KARP Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2011

M. Bhatia
Alcatel-Lucent
S. Hartman
Painless Security
D. Zhang
Huawei Technologies co., LTD.

February 14, 2011

Security Extension for OSPFv2 when using Manual Key Management
draft-bhatia-karp-ospf-ip-layer-protection-03

Abstract

The current OSPFv2 cryptographic authentication mechanism as defined in the OSPF standards is vulnerable to both inter-session and intra-session replay attacks when its uses manual keying. Additionally, the existing cryptographic authentication schemes do not cover the IP header. This omission can be exploited to carry out various types of attacks.

This draft proposes an authentication scheme based on a challenge-response mechanism that will protect OSPFv2 from both inter and intra replay attacks when its using manual keys for securing its protocol packets. For comparison, an approach based on making sequence numbers unique is presented. Later we also describe some changes in the cryptographic hash computation so that we eliminate most attacks that result because of OSPFv2 not protecting the IP header.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on August 18, 2011.

Copyright Notice

Table of Contents

1. Introduction 5

2. A Challenge and Response Solution 6
 2.1. Neighbor State Required 11
 2.2. Receiver Behavior 12
 2.3. Nonce Triggers 13

3. Packet Format 14
 3.1. Extensions to OSPF packets 14
 3.2. Extension of Hello Packet 15

4. Key Selection in Processing OSPF Packets 17
 4.1. Key Selection in Sending Unicast OSPF Packets 17
 4.2. Key Selection in Sending Multicast OSPF Packets 17
 4.3. Key Selection on Receiving OSPF Packets 18

5. Existing Cryptographic Authentication Mechanism 19

6. Mechanism to secure the IP header 20

7. Alternative Boot Count Approach 21

8. Security Considerations 22

9. IANA Considerations 23

10. Acknowledgements 24

11. References 25
 11.1. Normative References 25
 11.2. Informative References 25

Authors' Addresses 26

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 [RFC2119].

When used in lower case, these words convey their typical use in common language, and are not to be interpreted as described in RFC2119 [RFC2119].

1. Introduction

The OSPFv2 cryptographic authentication mechanism as described in [[RFC2328]] uses per-packet sequence numbers to provide protection against replay attacks. The sequence numbers increase monotonically so that the attempts to replay the stale packets can be thwarted. The sequence number values are maintained as a part of adjacency states. Therefore, if an adjacency is broken down, the associated sequence numbers get re-initiated and the neighbors start all over again. Additionally, the cryptographic authentication mechanism does not specify how to deal with the rollover of a sequence number when it reaches its maximum limit. These omissions can be taken advantage of by attackers to implement various replay attacks ([RFC6039]). In order to address these issues, we propose a challenge/ response mechanism that introduces two additional random numbers to help routers generate distinguishable new states when the sequence numbers need to be re-initiated. Compared with the cryptographic authentication mechanism proposed in [RFC5709], the solution proposed does not impose any more security presumption.

The cryptographic authentication as described in [RFC2328] and later updated in [RFC5709] does not include the IP header. This also can be exploited to launch several attacks as the source address in the IP header is no longer protected. The OSPF specification, in certain cases, requires the implementations to look at the source address carried in the IP header to determine the neighbor the packet was recieved from. Changing the source address of a packet can thus, confuse the reciever which can be exploited to produce a number of denial of service attacks [RFC6039]. If the packet is interpreted as coming from a different neighbor, the sequence number received from the neighbor may be updated. This may disrupt communication with the legitimate neighbor. Hello packets may be reflected to cause a neighbor to appear to have one-way communication. Old Database descriptions may be reflected in cases where the per-packet sequence numbers are sufficiently divergent in order to disrupt an adjacency [I-D.hartman-ospf-analysis]. This is referred to as the IP layer issue in [I-D.ietf-karp-threats-reqs].

[RFC2328] states that implementations MUST offer keyed MD5 authentication. It is likely that this will be deprecated in favor of the stronger algorithms described in [RFC5709] in future deployments [RFC6094].

This draft proposes a simple change in the cryptographic authentication mechanism, as currently described in [RFC5709], to prevent such IP layer attacks.

2. A Challenge and Response Solution

In OSPFv2, a non-decreasing sequence member is associated with each OSPF packet sent from a router in order to prevent replay attacks. However, as illustrated in [I-D.hartman-ospf-analysis] and [RFC6039], in the circumstances where automatic key management mechanisms are unavailable, any re-initiation of sequence numbers can potentially be taken advantage of to perform replay attacks. In this section, we introduce an extension of the OSPFv2 protocol, which uses challenge/ response to benefit the verification of the freshness of OSPF packets when the sequence numbers of routers are re-initiated. This solution eliminates the reliance on automatic key management mechanisms. However, it is assumed that a traffic key is shared between two communicating routers so that an attacker can play antique packets but lacks the capability to modify packets without being detected.

In this protocol, two random numbers (Session ID and Nonce) are introduced. The session ID is used to identify the session a packet is within and thus makes inter-session replay attacks difficult. The nonce is used to challenge the liveness of communicating routers so that states need not be maintained with routers that are not currently neighbors. In combination with the sequence number, the session ID can effectively resist intra-session replay attacks. When the sequence space is exhausted, a router simply chooses a new session ID.

Figure 1 illustrates how two routers A and B, challenge each other’s liveness when they are initially connected to a link. First, A selects a new session ID (X1) and a new nonce (N1), and sends them out within a hello packet (see step 1). Particularly, X1and N1 are encapsulated in the OSPF header of the packet. Note that if A is on a multicast LAN, the packet is sent using multicast. Similarly, B sends a hello packet with its new session ID (X2) and Nonce (N2) (step 2). Upon receiving the hello packet from B, A sends a hello packet with X1 and N1. In the neighbor field of the packet, the router ID of B, X2, and N2 is encapsulated (Step 3). Upon receiving the packet sent in step 3, B can ensure the freshness of the packet if the attached session ID and nonce values of both routers are correct.

In the same way, after receiving the hello packet from A, B sends a hello packet with X2 and N2 in the OSPF header, and in the neighbor field of the packet, the router ID of A, X1, and N1 is listed to identify that A has been discovered. After receiving the packet, A can make sure the packet is fresh if the session IDs and the nonce of both routers contained in the packet are correct. After A and B discover each other, they start exchanging their database information (steps 5 and 6). During the exchange, every packet from Router A is

associated with X1 and N1, while every packet from Router B is associated with X2 and N2. Each of these packets also contains a sequence number as part of the cryptographic authentication option. The sequence number MUST increase for every packet sent.

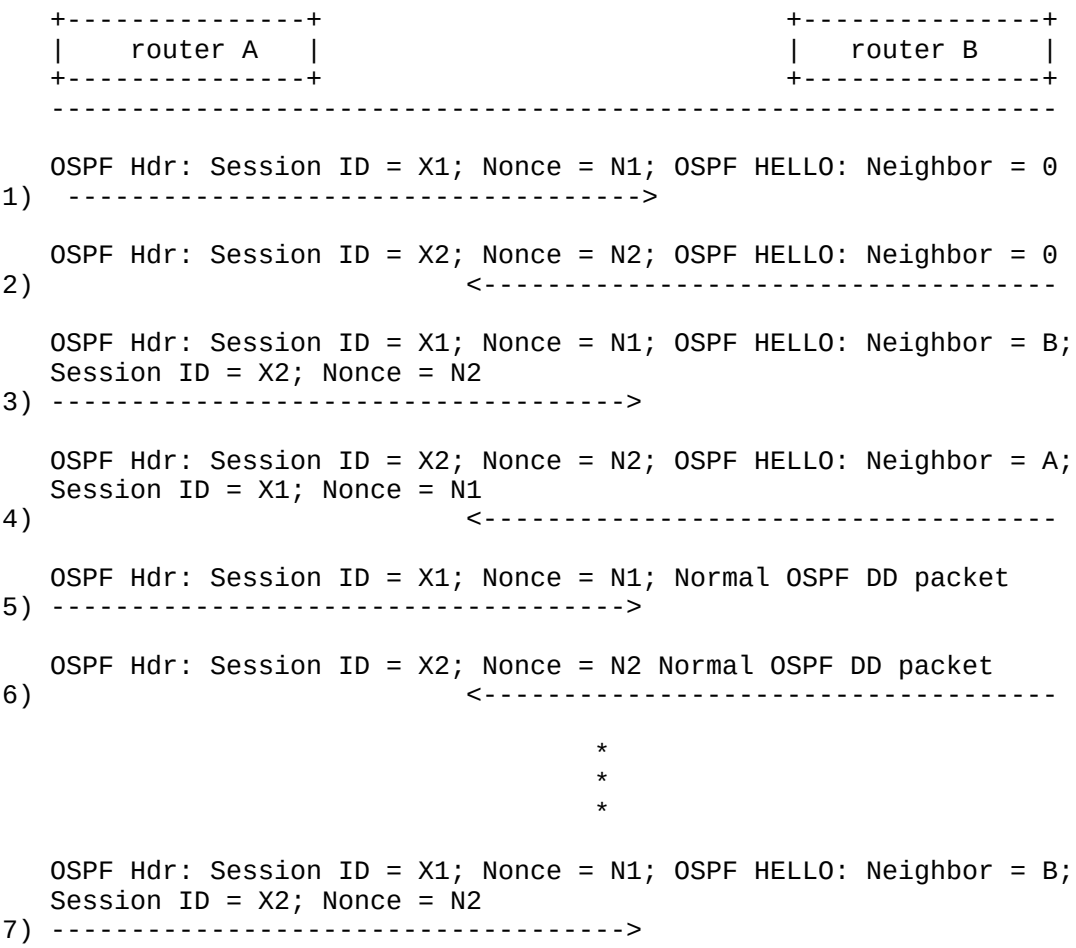


Figure 1 Scenario: two Routers coming up on a LAN

After A and B have generated a neighbor relationship, assume another router, C, is connected to the link. C finds the existence of A and intends to become a neighbor of A. The packets exchanged during this process are illustrated in Figure 2. Firstly, C selects a new session ID (X3) and a new nonce value (N3), and sends them out within a blank hello packet (see the second step of Figure 2). After receiving this packet, A sends out a hello packet with the information of C (router ID, X3, and N3) in the neighbor field.

Because A is challenging the liveness of a new neighbor, A selects a new nonce N1' and encapsulates it in the OSPF header of the hello packet to challenge whether the packet sent in step 2 is really from C. After receiving the packet from A, C can make sure the packet is valid since it consists of its current session ID and nonce (e.g., X3 and N3). Thus, C replies to A with a hello packet including the information of A (e.g., X1 and N1') in the neighbor field. After receiving this packet and checking the correctness of X1 and N1', A can ensure that the packet is fresh and C is currently online.

It worthwhile to note that during the challenge and response the hello packets sent immediately amnogst routers.

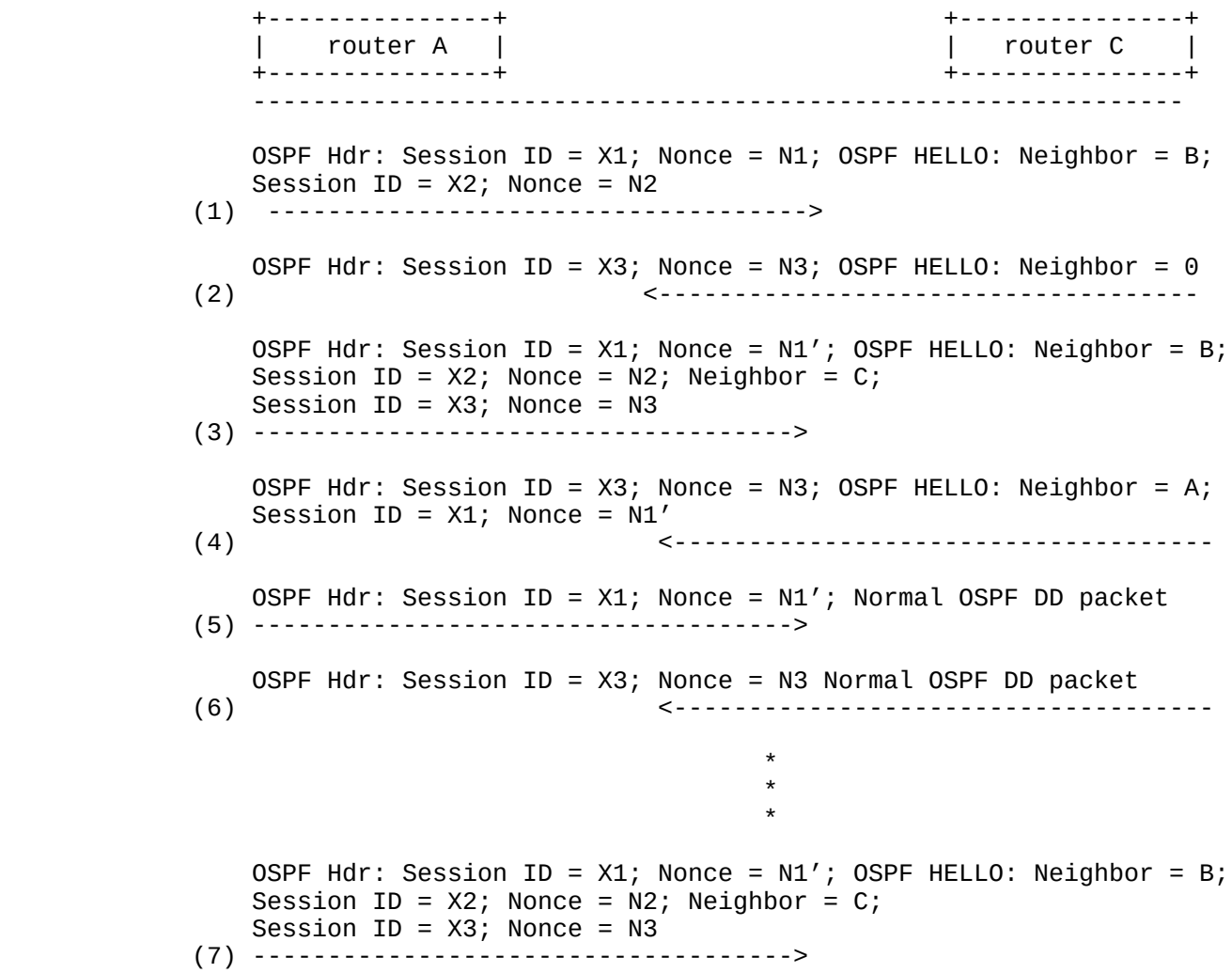


Figure 2. Scenario: another Router C comes up on that LAN

Figure 3 illustrates the scenario in which router A is rebooted. After the reboot, A lost its state and selects a new session ID (X4) and a new nonce value (N4). However, B still maintains the earlier session ID and nonce values of A (X1 and N1). In step 1, A sends a blank hello packet out with its new session ID and nonce value. After receiving the hello packet, B realizes that the session ID and the nonce value of A in the OSPF header are different from the ones maintained in its database. In order to distinguish a reboot from a replay of an old packet, B selects a new nonce value, N2’, and

transports it as well as its session ID (X2) in a hello packet to check whether the packet is from A. In the neighbor field of this packet, B continues listing A with the earlier session ID and nonce values (i.e., X1 and N1). Therefore, if an attacker attempts to send an antique packet to masquerade as A, A would update its database with the new nonce of B and send a hello packet with its existing Session ID and nonce values (X1 and N1). In step 3, B receives a new hello packet consisting of B's new nonce value from A. Since this packet lists B with the new nonce value in the neighbors field of the hello and since the nonce is new, this packet cannot be a replay. Now, B can safely assume that A has indeed restarted and can start using the new session ID and the nonce values sent by A in the neighbor field of its hellos.

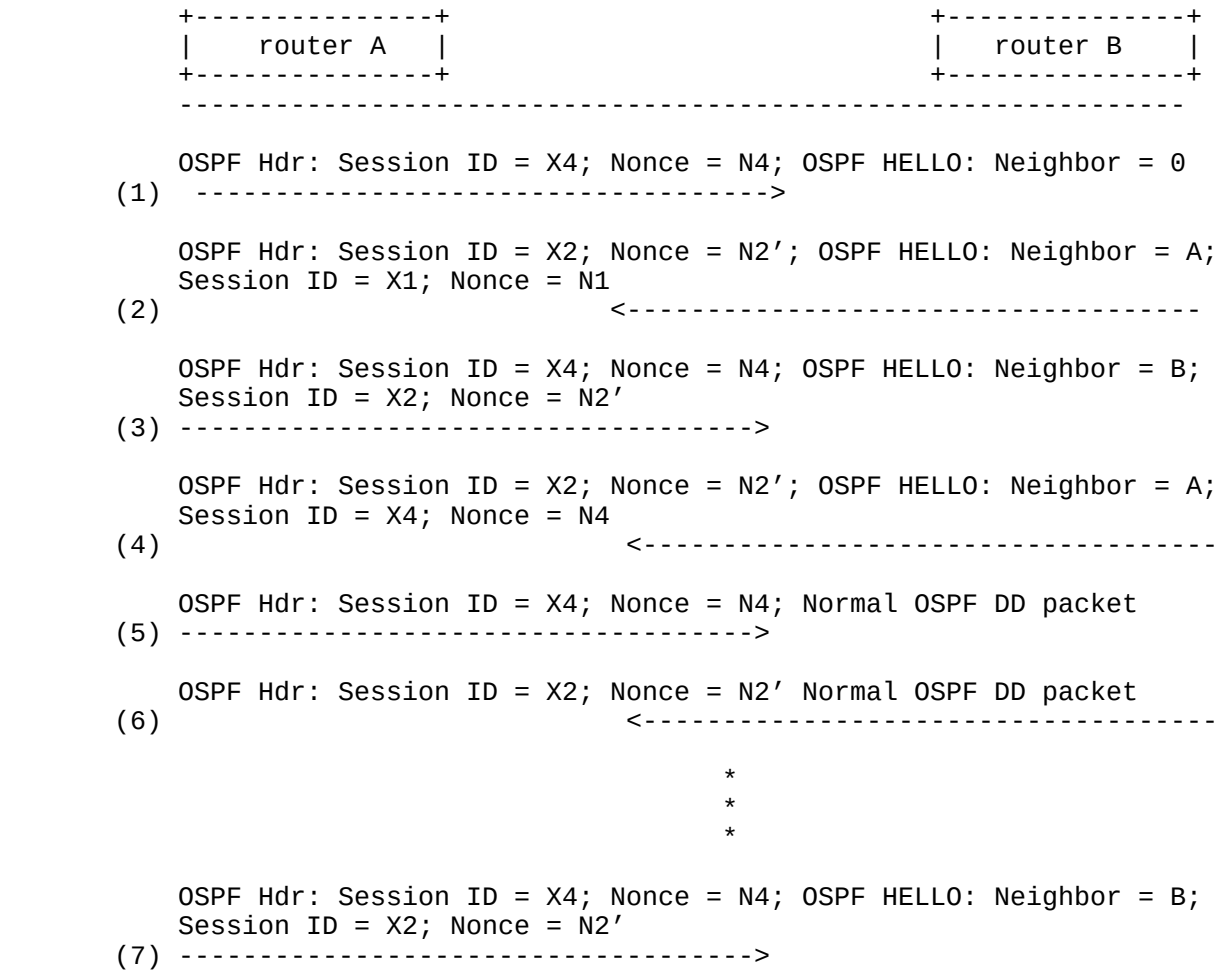


Figure 3. Scenario: Router A Reboot

2.1. Neighbor State Required

This authentication type requires the following additional fields be stored per neighbor:

- The session ID most recently received from a neighbor
- The nonce most recently received from a neighbor; this only needs to be kept up-to-date when the session ID changes or when establishing an adjacency

- o A set of sequence numbers for the neighbor; if packets are sometimes processed out of order, then a sequence number MAY be maintained for each type of packet

2.2. Receiver Behavior

This section describes how OSPF receivers will handle the reception of packets with the nonce and session ID.

If a packet is received for a neighbor in at least the 2-way state, then the session ID is compared to the one stored in the neighbor table. If the session ID does not match the session ID recorded with the neighbor, and the packet is not a hello, the packet is discarded. If the packet is a hello, then rules for hellos in following paragraphs apply. Otherwise, if the session ID matches, then if the sequence number in the cryptographic authentication option is not strictly greater than the sequence number associated with the neighbor for this type of packet, then the packet is discarded. If the cryptographic verification of the checksum fails, the packet is discarded. Otherwise, the packet is accepted by the cryptographic authentication and the sequence number associated with the neighbor for this packet type is updated to be the sequence number in the packet. The router MAY update the nonce associated with the neighbor to a nonce in a received hello packet. Updating the nonce is optional because the adjacency is already established. One case where a router implementationd would want to update nonces is where the router has recently changed session IDs without dropping all adjacencies. Such a session ID change is likely to be rare, either the result of a reboot that preserved adjacencies but might not preserve sequence numbers or running out of sequence number space.

If a hello is received for a neighbor that is not found or that has not reached 2-way state the following steps are performed. If a neighbor structure exists for the neighbor and the session ID match that stored in the neighbor structure, then the packet is processed as follows. The sequence number is checked and MUST be strictly greater than the sequence number in the neighbor structure. The cryptographic authentication is verified. If this router is listed in the set of neighbors in the hello packet, the nonce and session ID MUST match this router’s current nonce and session ID. If any of these checks fail, the packet is discarded. Otherwise the packet is accepted past cryptographic processing.

By this point, the router has received a hello packet. Either no neighbor structure exists or the session ID has changed. Before permitting communication with this router, its liveness needs to be challenged. If a neighbor has been deleted (because of a timeout) since the last nonce trigger, then a nonce trigger (see Section 2.3is

performed and the packet is discarded. If this router is listed in the list of neighbors, it MUST be listed with its current session ID and nonce otherwise the packet is discarded. If verification of the cryptographic checksum fails, the packet is discarded. If the neighbor is already in 2-way state or greater and this router is not listed in the set of neighbors, the packet is discarded. Otherwise, the session ID, nonce and all sequence numbers associated with the neighbor are updated from the packet and the packet is accepted by cryptographic authentication processing.

2.3. Nonce Triggers

The router keeps track of whether a nonce trigger has happened since the last time a neighbor is deleted.

In order to test liveness, a router updates its current nonce to a new value. As a side effect, all routers on the link that do not already have an adjacency with this router will update the nonce associated with this router. More importantly, though, the router we are testing liveness with will update the nonce in its hello entry for this router. That will allow this router to confirm that the session ID is correct and corresponds to current replay state.

As part of a nonce trigger, the router updates its current nonce. If a hello has not been sent too recently, then a hello is sent with the new nonce. The nonce trigger state is updated to indicate that no new neighbors have been deleted since the last nonce trigger.

3. Packet Format

In the challenge/ response mechanism, every OSPFv2 packet MUST carry the current Session ID and the associated Nonce value. This section describes how this information is carried in the OSPFv2 packets.

The OSPF packet header includes an authentication type field, and 64-bits of data for use by the appropriate authentication scheme (determined by the type field). Authentication types 0, 1 and 2 are defined in [RFC2328]. This document defines Authentication type 3.

When using this authentication scheme the 64 bit Authentication field in the OSPF packet header remains unchanged and is the same as defined in Section D.3 of [RFC2328]. NOTE to the WG: We can also increase the size of the Key ID. Currently it has been kept as, but nothing prevents us from changing this.

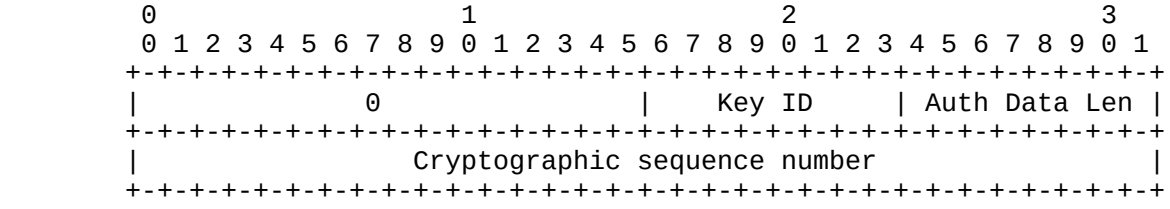


Figure 4.Usage of the Authentication field in the OSPF header when this mechanism is employed

The Session ID and the Nonce information is placed before the message digest that is appended to the OSPF packet. In this case too, the final Authentication data is not actually considered part of the OSPF protocol packet.

3.1. Extensions to OSPF packets

This section describes the new OSPFv2 packet format when this authentication scheme is being used.

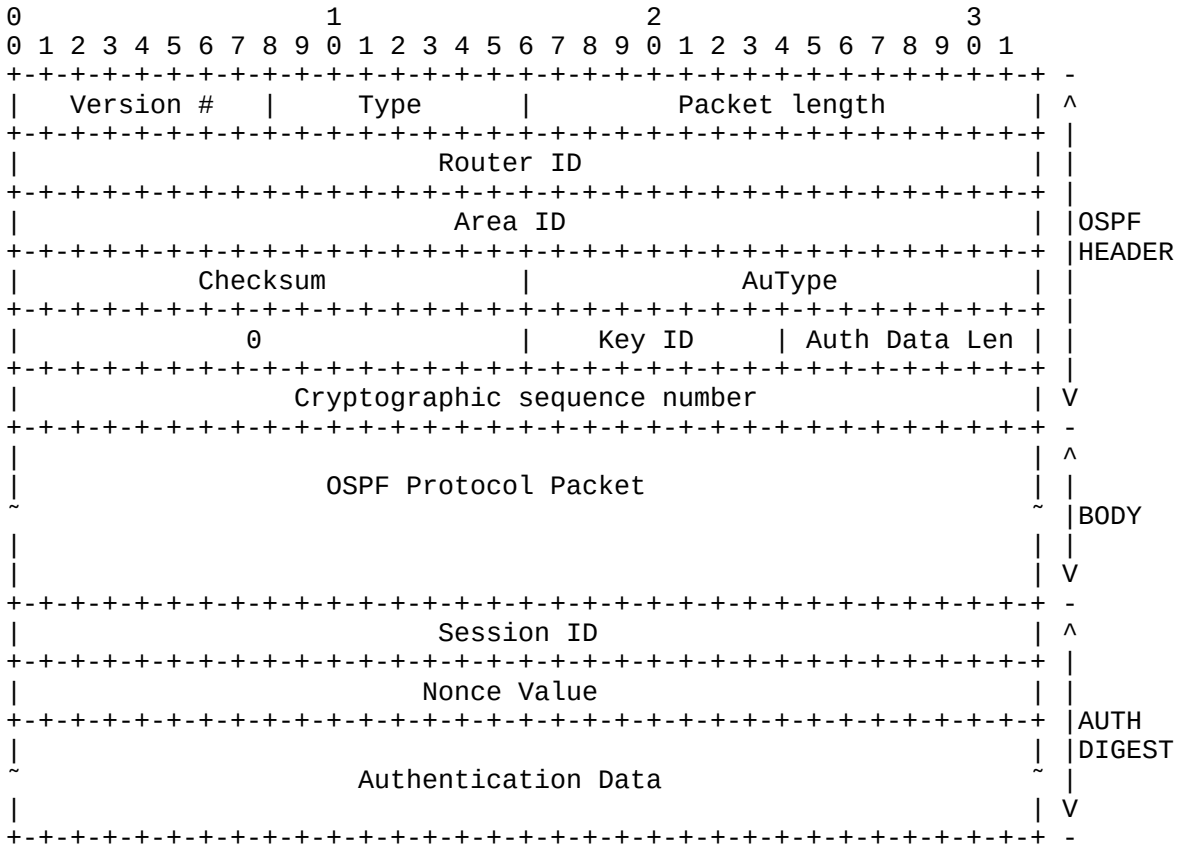
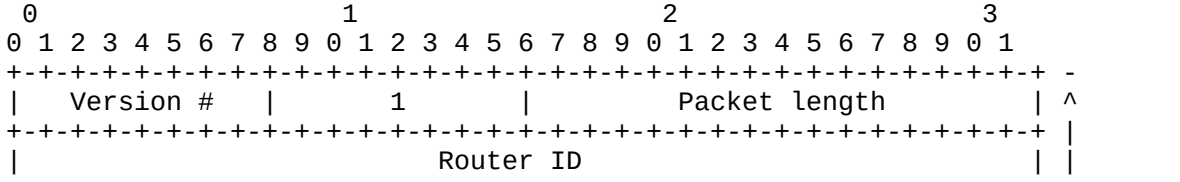


Figure 5.OSPFv2 Packet view

3.2. Extension of Hello Packet

The following figure shows an OSPF HELLO packet when this authentication scheme is being used. The HELLO payload has been modified to include each neighbor's Session ID and the Nonce value. The authentication data, as described above, carries the router's current Session ID and the Nonce value.



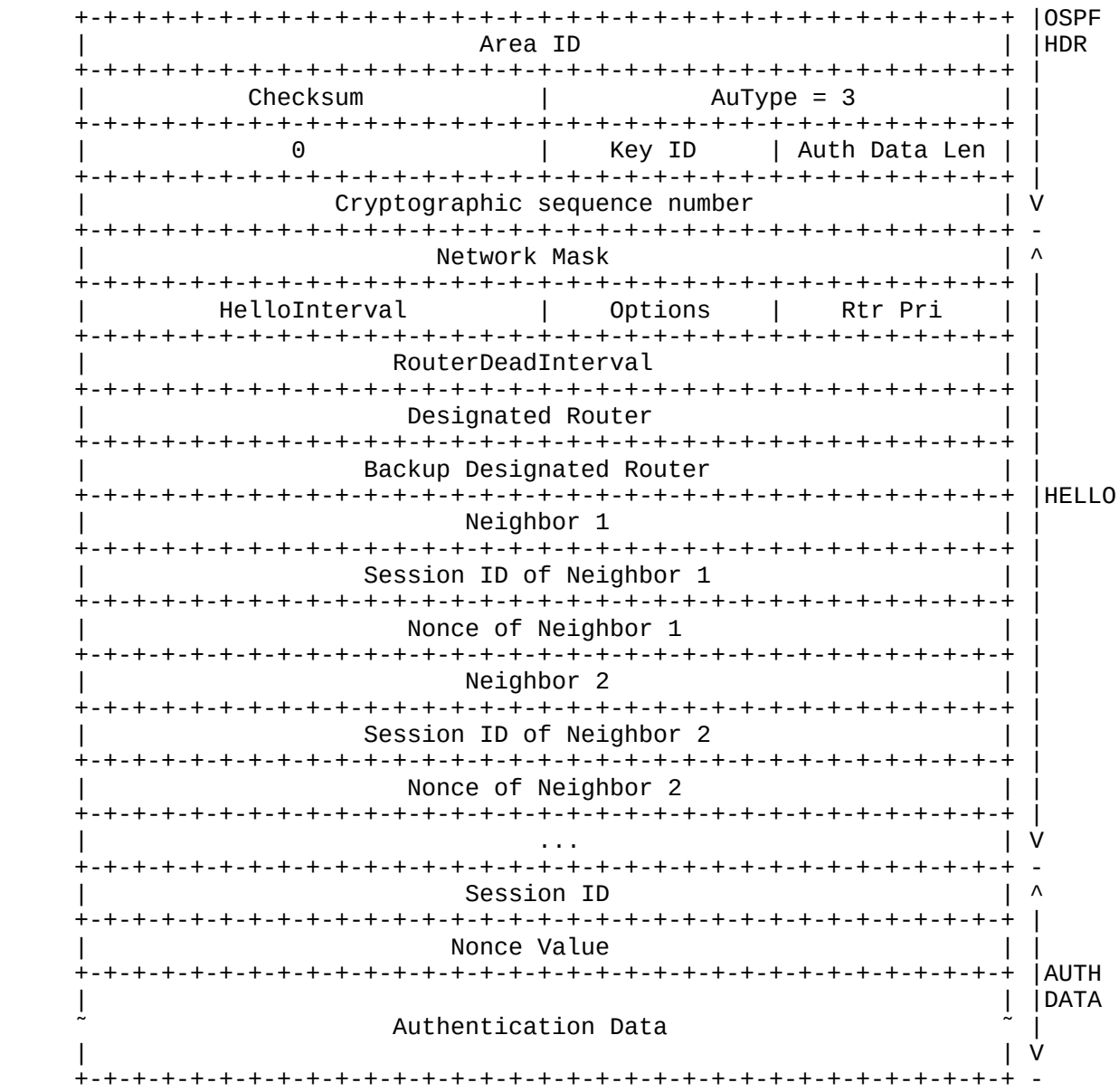


Figure 6.Extension of Protocol Packet

4. Key Selection in Processing OSPF Packets

This section introduces how the proposed security solution looks up long lived keys from key tables [I-D.ietf-karp-crypto-key-table]. Generally, a proper key selected to process an OSPFv2 packet should satisfy the requirements listed as follows:

the key is in its valid period; and

the key can be used for the desired security algorithm.

In the remainder of this section, other requirements that a selected key should particularly satisfy are depicted in different scenarios.

4.1. Key Selection in Sending Unicast OSPF Packets

Assume that a router R1 tries to send a unicast OSPF packet from its interface I1 to the interface R2 of a remote router R2 using security protocol P via interface I at time T. Firstly consider the circumstances where R1 and R2 are not connected with a virtual link. R1 then needs to select a long long-lived symmetric key from its key table. Because the key should be shared by the by both R1 and R2 to protect the communication between I1 and I2, the key should satisfy the following requirements:

the Peer field includes the router ID of R2;

the PeerKeyID field is not "unknown";

the Interfaces field includes I1; and

the Direction field is either "out" or "both".

When R1 and R2 are at the ends of a virtual link, the condition is a little more complex. Because the virtual link can be regarded as an unnumbered point-to-point network, the IP address of the interface actually used to send the packet (i.e., I1) is discovered during the routing table build process. Therefore, when the system operator deploys the keys to protect the virtual link, I1 has not been specified yet. Therefore, the key should be identified by the router IDs rather than by the interface originating the packet, and the third requirement introduced above should be changed to "the Interface field includes the router ID".

4.2. Key Selection in Sending Multicast OSPF Packets

If a router R1 sends an OSPF packet from its interface I1 to a multicast address (e.g., AllSPFRouters, AllDRouters), it needs to

select a key according to the following requirements:

the Peer field includes the multicast address;

the PeerKeyID field is "group";

the Interfaces field includes I1; and

the Direction field is either "out" or "both".

4.3. Key Selection on Receiving OSPF Packets

When Cryptographic Authentication is employed, the ID of the adopted key is encapsulated within the authentication field of an OSPF packet header. Using this ID, it is relatively easy for a receiver to locate the key. The requirement is relatively simple:

the Peer field includes the router ID of the sender; and

the PeerKeyID field includes the key ID obtained from the authentication field

5. Existing Cryptographic Authentication Mechanism

The overall cryptographic authentication process defined in [RFC5709] remains unchanged. To reduce the potential for confusion, this section minimises the repetition of text from RFC 5709 and is incorporated here by reference [RFC5709].

RFC 5709, Section 3.3, describes how the cryptographic authentication must be computed. It requires OSPFv2 packet's Authentication Trailer (which is the appendage described in RFC 2328, Section D.4.3, Page 233, items (6)(a) and (6)(d)) to be filled with the value Apad where Apad is a hexadecimal constant value 0x878FE1F3 repeated (L/4) times, where L is the length of the hash being used and is measured in octets rather than bits.

6. Mechanism to secure the IP header

This document updates the definition of Apad which is currently a constant defined in [RFC5709] to the source address thats carried in the IP header of the OSPFv2 protocol packet. Routers at the sending side must initialize Apad to a value of the source address that would be used when sending out the OSPFv2 packet, repeated L/4 times, where L is the length of the hash, measured in octets. The basic idea is to incorporate the source address from the IP header in the cryptographic authentication computation so that any change there can be detected.

At the recieving end implementations MUST initialize Apad as the source address that exists in the IP Header of the incoming OSPFv2 protocol packet, repeated L/4 times, instead of the constant that’s currently defined in [RFC5709]. Besides changing the value of Apad this document does not introduce any other changes to the authentication mechanism described in [RFC5709].

This would prevent all attacks where a rogue OSPF router changes the source address of the protocol packet and reflects it on some other interface as the authentication check would fail and all such packets would get rejected.

7. Alternative Boot Count Approach

During discussion of the challenge/response authentication approach, a desire was expressed to have a simpler alternative to consider. This section presents an alternative that obtains most advantages of the challenge/response mechanism. Instead of adding nonces and session IDs, OSPF implementations are required to keep a count of the number of times they have booted in non-volatile storage. This requirement is also placed on agents by the SNMPv3 security architecture; the same boot count can be used both for SNMP and for this OSPF mechanism.

The OSPF sequence number is extended to be 64-bits rather than 32-bits. The most significant 32-bits are the boot count. The least significant 32-bits is a counter that increases for every packet sent.

A receiver verifies that the sequence number on a received packet is strictly greater than the sequence number of the previous packet received.

Requiring that each packet have a strictly greater sequence number is a change from the current OSPF security model. However this change is required for a number of the security guarantees.

This mechanism requires fewer changes to the OSPF packet than the challenge/response mechanism. Also, the implementation complexity is somewhat less.

However there are disadvantages. First, this mechanism requires that the boot count be maintained succesfully in nonvolatile storage. If the boot count ever goes backwards without changing the encryption key, then all the attacks against the current OSPF protocol become possible against this protocol until the time that the boot count reaches a value greater than the largest value ever used for this client. This can be particularly problematic if equipment is replaced, using a router ID that has been used previously on a link but with a fresh boot count.

Another disadvantage is that the boot count mechanism does not protect against a session replayed while a router is down. If a router crashes or is taken out of service, then an attacker can replay packets as soon as the adjacencies with the router time out. The vulnerabilities of this have not been fully analyzed. Potential vulnerabilities include attacks on the designated router election process and replays of complete sessions. So far it looks like it is not likely that an attacker could bring up a replayed session far enough to inject routes from a down router.

8. Security Considerations

This document attempts to fix the manual key management procedure that currently exists within OSPFv2, as part of the Phase 1 of the KARP Working Group. This therefore, only considers manual key management mechanism to be used for OSPFv2. Any solution that takes advantage of the automatic key management mechanism is beyond the scope of this document.

This document also provides a solution to prevent certain denial of service attacks that can be launched by changing the source address in the IP header of the OSPFv2 protocol packet.

9. IANA Considerations

This document requests a new Auth Type to be defined for OSPFv2. It currently uses 3 to foster pre-standard deployments.

10. Acknowledgements

The authors would like to thank Acee Lindem for valuable contributions and helping to understand the tradeoffs surrounding various solutions.

11. References

11.1. Normative References

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

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, October 2009.

11.2. Informative References

[I-D.hartman-ospf-analysis] Hartman, S. and D. Zhang, "Analysis of OSPF Security According to KARP Design Guide", draft-hartman-ospf-analysis-02 (work in progress), December 2010.

[I-D.ietf-karp-crypto-key-table] Housley, R. and T. Polk, "Database of Long-Lived Symmetric Cryptographic Keys", draft-ietf-karp-crypto-key-table-00 (work in progress), November 2010.

[I-D.ietf-karp-threats-reqs] Lebovitz, G., Bhatia, M., and R. White, "The Threat Analysis and Requirements for Cryptographic Authentication of Routing Protocols' Transports", draft-ietf-karp-threats-reqs-01 (work in progress), October 2010.

[RFC6039] Manral, V., Bhatia, M., Jaeggli, J., and R. White, "Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010.

[RFC6094] Bhatia, M. and V. Manral, "Summary of Cryptographic Authentication Algorithm Implementation Requirements for Routing Protocols", RFC 6094, February 2011.

Authors' Addresses

Manav Bhatia
Alcatel-Lucent
Bangalore,
India

Phone:
Email: manav.bhatia@alcatel-lucent.com

Sam Hartman
Painless Security

Email: hartmans@painless-security.com

Dacheng Zhang
Huawei Technologies co., LTD.
Beijing,
China

Phone:
Fax:
Email: zhangdacheng@huawei.com
URI: