

On Supporting Multicast and Delegation in Hi3

Murugaraj Shanmugam, Technical University Hamburg-Harburg, Germany,

Email: murugaraj.shanmugam@tuhh.de

Franz Muenz, Fachhochschule Landshut, Germany,

Email: franz.muenz@fh-landshut.de

Hannes Tschofenig, Siemens AG, Germany,

Email: hannes.tschofenig@siemens.com

and Andrei Gurtov, Helsinki Institute for Information Technology, Finland,

Email: gurtov@cs.helsinki.fi

Abstract—Recently, much effort was applied to enable secure multihoming and mobility for Internet hosts. The Host Identity Indirection Infrastructure (Hi3) is a proposal that combines benefits of Secure-i3 and the Host Identity Protocol (HIP). In this paper, we extend the Hi3 architecture to enable multicast traffic and describe the delegation mechanism in detail. A prototype implementation and preliminary measurement results are provided.

I. INTRODUCTION

The Host Identity Protocol (HIP) [1] provides end-to-end authentication, basic DoS protection, and optionally confidentiality protection for the data traffic [2]. However, HIP is not intended to support multicast, anycast or service composition [3]. HIP resists Denial of Service (DoS) attacks to some extent, but severe flooding can temporarily isolate a HIP host from the Internet. HIP does not support simultaneous host mobility without a rendezvous server [5]. If a HIP host wishes to be contacted by other hosts, a Domain Name System (DNS) entry is mandatory and the host should register the IP address of its rendezvous server in the DNS.

The Secure Internet Indirection Infrastructure (Secure-i3) [3] relies on overlay network infrastructure. Secure-i3 enables communication without revealing IP addresses of end hosts and, thus, provides some DoS protection. Secure-i3 is inherently robust to failures as it uses a Distributed Hash Table (DHT), such as Chord [6]. By default, Secure-i3 requires that data traffic flows through an overlay server, thereby increasing the amount of network traffic and the latency experienced by end hosts [4]. End-to-end security is not currently provided.

Host Identity Indirection Infrastructure (Hi3) [4] is a proposal to integrate benefits and address shortcomings of HIP and Secure-i3. The idea is to use Secure-i3 for initial rendezvous, but let the data traffic flow directly between the end hosts over IPsec. This approach provides better DoS protection than HIP and better end-to-end security than Secure-i3.

In this paper, we propose Hi3 multicast support and a delegation mechanism based on Simple Public Key Infrastructure (SPKI) [10] certificates. We assume that an Hi3 host is located behind a middlebox that acts as a firewall for monitoring the host's traffic. The middlebox combines the functionality of the control and data planes of Hi3. After a registration procedure [8], the middlebox forwards HIP control packets as a Secure-i3 server and IPsec data traffic as an SPI-NAT.

An Hi3 host can use the middlebox for following purposes.

- *Registration Procedure.* A middlebox can require authentication and authorization of the host prior to allowing signalling or data traffic to bypass. The registration protocol mechanisms should not introduce new DoS attack possibilities at the middlebox.
- *Trigger Insertion.* The host can insert its public and private triggers using the middlebox. The middlebox then provides a rendezvous service and simultaneous mobility support to the host. For a definition of triggers refer to Section 3 of [3].
- *Multicast.* The Hi3 end can receive multicast traffic using the middlebox.

- *Delegation.* To obtain additional DoS protection, the end host can delegate replying to packets to the middlebox. As an example, the middlebox can reply to R1 packets or join requests to a multicast group.

The rest of the paper is organized as follows. Section II provides background on the middlebox registration procedure. In Section III and IV, the multicast and delegation mechanisms in Hi3 are described. In Section V, we show some initial implementation and measurement results. Section VI concludes the paper.

II. EXTENDED HIP REGISTRATION PROCEDURE

This section shortly explains the extended registration procedure of HIP [9]. It is a four-way handshake protocol based on public key cryptography. The protocol runs between the end host and the middlebox (or between two middleboxes if they are cascaded) and helps the middlebox to authenticate and authorize the end host. When the host moves, it uses a 3-way handshake UPDATE procedure to update the current location. We reuse the HIP registration protocol for inserting triggers. After the end host is authenticated and authorized, the middlebox inserts a host's trigger into Secure-i3.

To support the public/private trigger concept, we extend the registration procedure. We assume that the middlebox is attached to the Chord ring of Secure-i3 to support trigger insertion. The middlebox extracts the Host Identity Tag (HIT) of the Responder from the header field and inserts public and private triggers on behalf of the host. Now, other hosts can establish a communication with the host behind the middlebox.

To support multicast, we introduce a new field in the I2 message for signalling Security Parameter Index (SPI) to the middlebox in the registration procedure. This field is inherited from the HIP protocol [11]. Since there should an identifier to demultiplex the multicast data, in our case, the SPI helps the hosts to resolve the incoming data packets to an appropriate multicast group.

III. Hi3 MULTICAST

To enable multicast for Hi3 hosts, we have considered a basic approach and a delegation approach based on SPKI certificates. We make following assumptions.

- 1) The middlebox is responsible for executing the multicast. It will manage joins, leaves, group resolution, and perform packet replication for multicast.
- 2) During the registration procedure, the keys for the IPsec ESP data traffic will be established.
- 3) Multicast groups are formed by means of out-of-band mechanisms creating a table with HIT to group id mappings. The resolution table is used for assigning registered Responders to multicast groups, as well as for selecting the target group for incoming multicast packets.
- 4) The Initiator knows the HIT of the multicast group that acts as a multicast id.
- 5) The registration lifetime of participating Responders in a multicast group is limited and requires a continuous refresh.

A. Basic Multicast Approach

Because of arbitrary number of clients and their random event of joining and leaving in the group, providing a real end-to-end security, in the context of multicast, will be a daunting task. There are proposals to use a multicast centralized group controller to distribute the keys for the participating peers, but it poses complexity issues and does not fit well with the HIP architecture. The HIP base exchange optionally supports confidentiality protection for the data traffic. But executing a HIP base exchange with each client and establishing Security Associations (SAs) will not be a feasible solution for the Initiator.

The IPsec protection for the multicast traffic should be working for Hi3 hosts as well. The purpose of using IPsec is not only to secure the traffic but also to demultiplex the incoming traffic. This is achieved by means of using an unique SPI value, which is acting as a multicast group identifier. The middlebox will build up a resolution table for SPI to multicast group mapping. By using our "trusted" middleboxes, we can provide a layered independent IPsec protection to the clients. We use cascaded IPsec ESP protection for the data traffic. In this context, the Responder is the receiver for the multicast traffic.

Fig. 1 explains the flow diagram of the basic approach. Here, the middlebox1 is the owner of the multicast group. The middleboxes are cascaded and form a hierarchical structure for better scalability. Responders 1 and 2 run the HIP registration procedure with a neighbour middlebox, in this case

middlebox2. Their I2 messages contain an SPI value for receiving the data traffic. After the Responders are authenticated and authorized, their triggers are inserted by the middlebox.

Next, the middlebox2 will add SPI values of the Responders to form a multicast group list. The middlebox also sets up a IPsec security association for each Responder. The same procedure is repeated between the middlebox1 and middlebox2.

The Initiator and the middlebox1 set up a IPsec security association with each other after a base exchange. Then, the Initiator sends a data packet to the middlebox1 together with the SPI value, which is resolved by the middlebox1 to the unique multicast group identifier.

Once the integrity¹ of the data packet is verified, the middlebox1 strips off the IPsec header, adds the new IPsec header and forwards the packet to the middlebox2 with the corresponding SPI value. The middlebox2 performs the same procedure and forwards the packet with the corresponding SPI to its registered Responders. The Responders verify and, since the SPI value is unique, de-multiplex the incoming multicast packet.

B. Delegation Approach

In the second approach to multicast, Responders form a multicast group by exchanging SPKI certificates. This exchange will be triggered by a certain Responder, which will be the owner of the multicast group. This Responder delegates the multicast functionality to a middlebox. In our case, the Responder1 is the owner of the multicast group and registers its trigger and SPI value to the middlebox using the registration procedure and invites other hosts to participate in multicast by issuing a SPKI certificate. The SPKI certificate authorizes other hosts to use the trigger id of the Responder1. Responder2 runs the registration procedure with the middlebox1 and provides the SPKI certificate with the I2 message together with its SPI. The middlebox verifies the digital signature, performs validation and inserts the trigger id provided by the Responder1 with the IP address and SPI of Responder2 to a multicast group.

When the Initiator sends the packet to the middlebox, the middlebox resolves the group id, strips off the IPsec header, adds the new IPsec header,

¹Note that the keys are already established during the registration procedure.

and replicates the packet to registered Responders. The advantage of this approach is that the receiver can control the multicast group effectively. Unfortunately for this approach, certificate revocation for a large group poses higher implementation and protocol level complexity.

When a Responder wants to leave the multicast group, there are two ways: (a) an explicit approach by sending an UPDATE message to the middlebox to delete the Responder from the multicast group, and (b) an implicit approach where the Responder is deleted from the multicast group when the lifetime expires.

IV. HI3 DELEGATION

By executing the registration procedure to a middlebox, the Responder can delegate replying to R1 packets to the infrastructure. Fig. 2 shows the message flow in the delegation mechanism. The Responder executes the registration procedure with the middlebox2 to insert the private trigger (Priv_trig, IP). Next, the Responder executes the registration procedure with the middlebox1 to insert the public trigger pointing to a private trigger, (pub_trig, priv_trig). The Responder also issues an SPKI certificate to the middlebox1 by an out-of-band mechanism.

Base exchange packets are routed only through the overlay infrastructure, via a public trigger of the Responder. When the Initiator starts the base exchange with the Responder, it sends an I1 packet to the infrastructure. The middlebox1 storing the public trigger of the Responder, replies with an R1 packet and the SPKI certificate to prove that the packet is delegated from the Responder.

The Initiator verifies the Responder's certificate. Once the packet is processed correctly, the Initiator replies with an I2 packet to the middlebox1. The middlebox1 checks the trigger mapping and forwards the packet to the corresponding private trigger on middlebox2. The middlebox storing the private trigger verifies the solution and forwards the packet to the Responder. After the Responder receives the I2 packet, it sends the R2 packet that completes the base exchange.

To perform IP flooding, the attacker should know the Responder's IP address or the IP address of the Secure-i3 server storing the private trigger. With trigger chains, the Initiator never sends packets to

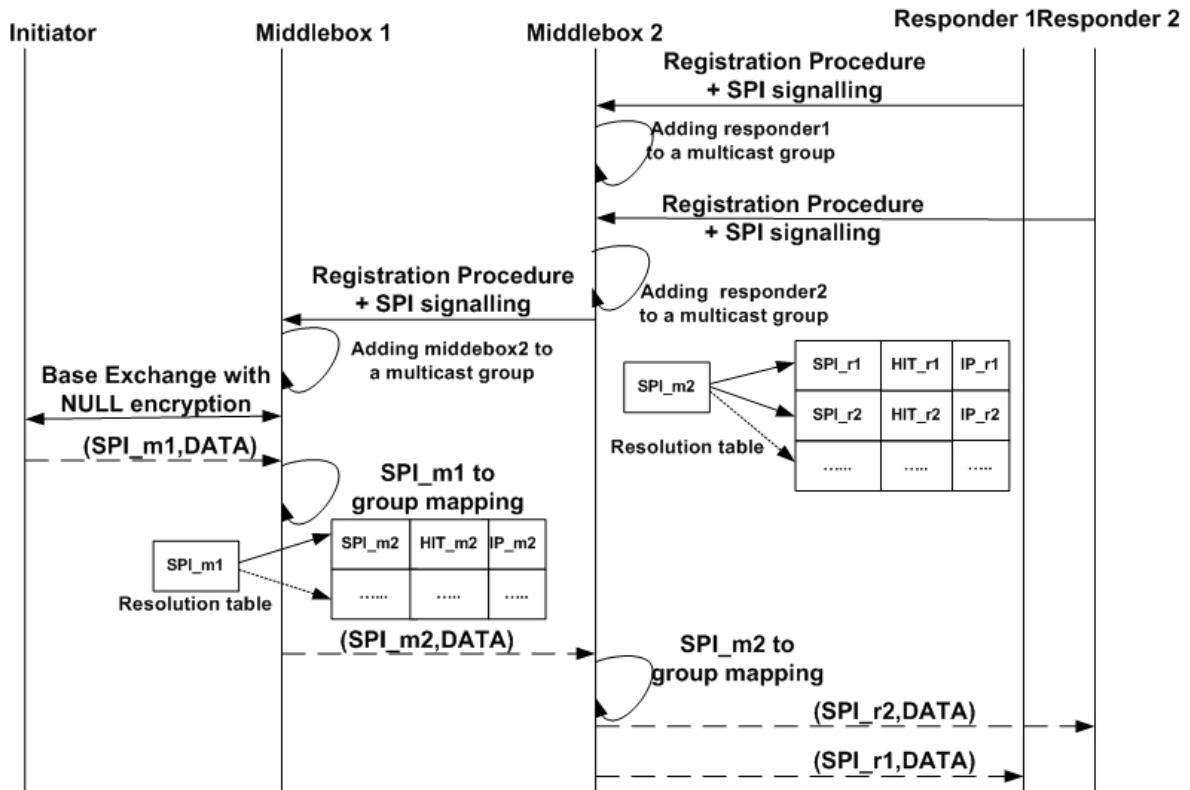


Fig. 1. The Hi3 multicast protocol.

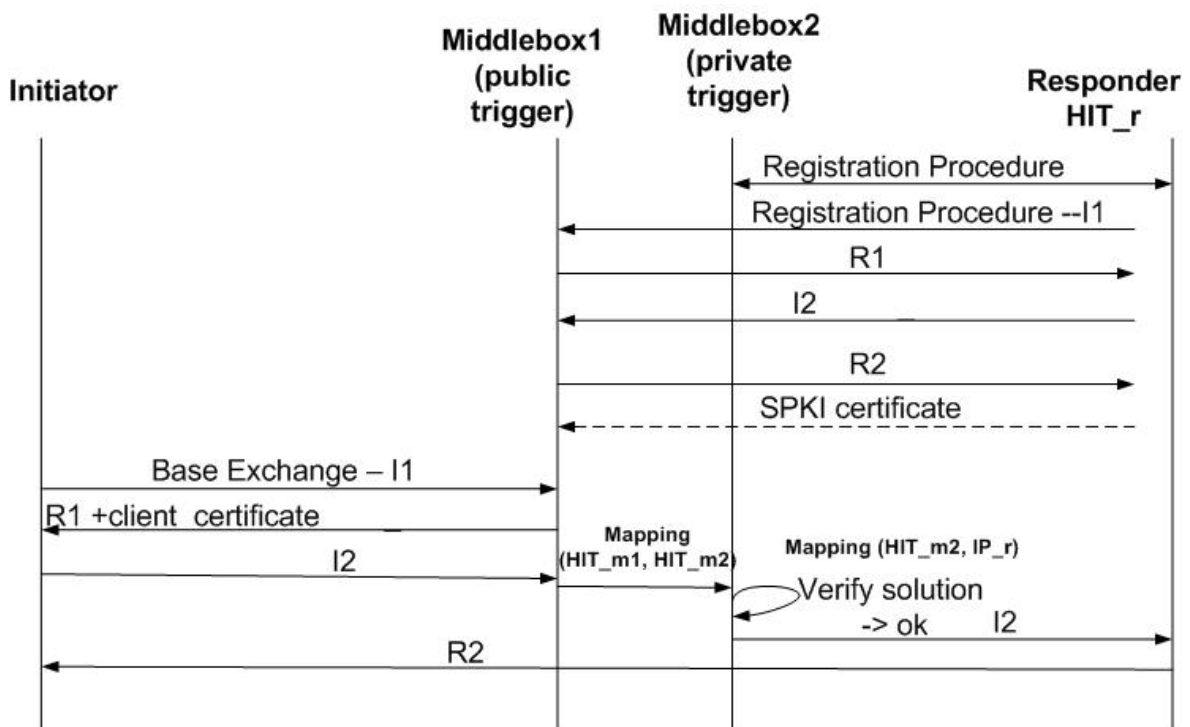


Fig. 2. The Hi3 delegation protocol.

the private trigger directly. Therefore, an attacker cannot obtain the IP address of the Secure-i3 server storing the private trigger or the IP address of the Responder. Flooding via a public trigger is mitigated with the HIP puzzle using the proposed delegation mechanism from a Responder to the infrastructure.

V. IMPLEMENTATION AND PERFORMANCE RESULTS

We have implemented a prototype of the registration procedure and the basic multicast approach². For simplicity, the current implementation assumes that SPKI certificates are exchanged by an out-of-band mechanism. The current prototype does not support delegation yet; we will implement it in the near future.

To receive multicast traffic, the Responder executes the registration procedure. The middlebox inserts the trigger into the trigger list, which is currently a single linked list. In the I2 message, a control flag is used to request a join to a multicast group. The middlebox adds the Responder's trigger to the multicast group list. The HIP daemon handling the registration and the Hi3 middlebox implementation run concurrently as two different threads. The middlebox replicates the packet for multicast.

We measured a trigger insertion delay of 0.44 ms, including inserting into the Chord ring and storing the trigger in a local multicast list. Multicasting a single message to 1000 receivers took 124 ms in realtime (CPU time 30 ms) and to 10000 receivers took 779 ms in realtime (CPU time 370 ms). The maximum throughput between the middlebox and Responder is 8.73 MB/sec in a 10 MB/sec, measured with `ttcp` tool.

In a next test, the Initiator transmitted a UDP packet (I1 message) 100 times to the middlebox that replicated it 1000 times to a single Responder over UDP. Out of the maximum number of 100000 packets, 83144 with a frame size of 137 bytes (95 bytes actual payload) were captured by `Ethereal`. The transmission time was 5.7 seconds from the first to the last packet sent from the middlebox. Throughput was 0.64 MB/s.

²We used Linux PCs as Initiator, Responder (PII 266 MHz) and Middlebox (PIII 450 MHz)

VI. CONCLUSIONS

Hi3 is a promising solution for providing secure mobility and multihoming for Internet applications. In this paper, we extended the Hi3 architecture to support multicast and delegation with the help of middleboxes. In future, we plan to implement the delegation mechanism for HIP association establishment and multicast based on delegation.

REFERENCES

- [1] R. Moskowitz, P. Nikander, P. Jokela and T. Henderson, Host Identity Protocol, draft-ietf-hip-base-02 (work in progress), February 2005.
- [2] R. Moskowitz and P. Nikander, Host Identity Protocol Architecture, draft-ietf-hip-arch-02 (work in progress), January 2004.
- [3] D. Adkins, K. Lakshminarayanan, A. Perrig and I. Stoica, Towards a More Functional and Secure Network Infrastructure, UCB Technical Report No. UCB/CSD-03-1242, 2003.
- [4] P. Nikander, J. Arkko and B. Ohlman, Host Identity Indirection Infrastructure, in Proc. of The Second Swedish National Computer Networking Workshop (SNCNW2004), November 2004.
- [5] J. Laganier and L. Eggert, Host Identity Protocol Rendezvous Extensions, draft-ietf-hip-rvs-01 (work in progress), February 2005.
- [6] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, in Proc. of ACM SIGCOMM'01, August 2001.
- [7] I. Stoica, D. Adkins, S. Zhaung, S. Shenker, and S. Surana, Internet Indirection Infrastructure, in Proc. of ACM SIGCOMM'02, August 2002.
- [8] H. Tschofenig, A. Nagarajan, V. Torvinnen, J. Ylitalo and M. Shanmugam, NAT and firewall Traversal for HIP, draft-tschofenig-hiprg-hip-natfw-traversal-01.txt (work in progress), February 2005.
- [9] H. Tschofenig, A. Gurtov, J. Ylitalo, A. Nagarajan and M. Shanmugam, Traversing Middleboxes with the Host Identity Protocol, in Proc. of ACISP 2005, July 2005.
- [10] Ellison C., Frantz B., Lampson B., Rivest R., Thomas B. and Ylonen T., SPKI Certificate Theory, RFC 2693, September 1999.
- [11] Jokela P., Moskowitz R., Nikander P., Using ESP transport format with HIP, draft-ietf-hip-esp-00-pre130505 (work in progress), June 2005.