HIPRG
Internet-Draft
Expires: January 19, 2006

H. Tschofenig
A. Nagarajan
Siemens
J. Ylitalo
Ericsson
M. Shanmugam
TUHH

July 18, 2005

Traversal of HIP aware NATs and Firewalls
draft-tschofenig-hiprg-hip-natfw-traversal-02.txt

Status of this Memo

Copyright Notice

Abstract

The Host Identity Protocol is a signaling protocol which adds another layer to the Internet model and (optionally) establishes IPsec ESP SAs to protect subsequent data traffic. HIP is designed to be a middlebox friendly protocol; it allows middleboxes (such as NATs and

Firewalls) to participate in the protocol exchange in order to learn the flow identifier. Additionally, adding authentication and authorization mechanisms can help the middlebox to prevent unauthorized end hosts to gain access to the network. This document provides a problem description, goals and lists a few scenarios.

Table of Contents

1. Introduction

An IP address serves the dual role of a locator and an identifier for every host on the Internet. Since, the transport layer connections are bound to the IP address, end systems that use IP addresses as identifiers cannot support dynamic changes in the mapping between the identifier and the locator in case of multi-homing and mobility.

The Host Identity Protocol (HIP) [I-D.ietf-hip-base] proposes to separate the identifier from the locator by adding an additional layer between the transport layer and the network layer. The transport layer uses a new, mobility-unrelated identifier called as Host Identity Tags (HITs), in place of IP addresses, while the network layer uses conventional IP addresses for routing. IPsec security associations are bound to the HITs and are not modified with IP address changes. In other words, a host despite being mobile or multi-homed can use a single transport layer connection associated to one HIT and multiple IP addresses.

One of the integral features of HIP protocol is, it provides a way to establish IPsec ESP which are subsequently used to encrypt data traffic between the two end hosts. HIP, being a mobility protocol, supports dynamic changes in IP addresses. Because of this, HIP is liable to all known incompatibilities of IPsec with middleboxes such as NATs [RFC3022] and firewalls. This draft investigates some of these problems and proposes a registration mechanism in order to support the secure traversal of NATs and Firewalls.

This document is organized as follows: Section 2 introduces some terms, Section 3 provides the problem statement, the goals are listed in Section 4, Section 5 explains the HIP base exchange, Section 6 and in Section 7 we provide some security considerations.

Appendix A illustrates a possible solution. This section will be moved into a separate document with a future draft version.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This draft used the terminology defined in [RFC3022], [I-D.ietf-hip-base], [I-D.ietf-hip-esp] and [I-D.ietf-hip-arch] and [RFC2401].

The term SPI refers to the Security Parameter Index value used in IPsec packets. The initiator selects one SPI(I) that can be found in the ESP_info parameter, which is then used by the responder to create an IPsec packet (ESP packet in this case) for traffic sent to the initiator. The responder selects one SPI(R)(using ESP_info(R) parameter) which is used by the initiator to encrypt all data sent to the responder.

Other relevant abbreviations can be found in [I-D.ietf-hip-base] and [I-D.ietf-hip-esp].

The concept of a flow identifier is described in [RFC4080].

2.1 Notation

[x] indicates that x is optional.

{x} indicates that x is encrypted.

<x>y indicates that "x" is encrypted with the key "y".

--> signifies "Initiator to Responder" communication (requests).

<-- signifies "Responder to Initiator" communication (replies).

3. Problem Statement

This version of the document assumes that the data traffic following the HIP base exchange is IPsec protected and uses the mechanism described in [I-D.ietf-hip-esp] for exchanging the IPsec parameters. However, this draft can also be extended to support other HIP based key exchange mechanisms.

Besides the communicating hosts in the Internet, the entities such as NATs and Firewalls play a major role in the event of delivering packets to an appropriate host, and each meant for specific functionality. For instance, NATs are used to combat the IPv4 address depletion problem, and Firewalls are erected to protect unsolicited information flowing in and out of a corporate network. NATs use <src IP ,dst IP, src port, dst port, protocol> as an flow identifier to identify a particular traffic or connection. Because of this, protocols like IPsec suffers from well-known NAT related problems. The NAT traversal approach described in [RFC3947] and [I-D.ietf-ipsec-ikev2] allows the end hosts to detect one or more NATs in between them and [RFC3948] proposes to use the UDP encapsulation of IPsec ESP packets to traverse NATs.

Since HIP uses IPsec protection for the data traffic, the flow identifier takes the shape of a <destination IP address, SPI and ESP> (in order to support smooth traversal of the middleboxes) and the middleboxes should learn this flow id in order to relay the data packets. To achieve this, HIP aims to interact with middleboxes actively whereby these devices need to understand the HIP protocol and they need to be involved in the protocol exchange. HIP also provides a way to deal with legacy NATs, as described in [draft-nikander-hip-path-00.txt]. To support this functionality, it is necessary to provide UDP encapsulation for both HIP signaling and IPsec packets. Legacy NAT traversal does not require NATs to be HIP aware or to understand the HIP message exchange.

Even though HIP allows the middleboxes to participate in the base exchange, but scenarios like routing asymmetry poses a serious challenge for the HIP to traverse a middlebox. Section 6 explains some possible scenarios which have routing asymmetry. The inability of HIP to handle routing asymmetry motivates to use an explicit signaling mechanism for the HIP hosts in order to support secure and smooth traversal of the middleboxes.

Although HIP is described as a two-party protocol, middleboxes are supposed to intercept these messages in order to learn the flow identifier and to process them correctly. In other words, a multi party protocol is created such that the flow identifier is available to middleboxes between the HIP hosts. To provide proper security,

middleboxes should not be subject to denial of service attacks and might want to authenticate or authorize entities which create state. Note that the IPsec SA is unidirectional and therefore two IPsec SAs (with two different SPIs, ESP_info contains the SPI value) have to be established.

4. Goals

The main goal of the draft is to find a suitable NAT/FW traversal solution for the Host Identity Protocol. In the context of middlebox signaling a few goals can be accomplished:

- Add some authentication and authorization capabilities to NAT traversal. Many NAT/Firewall traversal solutions do not allow the

end host to interact with the middlebox. As a consequence, some security vulnerabilities are introduced.

- Add secure firewall traversal functionality as another type of middlebox signaling by using <destination IP address, SPI and protocol> triplet. as a substitute for the typical < source IP, destination IP, source port, destination port, transport protocol> information.

Such a solution for HIP-based middlebox signaling has to provide the following properties:

- A HIP-aware NAT/FW MUST be able to authenticate the entity requesting a NAT binding or a firewall pinhole.

- A HIP-aware NAT/FW MUST authorize the entity requesting a NAT binding or a firewall pinhole before storing state information. This requirement might be accomplished by identity based authorization or an identity independent authorization mechanism.

- A HIP-aware NAT/FW MUST be able to intercept HIP messages in order to extract the flow identifier information and other related information. HIP messages are base exchange messages during context establishment and readdressing messages during IP address changes. A NAT/FW MUST be able to distinguish these messages.

- A NAT/FW node MUST NOT introduce new denial of service attacks based on authentication or key management mechanisms.

- A potential solution MUST respect the property of some middleboxes which do not allow traffic (data and signaling traffic) to traverse this middlebox without proper authorization.

Some requirements are taken from [I-D.ietf-nsis-nslp-natfw].

5. Overview of HIP Base Exchange with Middleboxes

This section explains the HIP base exchange together with the middleboxes and describes how the middleboxes behave during the base exchange.

5.1 HIP Base Exchange with NAT

Figure 1 shows the HIP base exchange traversing a NAT.

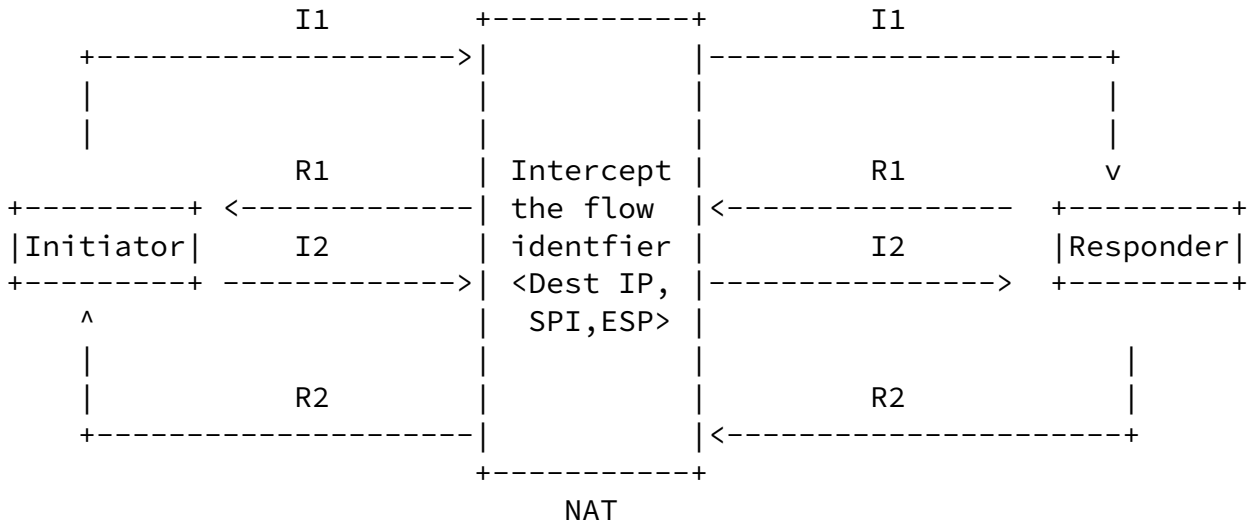


Figure 1: NAT and HIP Base Exchange

Subsequently, the HIP base exchange is described in more detail.

```
 I -> R: I1: Trigger exchange

 I <- R: R1: {Puzzle, D-H(R), HI(R), ESP Transform,
              HIP Transform }SIG

 I -> R: I2: {Solution, LSI(I), ESP_info(I), D-H(I),
              ESP_Transform, HIP Transform, {H(I)}SK }SIG

 I <- R: R2: {LSI(R), ESP_info(R), HMAC}SIG
```

A potential responsibility of the NAT, as shown in Figure 1, can be the following

- Intercept the signaling messages
- Authenticate and authorize the HIP nodes by verifying the signatures.

- Process the flow identifier information
- Perform actions according to the state machine
- Create state based on the content of message I2 with ESP_info(I) and R2 with ESP_info(R). Additionally, it might be necessary to include support for storing the respective HITs and host identities.

5.2 HIP Base Exchange with Firewall

In case of a firewall traversal, the routing asymmetry needs to be considered i.e., the fact that the messages I1 and I2 do not necessarily traverse the same devices as R1 and R2. The same is true with more complex network topologies with a mixture of NATs and Firewalls. This is an assumption made in the NSIS working group (and therefore also with NAT/Firewall traversal). Pure NAT traversal is therefore simpler to handle in comparison to middlebox traversal which also includes devices such as Firewalls. Figure 3 shows this circumstance graphically:

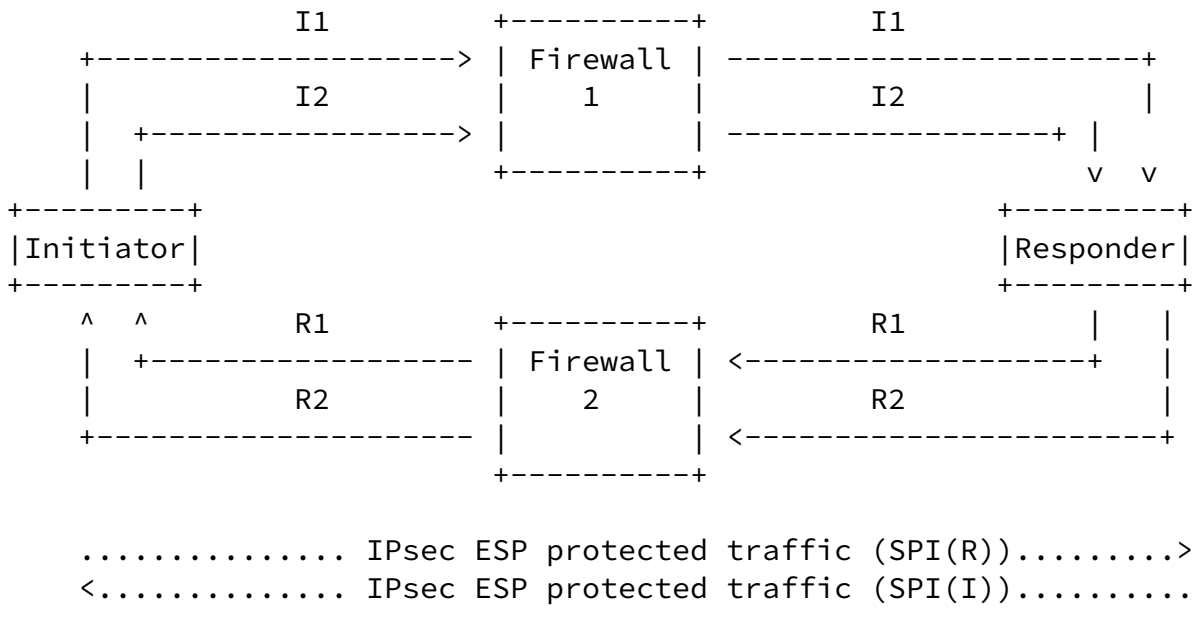


Figure 3: Firewall and HIP Base Exchange

With one single NAT between the HIP nodes, all messages of the base exchange are forced through it. With firewalls, it becomes obvious

that the nice property of a NAT with respect to the symmetric forwarding path is lost and the individual firewalls (Firewall 1 and Firewall 2) are unable to create the necessary firewall pinholes. SPI(I) is exchanged in I2 message (ESP_info(I)) through firewall 1, however firewall 2 only needs it. Similarly firewall 2 needs SPI (R) which is sent in message R2 (ESP_info(I)) through firewall 1.

6. Scenarios

The following section describes some sample scenarios and the possible solutions to learn the flow identifier:

6.1 Same Firewall at Initiator for both outgoing and incoming packets

This scenario assumes that the initiator I alone is behind a firewall named FW(I). This firewall is both for the outgoing and incoming packets and hence can look into all the base exchange messages. The FW(I) is expected to authenticate and authorize the initiator to send out going packets, receiver if necessary to let incoming packets and intercept the flow identifier from the base exchange. With the E2M messages, it can be achieved as follows. This is illustrated in Figure 4

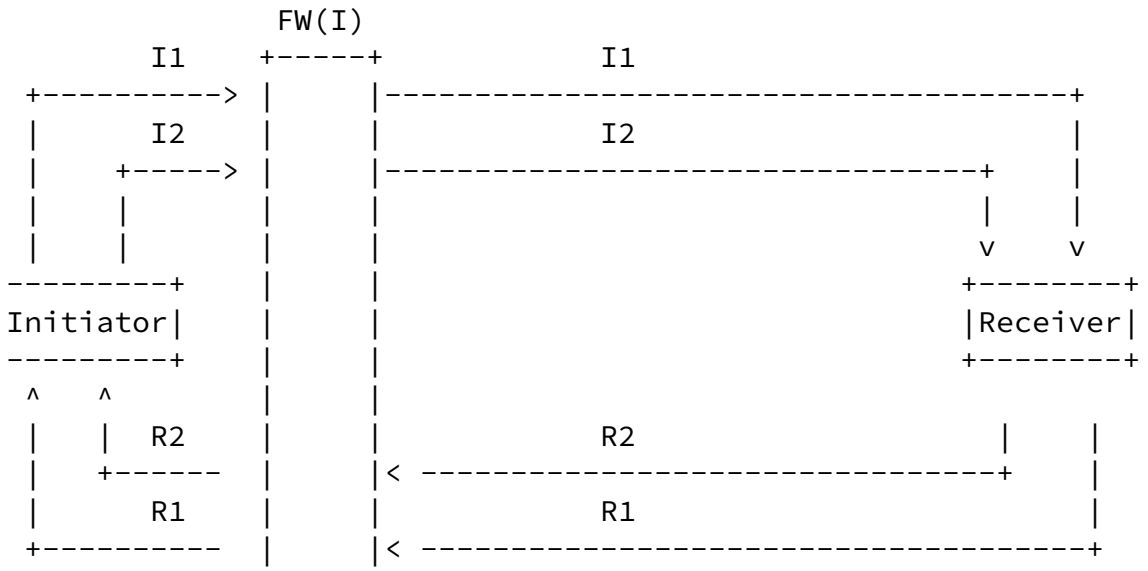


Figure 4: One FW only at initiator end

1. I1 packet is sent from the initiator I to receiver R.

2. FW(I) drops the packet and sends a R1' message back to I. This is the End host-to-Middlebox or E2M message exchange initiation.

3. I sends I2' message with CERT(I) parameters to FW(I). It requests the FW(I) to open up a pinhole.

4. FW verifies SPKI certificate and the signature of I. Accordingly, it either sends a R2' to acknowledge I that it can continue with the base exchange with message I1 or drops packet if verification fails.

5. On receiving R2',I sends message I1 to R again. Now the FW(I) will let the packet through.

6. R sends the message R1 to I.

7. On receiving R1, if FW(I) wishes to authenticate/authorize the receiver R, it should initiate E2M exchange here. It sends message R1' to R forcing R to send an I2' in exchange.

8. R sends the CERT(R) parameter in I2'.

9. FW verifies SPKI certificate and the signature of R. Accordingly, it either sends a R2' to acknowledge R that it can continue with the base exchange with message R1 or drops packet if verification fails.

10. On receiving R2', R sends message R1 to I again. Now the FW(I) will let it through.

11. The base exchange continues until complete. Since all messages I1,R1,I2 and R2 traverse through FW(I), it can look into these messages to learn the flow identifier information.

6.2 Different Firewalls at Initiator for outgoing and incoming packets

This scenario assumes that the initiator I alone is behind firewalls named FW1(I) and FW2(I).FW1(I) is for the incoming packets to I and FW2(I) is for the outgoing packets to R. The FW(I) is expected to authenticate and authorize the initiator to send out going packets, while FW2(I) would authenticate and authorize the receiver, if necessary to let incoming packets. It is sufficient that FW2(I) alone learns the flow identifier information of I. It should store the state <SPI(I),IP(I),HIT(I)> to forward IPsec protected payload packets. This scenario is illustrated in Figure 5

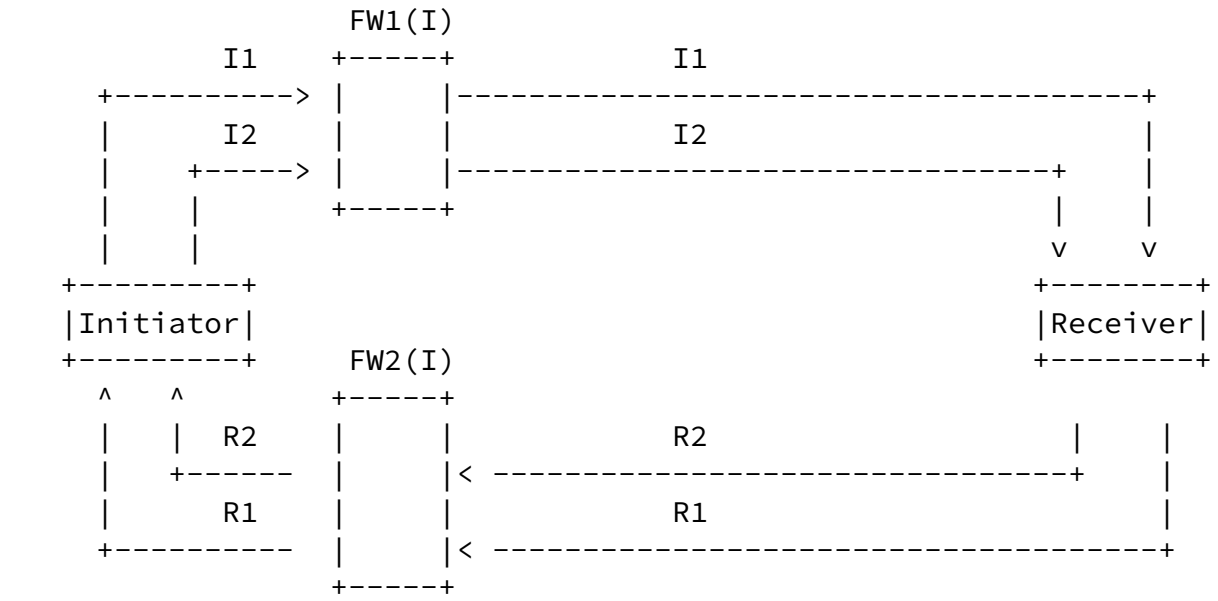


Figure 5: Two FWs at initiator's end

1. I1 packet is sent from the initiator I to receiver R.

2. FW1(I) drops the packet and sends a R1' message back to I. This is the E2M message exchange initiation.

3. I sends I2' message with CERT(I) parameters to FW1(I). It requests the FW1(I) to open up a pinhole.

4. FW1(I) verifies SPKI certificate and the signature of I. Accordingly, it either sends a R2' to acknowledge I that it can continue with the base exchange with message I1 or drops packet if verification fails.

5. On receiving R2',I sends message I1 to R again. Now the FW1(I) will let the packet through.

6. R sends the message R1 to I.

7. On receiving R1, if FW2(I) wishes to authenticate/authorize the receiver R, it should initiate E2M exchange here. It sends message R1' to R forcing R to send an I2' in exchange.

8. R sends the CERT(R) parameter in I2'.

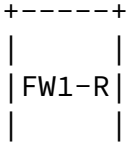
9. FW2(I) verifies SPKI certificate and the signature of R. Accordingly, it either sends a R2' to acknowledge R that it can continue with the base exchange with message R1 or drops packet if verification fails.

10. On receiving R2', R sends message R1 to I again. Now the FW2(I) will let it through.

11. Since FW2(I) needs the store the state, once the base exchange is complete, the initiator should inform the FW2(I) about the SPI it has chosen for the exchange. This way, FW2(I) can forward further IPsec payload packets from R to I

6.3 Different Firewalls at Initiator and Receiver

This scenario looks into a more complicated situation. Initiator I is behind multiple firewalls FW1(I) for outgoing packets and FW2(I) and FW3(I) are for incoming packets. Similarly, receiver R is behind FW1(R) and FW2(R) for incoming packets and FW3(R) for outgoing packets. The incoming firewalls are chosen based on the type of the application and the hosts are unaware from which firewall they receive packets. Here, however for our scenario we assume that FW2(R) and FW2(I) are chosen about which also the hosts are unaware of. The FW1(I) is expected to authenticate and authorize the initiator to send outgoing packets to R, while FW2(R) would authenticate and authorize the receiver to let outgoing packets to I. FW2(R) should store the state <SPI(R),IP(R),HIT(R)> for the receiver while FW2(I) should store the state <SPI(I),IP(I),HIT(I)> for the initiator. This scenario is illustrated in Figure 6



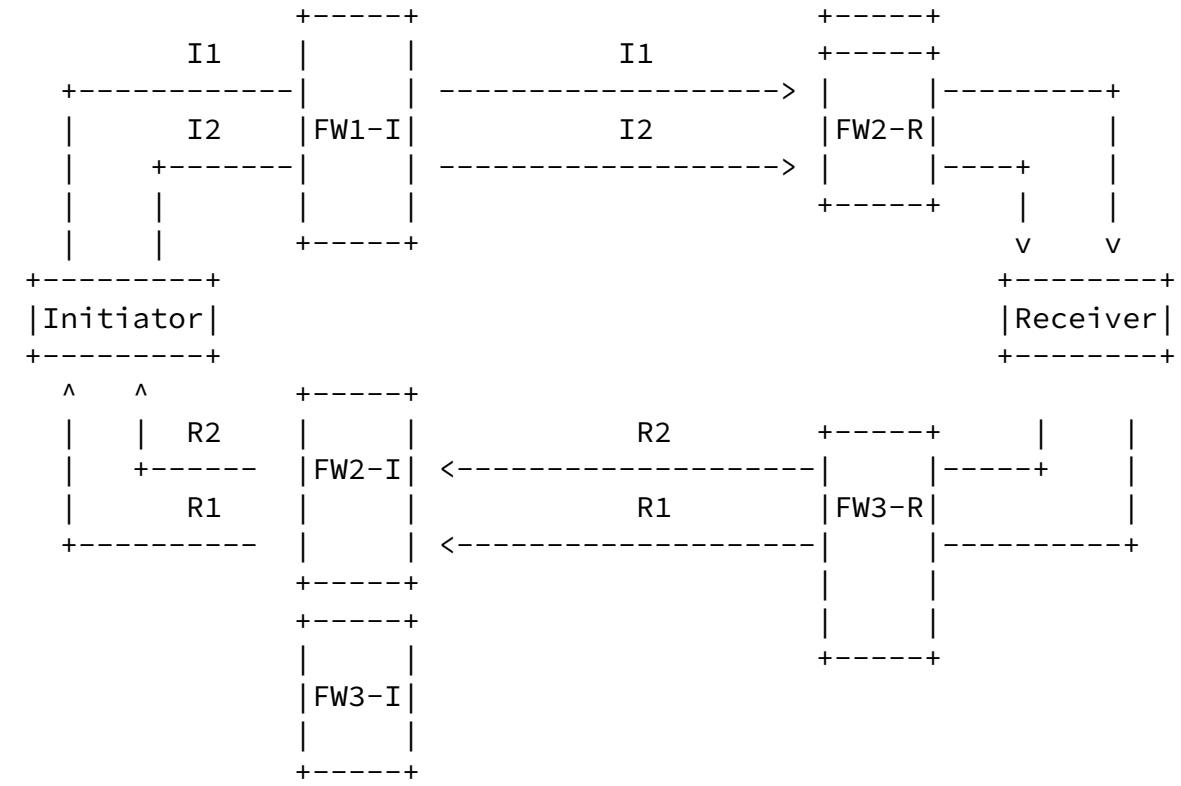


Figure 6: Multiple FWs at initiator's and receiver's end

1. I1 packet is sent from the initiator I to receiver R.

2. FW1(I) drops the packet and sends a R1' message back to I. This is the E2M message exchange initiation.

3. I sends I2' message with CERT(I) parameters to FW1(I). It requests the FW1(I) to open up a pinhole.

4. FW1(I) verifies SPKI certificate and the signature of I. Accordingly, it either sends a R2' to acknowledge I that it can continue with the base exchange with message I1 or drops packet if verification fails.

5. On receiving R2',I sends message I1 to R again. Now the FW1(I) will let the packet through.

6. This packet would reach FW2(R). If this firewall wishes to authenticate and authorize the initiator I, then it can start a E2M exchange with I. After this is successfully completed, FW2(R) would open up a pinhole to send packets to R.

7. R sends the message R1 to I.

8. When R sends R1 to I, FW3(R) would initiate a E2M message to authenticate and authorize the receiver R. After this is complete, it will forward the packet to the initiator. On receiving R1, if FW2(I) wishes to authenticate/authorize the receiver R, it should initiate E2M exchange here.

9. FW2(I) verifies SPKI certificate and the signature of R. Accordingly, it either sends a R2' to acknowledge R that it can continue with the base exchange with message R1 or drops packet if verification fails.

10. On receiving R2', R sends message R1 to I again. Now the FW2(I) will let it through.

11. This has completed only one round of authentication and authorization. However, the states are still not established at the firewalls. For this, the hosts have to signal their incoming firewalls about the SPI that they have chosen for IPsec ESP packets to follow.

When hosts are behind multiple incoming firewalls, there are unable to decide to which firewall they have to inform their SPI values to. The first option would be to somehow make the chosen FW to signal the host about its requirement for a state to forward IPsec protected packets (similar to a pull model). This could be possibly done along with the first incoming packet which is R1. R1 packet could include extra signaling as record route to the initiator. The second option would be to inform firewall about the SPI values (like the push model). Here, however it would be necessary to send an extra message I3 from the initiator to the receiver which would include the ESP_info(I) for FW(I) and to resend the ESP_info(R) in I2 message for FW(R).

The second problem is to secure the SPI signaling message from the end host to the FW. Since the end hosts authenticate and authorize to the FW that lets outgoing packets, they share keys only with them. However, they need to signal the SPI value to the FW on the other end which forwards incoming packets . For the sake of securing the SPI value, it might be necessary that the end hosts have to run a E2M exchange with the firewalls on the receiving end also.

7. Security Considerations

This document provides problem description and overview of the scenarios for the Host identity protocol, when deployed with the middleboxes. As motivated in the previous sections, in order to provide a smooth traversal of middleboxes, an explicit signaling mechanism is necessary. The solution approach should satisfy the following properties:

- SHOULD be resistant to denial of service attacks.
- MUST authenticate and authorize the end hosts.
- A potential solution MUST respect the property of some middleboxes which do not allow traffic (data and signaling traffic) to traverse this middlebox without proper authorization.
- MUST not be vulnerable to Man-in-the-Middle attacks.
- MUST protect the end hosts against replay attacks.

8. Acknowledgements

The authors would like to thank Pekka Nikander, Dieter Gollmann and Thomas Aura for their feedback to this document.

This document is a byproduct of the Ambient Networks Project, partially funded by the European Commission under its Sixth Framework Programme. It is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranties of fitness for a particular purpose. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks Project or the European Commission.

9. References

9.1 Normative References

[I-D.ietf-hip-base]
Moskowitz, R., "Host Identity Protocol", draft-ietf-hip-base-03 (work in progress), June 2005.

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

9.2 Informative References

[I-D.ietf-hip-arch]
Moskowitz, R., "Host Identity Protocol Architecture", draft-ietf-hip-arch-02 (work in progress), January 2005.

[I-D.ietf-hip-esp]
Jokela, P., "Using ESP transport format with HIP", draft-ietf-hip-esp-00 (work in progress), July 2005.

[I-D.ietf-ipsec-ikev2]
Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-17 (work in progress), October 2004.

[I-D.ietf-nsis-nslp-natfw]
Stiemerling, M., "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", draft-ietf-nsis-nslp-natfw-06 (work in progress), May 2005.

[I-D.saml-tech-overview-1.1-03]
Maler, E. and J. Hughes, "Technical Overview of the OASIS Security Assertion Markup Language (SAML) V1.1", March 2004.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC3947] Kivinen, T., Swander, B., Huttunen, A., and V. Volpe,

"Negotiation of NAT-Traversal in the IKE", RFC 3947, January 2005.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

[RFC4080] Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, June 2005.

[SPINAT] Ylitalo, J., Melen, J., Nikander, P., and V. Torvinen, "Re-thinking Security in IP based Micro-Mobility", 7th Information Security Conference (ISC-04), Palo Alto,", September 2004.

[SPINAT1] Ylitalo, J., Melen, J., and P. Nikander, "SPINAT: A Security Framework for Local IP Mobility Management, unpublished manuscript", November 2003.

[draft-ietf-ipsec-esp-v3-08]
Kent, S., "IP Encapsulating Security Payload (ESP)", draft-ietf-ipsec-esp-v3-10 (work in progress) (work in progress), March 2005.

[draft-koponen-hip-registration-00.txt]
Laganier, J., Koponen, T., and L. Eggert, "Host Identity Protocol (HIP) Registration Extension", draft-koponen-hip-registration-00.txt (work in progress), February 2005.

[draft-nikander-hip-path-00.txt]
Nikander, P., Tschofenig, H., Henderson, T., Eggert, L., and J. Laganier, "Preferred Alternatives for Tunnelling HIP (PATH)", draft-nikander-hip-path-00.txt (work in progress) (work in progress), February 2005.

Authors' Addresses

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Hannes.Tschofenig@siemens.com
URI: http://www.tschofenig.com

Aarthi Nagarajan
Siemens
Otto-Hahn-Ring 6
Munich, Bayern 81739
Germany

Email: aarthi.nagarajan@tuhh.de

Jukka Ylitalo
Ericsson Research Nomadiclab
Jorvas FIN 02420
Finland

Phone: +358 9 299 1
Email: jukka.ylitalo@ericsson.com

Murugaraj Shanmugam
Technical Univeristy Hamburg-Harburg
Schwarzenbergstrasse 95
Harburg, Hamburg 21073
Germany

Email: murugaraj.shanmugam@tuhh.de

Appendix A. Solution Approach

A.1 Flow identifier interception

The most important issue with the HIP NAT/FW traversal is to make the

flow identifier <destination IP address, SPI and ESP> available to the middleboxes. In the presence of NATs, we are always sure that the forward path and the backward path are same, since the NAT forces the IP packets to flow through these devices. Hence all the 4 messages I1, R1, I2 and R2 traverse through a single NAT. This makes it possible for the NATs to intercept the messages for the relevant flow identifier information. But, in the presence of firewalls, routing asymmetry has to be taken into consideration.

To enable the firewalls intercept the correct mapping triplet < dest IP, SPI, ESP > certain values have to be resent with the base exchange messages. This is illustrated in the Figure 7. While the IP value of the flow identifier can be intercepted from the IP header of any base exchange message, the SPI value can be intercepted only in messages I2 (using ESP_info(I)) and R2 (using ESP_info(R)). I generates its SPI(I) and sends it to R through FW-R. However, FW-I needs this information to forward all packets from R to I. Therefore there has to be someway FW-I can learn this information. One possible method would be that message R2 could include the ESP_info(I) value. However, changes to the base exchange are not desired and we try to keep the base exchange unaffected. The only other possibility would be that once the base exchange is complete, the HIP host I could inform the FW-I in its domain about its SPI(I) value. Similarly, the receiver R could inform the FW-R local to it about its SPI(R).This way, the firewalls will be able to learn the SPI values needed to create the state.

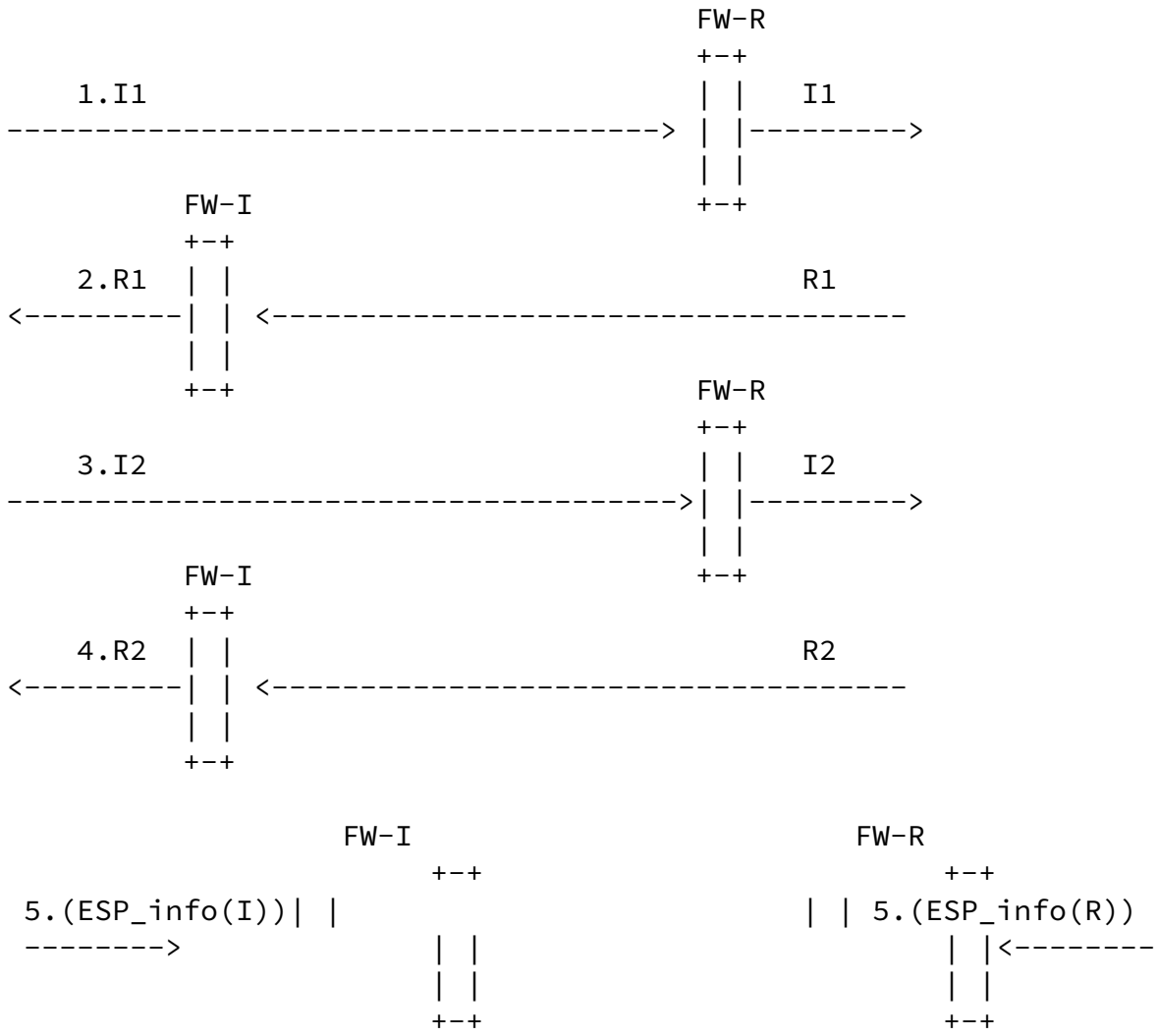


Figure 7: Firewalls and mapping information during Base exchange

A.2 Sender Invariance

The NAT/Firewall HIP node establishes state at possibly several entities between the HIP Initiator and the HIP Responder. Providing authentication of the signaling initiator to each individual HIP node along the path might be possible but not particularly useful, since the initiator is most likely unknown to some middlebox along the path. Hence, authentication per se does not solve the security problem.

With mobility, it might be possible that the intermediate HIP-aware middlebox need some assurance that a particular node is the allowed to modify existing state. No other entity should be allowed to modify state since this would allow certain attacks (such as denial of service or third party flooding). In some respect this issue is similar to the authorization property in Mobile IP where the mobility

binding state established at the CN needs to be protected against unauthorized modifications.

It seems that the property of "sender Invariance" is required in this case: "A party is assured that the source of the communication has remained the same as the one that started the communication, although the actual identity of the source is not important to the recipient."

This property is particularly important in the context of mobility which requires a change in the NAT binding or the packet filter. SPINAT (see [SPINAT] and [SPINAT1]) provided innovative aspects by using a hash chain approach in combination with delayed authorization to secure state modifications at NAT devices.

A future version of this document will address the aspect of sender invariance in more details.

A.3 Authentication and Authorization

Before a middlebox can allocate a NAT binding or a pin hole, the HIP nodes requesting them may need to be authenticated. Middleboxes could potentially use information stored in the DNS to authenticate the HIP end points. Since Host Identities are used to identify HIP nodes, middleboxes can also use signature verification at relevant HIP messages for authentication. This might raise some issues on denial of service attacks at the middleboxes and these need to be determined. Authorization is certainly more important than authentication particularly since HIP supports ephemeral host identities as a mechanism to preserve privacy. As such it would be useful to use identity independent authorization assertions. SPKI certificates, attribute certificates or similar mechanisms could be of particular use, especially in cases where the HIP nodes prefer to remain anonymous.

A.3.1 What is SPKI?

SPKI authorization certificates are used in access control and are identity independent. Issuing and receiving an SPKI certificate is completely local to the network domain and there is no need for a higher certification authority to issue them. For a HIP protocol this would mean whenever a HIP host wishes to create a NAT binding or a FW pinhole, it can locally obtain the SPKI certificate for

authorization at middleboxes. The structure of the SPKI certificate is shown in Figure 8.

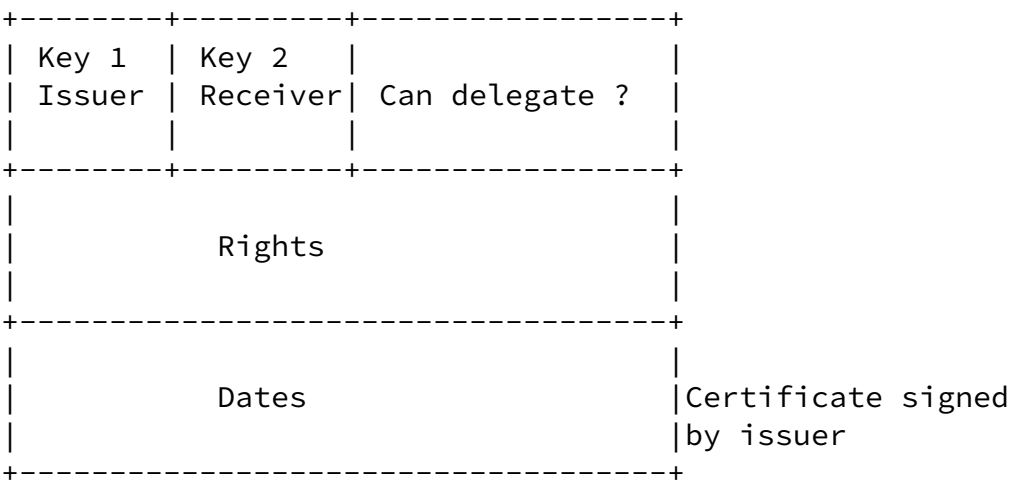


Figure 8: SPKI certificate structure

- Key 1 is the public key of the certificate issuer.
- Key 2 is the public key of the certificate receiver.
- If a subject gets the right to re-delegate its rights, it can re-delegate its certificates to other subjects. In addition, he can freely sign new certificates to other subjects.
- Rights define access control of the receiver.
- Dates define the validity period of the certificate.
- The complete certificate is signed by the private key of the issuer.

When a subject wishes to use his certificates, it sends a request that is signed by the subject's private key. Attached are a chain of certificates that belong not only to it but also to those of its delegates. When a verifier receives requests along with a chain of

certificates from a subject, the verifier verifies the requests and the certificates. If the verifier is satisfied with the certificates, then the requested operation is allowed. Otherwise, the requests will be refused.

A.3.2 SAML Usage in HIP

Security Assertion Markup Language (SAML) [I-D.saml-tech-overview-1.1-03] is an XML extension for security information exchange. It is being developed by OASIS. SAML enables entities to access resources by providing assertions which allow authorization.

A.3.2.1 Assertions

An Assertion is a package of information including authentication statements, attribute statements and authorization decision statements. All kinds of statements do not have to be present, but at least one. An Assertion contains several elements:

Issuing information:

Who issued the assertion, when was it issued and the assertion identifier.

Subject information:

The name of the subject, the security domain and optional subject information, like public key.

Conditions under which the assertion is valid:

special kind of conditions like assertion validity period, audience restriction and target restriction.

Additional advice:

explaining how the assertion was made, for example.

In an authentication statement, an issuing authority asserts that a certain subject was authenticated by certain means at a certain time.

In an attribute statement, an issuing authority asserts that a certain subject is associated with certain attributes which has certain values. For example, user jon@cs.example.com is associated with the attribute 'Department', which has the value 'Computer Science'.

In an authorization decision statement, a certain subject with a certain access type to a certain resource has given certain evidence that the identity is correct. Based on this, the relying party then makes the decision on giving access or not. The subject could be a human or a program, the resource could be a webpage or a web service, for example.

A.3.2.2 Artifact

The Artifact is a base-64 encoded string which is 40 bytes long. 20

bytes consists of the typecode, which is the source id. The remaining 20 bytes consists of a 20-byte random number that servers use to look up an assertion. The entity creating an Assertion stores it temporarily. The entity performing the authorization decision uses the received Artifact to retrieve the assertion. The purpose of the Artifact is to act as a token which references an Assertion.

SAML also defines a request/response protocol for obtaining Assertions. The request asks for an Assertion or makes queries for authentication, attribute and authorization decisions. The response is carrying back the requested Assertion. The XML format for protocol messages are defined within an XML schema.

A HIP-aware NAT/Firewall can use this request/response protocol to fetch assertions from the indicated place.

HIP can use SAML Assertions in CER payloads to provide a mechanism for HIP end points to authorize them towards middlebox using an emerging technology. Furthermore, SAML Assertions can be used to bind the authorization decision of different protocols sessions from different layers in the ISO-OSI model together. As an example, the

authorization decision by an application layer entity can be used to bind it to a subsequent HIP exchange. SAML provides a complete solution for authorization using Artifacts and Assertions and the corresponding protocols to obtain them. The assertions are based on XML which allows extensibility beyond the initially envisioned deployment area.

A.3.3 SPKI usage for HIP

HIP has already defined the CERT parameter that can carry certificates. The HIP nodes requesting a NAT/FW traversal can send their base exchange message with the CERT parameter. The CERT will carry the SPKI certificate and the packet will be signed by the requesting HIP node. This would mean, messages I2 and R2 should include the CERT parameter to get them authorized at the middleboxes. The structure of the SPKI certificate for HIP is shown in Figure 9.

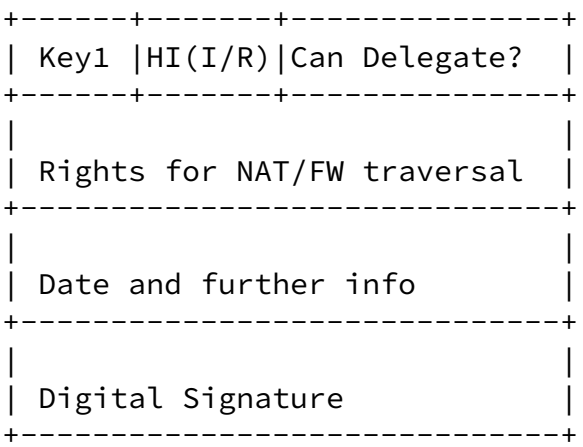


Figure 9: SPKI certificate structure for HIP

A.3.4 Authentication and authorization for Base Exchange

When a HIP host requests a NAT binding or a FW pinhole, it has to be first authenticated and authorized by the middleboxes. Authentication is necessary in many cases,because, in case of mobility, the middlebox should be authorized to change the flow id and no other party forge the middlebox to change. Since all HIP packets are signed using the private keys of the HIP hosts, middleboxes can verify these packets using the signature verifications. This, of course, will introduce certain kinds of denial of service attacks. Denial of service attacks for signature verification at middleboxes can be prevented by using the client puzzle concept used by the HIP protocol. For more details the HIP protocol [I-D.ietf-hip-base] can be consulted. This will force the middleboxes to delay state creation and to also delay expensive computational operations. As explained in the previous sections, we seek to use non-identity based authorization mechanisms that can be verified by the middleboxes before creating a NAT binding or FW pinhole. Since NATs force the outbound and inbound packets to flow through them, they are much easier to handle. For instance, the mechanism used by SPINAT [SPINAT] can be used for authorization of state modifications by utilizing hash chains and delayed authentication with NATS. However, this is not presently suitable for firewalls with asymmetric paths. More work needs to be done towards extending this idea for a combination of NATs and firewalls with routing asymmetry.

A HIP host behind a firewall might need to register itself with local middleboxes before the base exchange can be initiated or completed. Firewalls might not allow the traffic to bypass the firewall. For this, we consider using messages I1',R1',I2' and R2' which are an extended version of the normal base exchange messages used in HIP.

However, these messages are exclusively used only for configuring the HIP host with the firewalls such that authentication and authorization is complete before the firewall opens up a pinhole. With this approach, we make fewer changes to the base exchange by avoiding the inclusion of certain authorization parameters into them. We refer to this exchange as 'Registration Procedure', defined in [draft-koponen-hip-registration-00.txt], as shown in Figure 10 which

provides more details of this lightweight protocol exchange.

```
 End host-to-Middlebox or E2M messages

 I -> R: I1: Trigger exchange
      OR
 I -> FW1: I1': Trigger exchange

 I <- FW1: R1': {Puzzle, D-H(R), HI(R),HIP Transform}SIG

 I -> FW1: I2': {Solution,D-H(I),HIP Transform,{H(I)},CERT(I)}SIG

 I <- FW1: R2': {HMAC}SIG
```

Figure 10: HIP NAT/Firewall Registration Procedure

As an overview, we modify the HIP exchange protocol to authenticate the middlebox towards the initiator, to authorize (and possibly authenticate) the initiator towards the firewall and to establish a security association between the initiator and the responder. We reuse the HIP protocol for this purpose to use the same infrastructure and to benefit from a lightweight protocol. Note that the message flow in Figure 10 does not establish IPsec security associations. These security associations are not necessary in most scenarios.

When a host I wishes to create a pinhole with a FW on its side (named as FW-I), it has two choices:

- It sends a regular I1 message to the firewall. This assumes that the end host knows that a firewall is located in the network and additionally the address of this firewall is also known to the end host. This might be the case in a corporate network environment. This is shown as the I1' message.

- The initiator I can also send a regular HIP I1 message towards a destination host (denoted as R). This message will then be intercepted by the firewall and a R1' is returned.

With R1' the firewall sends a puzzle to the initiator similar to the

one sent from a HIP receiver to a HIP initiator. The initiator solves the puzzle and sends the solution back to the FW along with its SPKI certificate using the I2' message. Note that the Initiator can send its certificate in the I1/I1' message. This will, however, create a state even before the client puzzle solution is obtained from the initiator. This raises some denial of service concerns. The FW can validate this SPKI certificate and authorize the HIP host I1. This packet is not liable to any denial of service or replay attacks as the solution is dependent on the cookie that R1' included. Hence, the FW can look into the cookie index to avoid unwanted signature verifications. The ESP transforms are also dropped here as the there will be no IPsec ESP packets exchanged between the HIP host and the FW. There is also no need for the ESP_info values in I2' and R2' messages.

Once the FW receives the I2' packet, it verifies the solution to make sure that it is the entity to which it sent the R1' packet. It sends a R2' packet back to the initiator as an acknowledgement for authorization. The R2' packet however should include a HMAC to prevent denial of service attacks on I.

After I receives the R2' packet, it can now initiate the normal base exchange that the FW will forward to R.

On receiving I1, receiver R will send a R1 message back to the initiator. However, since the FW-R at the receiver end also needs to authenticate and authorize the receiver, we run the registration procedure with the E2M messages similar to the previous step between FW-R and receiver R. Once receiver R receives the acknowledgement R2', it now sends packet R1 to the initiator that the FW-R will forward. The rest of the base exchange continues as usual. However for the sake of the ESP_info interception at the firewalls, as mentioned before signaling messages have to be sent from the HIP hosts to their local middleboxes about the SPI values (using ESP_info parameter)they have agreed on.

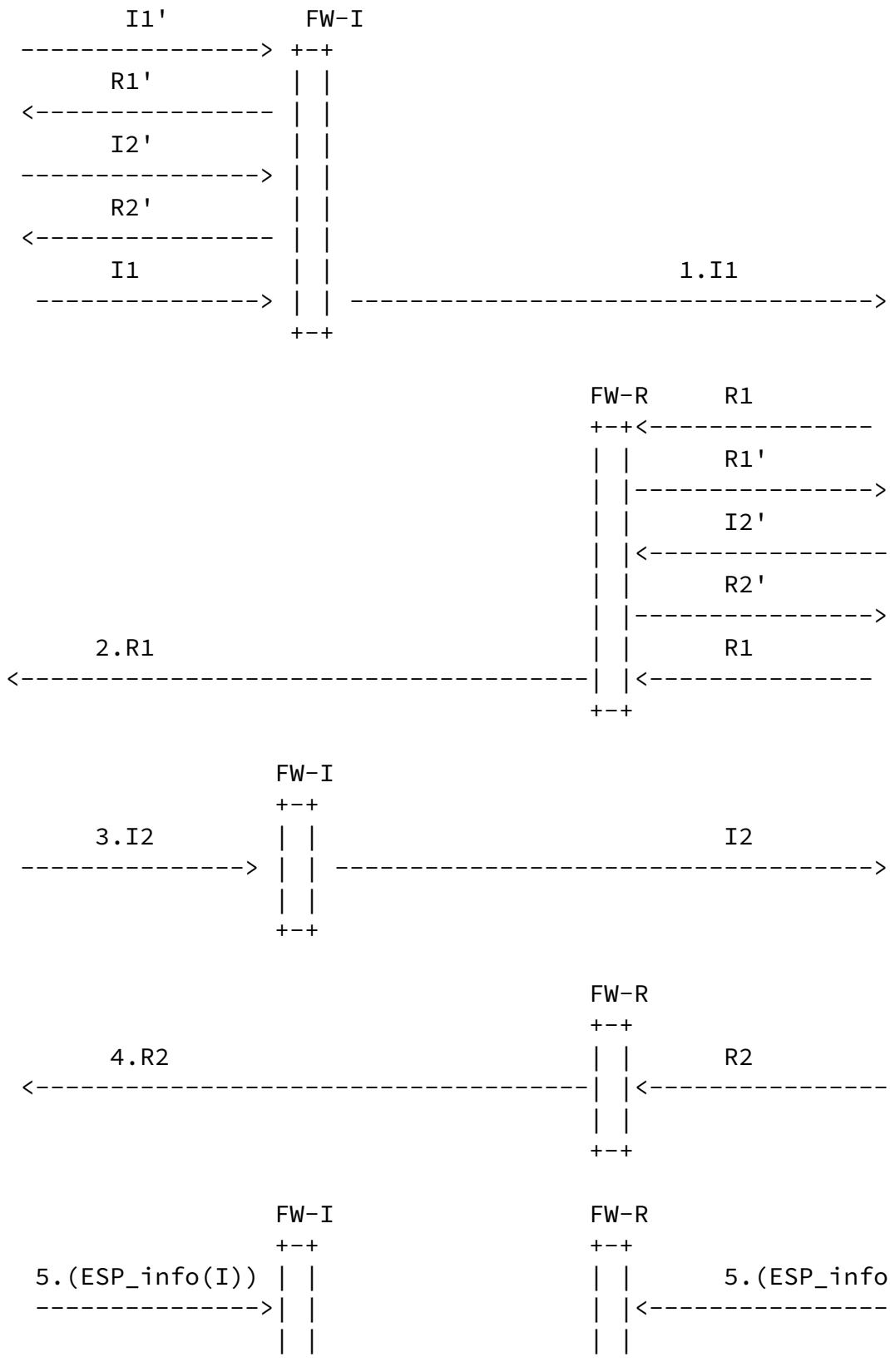
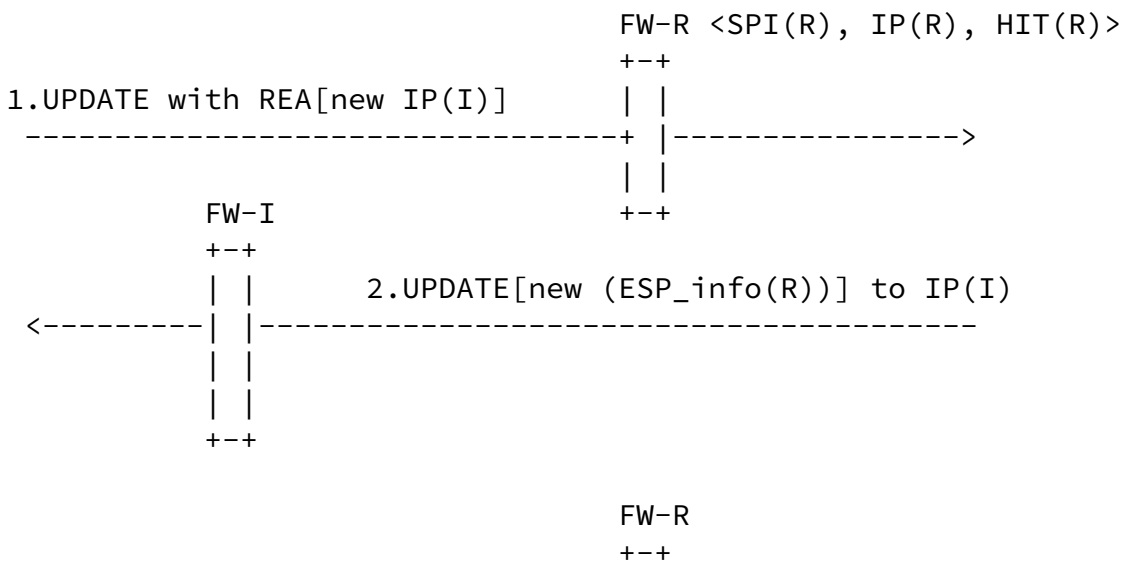

I1'
FW-I
R1'
I2'
R2'
I1
1.I1
FW-R
R1
R1'
I2'
R2'
R1
2.R1
FW-I
3.I2
I2
FW-R
4.R2
R2
FW-I
FW-R
5.(ESP_info(I))
5.(ESP_info(R))

Figure 11: Authentication and authorization for base exchange messages

A.3.5 Authentication and authorization for Readdressing

After the base exchange is complete, IPsec payload packets are exchanged among the HIP hosts. The middleboxes use the state that is established with them to forward such packets to the HIP hosts. The state at FW-R is < SPI(R), IP(R), HIT(R) > and state at FW-I will be < SPI(I), IP(I), HIT(I) >.When one of the HIP hosts moves, it sends an UPDATE message to its peer informing about the new IP addresses. The peer will send a new SPI value back to the initiator to make a return routability check. If the peer receives data from the initiator on the new security association with this new SPI, it confirms the mobile node has moved and is indeed reachable at the new IP address. For middleboxes that use <destination IP address, SPI and ESP> as the flow identifier to forward HIP packets, this information needs to be updated with every UPDATE message. FW-I (assuming that I is mobile) has to intercept the new IP address of I while FW-R (behind which is the peer R) has to update the new SPI(R) (using ESP_info(R) parameter(to forward packets correctly. This is illustrated in Figure 12.



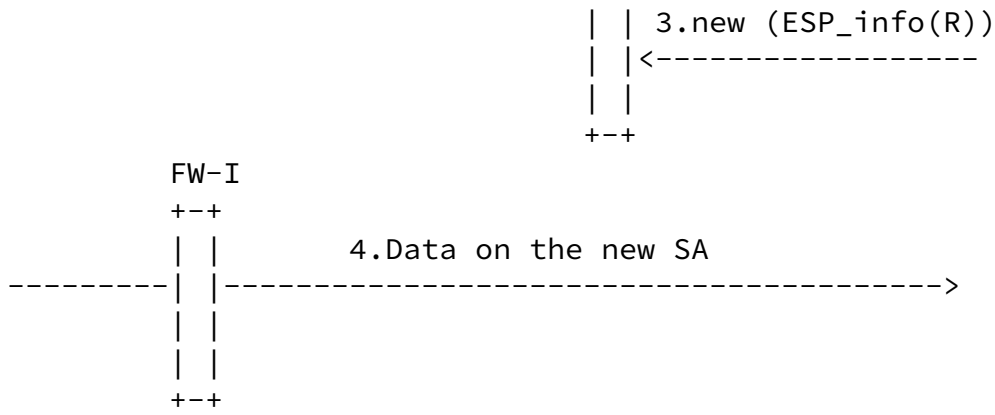


Figure 12: Authentication and authorization for UPDATE messages

As seen, FW-R has the flow identifier information for receiver R and FW-I has the flow identifier information for initiator I. When I sends a UPDATE message with a REA parameter, R sends a new ESP_info(R) parameter, which contains SPI(R), to check the reachability of the new IP address. FW-I can intercept the destination IP address from this message and can update its information. After both the UPDATE and UPDATE reply messages have been sent out, the receiver needs to signal the FW-R about it new SPI(R). Denial of service attacks and replay attacks are considerably reduced at firewalls if the firewalls keep track of the UPDATE ID that is sent in the UPDATE messages. Every UPDATE REPLY message carries the same number as the UPDATE message and hence the middleboxes are able to keep up the sequence. Issues as to how the receiver can inform the FW-R about its new SPI(R) even before it has received a confirmation on the return routability test have to be considered.

However, even this is true only if the new access point has the same set of middleboxes. If the mobile node is behind a new firewall while sending an UPDATE message, the firewall does not have any state information to create a pin hole. Hence, it should send a trigger message that will reinitiate the extended E2M messages between the mobile node and the firewall as in Figure 11.

Intellectual Property Statement

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.

Disclaimer of Validity

Copyright Statement

Acknowledgment

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