Network Working Group	Y. Sheffer	
Internet-Draft	Check Point	
Intended status: Standards Track	H. Tschofenig	
Expires: January 13, 2009	Nokia Siemens Networks	
	L. Dondeti	
	V. Narayanan	
	QUALCOMM, Inc.	
	July 12, 2008	

TOC

IPsec Gateway Failover Protocol
draft-sheffer-ipsec-failover-04.txt

Status of this Memo

Abstract

The Internet Key Exchange version 2 (IKEv2) protocol has a certain computational and communication overhead with respect to the number of round-trips required and the cryptographic operations involved. In remote access situations, the Extensible Authentication Protocol is used for authentication, which adds several more round trips and therefore latency.
To re-establish security associations (SA) upon a failure recovery condition is time consuming, especially when an IPsec peer, such as a VPN gateway, needs to re-establish a large number of SAs with various

end points. A high number of concurrent sessions might cause additional problems for an IPsec peer during SA re-establishment.
In many failure cases it would be useful to provide an efficient way to resume an interrupted IKE/IPsec session. This document 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.
A client can reconnect to a gateway from which it was disconnected, or alternatively migrate to another gateway that is associated with the previous one. The proposed approach conveys IKEv2 state information, in the form of an encrypted ticket, to a VPN client that is later presented to the VPN gateway for re-authentication. The encrypted ticket can only be decrypted by the VPN gateway in order to restore state for faster session setup.

Table of Contents

---

1. Introduction

TOC

The Internet Key Exchange version 2 (IKEv2) protocol protocol has a certain computational and communication overhead with respect to the number of round-trips required and the cryptographic operations involved. In particular the Extensible Authentication Protocol is used for authentication in remote access cases, which increases latency. To re-establish security associations (SA) upon a failure recovery condition is time-consuming, especially when an IPsec peer, such as a VPN gateway, needs to re-establish a large number of SAs with various end points. A high number of concurrent sessions might cause additional problems for an IPsec peer.

In many failure cases it would be useful to provide an efficient way to resume an interrupted IKE/IPsec session. This document 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.

A client can reconnect to a gateway from which it was disconnected, or alternatively migrate to another gateway that is associated with the previous one. This document proposes to maintain IKEv2 state in a "ticket", an opaque data structure created and used by a server and stored by a client, which the client cannot understand or tamper with. The IKEv2 protocol is extended to allow a client to request and present a ticket. When two gateways mutually trust each other, one can accept a ticket generated by the other.

This approach is similar to the one taken by TLS session resumption [RFC4507] (Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State," May 2006.) with the required adaptations for IKEv2, e.g., to

accommodate the two-phase protocol structure. We have borrowed heavily from that specification.

1.1. Goals

TOC

The high-level goal of this extension is to provide an IPsec failover solution, according to the requirements defined in [I-D.vidya-ipsec-failover-ps] (Narayanan, V., "IPsec Gateway Failover and Redundancy - Problem Statement and Goals," December 2007.). Specifically, the proposed extension should allow IPsec sessions to be recovered from failures in remote access scenarios, in a more efficient manner than the basic IKE solution. This efficiency is primarily on the gateway side, since the gateway might have to deal with many thousands of concurrent requests. We should enable the following cases:

- Failover from one gateway to another, where the two gateways do not share state but do have mutual trust. For example, the gateways may be operated by the same provider and share the same keying materials to access an encrypted ticket.

- Recovery from an intermittent connectivity, where clients reconnect into the same gateway. In this case, the gateway would typically have detected the clients' absence and removed the state associated with them.

- Recovery from a gateway restart, where clients reconnect into the same gateway.

The proposed solution should additionally meet the following goals:

- Using only symmetric cryptography to minimize CPU consumption.

- Allowing a gateway to push state to clients.

- Providing cryptographic agility.

- Having no negative impact on IKEv2 security features.

1.2. Non-Goals

TOC

The following are non-goals of this solution:

- Providing load balancing among gateways.

*Specifying how a client detects the need for a failover.

---

2. Terminology

TOC

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] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).
This document uses terminology defined in [RFC4301] (Kent, S. and K. Seo, "Security Architecture for the Internet Protocol," December 2005.), [RFC4306] (Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," December 2005.), and [RFC4555] (Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)," June 2006.). In addition, this document uses the following terms:

Secure domain: A secure domain comprises a set of gateways that are able to resume an IKEv2 session that may have been established by any other gateway within the domain. All gateways in the secure domain are expected to share some secrets, so that they can generate an IKEv2 ticket, verify the validity of the ticket and extract the IKEv2 policy and session key material from the ticket.

IKEv2 ticket: An IKEv2 ticket is a data structure that contains all the necessary information that allows any gateway within the same secure domain as the gateway that created the ticket to verify the validity of the ticket and extract IKEv2 policy and session keys to re-establish an IKEv2 session.

Stateless failover: When the IKEv2 session state is stored at the client, the IKEv2 responder is "stateless" until the client restores the SA with one of the gateways within the secure domain; thus, we refer to SA resumption with SA storage at the client as stateless session resumption.

Stateful failover: When the infrastructure maintains IKEv2 session state, we refer to the process of IKEv2 SA re-establishment as stateful session resumption.

TOC

3. Usage Scenarios

This specification envisions two usage scenarios for efficient IKEv2 and IPsec SA session re-establishment.
The first is similar to the use case specified in Section 1.1.3 of the IKEv2 specification [RFC4306] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.), where the IPsec tunnel mode is used to establish a secure channel between a remote access client and a gateway; the traffic flow may be between the client and entities beyond the gateway.
The second use case focuses on the usage of transport (or tunnel) mode to secure the communicate between two end points (e.g., two servers). The two endpoints have a client-server relationship with respect to a protocol that runs using the protections afforded by the IPsec SA.

3.1. Recovering from a Remote Access Gateway Failover

TOC

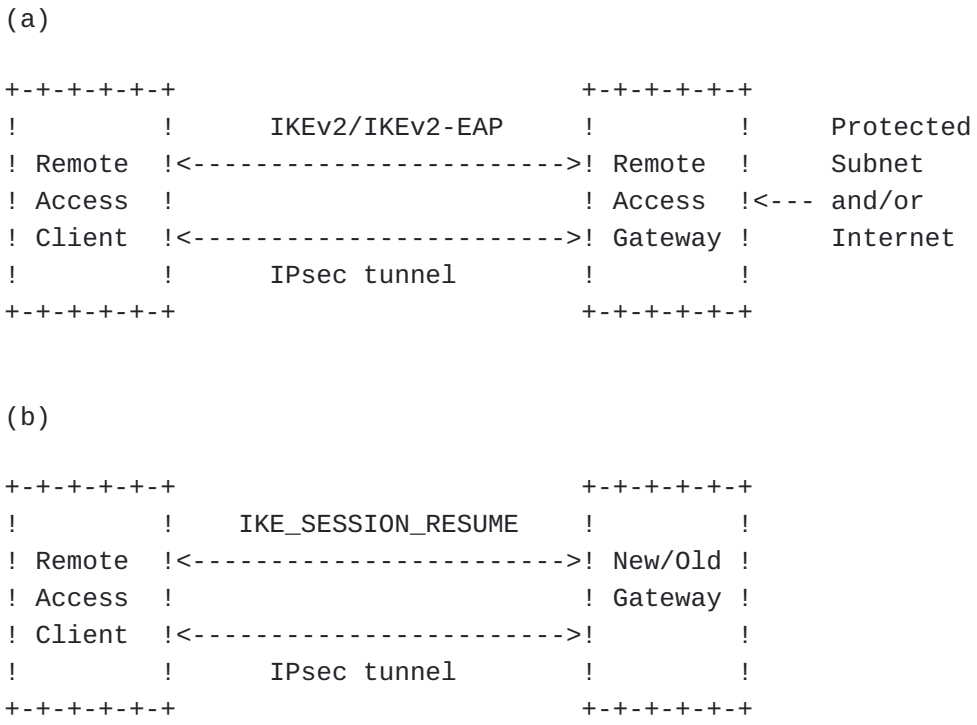

Figure 1: Remote Access Gateway Failure

In this scenario, an end-host (an entity with a host implementation of IPsec [RFC4301] (Kent, S. and K. Seo, “Security Architecture for the Internet Protocol,” December 2005.)) establishes a tunnel mode IPsec SA with a gateway in a remote network using IKEv2. The end-host in this scenario is sometimes referred to as a remote access client. When the remote gateway fails, all the clients associated with the gateway either need to re-establish IKEv2 sessions with another gateway within the same secure domain of the original gateway, or with the original gateway if the server is back online soon.

The clients may choose to establish IPsec SAs using a full IKEv2 exchange or the IKE_SESSION_RESUME exchange (shown in Figure 1 (Remote Access Gateway Failure)).

In this scenario, the client needs to get an IP address from the remote network so that traffic can be encapsulated by the remote access gateway before reaching the client. In the initial exchange, the gateway may acquire IP addresses from the address pool of a local DHCP server. The new gateway that a client gets associated may not receive addresses from the same address pool. Thus, the session resumption protocol needs to support the assignment of a new IP address.

The protocol defined in this document supports the re-allocation of an IP address to the client, if this capability is provided by the network. For example, if routing tables are modified so that traffic is rerouted through the new gateway. This capability is implicit in the use of the IKE Config mechanism, which allows the client to present its existing IP address and receive the same address back, if allowed by the gateway.

The protocol defined here supports both stateful and stateless scenarios. In other words, tickets can be stored wholly on the client, or the ticket can be stored on the gateway (or in a database shared between multiple gateways), with the client only presenting a handle that identifies a particular ticket. In fact these scenarios are transparent to the protocols, with the only change being the non-mandatory ticket format.

3.2. Recovering from an Application Server Failover

TOC

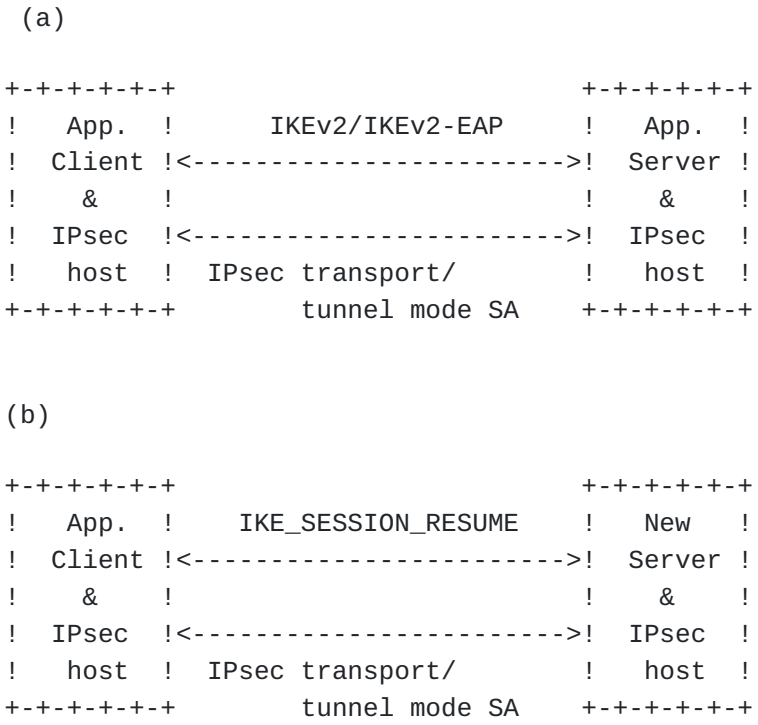

Figure 2: Application Server Failover

The second usage scenario is as follows: two entities with IPsec host implementations establish an IPsec transport or tunnel mode SA between themselves; this is similar to the model described in Section 1.1.2. of [RFC4306] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.). At the application level, one of the entities is always the client and the other is a server. From that view point, the IKEv2 exchange is always initiated by the client. This allows the Initiator (the client) to authenticate itself using EAP, as long as the Responder (or the application server) allows it.

If the application server fails, the client may find other servers within the same secure domain for service continuity. It may use a full IKEv2 exchange or the IKE_SESSION_RESUME exchange to re-establish the IPsec SAs for secure communication required by the application layer signaling.

The client-server relationship at the application layer ensures that one of the entities in this usage scenario is unambiguously always the Initiator and the other the Responder. This role determination also allows the Initiator to request an address in the Responder's network using the Configuration Payload mechanism of the IKEv2 protocol. If the client has thus received an address during the initial IKEv2 exchange, when it associates with a new server upon failure of the original server, it needs to request an address, specifying its assigned

address. The server may allow the client to use the original address or if it is not permitted to use that address, assign a new address.

---

4. Protocol Details

TOC

This section provides protocol details and contains the normative parts. This document defines two protocol exchanges, namely requesting a ticket and presenting a ticket. Section 4.1 (Requesting a Ticket) describes the procedure to request a ticket and Section 4.2 (Presenting a Ticket) illustrates how to present a ticket.

4.1. Requesting a Ticket

TOC

A client MAY request a ticket in the following exchanges:

- In an IKE_AUTH exchange, as shown in the example message exchange in Figure 3 (Example Message Exchange for Requesting a Ticket) below.
- In a CREATE_CHILD_SA exchange, when an IKE SA is rekeyed.
- In an Informational exchange, if the gateway previously replied with an N(TICKET_ACK) instead of providing a ticket.
- In an Informational exchange, when the ticket lifetime is about to expire.
- In an IKE_SESSION_RESUME exchange, see Section 4.2.3 (Requesting a ticket during resumption).

Normally, a client requests a ticket in the third message of an IKEv2 exchange (the first of IKE_AUTH). Figure 3 (Example Message Exchange for Requesting a Ticket) shows the message exchange for this typical case.

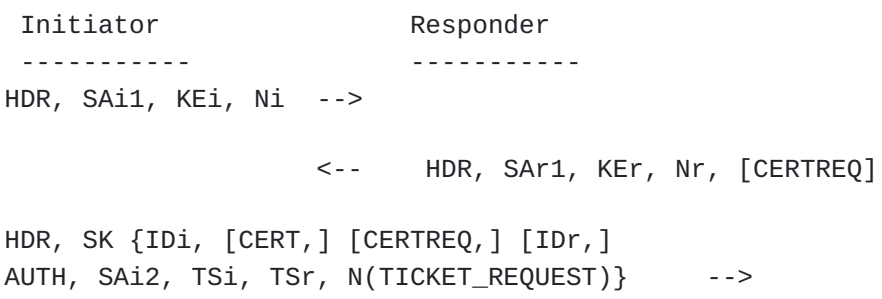

Figure 3: Example Message Exchange for Requesting a Ticket

The notification payloads are described in Section 4.3 (IKE Notifications). The above is an example, and IKEv2 allows a number of variants on these messages. A complete description of IKEv2 can be found in [RFC4718] (Eronen, P. and P. Hoffman, "IKEv2 Clarifications and Implementation Guidelines," October 2006.).
When an IKEv2 responder receives a request for a ticket using the N(TICKET_REQUEST) payload it MUST perform one of the following operations if it supports the extension defined in this document:

- it creates a ticket and returns it with the N(TICKET_OPAQUE) payload in a subsequent message towards the IKEv2 initiator. This is shown in Figure 4 (Receiving a Ticket).

- it returns an N(TICKET_NACK) payload, if it refuses to grant a ticket for some reason.

- it returns an N(TICKET_ACK), if it cannot grant a ticket immediately, e.g., due to packet size limitations. In this case the client MAY request a ticket later using an Informational exchange, at any time during the lifetime of the IKE SA.

Provided the IKEv2 exchange was successful, the IKEv2 initiator can accept the requested ticket. The ticket may be used later with an IKEv2 responder that supports this extension. Figure 4 (Receiving a Ticket) shows how the initiator receives the ticket.

```
Initiator                Responder
-----------              -----------
       <--    HDR, SK {IDr, [CERT,] AUTH, SAr2, TSi,
                  TSr, N(TICKET_OPAQUE) [,N(TICKET_GATEWAY_LIST)]}
```

Figure 4: Receiving a Ticket

4.2. Presenting a Ticket

TOC

Following a communication failure, a client re-initiates an IKE exchange to the same gateway or to a different one, and includes a ticket in the first message. A client MAY initiate a regular (non-ticket-based) IKEv2 exchange even if it is in possession of a valid ticket. A client MUST NOT present a ticket after the ticket's lifetime has expired.

It is up to the client's local policy to decide when the communication with the IKEv2 responder is seen as interrupted and a new exchange needs to be initiated and the session resumption procedure to be initiated.

Tickets are intended for one-time use: a client MUST NOT reuse a ticket, either with the same or with a different gateway. A gateway SHOULD reject a reused ticket. Note, however, that a gateway can elect not to retain a list of already-used tickets. Potential replay attacks on such gateways are mitigated by the cookie mechanism described in Section 4.2.2 (Presenting a Ticket: The DoS Case).

This document specifies a new IKEv2 exchange type called IKE_SESSION_RESUME whose value is TBA by IANA. This exchange is somewhat similar to the IKE_AUTH exchange, and results in the creation of a Child SA. The client SHOULD NOT use this exchange type unless it knows that the gateway supports it, either through configuration, by out-of-band means or by using the Gateway List provision.

```
Initiator                Responder
-----------              -----------
HDR, Ni, N(TICKET_OPAQUE), [N+,]
     SK {IDi, [IDr,] SAi2, TSi, TSr [, CP(CFG_REQUEST)]} -->
```

The exchange type in HDR is set to 'IKE_SESSION_RESUME'.

See Section 4.2.1 (Protection of the IKE_SESSION_RESUME Exchange) for details on computing the protected (SK) payload.

When the IKEv2 responder receives a ticket using the N(TICKET_OPAQUE) payload it MUST perform one of the following steps if it supports the extension defined in this document:

*If it is willing to accept the ticket, it responds as shown in Figure 5 (IKEv2 Responder accepts the ticket).

*It responds with an unprotected N(TICKET_NACK) notification, if it rejects the ticket for any reason. In that case, the initiator should re-initiate a regular IKE exchange. One such case is when the responder receives a ticket for an IKE SA that has previously been terminated on the responder itself, which may indicate inconsistent state between the IKEv2 initiator and the responder. However, a responder is not required to maintain the state for terminated sessions.

*When the responder receives a ticket for an IKE SA that is still active and if the responder accepts it, then the old SAs SHOULD be silently deleted without sending a DELETE informational exchange.

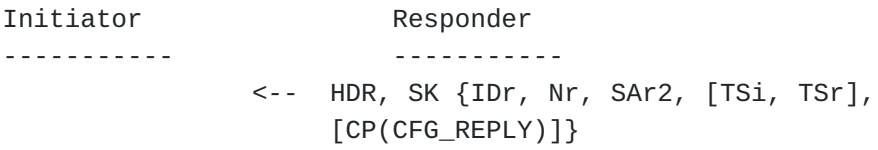

Figure 5: IKEv2 Responder accepts the ticket

Again, the exchange type in HDR is set to 'IKE_SESSION_RESUME'.
The SK payload is protected using the cryptographic parameters derived from the ticket, see Section 4.2.1 (Protection of the IKE_SESSION_RESUME Exchange) below.
At this point a new IKE SA is created by both parties, see Section 4.6 (Processing Guidelines for IKE SA Establishment). This is followed by normal derivation of a child SA, per Sec. 2.17 of [RFC4306] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.).

TOC

4.2.1. Protection of the IKE_SESSION_RESUME Exchange

The two messages of this exchange are protected by a "subset" IKE SA. The key material is derived from the ticket, as follows:

```
{SK_d2 | SK_ai | SK_ar | SK_ei | SK_er} = prf+(SK_d_old, Ni)
```

where SK_d_old is the SK_d value of the original IKE SA, as retrieved from the ticket. Ni guarantees freshness of the key material. SK_d2 is used later to derive the new IKE SA, see Section 4.6 (Processing Guidelines for IKE SA Establishment).
See [RFC4306] (Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," December 2005.) for the notation. "prf" is determined from the SA value in the ticket.

4.2.2. Presenting a Ticket: The DoS Case

TOC

When receiving the first message of the IKE_SESSION_RESUME exchange, the gateway may decide that it is under a denial-of-service attack. In such a case, the gateway SHOULD defer the establishment of session state until it has verified the identity of the client. We use a variation of the IKEv2 Cookie mechanism, whereby the cookie is protected.
In the two messages that follow, the gateway responds that it is unwilling to resume the session until the client is verified, and the client resubmits its first message, this time with the cookie:

```
      Initiator                Responder
      -----------              -----------
                     <-- HDR, SK{N(COOKIE)}

     HDR, Ni, N(TICKET_OPAQUE), [N+,]
          SK {N(COOKIE), IDi, [IDr,] SAi2, TSi, TSr [, CP(CFG_REQUEST)]} -->
```

Assuming the cookie is correct, the gateway now replies normally.
This now becomes a 4-message exchange. The entire exchange is protected as defined in Section 4.2.1 (Protection of the IKE_SESSION_RESUME Exchange).
See Sec. 2.6 and Sec. 3.10.1 of [RFC4306] (Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," December 2005.) for more guidance regarding the usage and syntax of the cookie. Note that the cookie is completely independent of the IKEv2 ticket.

4.2.3. Requesting a ticket during resumption

TOC

When resuming a session, a client will typically request a new ticket immediately, so it is able to resume the session again in the case of a second failure. Therefore, the N(TICKET_REQUEST), N(TICKET_OPAQUE) and N(TICKET_GATEWAY_LIST) notifications may be piggybacked as protected payloads to the IKE_SESSION_RESUME exchange.
The returned ticket (if any) will correspond to the IKE SA created per the rules described in Section 4.6 (Processing Guidelines for IKE SA Establishment).

4.3. IKE Notifications

TOC

This document defines a number of notifications. The notification numbers are TBA by IANA.

Notification Name	Number	Data
TICKET_OPAQUE	TBA1	See Section 4.4 (TICKET_OPAQUE Notify Payload)
TICKET_REQUEST	TBA2	None
TICKET_ACK	TBA3	None
TICKET_NACK	TBA4	None
TICKET_GATEWAY_LIST	TBA5	See Section 4.5 (TICKET_GATEWAY_LIST Notify Payload)

4.4. TICKET_OPAQUE Notify Payload

TOC

The data for the TICKET_OPAQUE Notify payload consists of the Notify message header, a lifetime field and the ticket itself. The four octet lifetime field contains the number of seconds until the ticket expires (encoded as an unsigned integer). Section 5.2 (Ticket Format) describes a possible ticket format, and Section 5.3 (Ticket Identity and Lifecycle) offers further guidelines regarding the ticket's lifetime.

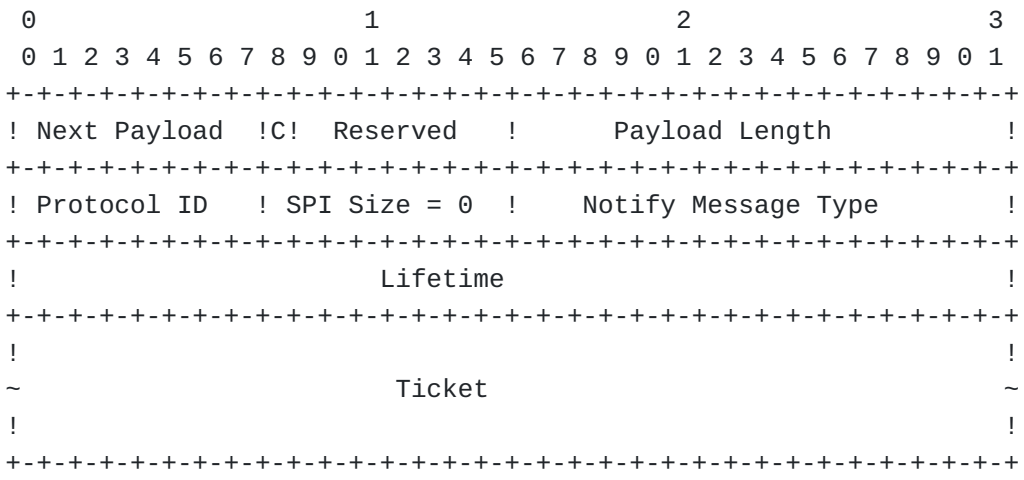

Figure 6: TICKET_OPAQUE Notify Payload

4.5. TICKET_GATEWAY_LIST Notify Payload

TOC

The TICKET_GATEWAY_LIST Notify payload contains the Notify payload header followed by a sequence of one or more gateway identifiers, each of the format depicted in Figure 8 (Gateway Identifier for One Gateway).

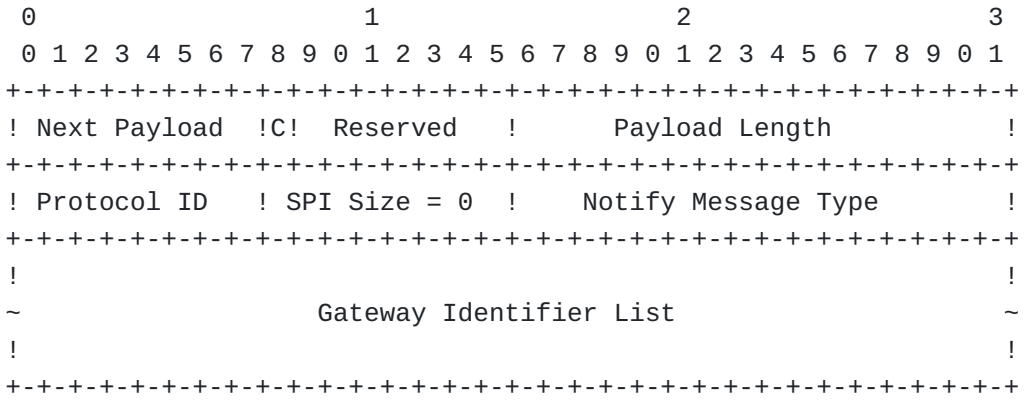

Figure 7: TICKET_GATEWAY_LIST Notify Payload

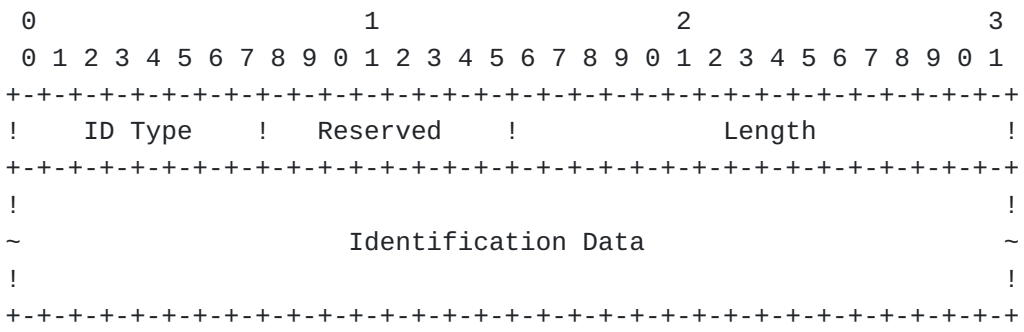

Figure 8: Gateway Identifier for One Gateway

ID Type:
The ID Type contains a restricted set of the IKEv2 ID payloads (see [RFC4306] (Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," December 2005.), Section 3.5). Allowed ID types are: ID_IPV4_ADDR, ID_IPV6_ADDR, ID_FQDN and the various reserved values.

Reserved:
This field MUST be sent as 0 and MUST be ignored when received.

Length:
The length field indicates the total size of the Identification data.

Identification Data:
The Identification Data field is of variable length and depends on the ID type. The length is not necessarily a multiple of 4.

4.6. Processing Guidelines for IKE SA Establishment

TOC

When a ticket is presented, the gateway parses the ticket to retrieve the state of the old IKE SA, and the client retrieves this state from

its local store. Both peers now create state for the new IKE SA as follows:

- *The SA value (transforms etc.) is taken directly from the ticket.
- *The sequence numbers are reset to 0.
- *The IDi value is obtained from the ticket.
- *The IDr value is obtained from the new exchange. The gateway MAY make policy decisions based on the IDr value encoded in the ticket.
- *The SPI values are created anew, similarly to a regular IKE exchange. SPI values from the ticket SHOULD NOT be reused. This restriction is to avoid problems caused by collisions with other SPI values used already by the initiator/responder. The SPI value should only be reused if collision avoidance can be ensured through other means.

The cryptographic material is refreshed based on the ticket and the nonce values, Ni, and Nr, from the current exchange. A new SKEYSEED value is derived as follows:

```
SKEYSEED = prf(SK_d2, Ni | Nr)
```

where SK_d2 was computed earlier (Section 4.2.1 (Protection of the IKE_SESSION_RESUME Exchange)).
The keys are derived as follows, unchanged from IKEv2:

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

where SPIi, SPIr are the SPI values created in the new IKE exchange. See [RFC4306] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.) for the notation. "prf" is determined from the SA value in the ticket.

---

5. The IKE Ticket

TOC

This section lists the required contents of the ticket, and recommends a non-normative format. This is followed by a discussion of the ticket's lifecycle.

5.1. Ticket Contents

TOC

The ticket MUST encode at least the following state from an IKE SA. These values MUST be encrypted and authenticated.

- *IDi, IDr.
- *SPIi, SPIr.
- *SAr (the accepted proposal).
- *SK_d.

In addition, the ticket MUST encode a protected ticket expiration value.

5.2. Ticket Format

TOC

This document does not specify a mandatory-to-implement or a mandatory-to-use ticket format. The following format is RECOMMENDED, if interoperability between gateways is desired.

```
struct {
    [authenticated] struct {
        octet format_version;    // 1 for this version of the protocol
        octet reserved[3];       // sent as 0, ignored by receiver.
        octet key_id[8];         // arbitrary byte string
        opaque IV[0..255];       // actual length (possibly 0) depends
                                 // on the encryption algorithm

        [encrypted] struct {
            opaque IDi, IDr;     // the full payloads
            octet SPIi[8], SPIr[8];
            opaque SA;           // the full SAr payload
            octet SK_d[0..255];  // actual length depends on SA value
            int32 expiration;    // an absolute time value, seconds
                                 // since Jan. 1, 1970
        } ikev2_state;
    } protected_part;
    opaque MAC[0..255];          // the length (possibly 0) depends
                                 // on the integrity algorithm
} ticket;
```

Note that the key defined by "key_id" determines the encryption and authentication algorithms used for this ticket. Those algorithms are unrelated to the transforms defined by the SA payload.
The reader is referred to a recent draft [I-D.rescorla-stateless-tokens] (Rescorla, E., “How to Implement Secure (Mostly) Stateless Tokens,” March 2007.) that recommends a similar (but not identical) ticket format, and discusses related security considerations in depth.

5.3. Ticket Identity and Lifecycle

TOC

Each ticket is associated with a single IKE SA. In particular, when an IKE SA is deleted, the client MUST delete its stored ticket.
A ticket is therefore associated with the tuple (IDi, IDr). The client MAY, however, present a ticket to other gateways that are willing to accept it. How a client discovers such gateways is outside the scope of this document.
The lifetime of the ticket carried in the N(TICKET_OPAQUE) notification SHOULD be the minimum of the IKE SA lifetime (per the gateway's local policy) and its re-authentication time, according to [RFC4478] (Nir, Y., “Repeated Authentication in Internet Key Exchange (IKEv2) Protocol,” April 2006.). Even if neither of these are enforced by the gateway, a finite lifetime MUST be specified for the ticket.

5.4. Exchange of Ticket-Protecting Keys

TOC

This document does not define an interoperable mechanism for the generation and distribution of the keys that protect IKE keys. Note that there are no significant performance requirements on such a protocol, as key rollover can be at a daily or even more leisurely rate.

---

6. IANA Considerations

TOC

This document requires a number of IKEv2 notification status types in Section 4.3 (IKE Notifications), to be registered by IANA. The corresponding registry was established by IANA.
The document defines a new IKEv2 exchange in Section 4.2 (Presenting a Ticket). The corresponding registry was established by IANA.

---

7. Security Considerations

TOC

This section addresses security issues related to the usage of a ticket.

7.1. Stolen Tickets

TOC

An eavesdropper or man-in-the-middle may try to obtain a ticket and use it to establish a session with the IKEv2 responder. This can happen in different ways: by eavesdropping on the initial communication and copying the ticket when it is granted and before it is used, or by listening in on a client's use of the ticket to resume a session. However, since the ticket's contents is encrypted and the attacker does not know the corresponding secret key (specifically, SK_d), a stolen ticket cannot be used by an attacker to succesfully resume a session. An IKEv2 responder MUST use strong encryption and integrity protection of the ticket to prevent an attacker from obtaining the ticket's contents, e.g., by using a brute force attack.

7.2. Forged Tickets

TOC

A malicious user could forge or alter a ticket in order to resume a session, to extend its lifetime, to impersonate as another user, or to gain additional privileges. This attack is not possible if the ticket is protected using a strong integrity protection algorithm.

7.3. Denial of Service Attacks

TOC

The key_id field, defined in the recommended ticket format, helps the server to detect tickets that it did not issue. However, an adversary could generate and send a large number of tickets to a gateway for verification. To minimize the possibility of such denial of service, ticket verification should be lightweight (e.g., using efficient symmetric key cryptographic algorithms).

TOC

7.4. Ticket Protection Key Management

A full description of the management of the keys used to protect the ticket is beyond the scope of this document. A list of RECOMMENDED practices is given below.

- The keys should be generated securely following the randomness recommendations in [RFC4086] (Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security," June 2005.).
- The keys and cryptographic protection algorithms should be at least 128 bits in strength.
- The keys should not be used for any other purpose than generating and verifying tickets.
- The keys should be changed regularly.
- The keys should be changed if the ticket format or cryptographic protection algorithms change.

7.5. Ticket Lifetime

TOC

An IKEv2 responder controls the lifetime of a ticket, based on the operational and security requirements of the environment in which it is deployed. The responder provides information about the ticket lifetime to the IKEv2 initiator, allowing it to manage its tickets.
An IKEv2 client may present a ticket in its possession to a gateway, even if the IKE SA associated with this ticket had previously been terminated by another gateway (the gateway that originally provided the ticket). Where such usage is against the local security policy, an Invalid Ticket List (ITL) may be used, see [I-D.rescorla-stateless-tokens] (Rescorla, E., "How to Implement Secure (Mostly) Stateless Tokens," March 2007.). Management of such lists is outside the scope of the current document. Note that a policy that requires tickets to have shorter lifetimes (e.g., 1 hour) significantly mitigates this risk.

7.6. Alternate Ticket Formats and Distribution Schemes

TOC

If the ticket format or distribution scheme defined in this document is not used, then great care must be taken in analyzing the security of the solution. In particular, if confidential information, such as a

secret key, is transferred to the client, it MUST be done using secure communication to prevent attackers from obtaining or modifying the key. Also, the ticket MUST have its integrity and confidentiality protected with strong cryptographic techniques to prevent a breach in the security of the system.

7.7. Identity Privacy, Anonymity, and Unlinkability

TOC

This document mandates that the content of the ticket MUST be encrypted in order to avoid leakage of information, such as the identities of an IKEv2 initiator and a responder. Thus, it prevents the disclosure of potentially sensitive information carried within the ticket.
When an IKEv2 initiator presents the ticket as part of the IKE_SESSION_RESUME exchange, confidentiality is not provided for the exchange. Although the ticket itself is encrypted there might still be a possibility for an on-path adversary to observe multiple exchange handshakes where the same ticket is used and therefore to conclude that they belong to the same communication end points. Administrators that use the ticket mechanism described in this document should be aware that unlinkability may not be provided by this mechanism. Note, however, that IKEv2 does not provide active user identity confidentiality for the IKEv2 initiator either.

7.8. Replay Protection in the IKE_SESSION_RESUME Exchange

TOC

A major design goal of this protocol extension has been the two-message exchange for session resumption. There is a tradeoff between this abbreviated exchange and replay protection. It is RECOMMENDED that the gateway should cache tickets, and reject replayed ones. However some gateways may not do that in order to reduce state size. In addition, an adversary may replay a ticket last presented to gateway A, into gateway B. The cookie-based mechanism (see Section 4.2.2 (Presenting a Ticket: The DoS Case)) mitigates these risks: a client may be required by the gateway to show that it knows the ticket's secret, before any state is committed on the gateway side. Note that this is a stronger guarantee than the regular IKE cookie mechanism, which only proves IP return routability of the client. This is enabled by including the cookie in the protected portion of the message.
For performance reasons, the cookie mechanism is optional, and invoked by the gateway only when it suspects that it is the subject of a denial-of-service attack.
In any case, a ticket replayed by an adversary only causes partial IKE state to be created on the gateway. The IKE exchange cannot be

completed and an IKE SA cannot be created unless the client knows the ticket's secret values.

8. Acknowledgements

TOC

We would like to thank Paul Hoffman, Pasi Eronen, Florian Tegeler, Yoav Nir and Tero Kivinen for their many helpful comments.

9. References

TOC

9.1. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC4306]	Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” RFC 4306, December 2005 (TXT).

9.2. Informative References

TOC

[I-D.friedman-ike-short-term-certs]	Friedman, A., “Short-Term Certificates,” draft-friedman-ike-short-term-certs-02 (work in progress), June 2007 (TXT).
[I-D.rescorla-stateless-tokens]	Rescorla, E., “How to Implement Secure (Mostly) Stateless Tokens,” draft-rescorla-stateless-tokens-01 (work in progress), March 2007 (TXT).
[I-D.vidya-ipsec-failover-ps]	Narayanan, V., “IPsec Gateway Failover and Redundancy - Problem Statement and Goals,” draft-vidya-ipsec-failover-ps-02 (work in progress), December 2007 (TXT).
[RFC4086]	Eastlake, D., Schiller, J., and S. Crocker, “Randomness Requirements for Security,” BCP 106, RFC 4086, June 2005 (TXT).
[RFC4301]	Kent, S. and K. Seo, “Security Architecture for the Internet Protocol,” RFC 4301, December 2005 (TXT).
[RFC4478]	Nir, Y., “Repeated Authentication in Internet Key Exchange (IKEv2) Protocol,” RFC 4478, April 2006 (TXT).

[RFC4507]	Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State," RFC 4507, May 2006 (TXT).
[RFC4555]	Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)," RFC 4555, June 2006 (TXT).
[RFC4718]	Eronen, P. and P. Hoffman, "IKEv2 Clarifications and Implementation Guidelines," RFC 4718, October 2006 (TXT).

Appendix A. Related Work

TOC

[I-D.friedman-ike-short-term-certs] (Friedman, A., "Short-Term Certificates," June 2007.) is on-going work that discusses the use of short-term certificates for client re-authentication. It is similar to the ticket approach described in this document in that they both require enhancements to IKEv2 to allow information request, e.g., for a certificate or a ticket. However, the changes required by the former are fewer since an obtained certificate is valid for any IKE responder that is able to verify them. On the other hand, short-term certificates, while eliminating the usability issues of user re-authentication, do not reduce the amount of effort performed by the gateway in failover situations.

Appendix B. Change Log

TOC

B.1. -04

TOC

Editorial fixes; references cleaned up; updated author's contact address

B.2. -03

TOC

Removed counter mechanism. Added an optional anti-DoS mechanism, based on IKEv2 cookies (removed previous discussion of cookies). Clarified that gateways may support reallocation of same IP address, if provided by network. Proposed a solution outline to the problem of key exchange

for the keys that protect tickets. Added fields to the ticket to enable interoperability. Removed incorrect MOBIKE notification.

B.3. -02

TOC

Clarifications on generation of SPI values, on the ticket's lifetime and on the integrity protection of the anti-replay counter. Eliminated redundant SPIs from the notification payloads.

B.4. -01

TOC

Editorial review. Removed 24-hour limitation on ticket lifetime, lifetime is up to local policy.

B.5. -00

TOC

Initial version. This draft is a selective merge of draft-sheffer-ike-session-resumption-00 and draft-dondeti-ipsec-failover-sol-00.

Authors' Addresses

TOC

	Yaron Sheffer
	Check Point Software Technologies Ltd.
	5 Hasolelim St.
	Tel Aviv 67897
	Israel
Email:	yaronf@checkpoint.com
	Hannes Tschofenig
	Nokia Siemens Networks
	Linnoitustie 6
	Espoo 02600
	Finland
Phone:	+358 (50) 4871445
Email:	Hannes.Tschofenig@gmx.net
URI:	http://www.tschofenig.priv.at
	Lakshminath Dondeti
	QUALCOMM, Inc.

	5775 Morehouse Dr
	San Diego, CA
	USA
Phone:	+1 858-845-1267
Email:	ldondeti@qualcomm.com
	Vidya Narayanan
	QUALCOMM, Inc.
	5775 Morehouse Dr
	San Diego, CA
	USA
Phone:	+1 858-845-2483
Email:	vidyan@qualcomm.com

Full Copyright Statement

TOC

Intellectual Property

standard. Please address the information to the IETF at ietf-ipr@ietf.org.