

Network Working Group
Request for Comments: 4030
Category: Standards Track

M. Stapp
Cisco Systems, Inc.
T. Lemon
Nominum, Inc.
March 2005

The Authentication Suboption for the
Dynamic Host Configuration Protocol (DHCP) Relay Agent Option

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

The Dynamic Host Configuration Protocol (DHCP) Relay Agent Information Option ([RFC 3046](#)) conveys information between a DHCP Relay Agent and a DHCP server. This specification defines an authentication suboption for that option, containing a keyed hash in its payload. The suboption supports data integrity and replay protection for relayed DHCP messages.

Table of Contents

1.	Introduction	2
2.	Requirements Terminology	3
3.	DHCP Terminology	4
4.	Suboption Format	4
5.	Replay Detection	5
6.	The Relay Identifier Field	5
7.	Computing Authentication Information	6
	7.1. The HMAC-SHA1 Algorithm	6
8.	Procedures for Sending Messages	7
	8.1. Replay Detection	7
	8.2. Packet Preparation	8
	8.3. Checksum Computation	8
	8.4. Sending the Message	8
9.	Procedures for Processing Incoming Messages	8
	9.1. Initial Examination	8
	9.2. Replay Detection Check	9
	9.3. Testing the Checksum	9
10.	Relay Agent Behavior	9
	10.1. Receiving Messages from Other Relay Agents	10
	10.2. Sending Messages to Servers	10
	10.3. Receiving Messages from Servers	10
11.	DHCP Server Behavior	10
	11.1. Receiving Messages from Relay Agents	10
	11.2. Sending Reply Messages to Relay Agents	11
12.	IANA Considerations	11
13.	Security Considerations	11
	13.1. The Key ID Field	12
	13.2. Protocol Vulnerabilities	12
14.	Acknowledgements	13
15.	References	13
	15.1. Normative References	13
	15.2. Informative References	13
	Authors' Addresses	14
	Full Copyright Statement	15

[1.](#) Introduction

DHCP ([RFC 2131](#) [6]) provides IP addresses and configuration information for IPv4 clients. It includes a relay-agent capability ([RFC 951](#) [7], [RFC 1542](#) [8]) in which processes within the network infrastructure receive broadcast messages from clients and forward

them to servers as unicast messages. In network environments such as DOCSIS data-over-cable and xDSL, for example, it has proven useful for the relay agent to add information to the DHCP message before forwarding it, by using the relay-agent information option ([RFC 3046](#) [1]). The kind of information that relays add is often used in the

server's decision-making about the addresses and configuration parameters that the client should receive. The way that the relay-agent data is used in server decision-making tends to make that data very important, and it highlights the importance of the trust relationship between the relay agent and the server.

The existing DHCP Authentication specification ([RFC 3118](#)) [9] only covers communication between the DHCP client and server. Because relay-agent information is added after the client has sent its message, the DHCP Authentication specification explicitly excludes relay-agent data from that authentication.

The goal of this specification is to define methods that a relay agent can use to

1. protect the integrity of relayed DHCP messages,
2. provide replay protection for those messages, and
3. leverage existing mechanisms, such as DHCP Authentication.

In order to meet these goals, we specify a new relay-agent suboption, the Authentication suboption. The format of this suboption is very similar to the format of the DHCP Authentication option, and the specification of its cryptographic methods and hash computation is also similar.

The Authentication suboption is included by relay agents that seek to ensure the integrity of the data they include in the Relay Agent option. These relay agents are configured with the parameters necessary for generating cryptographic checksums of the data in the DHCP messages that they forward to DHCP servers. A DHCP server configured to process the Authentication suboption uses the information in the suboption to verify the checksum in the suboption and continues processing the relay agent information option only if the checksum is valid. If the DHCP server sends a response, it includes an Authentication suboption in its response message. Relay agents test the checksums in DHCP server responses to decide whether

to forward the responses.

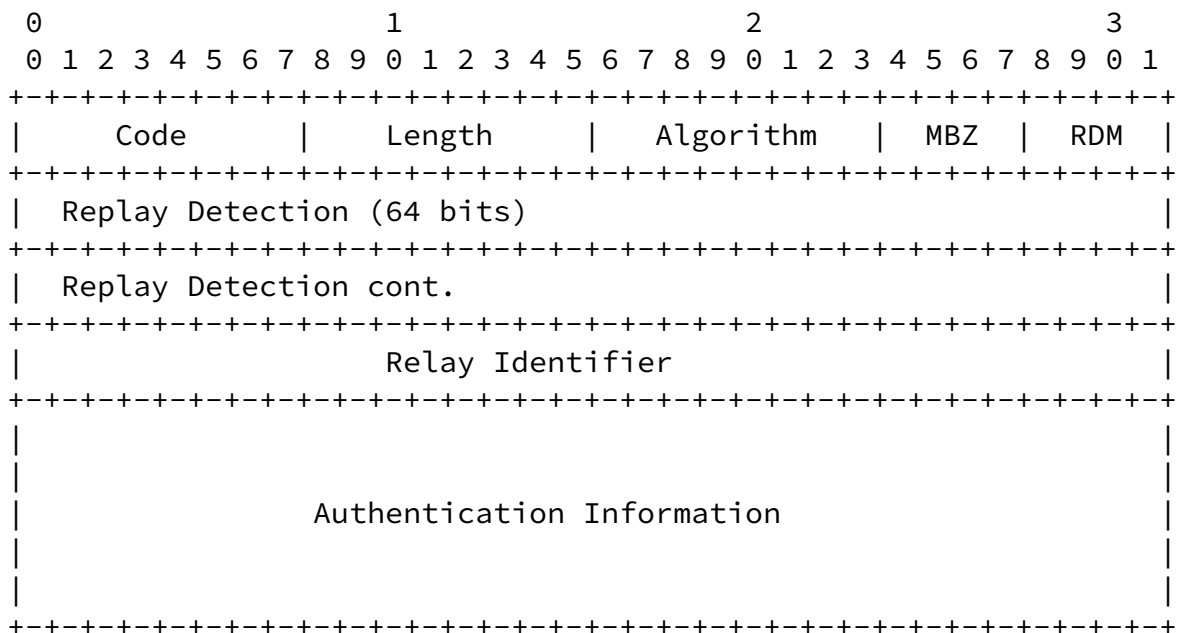
2. Requirements Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC 2119](#) [2].

3. DHCP Terminology

This document uses the terms "DHCP server" (or "server") and "DHCP client" (or "client") as defined in [RFC 2131](#) [6]. The term "DHCP relay agent" refers to a "BOOTP relay agent" as defined in [RFC 2131](#).

4. Suboption Format



The code for the suboption is 8. The length field includes the lengths of the algorithm, the RDM, and all subsequent suboption fields in octets.

The Algorithm field defines the algorithm used to generate the authentication information.

Four bits are reserved for future use. These bits SHOULD be set to zero and MUST NOT be used when the suboption is processed.

The Replay Detection Method (RDM) field defines the method used to generate the Replay Detection Data.

The Replay Detection field contains a value used to detect replayed messages, which are interpreted according to the RDM.

The Relay Identifier field is used by relay agents that do not set giaddr, as described in [RFC 3046](#) [1], section 2.1.

The Authentication Information field contains the data required to communicate algorithm-specific parameters, as well as the checksum. The checksum is usually a digest of the data in the DHCP packet computed by using the method specified by the Algorithm field.

[5.](#) Replay Detection

The replay-detection mechanism is designed on the notion that a receiver can determine whether a message has a valid replay token value. The default RDM, with value 1, specifies that the Replay Detection field contains an increasing counter value. The receiver associates a replay counter with each sender and rejects any message containing an authentication suboption with a Replay Detection counter value less than or equal to the last valid value. DHCP servers MAY identify relay agents by giaddr value or by other data in the message (e.g., data in other relay agent suboptions). Relay agents identify DHCP servers by source IP address. If the message's replay detection value, and the checksum are valid, the receiver updates its notion of the last valid replay counter value associated with the sender.

All implementations MUST support the default RDM. Additional methods

may be defined in the future, following the process described in [section 12](#).

Receivers SHOULD perform the replay-detection check before testing the checksum. The keyed hash calculation is likely to be much more expensive than the replay-detection value check.

DISCUSSION:

This places a burden on the receiver to maintain some run-time state (the most-recent valid counter value) for each sender, but the number of members in a DHCP agent-server system is unlikely to be unmanageably large.

[6](#). The Relay Identifier Field

The Relay Agent Information Option [[1](#)] specification permits a relay agent to add a relay agent option to relayed messages without setting the giaddr field. In this case, the eventual receiver of the message needs a stable identifier to use in order to associate per-sender state such as Key ID and replay-detection counters.

A relay agent that adds a relay agent information option and sets giaddr MUST NOT set the Relay ID field. A relay agent that does not set giaddr MAY be configured to place a value in the Relay ID field. If the relay agent is configured to use the Relay ID field, it MAY be configured with a value to use, or it MAY be configured to generate a

value based on some other data, such as its MAC or IP addresses. If a relay generates a Relay ID value, it SHOULD select a value that it can regenerate reliably; e.g., across reboots.

Servers that process an Authentication Suboption SHOULD use the giaddr value to identify the sender if the giaddr field is set. Servers MAY be configured to use some other data in the message to identify the sender. If giaddr is not set, the server SHOULD use the Relay ID field if it is nonzero. If neither the giaddr nor the Relay ID field is set, the server MAY be configured to use some other data in the message, or it MAY increment an error counter.

[7](#). Computing Authentication Information

The Authentication Information field contains a keyed hash generated

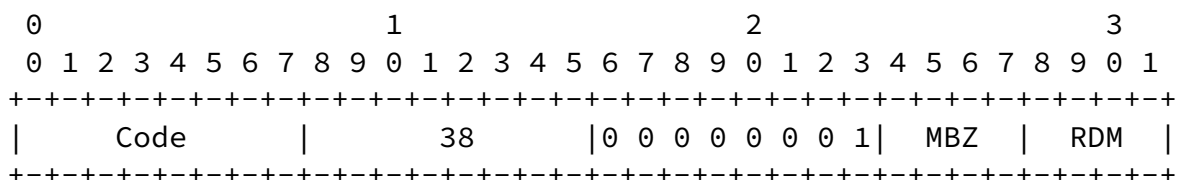
by the sender. All algorithms are defined to process the data in the DHCP messages in the same way. The sender and receiver compute a hash across a buffer containing all of the bytes in the DHCP message, including the fixed DHCP message header, the DHCP options, and the relay agent suboptions, with the following exceptions. The value of the 'hops' field MUST be set to zero for the computation because its value may be changed in transmission. The value of the 'giaddr' field MUST also be set to zero for the computation because it may be modified in networks where one relay agent adds the relay agent option but another relay agent sets 'giaddr' (see [RFC 3046, section 2.1](#)). In addition, because the relay agent option is itself included in the computation, the 'authentication information' field in the Authentication suboption is set to all zeros. The relay agent option length, the Authentication suboption length and other Authentication suboption fields are all included in the computation.

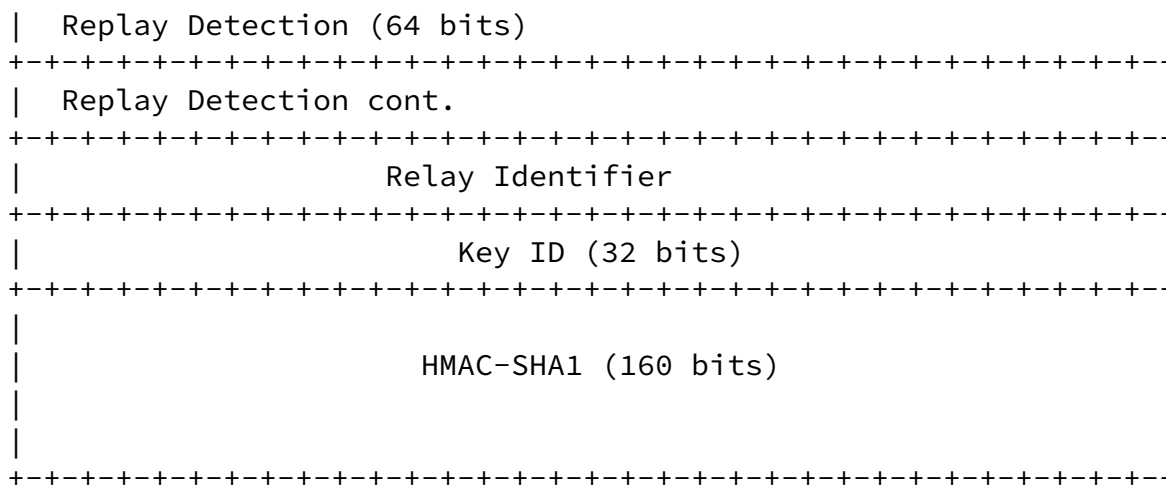
All implementations MUST support Algorithm 1, the HMAC-SHA1 algorithm. Additional algorithms may be defined in the future, following the process described in [section 12](#).

[7.1](#). The HMAC-SHA1 Algorithm

Algorithm 1 is assigned to the HMAC [3] protocol by using the SHA-1 [4] hash function. This algorithm requires that a shared secret key be configured at the relay agent and the DHCP server. A 32-bit Key Identifier is associated with each shared key, and this identifier is carried in the first 4 bytes of the Authentication Information field of the Authentication suboption. The HMAC-SHA1 computation generates a 20-byte hash value, which is placed in the Authentication Information field after the Key ID.

When Algorithm 1 is used, the format of the Authentication suboption is as follows:





The suboption length is 38. The RDM and Replay Detection fields are as specified in [section 5](#). The Relay ID field is set as specified in [section 6](#). The Key ID is set by the sender to the ID of the key used in computing the checksum, as an integer value in network byte order. The HMAC result follows the Key ID.

The Key ID exists only to allow the sender and receiver to specify a shared secret in cases where more than one secret is in use among a network's relays and DHCP servers. The Key ID values are entirely a matter of local configuration; they only have to be unique locally. This specification does not define any semantics or impose any requirements on this algorithm's Key ID values.

[8.](#) Procedures for Sending Messages

[8.1.](#) Replay Detection

The sender obtains a replay-detection counter value to use based on the RDM it is using. If the sender is using RDM 1, the default RDM, the value MUST be greater than any previously sent value.

[8.2.](#) Packet Preparation

The sender sets the 'giaddr' field and the 'hops' field to all zeros. The sender appends the relay agent information option to the client's packet, including the Authentication suboption. The sender selects an appropriate Replay Detection value. The sender places its identifier into the Relay ID field, if necessary, or sets the field to all zeros. The sender sets the suboption length, places the Replay Detection value into the Replay Detection field of the suboption, and sets the algorithm to the algorithm number that it is using. If the sender is using HMAC-SHA1, it sets the Key ID field to the appropriate value. The sender sets the field that will contain the checksum to all zeros. Other algorithms may specify additional preparation steps.

8.3. Checksum Computation

The sender computes the checksum across the entire DHCP message, using the algorithm it has selected. The sender places the result of the computation into the Authentication Information field of the Authentication suboption.

8.4. Sending the Message

The sender restores the values of the 'hops' and 'giaddr' fields and sends the message.

9. Procedures for Processing Incoming Messages

9.1. Initial Examination

The receiver examines the message for the value of the giaddr field and determines whether the packet includes the relay agent information option. The receiver uses its configuration to determine whether it should expect an Authentication suboption. The receiver MUST support a configuration that allows it to drop incoming messages that do not contain a valid relay agent information option and Authentication suboption.

If the receiver determines that the Authentication suboption is present and that it should process the suboption, it uses the data in the message to determine which algorithm, key, and RDM to use in validating the message. If the receiver cannot determine which algorithm, key, and RDM to use, or if it does not support the value indicated in the message, it SHOULD drop the message. Because this situation could indicate a misconfiguration that could deny service to clients, receivers MAY attempt to notify their administrators or to log an error message.

9.2. Replay Detection Check

The receiver examines the RDM field. Receivers MUST discard messages containing RDM values that they do not support. Because this may indicate a misconfiguration at the sender, an attempt SHOULD be made to indicate this condition to the administrator by incrementing an error counter or writing a log message. If the receiver supports the RDM, it examines the value in the Replay Detection field by using the procedures in the RDM and in [section 5](#). If the Replay value is not valid, the receiver MUST drop the message.

Note that at this point the receiver MUST NOT update its notion of the last valid Replay Detection value for the sender. Until the checksum has been tested, the Replay Detection field cannot be trusted. If the receiver trusts the Replay Detection value without testing the checksum, a malicious host could send a replayed message with a Replay Detection value that was very high, tricking the receiver into rejecting legitimate values from the sender.

9.3. Testing the Checksum

The receiver prepares the packet in order to test the checksum by setting the 'giaddr' and 'hops' fields to zero, and by setting the Authentication Information field of the suboption to all zeros. Using the algorithm and key associated with the sender, the receiver computes a hash of the message. The receiver compares the result of its computation with the value sent. If the checksums do not match, the receiver MUST drop the message. Otherwise, the receiver updates its notion of the last valid Replay Detection value associated with the sender and processes the message.

10. Relay Agent Behavior

DHCP Relay agents are typically configured with the addresses of one or more DHCP servers. A relay agent that implements this suboption requires an algorithm number for each server, as well as appropriate credentials (i.e., keys). Relay implementations SHOULD support a configuration that indicates that all relayed messages should include the authentication suboption. Use of the authentication suboption SHOULD be disabled by default. Relay agents MAY support configuration that indicates that certain destination servers support the authentication suboption and that other servers do not. Relay agents MAY support configuration of a single algorithm number and key to be used with all DHCP servers, or they MAY support configuration of different algorithms and keys for each server.

[10.1.](#) Receiving Messages from Other Relay Agents

There are network configurations in which one relay agent adds the relay agent option and then forwards the DHCP message to another relay agent. For example, a layer-2 switch might be directly connected to a client, and it might forward messages to an aggregating router, which sets giaddr and then forwards the message to a DHCP server. When a DHCP relay that implements the Authentication suboption receives a message, it MAY use the procedures in [section 9](#) to verify the source of the message before forwarding it.

[10.2.](#) Sending Messages to Servers

When the relay agent receives a broadcast packet from a client, it determines which DHCP servers (or other relay agents) should receive copies of the message. If the relay agent is configured to include the Authentication suboption, it determines which Algorithm and RDM to use, and then it performs the steps in [section 8](#).

[10.3.](#) Receiving Messages from Servers

When the relay agent receives a message, it determines from its configuration whether it expects the message to contain a relay agent information option and an Authentication suboption. The relay agent MAY be configured to drop response messages that do not contain the Authentication suboption. The relay agent then follows the procedures in [section 9](#).

[11.](#) DHCP Server Behavior

DHCP servers may interact with multiple relay agents. Server implementations MAY support a configuration that associates the same algorithm and key with all relay agents. Servers MAY support a configuration that specifies the algorithm and key to use with each relay agent individually.

[11.1.](#) Receiving Messages from Relay Agents

When a DHCP server that implements the Authentication suboption receives a message, it performs the steps in [section 9](#).

[11.2](#). Sending Reply Messages to Relay Agents

When the server has prepared a reply message, it uses the incoming request message and its configuration to determine whether it should include a relay agent information option and an Authentication suboption. If the server is configured to include the Authentication suboption, it determines which Algorithm and RDM to use and then performs the steps in [section 8](#).

DISCUSSION:

This server behavior represents a slight variance from [RFC 3046 \[1\]](#), section 2.2. The Authentication suboption is not echoed back from the server to the relay; the server generates its own suboption.

[12](#). IANA Considerations

[Section 4](#) defines a new suboption for the DHCP relay agent option called the Authentication Suboption. IANA has allocated a new suboption code from the relay agent option suboption number space.

This specification introduces two new number spaces for the Authentication suboption's 'Algorithm' and 'Replay Detection Method' fields. These number spaces have been created and will be maintained by IANA.

The Algorithm identifier is a one-byte value. The Algorithm value 0 is reserved. The Algorithm value 1 is assigned to the HMAC-SHA1 keyed hash, as defined in [section 7.1](#). Additional algorithm values will be allocated and assigned through IETF consensus, as defined in [RFC 2434 \[5\]](#).

The RDM identifier is a four-bit value. The RDM value 0 is reserved. The RDM value 1 is assigned to the use of a monotonically increasing counter value, as defined in [section 5](#). Additional RDM values will be allocated and assigned through IETF consensus, as defined in [RFC 2434](#) [5].

[13.](#) Security Considerations

This specification describes a protocol that adds source authentication and message integrity protection to the messages between DHCP relay agents and DHCP servers.

The use of this protocol imposes a new computational burden on relay agents and servers, because they must perform cryptographic hash calculations when they send and receive messages. This burden may add latency to DHCP message exchanges. Because relay agents are

involved when clients reboot, periods of very high reboot activity will result in the largest number of messages that have to be processed. During a cable MSO head-end reboot event, for example, the time required for all clients to be served may increase.

[13.1.](#) The Key ID Field

The Authentication suboption contains a four-byte Key ID, following the example of the DHCP Authentication RFC. Other authentication protocols, such as DNS TSIG [10], use a key name. A key name is more flexible and potentially more human readable than a key id. DHCP servers may well be configured to use key names for DNS updates using TSIG, so it might simplify DHCP server configuration if some of the key management for both protocols could be shared.

On the other hand, it is crucial to minimize the size expansion caused by the introduction of the relay agent information option. Named keys would require more physical space and would entail more complex suboption encoding and parsing implementations. These considerations have led us to specify a fixed-length Key ID instead of a variable-length key name.

[13.2.](#) Protocol Vulnerabilities

Because DHCP is a UDP protocol, messages between relays and servers

may be delivered in an order different from that in which they were generated. The replay-detection mechanism will cause receivers to drop packets that are delivered 'late', leading to client retries. The retry mechanisms that most clients implement should not cause this to be an enormous issue, but it will cause senders to do computational work which will be wasted if their messages are re-ordered.

The DHC WG has developed two documents describing authentication of DHCP relay agent options to accommodate the requirements of different deployment scenarios: this document and "Authentication of Relay Agent Options Using IPsec" [[11](#)]. As we note in [section 11](#), the Authentication suboption can be used without pairwise keys between each relay and each DHCP server. In deployments where IPsec is readily available and pairwise keys can be managed efficiently, the use of IPsec as described in that document may be appropriate. If IPsec is not available or there are multiple relay agents for which multiple keys must be managed, the protocol described in this document may be appropriate. As is the case whenever two alternatives are available, local network administration can choose whichever is more appropriate. Because the relay agents and the DHCP

server are all in the same administrative domain, the appropriate mechanism can be configured on all interoperating DHCP server elements.

[14.](#) Acknowledgements

The need for this specification was made clear by comments made by Thomas Narten and John Schnizlein, and the use of the DHCP Authentication option format was suggested by Josh Littlefield, at IETF 53.

[15.](#) References

[15.1.](#) Normative References

- [1] Patrick, M., "DHCP Relay Agent Information Option", [RFC 3046](#), January 2001.

- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [3] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), February 1997.
- [4] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", [RFC 3174](#), September 2001.
- [5] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 2434](#), October 1998.

15.2. Informative References

- [6] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [7] Croft, W. and J. Gilmore, "Bootstrap Protocol", [RFC 951](#), September 1985.
- [8] Wimer, W., "Clarifications and Extensions for the Bootstrap Protocol", [RFC 1542](#), October 1993.
- [9] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", [RFC 3118](#), June 2001.
- [10] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", [RFC 2845](#), May 2000.

- [11] Droms, R., "Authentication of Relay Agent Options Using IPsec", Work in Progress, February 2004.

Authors' Addresses

Mark Stapp
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Phone: 978.936.0000
EMail: mjs@cisco.com

Ted Lemon
Nominum, Inc.
950 Charter St.
Redwood City, CA 94063
USA

EMail: Ted.Lemon@nominum.com

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 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.

Acknowledgement

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