESS_TOKEN)(Section 7) and optional N(TICKET_OPAQUE) [IKEv2RESUMPTION] payloads.

7. ACCESS_TOKEN

Client MUST use ACCESS_TOKEN as the pre-shared key for password based authentication between server and client if client includes N(GATE_PASS) in the IKE_AUTH exchange. AUTH is computed using the same mechanism specified in Section 2.15 of IKEv2 RFC [RFC4306] using pre-shared keys. Server computes the same ACCESS_TOKEN using client's identity, GATE_PASS and GP_SECRET.

ACCESS_TOKEN = prf(encrypt(GP_SECRET, GATE_PASS | Client's Identity))

Where prf() and encrypt() are the pseudo random number generator algorithm and encryption algorithm negotiated between server and trust anchor during IKE_SA. It MUST be picked from the same IKE channel through which the server has distributed GP_SECRET and GATE_PASS to Trust Anchor.

Client's Identity is the identity specified by IDi payload during IKE_AUTH exchange.

Trust anchor sends ACCESS_TOKEN to client using N(ACCESS_TOKEN) payload along with N(GATE_PASS) and N(REDIRECT) payloads to redirect it back to the server.

N(REDIRECT) is defined in [IKEv2REDIRECT].

N(ACCESS_TOKEN) payloads can be included only after successful

verification of client's identity.

8. PROXY_TICKET

PROXY_TICKET is an IKEv2 ticket [IKEv2RESUMPTION] negotiated between client and trust anchor. IKEv2 Auth offload mechanism explained in this memo allows proxy session resumption with an IDr which is different from the one mentioned in the IKEv2 ticket. Clients can include N(PROXY_TICKET_SUPPORTED) along with N(REDIRECT_SUPPORTED) and N(GATE_PASS_SUPPORTED) in the IKE_SA_INIT exchange to inform proxy support for IKEv2 Session Resumption [IKEv2RESUMPTION] capability to the servers or trust anchors. Server redirects clients to trust anchors for authentication and proxy session negotiations. Trust anchor redirects clients back to server after successful authentication and proxy session negotiations. Trust anchors MUST not use Ticket by reference [IKEv2RESUMPTION] for passing Tickets to clients, instead, it MUST use Ticket by value [IKEv2RESUMPTION] method. If proxy session resumption is supported trust anchor MUST encrypt and integrity protect the Ticket using the GP_SECRET associated with the GATE_PASS. Client MUST include N(GATE_PASS) along with N(TICKET_OPAQUE) [IKEv2RESUMPTION] payload in the IKE_SESSION_RESUME exchange [IKEv2RESUMPTION] to indicate that the resumed session is a proxy session and the identities (old IDi of client and new IDr of server) are authenticated separately using GATE_PASS in the IKE_AUTH exchange. Server MUST not accept the Tickets with different IDr if N(GATE_PASS) is not included. If N(GATE_PASS) is included in the IKE_SESSION_RESUME exchange then the IDr contained in the Ticket MUST be the same as Trust Anchor ID contained in the GATE_PASS, otherwise this MUST be considered as normal IKE_SESSION_RESUME exchange and IDr MUST be the same as that of the responder (server).

9. GP_SECRET and GATE_PASS Distribution to Trust Anchor

Server MAY periodically update GP_KEY and GP_SECRET and each time it updates it MUST recompute and redistributes the associated GATE_PASS and GP_SECRET to all trust anchors (if they support this feature, indicated by N(GATE_PASS_SUPPORTED) in the INIT exchange). This way servers can restrict the life time of tokens issued to clients. Server MAY maintain a history of GP_KEYs and GP_SECRETs for a duration decided by server's policy to support clients who bring old GATE_PASS but still valid based on GP_KEY history.

Server uses an INFORMATIONAL exchange to distribute GATE_PASS and GP_SECRET. Server and trust anchor need not retain the IKE SAs but

in such cases they MUST remember the prf and encryption algorithms negotiated.

```
Initiator (Server)                Responder (Trust Anchor)
------------------                ------------------------

HDR, SK {N(GATE_PASS),
    N(GP_SECRET)
    }                          -->


                               <-- HDR, SK {}
```

Server distribution of GATE_PASS and GP_SECRET to Trust Anchor

The INFORMATIONAL message exchange described above is protected by the existing IKEv2 SA between the server and trust anchor. Server MUST maintain mappings between GATE_PASS, GP_SECRET and GP_KEY for the items present in its history. Trust anchor need not maintain any such history and keep only the latest GATE_PASS and GP_SECRET.

10. IKEv2 First Init exchange with Server and Server Redirection

Apart from the items specified in section 3 of [IKEv2REDIRECT] this exchange includes N(GATE_PASS_SUPPORTED) and N(PROXY_TICKET_SUPPORTED) payloads.

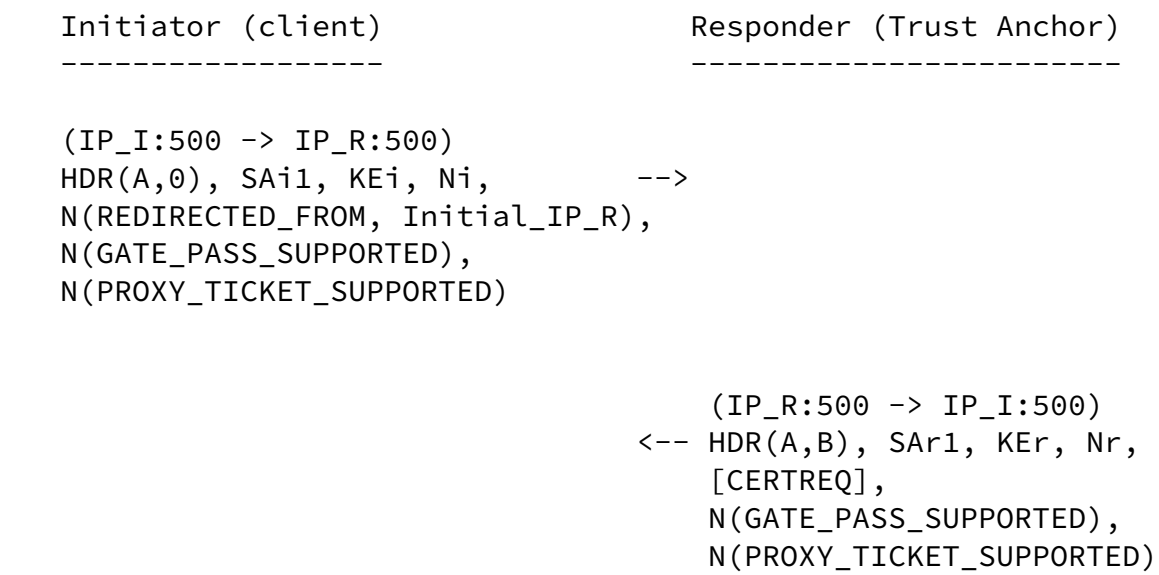
```
Initiator(client)                 Responder (Server)
-----------------                 ------------------

(IP_I:500 -> Initial_IP_R:500)
HDR(A,0), SAi1, KEi, Ni,    -->
N(REDIRECT_SUPPORTED),
N(GATE_PASS_SUPPORTED),
N(PROXY_TICKET_SUPPORTED)


                                    (Initial_IP_R:500 -> IP_I:500)
                               <-- HDR(A,0), N(REDIRECT,
                                    Trust_Anchor_ID,
                                    Ni_data),
                                    N(GATE_PASS_SUPPORTED),
                                    N(PROXY_TICKET_SUPPORTED)
```

IKEv2 First Init exchange with Server and Server Redirection

Subsequently client initiates a new IKE_SA_INIT exchange with the

trust anchor listed in the REDIRECT payload.

Client MUST include N(GATE_PASS_SUPPORTED)and N(PROXY_TICKET_SUPPORTED) payloads in this exchange if both the client and server (as indicated in the earlier redirect from server) support this feature.



IKEv2 Init exchange with Trust Anchor

11. IKEv2 Second Redirection by the Trust Anchor during IKE_AUTH Exchange

Clients and trust anchors have to adhere to the rules specified in Section 6 of [IKEv2REDIRECT].

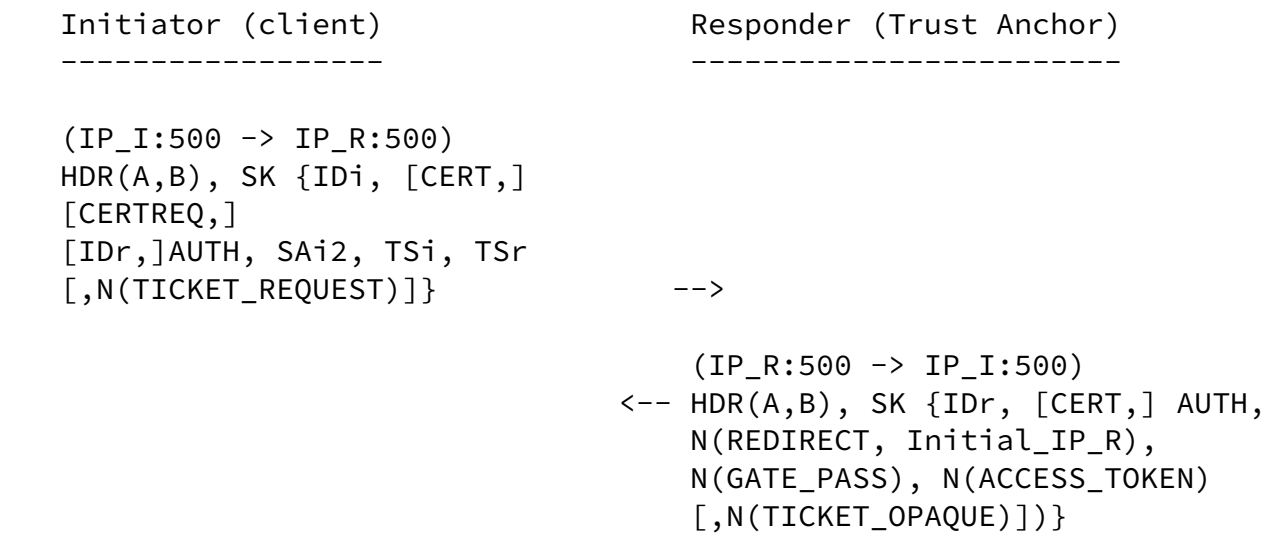
Additionally, trust anchor MAY include N(GATE_PASS), N(ACCESS_TOKEN) and N(TICKET_OPAQUE) payloads along with N(REDIRECT, Initial_IP_R)in the IKE_AUTH if client specified these capabilities in the INIT exchange. In such cases trust anchor MUST use the same IP of

Initial_IP_R received from N(REDIRECTED_FROM, Initial_IP_R) to redirect it back to the same server.

N(ACCESS_TOKEN) payloads MUST be included only after verifying client's identity.

If trust anchor decides to include N(TICKET_OPAQUE) along with N(REDIRECT, Initial_IP_R) in the IKE_AUTH exchange it MUST also include N(GATE_PASS)and N(ACCESS_TOKEN) in the exchange.

If trust anchor had redirected the client with N(TICKET_OPAQUE) payload, client MUST use IKEv2 Session Resumption Exchange (Section 12) to resume the proxy session (established between client and trust anchor) with the server, otherwise client MUST proceed with IKE_SA_INIT exchange (Section 13) and SHOULD include N(GATE_PASS) payload in the exchange to prevent it from getting redirected again.

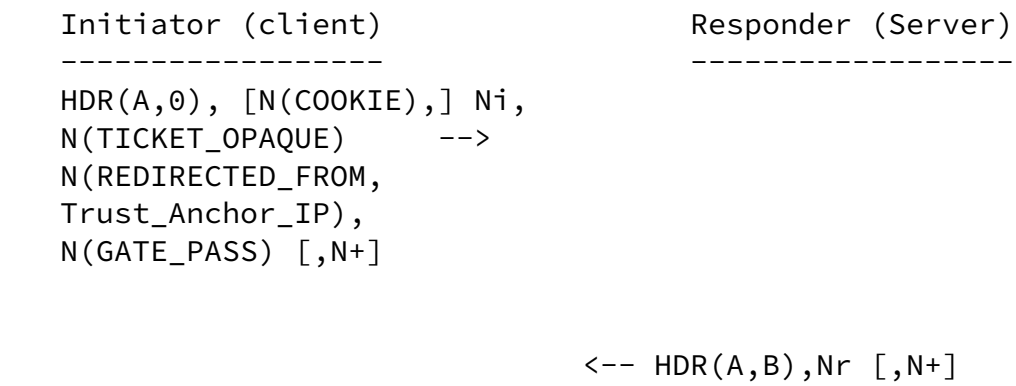


IKEv2 Auth exchange with Trust Anchor

12. IKEv2 Proxy Session Resumption Exchange with Server with N(GATE_PASS)

Client can resume a proxy session with server if it holds a valid PROXY_TICKET, GATE_PASS and ACCESS_TOKEN and in such cases client MUST include N(GATE_PASS) in the IKE_SESSION_RESUME exchange to indicate that the resumed session is a proxy session and the

identities (old IDi of client and new IDr of server) are authenticated separately using GATE_PASS in the IKE_AUTH exchange. Client and Server MUST use GATE_PASS based IKE_AUTH exchange (Section 14) to resume the proxy session. Server MUST not accept the Tickets with different IDr if N(GATE_PASS) is not included. Server MUST not accept the N(GATE_PASS) if Trust_Anchor_IP contained in N(REDIRECTED_FROM,Trust_Anchor_IP)is different from Trust Anchor IP contained in N(GATE_PASS). If N(GATE_PASS) is included in the IKE_SESSION_RESUME exchange then the IDr contained in the Ticket MUST be the same as Trust Anchor ID contained in the GATE_PASS otherwise this MUST be considered as normal IKE_SESSION_RESUME exchange and IDr MUST be the same as that of the responder (server) and both parties MUST use the IKE_AUTH Exchange mentioned in section 4.3.3 of IKEv2 Session Resumption [IKEv2RESUMPTION].



IKEv2 Proxy Session Resumption Exchange with Server

13. IKEv2 Second Init exchange with Server with N(GATE_PASS)

Clients MUST includes N(GATE_PASS) payload in this exchange to prevent it from getting redirected again.

In such cases server MAY not include [CERTREQ] in the INIT reply as the proof is based on GATE_PASS based authentication.

```
(IP_I:500 -> IP_R:500)
HDR(A,0), SAi1, KEi, Ni,     -->
N(REDIRECTED_FROM,
Trust_Anchor_IP),
N(GATE_PASS)


                                    (IP_R:500 -> IP_I:500)
                                <-- HDR(A,B), SAr1, KEr, Nr,
```

IKEv2 Second INIT exchange with Server

14. IKEv2 Auth exchange with Server with N(GATE_PASS)

Client includes GATE_PASS in the AUTH exchange and compute AUTH using ACCESS_TOKEN as the pre-shared key.

Server computes ACCESS_TOKEN based on IDi, GATE_PASS and GP_SECRET the same way TRUST_ANCHOR computed it earlier.

ACCESS_TOKEN = prf(encrypt(GP_SECRET, GATE_PASS | Client's Identity

(IDi)))

Client MUST use the same identity that it used during its earlier IKE_AUTH exchange with trust anchor. Otherwise the pre-shared key computed will be different and IKE_AUTH will fail.

Where prf() and encrypt() are the pseudo random number generator algorithm and encryption algorithm negotiated between server and trust anchor during IKE_SA. It MUST be picked from the same IKE channel through which the server has distributed GP_SECRET and GATE_PASS to Trust Anchor.

If server prefers to provide AUTH based on GATE_PASS it MUST include the same GATE_PASS received in the reply.

```
Initiator (client)                Responder (Server)
------------------                ------------------
(IP_I:500 -> IP_R:500)
```

```
HDR(A,B), SK {IDi,
N(GATE_PASS)
[IDr,]AUTH, SAi2,
TSi, TSr}                    -->

                                    (IP_R:500 -> IP_I:500)
                             <-- HDR(A,B), SK {IDr, AUTH,
                                 N(GATE_PASS),
                                 SAr2, TSi, TSr}
```

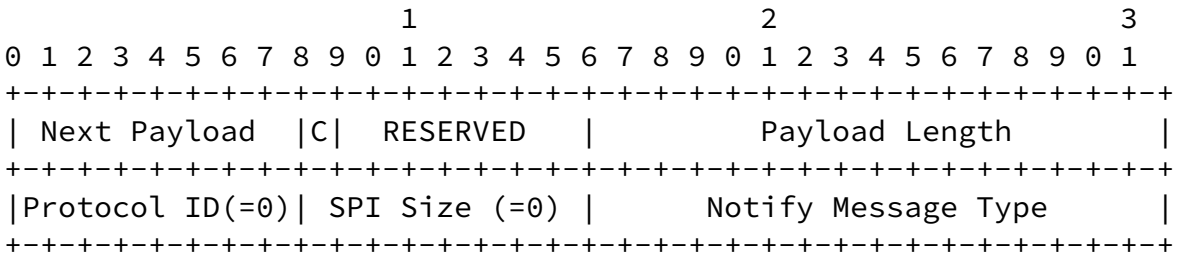
IKEv2 AUTH exchange with Server

The responder MAY include N(AUTH_LIFETIME) [RFC4478] in the Auth reply. Such cases client MUST not include N(GATE_PASS) in the INIT exchange when it restarts INIT exchange. Instead it MAY include N(GATE_PASS_SUPPORTED) if it wants to continue using this mechanism.

15. IKEv2 Auth Offload and Proxy Session Resumption Messages

15.1. GATE_PASS_SUPPORTED

The GATE_PASS_SUPPORTED payload is included in the initial IKE_SA_INIT request by the initiator to indicate support for the IKEv2 auth offload mechanism described in this memo.



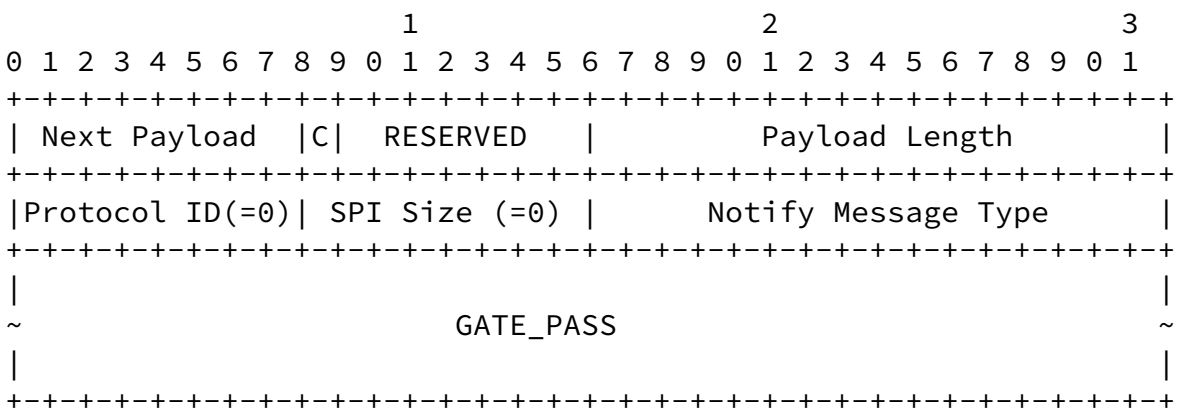
GATE_PASS_SUPPORTED

The 'Next Payload', 'Payload Length', 'Protocol ID', 'SPI Size' and the 'Notify Message Type' fields are the same as described in Section 3.10 of RFC 4306. The 'SPI Size' field MUST be set to 0 to indicate that the SPI is not present in this message. The 'Protocol ID' MUST be set to 0, since the notification is not specific to a particular security association.

The 'Payload Length' field is set to the length in octets of the entire payload, including the generic payload header. The Notify Message Type' field is set to indicate the GATE_PASS_SUPPORTED payload.

15.2. GATE_PASS

GATE_PASS payload is included in the initial IKE_SA_AUTH reply by the trust anchor to client, or the initial IKE_SA_AUTH request by the client to server to carry the GATE_PASS.



GATE_PASS

The 'Next Payload', 'Payload Length', 'Protocol ID', 'SPI Size' and

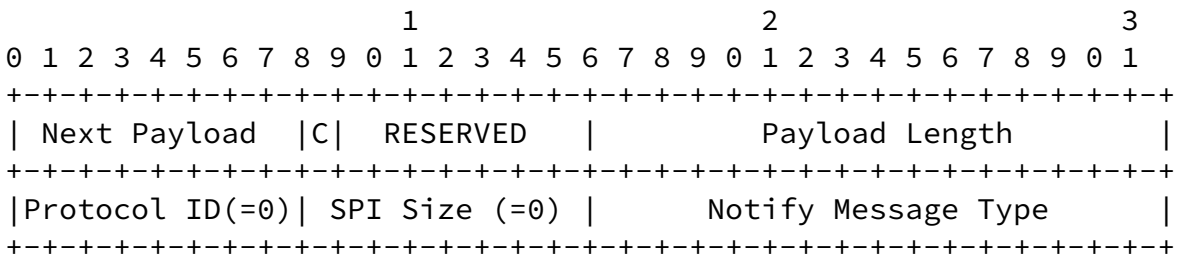
the 'Notify Message Type' fields are the same as described in Section 3.10 of RFC 4306. The 'SPI Size' field MUST be set to 0 to indicate that the SPI is not present in this message. The 'Protocol ID' MUST be set to 0, since the notification is not specific to a particular

security association.

The 'Payload Length' field is set to the length in octets of the entire payload, including the generic payload header. The Notify Message Type' field is set to indicate the GATE_PASS payload.

15.3. PROXY_TICKET_SUPPORTED

The PROXY_TICKET_SUPPORTED payload is included in the initial IKE_SA_INIT request by the initiator to indicate support for the IKEv2 proxy session resumption mechanism described in this memo. Note that proxy session resumption is dependent on auth offload capability and client MUST also include GATE_PASS_SUPPORTED payload in the exchange.



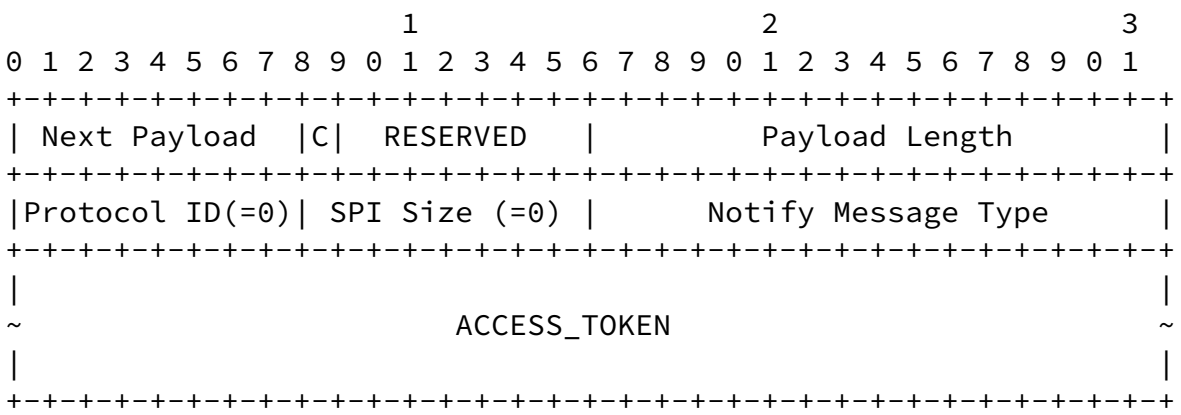
PROXY_TICKET_SUPPORTED

The 'Next Payload', 'Payload Length', 'Protocol ID', 'SPI Size' and the 'Notify Message Type' fields are the same as described in Section 3.10 of RFC 4306. The 'SPI Size' field MUST be set to 0 to indicate that the SPI is not present in this message. The 'Protocol ID' MUST be set to 0, since the notification is not specific to a particular security association.

The 'Payload Length' field is set to the length in octets of the entire payload, including the generic payload header. The Notify Message Type' field is set to indicate the GATE_PASS_SUPPORTED payload.

15.4. ACCESS_TOKEN

ACCESS_TOKEN payload is included in the initial IKE_SA_AUTH reply by the trust anchor to client.



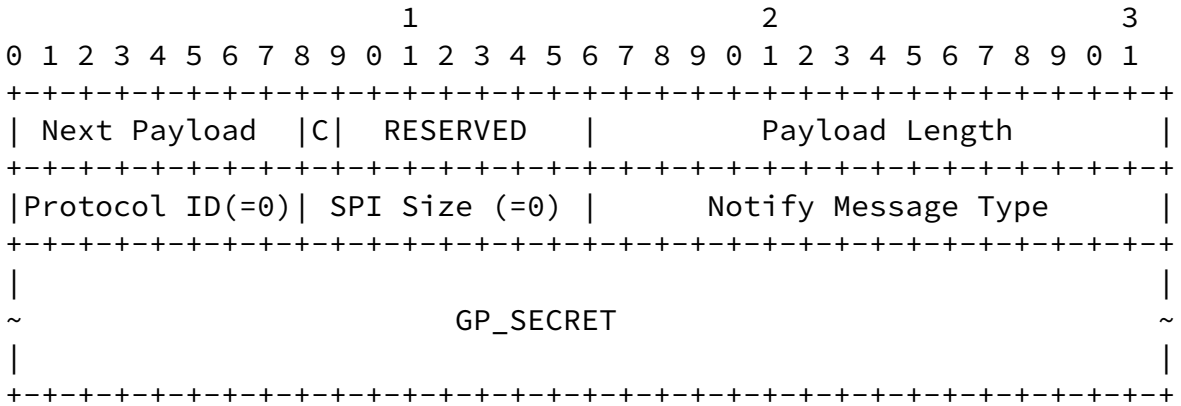
ACCESS_TOKEN

The 'Next Payload', 'Payload Length', 'Protocol ID', 'SPI Size' and the 'Notify Message Type' fields are the same as described in Section 3.10 of RFC 4306. The 'SPI Size' field MUST be set to 0 to indicate that the SPI is not present in this message. The 'Protocol ID' MUST be set to 0, since the notification is not specific to a particular security association.

The 'Payload Length' field is set to the length in octets of the entire payload, including the generic payload header. The Notify Message Type' field is set to indicate the ACCESS_TOKEN payload.

15.5. GP_SECRET

Server uses an INFORMATIONAL exchange to distribute GP_SECRET.



GP_SECRET

The 'Next Payload', 'Payload Length', 'Protocol ID', 'SPI Size' and the 'Notify Message Type' fields are the same as described in Section 3.10 of RFC 4306. The 'SPI Size' field MUST be set to 0 to indicate that the SPI is not present in this message. The 'Protocol ID' MUST be set to 0, since the notification is not specific to a particular security association.

The 'Payload Length' field is set to the length in octets of the entire payload, including the generic payload header. The Notify Message Type' field is set to indicate the GP_SECRET payload.

16. Acknowledgements

.

17. IANA Considerations

Section 15 defines five new IKEv2 notifications whose Message Type values are to be allocated from the "IKEv2 Notify Message Types - Status Types" registry.

- GATE_PASS_SUPPORTED
- PROXY_TICKET_SUPPORTED
- GATE_PASS
- ACCESS_TOKEN
- GP_SECRET

18. Security Considerations

Servers MUST periodically update GATE_PASS. This is required to prevent clients from reusing the tokens. It is a good idea to force clients to reauthenticate when the GATE_PASS expires using N(AUTH_LIFETIME). [RFC4478]

19. References

19.1. Normative References

[IKEv2REDIRECT]
Devarapalli, V. and K. Weniger, "Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)", November 2009.

[IKEv2RESUMPTION]
Sheffer, Y. and H. Tschofenig, "Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption", December 2009.

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

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4478] Nir, Y., "Repeated Authentication in Internet Key Exchange (IKEv2) Protocol", April 2006.

19.2. Informative References

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC4718] Eronen, P. and P. Hoffman, "IKEv2 Clarifications and Implementation Guidelines", RFC 4718, October 2006.

Authors' Addresses

Padmakumar A.V.

Cisco Systems, Inc.
O'Shaugnessy Road
Bangalore, Karnataka 560025
India

Phone: +91 80 4103 3184
Email: paav@cisco.com

Manikchand Bafna
Cisco Systems, Inc.
O'Shaugnessy Road
Bangalore, Karnataka 560025
India

Phone: +91 80 4154 1365
Email: manikrb@cisco.com

Pratima Sethi
Cisco Systems, Inc.
O'Shaugnessy Road
Bangalore, Karnataka 560025
India

Phone: +91 80 4154 1654
Email: psethi@cisco.com