

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 23, 2007

Y. Sheffer
Check Point
H. Tschofenig
Siemens Networks GmbH & Co KG
T. Wan
Nortel
January 19, 2007

Stateless Session Resumption for the IKE Protocol
draft-sheffer-ike-session-resumption-00.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of 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/lid-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 July 23, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

The Internet Key Exchange version 2 (IKEv2) protocol is computationally intensive with respect to the number of round-trips required and cryptographic operations involved. In particular the Extensible Authentication Protocol is used for authentication, which adds additional computational intensity.

Internet-Draft

IKE Session Resumption

January 2007

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 reestablishment.

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. An encrypted ticket cannot be decrypted by a VPN client but allows a VPN gateway to restore state for faster session state setup.

Internet-Draft

IKE Session Resumption

January 2007

Table of Contents

1.	Introduction	4
1.1.	Goals	4
1.2.	Non-Goals	5
2.	Requirements Notation	5
3.	Protocol Details	5
3.1.	Requesting a Ticket	5
3.2.	Presenting a Ticket	7
3.3.	IKE Notifications	9
3.4.	Processing Guidelines for IKE SA Establishment	9
3.5.	Computing the AUTH Payload	10
4.	The IKE Ticket	10
4.1.	Ticket Contents	10
4.2.	Ticket Format	11
4.3.	Ticket Identity and Lifecycle	11
5.	IANA Considerations	12
6.	Security Considerations	12
6.1.	Stolen Tickets	12
6.2.	Forged Tickets	12
6.3.	Denial of Service Attacks	12
6.4.	Ticket Protection Key Management	12
6.5.	Ticket Lifetime	13
6.6.	Alternate Ticket Formats and Distribution Schemes	13
6.7.	Identity Privacy, Anonymity, and Unlinkability	13
6.8.	Usage of IKE Cookies	14
7.	Acknowledgements	14
8.	References	14
8.1.	Normative References	14
8.2.	Informative References	14
Appendix A.	Related Work	15
Appendix B.	Change Log	15
B.1.	-00	15
	Authors' Addresses	15
	Intellectual Property and Copyright Statements	17

1. Introduction

The Internet Key Exchange version 2 (IKEv2) protocol is computationally intensive with respect to the number of round-trips required and cryptographic operations involved. In particular the Extensible Authentication Protocol is used for authentication, which adds additional computational intensity.

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 an 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 needs to be 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](#)] 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

The high-level goal of this extension is to become a building block of an overall IPsec failover solution, according to the requirements defined in [[I-D.vidya-ipsec-failover-ps](#)].

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:

- o 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.
- o Recovery from an intermittent connectivity failure ("network reboot"), 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.
- o Recovery from a gateway restart, where clients reconnect into the same gateway.

The proposed solution should additionally meet the following goals:

- o Using only symmetric cryptography to minimize CPU consumption.
 - o Allowing a gateway to push state to clients.
 - o Providing cryptographic agility.
 - o Having no negative impact on IKEv2 security features.
- Specifically, the solution must not introduce new security vulnerabilities, such as denial-of-service.

[1.2.](#) Non-Goals

The following are non-goals of this solution:

- o Providing load balancing among gateways.
- o Specifying how a client detects the need for a failover.
- o Establishing trust among gateways.

[2.](#) Requirements Notation

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](#)].

[3.](#) Protocol Details

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 3.1](#) describes the procedure to request a ticket and [Section 3.2](#) illustrates how to present a ticket.

[3.1.](#) Requesting a Ticket

A client MAY request a ticket in the following exchanges:

- o In an IKE_AUTH exchange, as shown in the example message exchange in Figure 1 below.
- o In a CREATE_CHILD_SA exchange, when an IKE SA is rekeyed.
- o In an Informational exchange, if the gateway previously replied with an N(TICKET/Ack) instead of providing a ticket.
- o In an Informational exchange, when the ticket lifetime is about to expire.

Normally, a client requests a ticket in the third message of an IKEv2 (the first of IKE_AUTH). Figure 1 shows the message exchange for this typical case.

```

-----
HDR, SAi1, KEi, Ni    -->

      <--    HDR, SAR1, KEr, Nr, [CERTREQ]

HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,]
AUTH, SAi2, TSi, TSr, N(TICKET/Request)}    -->

```

Figure 1: Example Message Exchange for Requesting a Ticket

The notification payloads are described in [Section 3.3](#). For editorial reasons a number of IKEv2-specific details are omitted. A complete description of IKEv2 can be found in [\[RFC4718\]](#).

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:

- o it creates a ticket and returns it with the N(TICKET/Opaque) payload in a subsequent message towards the IKEv2 initiator. This exchange is shown in Figure 2.
- o it returns an N(TICKET/Nack) payload, if it refuses to grant a ticket for some reason.
- o 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.

The IKEv2 initiator receives the requested ticket if the IKEv2 exchange was successful, and the ticket later be used with an IKEv2 responder which supports this extension. The ticket exchange is shown in Figure 2

```

Initiator              Responder
-----
      <--    HDR, SK {IDr, [CERT,] AUTH,
              SAR2, TSi, TSr, N(TICKET/Opaque)}

```

Figure 2: Receiving a Ticket

3.2. Presenting a Ticket

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 of IKE_SA_INIT. 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 purely a local decision of the client when and based on what information to decide that a communication failure has happened and that a new exchange including the ticket is needed.

Initiator	Responder
-----	-----
HDR, SAi1, KEi, Ni, N(TICKET/Opaque)	-->

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:

- o It responds with an N(TICKET/Ack), if it supports this extension and is willing to accept the ticket. This message is shown in Figure 4.
- o It responds with an N(TICKET/Nack) notification, if it does not accept the ticket for any reason. The responder SHOULD respond with a regular IKE_INIT message #2 including the said notification, and the two IKEv2 peers SHOULD continue with a full, 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.
- o If the responder receives a ticket for an IKE SA that is still active and accepts it, it SHOULD silently delete the old SA without sending a DELETE Informational exchange.

If the responder replies with a standard IKE_INIT message #2 then the

initiator can assume that the responder does not implement the

extension described in this document.

```
Initiator                      Responder
-----
<-- HDR, SAr1, Nr, N(TICKET/Ack)
```

Figure 4: IKEv2 Responder responds with N(TICKET/Ack)

The KE payload is omitted in the response, but Ni and Nr are still exchanged to assure the freshness of subsequently derived keying materials.

At this point a new IKE SA is created by both parties (see [Section 3.4](#)), and used for the following IKE_AUTH exchange. Both parties thereby exchange AUTH payloads, as shown below:

```
Initiator                      Responder
-----
--> HDR, SK {IDi, [IDr,] AUTH}
<-- HDR, SK {IDr, AUTH}
```

Figure 5: A Stand-Alone IKE_AUTH Exchange

See [Section 3.5](#) for the derivation of the AUTH values from the message bytes, the ticket and the nonce values. The initiator and the responder MUST verify the AUTH values they receive. The new IKE SA only becomes fully valid if both parties have verified each other's AUTH payload. If either side receives an incorrect AUTH value, it MUST terminate the protocol.

the IKE_AUTH exchange can also be piggybacked into a CREATE_CHILD_SA exchange, as shown in Figure 6. It is up to the client to decide whether to piggyback the exchange.

```
--> HDR, SK {IDi, [IDr,] AUTH, SAi2, Ni2, TSi, TSr [, CP(CFG_REQUEST)]}
<-- HDR, SK {IDr, AUTH, SAR2, Nr2, TSi, TSr [, CP(CFG_REPLY)]}
```

Figure 6: IKE_AUTH piggybacked on top of a CREATE_CHILD_SA exchange

3.3. IKE Notifications

This document defines a single notification type, TICKET, with a number of subtypes. The notification number is TBD and the subtypes are listed below:

Subtype Name	Number	Data
Opaque	1	See below
Request	2	None
Ack	3	None
Nack	4	None
Reserved	5-127	
Private Use	128-255	

The data for the Opaque subtype consists of a lifetime duration in seconds (4 octets, denoting the time until the ticket expires), followed by the ticket content. See [Section 4.3](#) for further guidelines regarding the ticket's lifetime, and [Section 4.2](#) for a recommended ticket format.

3.4. Processing Guidelines for IKE SA Establishment

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:

- o The SA value (transforms etc.) is taken directly from the ticket.
- o The sequence numbers are reset to 0.
- o The IDi value is obtained from the ticket.
- o The IDr value is obtained from the new exchange. The gateway MAY make policy decisions based on the IDr value encoded in the ticket.
- o The SPI values are created anew, similarly to a regular IKE exchange. SPI values from the ticket MUST NOT be reused.

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:

$$\text{SKEYSEED} = \text{prf}+(\text{SK_d_old}, \text{Ni} \parallel \text{Nr})$$

where SK_d_old is the value retrieved from the ticket.

The keys are derived as follows, unchanged from IKEv2:

$$\{\text{SK_d} \mid \text{SK_ai} \mid \text{SK_ar} \mid \text{SK_ei} \mid \text{SK_er} \mid \text{SK_pi} \mid \text{SK_pr}\} = \text{prf}+(\text{SKEYSEED}, \text{Ni} \mid \text{Nr} \mid \text{SPIi} \mid \text{SPIr})$$

where SPIi, SPIr are the SPI values created in the new IKE exchange.

See [\[RFC4306\]](#) for the notation. "prf" is determined from the SA value in the ticket.

[3.5.](#) Computing the AUTH Payload

The value for the AUTH payload is derived in a manner similar to the usage of IKEv2 pre-shared secret-based authentication, as shown below:

$$\text{AUTH} = \text{prf}(\text{SK_px}, \langle \text{msg octets} \rangle)$$

Each of the initiator and responder uses its own SK_p value, taken from the newly generated IKE SA.

The exact material to be signed is defined in [Section 2.15 of \[RFC4306\]](#). The notation "IDr" in [RFC 4306](#) should be applied to the new IDr value included in the exchange, rather than the value in the ticket.

[4.](#) The IKE Ticket

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.

[4.1.](#) Ticket Contents

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

- o IDi, IDr.
- o SPIi, SPIr.
- o SAR (the accepted proposal).
- o SK_d.

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

[4.2.](#) Ticket Format

This document does not specify a ticket format. The following format (this entire current section) is a RECOMMENDED implementation. The client SHOULD NOT attempt to decode the ticket.

```
struct {
    opaque key_name[16];           // ASCII, null-terminated
    opaque IV[0..255];            // the length (possibly 0) depends
                                // on the encryption algorithm

    [protected] struct {
        opaque IDi, IDr;          // the full payloads
        opaque SPIi, SPIr;
        opaque SA;                // the full payload, returned as SAR
        opaque SK_d;
        opaque expiration;        // an absolute time value
    } ikev2_state;                // encrypted and authenticated
    opaque MAC[0..255];           // the length (possibly 0) depends
                                // on the integrity algorithm
} ticket;
```

Note that the key defined by "key_name" 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 [\[I-D.rescorla-stateless-tokens\]](#) that recommends a similar (but not identical) ticket format, and discusses related security considerations in depth.

[4.3.](#) Ticket Identity and Lifecycle

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 use a ticket to approach other gateways that are willing to accept it. How a client discovers such gateways is outside the scope of this document.

The lifetime included with the ticket 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 [\[RFC4478\]](#). Even if neither of these is enforced by the gateway, a finite lifetime MUST be specified for the ticket and SHOULD be less than 24 hours.

[5.](#) IANA Considerations

This document requires the following notifications to be registered by IANA. The corresponding registry was established by IANA.

- o Notification type ([Section 3.3](#)).
- o Notification subtypes ([Section 3.3](#)).

[6.](#) Security Considerations

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

[6.1.](#) Stolen Tickets

An eavesdropper or man-in-the-middle may try to obtain a ticket and use it to establish a session with the IKEv2 responder. However, since the ticket 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 resume a session. An IKEv2 responder MUST use strong encryption and integrity protection for the ticket to prevent an attacker from obtaining the ticket's contents, e.g., by using a brute force attack.

[6.2.](#) Forged Tickets

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 such as a keyed HMAC-SHA1.

[6.3.](#) Denial of Service Attacks

The key_name field defined in the recommended ticket format helps the server efficiently reject 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). See also [Section 6.8](#).

[6.4.](#) 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.

- o The keys should be generated securely following the randomness recommendations in [\[RFC4086\]](#).
- o The keys and cryptographic protection algorithms should be at least 128 bits in strength.
- o The keys should not be used for any other purpose than generating and verifying tickets.
- o The keys should be changed regularly.
- o The keys should be changed if the ticket format or cryptographic protection algorithms change.

[6.5.](#) Ticket Lifetime

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](#)]. 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.

[6.6.](#) Alternate Ticket Formats and Distribution Schemes

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.

[6.7.](#) Identity Privacy, Anonymity, and Unlinkability

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 first message of the IKE_SA_INIT 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.

[6.8.](#) Usage of IKE Cookies

The extension specified in this document eliminates most potential denial-of-service attacks that the cookie mechanism aims to solve. However, memory exhaustion remains possible. Therefore the cookie mechanism described in [Section 2.6 of \[RFC4306\]](#) MAY be invoked by the gateway for the modified IKE_INIT exchange described in [Section 3.2](#).

[7.](#) Acknowledgements

We would like to thank Pasi Eronen and Yoav Nir for their review of early drafts.

[8.](#) References

[8.1.](#) Normative References

[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.

[8.2.](#) Informative References

[I-D.friedman-ike-short-term-certs]
Friedman, A., "Short-Term Certificates",
[draft-friedman-ike-short-term-certs-01](#) (work in progress),
December 2006.

[I-D.rescorla-stateless-tokens]
Rescorla, E., "How to Implement Secure (Mostly) Stateless Tokens", [draft-rescorla-stateless-tokens-00](#) (work in progress), January 2007.

[I-D.vidya-ipsec-failover-ps]
Narayanan, V., "IPsec Gateway Failover and Redundancy -

- [RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", [BCP 106](#), [RFC 4086](#), June 2005.
- [RFC4478] Nir, Y., "Repeated Authentication in Internet Key Exchange (IKEv2) Protocol", [RFC 4478](#), April 2006.
- [RFC4507] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", [RFC 4507](#), May 2006.
- [RFC4718] Eronen, P. and P. Hoffman, "IKEv2 Clarifications and Implementation Guidelines", [RFC 4718](#), October 2006.

[Appendix A](#). Related Work

[I-D.friedman-ike-short-term-certs] 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

[B.1](#). -00

Initial version.

Authors' Addresses

Yaron Sheffer
Check Point Software Technologies Ltd.
3A Jabotinsky St.
Ramat Gan 52520
Israel

Email: yaronf at checkpoint dot com

Hannes Tschofenig
Siemens Networks GmbH & Co KG
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Phone: +49 89 636 40390
Email: Hannes.Tschofenig@siemens.com
URI: <http://www.tschofenig.com>

Tao Wan
Nortel Networks
250 Sidney Street
Belleville, Ontario K8N 5B7
Canada

Phone: +1 613 961 2350
Email: twan (at) nortel (dot) com

Internet-Draft

IKE Session Resumption

January 2007

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).