End-host Authentication and Authorization for Middleboxes based on a Cryptographic Namespace

Tobias Heer*, René Hummen*, Miika Komu†, Stefan Götz*, and Klaus Wehrle*
* RWTH Aachen University, Distributed Systems Group
{heer, hummen, goetz, wehrle}@cs.rwth-aachen.de
† Helsinki University of Technology / HIIT, miika@iki.fi

Abstract—**Today, middleboxes such as firewalls and network address translators have advanced beyond simple packet forwarding and address mapping. They also inspect and filter traffic, detect network intrusion, control access to network resources, and enforce different levels of quality of service. The cornerstones for these security-related network services are end-host authentication and authorization. Using a cryptographic namespace for end-hosts simplifies these tasks since it gives them an explicit and verifiable identity. The Host Identity Protocol (HIP) is a key-exchange protocol that introduces such a cryptographic namespace for secure end-to-end communication. Although HIP was designed with middleboxes in mind, these cannot securely use its namespace because the on-path identity verification is susceptible to replay attacks. Moreover, the binding between HIP as an authentication protocol and IPsec as payload transport is insufficient because on-path middleboxes cannot securely map payload packets to a HIP association. In this paper, we propose to prevent replay attacks by allowing packet-forwarding middleboxes to directly interact with end-hosts. Also we propose a method for strengthening the binding between the HIP authentication process and its payload channel with hash-chain-based authorization tokens for IPsec. Our solution allows on-path middleboxes to efficiently leverage cryptographic end-host identities and integrates cleanly into existing standards.**

I. Introduction

In recent years, two complementary developments have surfaced in the area of network security. On the one hand, end-systems implement an increasing number of security features because networks – especially in the wireless domain – have become inherently insecure. On the other hand, middleboxes (MBs) realize more and more security-related services within the network to prevent intrusion, Denial of Service (DoS) attacks, and misuse of resources. Thus, security protocols need to cater to the two conflicting goals of protecting end-systems from on-path entities and assisting MBs in authentication, authorization, and accounting (AAA).

On an end-to-end basis, authentication and the bootstrapping of security associations is typically managed by key-exchange protocols, such as the family of Sign-and-MAC (SIGMA) [1] protocols. The Host Identity Protocol (HIP) as specified in RFC 5201 [2] is a SIGMA-compliant key-exchange protocol that sets up IPsec Security Associations (SAs) to protect the integrity and confidentiality of application payload. HIP uses self-certifying public-key-based identities to address hosts, thereby creating a new cryptographic namespace. These *Host Identities* (HIs) are represented by RSA or DSA public keys (PK). A host then can prove that it is the owner of an HI and the corresponding private key by using PK signatures. HIP supports MBs in using the HI namespace by, for example, applying additional public-key signatures to its control packets, thus enabling MBs to verifying end-host identities. Moreover, the HIP control-channel is intentionally left unencrypted to allow MBs to inspect and process HIP-related information. However, these measures do not suffice for MBs to use the HI namespace because: i) The current public-key based authentication scheme is prone to replay attacks and allows full impersonation of the end host towards the MB. ii) The binding between the HIP control-channel and the IPsec-protected payload channel is insufficient to prevent attackers from injecting forged packets into the payload flow. In this paper, we develop a path-coupled signaling approach [3] for security protocols which alleviates these problems on the practical example of HIP and IPsec. In Section II, we first give an overview of HIP and its cryptographic namespace to clarify the benefits of using the HI namespace for both end-hosts and MBs. Second, we identify a replay attack against MBs and discuss its implications in Section III. Third, we propose an approach for MBs to eliminate this replay vulnerability in Section III-B. Finally, we discuss the binding between the HIP control and payload channel and present a mechanism to strengthen this binding in Sections III-D3 and IV.

Although this paper primarily addresses HIP, the proposed MB extensions also apply to other end-to-end key-exchange protocols that are based on public-key identities (e.g. the family of SIGMA protocols) if the protocol exposes the identities and signatures to enable PK verification by the MB.

II. Host Identity Namespace

HIP transparently slots in between the network and the transport layer. Hence, it provides the new namespace and its services to protocols of the transport layer and above. It achieves compatibility between the HI namespace and IPv6 addresses through 128-bit hashes, so-called Host Identity Tags (HITs), of the potentially long HIs (RFC 5338 [4] discusses IPv4 compatibility on a similar principle).

HIP's cryptographic namespace elegantly addresses a number of security and trust-related problems that have previously been tackled separately without further thought about interoperability: end host macro mobility, multihoming, NAT traversal, migration of processes and hosts, and numerous trust and authentication-related issues in today's Internet. MBs can also benefit from HIP for the majority of AAA-related tasks based on IP: a) Hosts can be *authenticated* without additional authentication protocols and authentication infrastructure. b) *Authorization* can be performed based on strong cryptographic HIs rather than on implicit and often unreliable information about the network topology (e.g., IP-address-based and ethernet-port-based filtering) and non-cryptographic protocol properties (e.g., protocol numbers and statistical analysis). c) Finally, *accounting*, e.g. the tracking of the use of certain resources, is greatly aided by strong cryptographic identities. Examples for the practical benefit of using the HI namespace on MBs are HI-based access control, identity-based QoS and flow prioritization, active state removal, and admission control.

III. End-host Authentication by HIP-aware MBs

HIP uses two distinct channels between a pair of communicating hosts: the *control channel* is a signaling channel for establishing and maintaining the HIP association, while the *payload channel* carries the data of the transport layer. HIP protects the latter through IPsec Encapsulated Security Payload (ESP) tunnels [5] to offer end-to-end authentication, encryption, and integrity protection for the transport layer. At the end hosts, the IPsec packets are mapped to the HIP association by their Security Parameter Index (SPI).

HIP end-hosts use the control channel for authentication, key-exchange, and negotiating HIP association-related parameters, such as end-point addresses (IP addresses), cryptographic algorithms, or keys. End-hosts sign control-channel packets with PK signatures to support the verification of association-related information by MBs on the path[1]. Although MBs can use these PK signatures to map HIP packets to the identities of two communicating end-hosts, a pair of collaborating attackers can replay HIP control-channel packets and impersonate legitimate end-hosts towards the MB as described below.

A. End-host Impersonation Attack

In preparation of the attack, an unprivileged attacker observes and records a HIP handshake between two privileged legitimate hosts (the *victims*) and shares this information with its peer-attacker. Such data can easily be acquired by eavesdropping on an unencrypted wireless link or by installing monitoring devices in the network. At any later point in time, the colluding attackers can impersonate the victims and gain their privileges by replaying the recorded packet exchange through any MB. Thus, MBs cannot safely rely on the HI namespace to authenticate hosts for access control, QoS, or accounting since they cannot detect that the handshake occurs between hosts that are not the legitimate owners of the corresponding HIs. The attack is severe because there are no temporal or spatial restrictions to it. Using the attack, *any pair of attackers* can attack *any MB* by using *any recorded HIP handshake* at *any time* from *an arbitrary network location*.

The root of the problem is the absence of any discriminative information (e.g., time stamps or network-level identifiers) within the PK signatures of the HIP control packets. However, the lack of such information was intentional in HIP because both options require unrealistic assumptions. First, the secure use of time stamps requires global time synchronization, which leads to the problem of dealing with un-synchronized end-systems or MBs. Second, tying HIP associations to network-level identifiers by means of end-host-generated signatures leads to severe incompatibilities with network address translators (NATs) that modify network addresses in the packet headers. For end-hosts, HIP counters replay attacks by utilizing nonces and a shared secret derived from a Diffie-Hellman (DH) key exchange. However, these anti-replay measures are not applicable to MBs, leaving these devices with no measures to verify the genuineness and freshness of a HIP handshake.

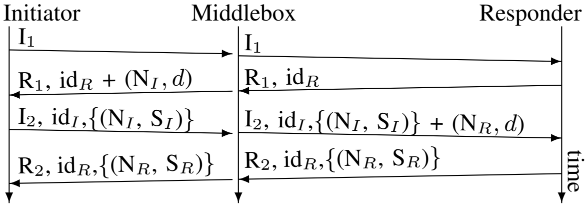
Initiator
Middlebox
Responder
I_1
I_1
R_1, id_R + (N_I, d)
R_1, id_R
I_2, id_I,$\{(N_I, S_I)\}$
I_2, id_I,$\{(N_I, S_I)\}$ + (N_R, d)
R_2, id_R,$\{(N_R, S_R)\}$
R_2, id_R,$\{(N_R, S_R)\}$
time

Fig. 1.Modified HIP handshake with MB. Only relevant parameters are shown.

B. Multilateral Authentication in HIP

To enable secure end-host authentication by MBs and to prevent the replay attack, we let MBs participate in the HIP handshake, mobility, and multihoming signaling. There are two viable options for including MBs in the on-path protocol. As the first option, an MB can act as a first-class object that establishes a full HIP association with both end-hosts. However, the increased computational complexity for computing the DH key-exchange with every MB on the path and the additional packet-space requirements for transferring the additional DH public keys rule out this option. As the second option, MBs can implement a challenge-response mechanism by injecting their own nonces into the end-to-end HIP control packets they forward. End-systems are then expected to sign these MB nonces with their private keys. As MBs select the contents of the nonces in the packets, they can verify whether the legitimate hosts are involved in the packet exchange and detect replays. In the remainder of this section we discuss this multilateral authentication mechanism for HIP in detail. Figure 1 illustrates an extended HIP handshake. For the sake of conciseness, we discuss only the packets, parameters, and operations that are relevant for MBs. Text in curly braces represents packet contents that are signed with the HI of the original sender of a packet. The four handshake packets are denoted as I_1, R_1, I_2, and R_2. The letters I and R indicate the origin of the message: The *Initiator* of the handshake or its *Responder*. The I_1 packet does not contain any public-key or DH related parameters. This is a precaution that protects the Responder from resource exhaustion attacks (cf. [2] Section 5.3.1) that would be possible if the I_1 packet invoked costly operations. To preserve the DoS resilience property, the MB forwards the I_1 packet without modification.

The R_1 packet contains the Responder's HI as well as its HIT. In order to challenge the Initiator, the MB adds a

[1] Note that the use of encrypted HIs is not possible when HIP is used for HI-based authentication to MBs.

nonce N_I to the R_1 packet. The Initiator must return N_I in a signed envelope. Additionally, N_I serves as seed value for a cryptographic puzzle of difficulty d that serves as DoS protection for the MB (cf. Section III-D1)[2]. Upon receiving the R_1, the Initiator solves the puzzle and returns the solution S_I as well as the signed N_I in the I_2 packet. As the HIP standard requires the Initiator to sign part of the I_2 packet to prove its identity to the Responder, the Initiator can simply append N_I to the signed part of the I_2 packet without additional signature overhead for a second signature. On receipt of the I_2 packet, the MB validates the puzzle before checking the nonce and verifying the PK signature to authenticate the end-host. Identity verification of the Responder is achieved in the same way by using the nonce and puzzle seed N_R[3]. Depending on the desired security function of the MB, it can authenticate the Initiator, the Responder, or both by injecting the nonces into the R_1 or I_2 packet respectively.

The extension provides two security benefits to MBs:

i) Host authentication: MBs can verify whether a certain host is involved in the establishment of a HIP association and thus, in the establishment of the payload channel. This property can be used to restrict access to network resources to authorized hosts. Moreover, traffic handling policies (e.g. QoS, or service restrictions) can be enforced on a per-association level.

ii) Attribute binding: Hosts can bind certain properties (e.g., payload channel attributes such as source and destination IP addresses or IPsec SPIs) to their HI and the HIP association. These attributes can either be stated explicitly in the HIP control packets or can be derived from the IP or UDP packets carrying HIP control messages. MBs can verify these bindings and use the specified attributes to enforce certain restrictions on the payload channel, e.g., to exhibit the same attributes as the control channel.

Since HIP supports end-host mobility and multihoming, HIP hosts may also have to prove their identity to MBs after a change of their network attachment. Due to space restrictions, we cannot elaborate on the authentication process for these cases in this document. A more detailed discussion can be found in [6] currently under discussion at the IETF.

C. Service Discovery and MB Identification

When authenticating towards an MB, the end-host implicitly subscribes to its services. For security or policy reasons, hosts may decide to not use the services of a particular MB. Since service discovery, service signaling, and service negotiations are out of scope for this work, we only outline a possible solution and refer to the IETF companion documents [6] and [7] for further information. In order to enable an end-host to determine the purpose and identity of an MB, the MB can add a service identifier (a so-called *service offer*) to the HIP packets carrying the N_I parameters. The service identifier states the nature, properties, and requirements of the service. By replying the hashed service offer in the signed part of the next HIP packet, an end-host clearly indicates that it is aware of the service and its function and that it accepts its terms of usage.

D. DoS Protection for Middleboxes

Although the performance effects of our extension are marginal for end-hosts, MBs must perform new security-related tasks. Especially HI verification through PK-signatures and maintenance of HIP-related state information can be exploited to create DoS attacks. Hence, we dedicate the next sections to the mitigation of such attacks.

1) Defense against CPU Exhaustion Attacks: Floods of forged I_2 and R_2 messages can easily exceed the computational capabilities of an MB, because for each packet the MB must verify the PK-signature belonging to the HI of the sender. Therefore, defenses against malicious flooding are essential for keeping an MB functional in the face of an attack. To frustrate CPU-targeted attacks, we use the well-known technique of client puzzles [8]. They are computational puzzles that are difficult to solve, but, for which the solution can be verified in a computationally inexpensive way. When an MB suspects such an attack, it increases the puzzle difficulty d, forcing the end-hosts to solve a more complex puzzle. The receiver of the nonce N must solve the puzzle with N as input value and d as puzzle difficulty before the MB performs PK verification for the handshake packets. It sends the solution S to the MB in the subsequent HIP handshake packet. Note that because of the dual functionality of N as the nonce and puzzle input value, it must exhibit sufficient randomness to prevent malicious re-use of the solution. As a puzzle increases the delay and computational cost of establishing a HIP association for the end-hosts, an MB should only demand solutions (set $d > 0$) when it is under attack.

2) Defense against Memory Exhaustion Attacks: Memory exhaustion attacks are a second threat for many MBs like routers and firewalls. Especially unauthenticated establishment of state (e.g. for half-open connections) can be exploited with flooding attacks to exceed the memory capabilities of MBs. Tracking the full handshake beginning with the I_1 and R_1 messages would require the MB to establish state based on unauthenticated packets. At this point neither the identity of the hosts nor the return-routability of the IP addresses are ensured, which opens a large window for attacks. Hence, MBs should delay the state establishment, especially the tracking of a connection, until one or both end-hosts prove their identity.

Receiving an I_2 packet with the signed nonce and puzzle solution testifies a) the return-routability of the Initiator's IP address, b) the ability of the Initiator to spend CPU cycles for the puzzle solution, and c) the identity of the peer before the MB establishes state. However, the nonce mechanism allows the MB to even stay stateless at this point by transferring the MB's state information as nonce to the Responder in the R_2 packet. As the required state information is small for most

[2] The duality of N_I as challenge and puzzle is a precaution to keep the amount of MB data in each packet low to avoid segmentation and MTU issues with multiple MBs on the path.

[3] Responders become vulnerable to DoS attacks when solving difficult MB puzzles unconditionally. Hence, they should prioritize BEX packets with low puzzle difficulty.

cases (authentication state and time), it can be enclosed in a small encrypted envelope for which only the MB possesses the key. Using this envelope as nonce N_R allows the MB to send its state information to the Responder and to receive it in the R_2 packet. Hence, state variables don't require buffer space on the MB. Late state establishment requires hosts to use valid network-layer addresses and HIs, and thus, significantly strengthens HIP MBs against memory exhaustion attacks.

3) Channel Binding for Middleboxes: Although the binding between the IPsec payload channel and the HIP control channel is cryptographically strong from the perspective of the end-hosts, an MB has more limited options for mapping IPsec ESP traffic a to the corresponding HIP associations. As MBs do not have access to the shared keys that are used within an end-to-end IPsec SA, the MB can only use non-cryptographic packet properties to map IPsec packets to a HIP association. Using such properties for mapping only enables decisions at *association-level* granularity. However, the access control can be enhanced to *packet-level* granularity where a single IPsec packet can be mapped to a HIP association in a secure way. Since the achievable level of the security services differ greatly for these two granularities, we discuss them separately in the remainder of this section.

4) Association-level Mapping: Although the MB can observe and verify the establishment and modification of a HIP and IPsec association, the MB cannot securely map a single payload packet to these. The MB cannot verify the packet source, destination, and integrity of payload packets in a cryptographic sense due to lack of middlebox-friendly authentication information. Therefore, decisions for a payload flow can only be taken at the association level. The MB can extract cryptographically-protected association-relevant information (e.g., the HIs and the SPI number assignment) from the control channel. Optionally, other information (e.g., IPsec sequence numbers) could be communicated within the signed HIP control packets to create a strong attribute binding. Moreover, an implicit binding between the IP address and the HI of a host can be derived from the use of the IP address in packets carrying the HIP control packets.

To show the practical use of *association-level* mappings, we use the example of a HIP-aware firewall that blocks traffic from unauthorized or unauthenticated hosts. As our proposed MB authentication extension for HIP prevents an unauthorized attacker from opening new HIP control and payload connections, our main concern is to prevent adversaries from injecting packets into already established payload channels of legitimate hosts. For example, a HIP-aware firewall can use the given attribute bindings (e.g. HIs or source- and destination addresses) to map an IPsec ESP packet to an HIP association and verdict the packet based on this. Therefore, an attacker must fulfill the requirements of the attribute bindings when it abuses an existing HIP association to send ESP payload traffic. An attacker can only inject ESP packets from the address matching the HIP association which limits the opportunity of a successful attack. Likewise, only hosts that can receive packets addressed to a legitimate host can receive the injected packets. These restrictions make exploitable injection attacks considerably more difficult.

The location of the attacker makes a difference to its attack opportunities. On-path attackers can read, modify, drop, and forge packets, whereas an attacker besides the path (e.g., at the local network of the victim) can only read and send forged packets. Although parameter binding cannot completely prevent packet injection, it enables MBs to detect this attack from off-the-path attackers if the ESP sequence numbers are bound to the HIP association. The MB can easily detect injected ESP packets by observing duplicate ESP sequence numbers. However, the MB can only detect the presence of duplicates but cannot filter forged packets since the it cannot determine which duplicate is authentic.

5) Packet-level Mapping: Additional security measures, such as per-packet public-key signatures as used by Packet-Level Authentication (PLA) [9] or pair-wise link-layer security measures such as IEEE 802.1x, can enable an MB to distinguish payload traffic from different hosts. Strong bindings between the HIP control channel and the payload channel can be achieved by performing attribute binding for the public keys and credentials of the additional payload security protocols. In PLA, every payload packet is signed with elliptic curve cryptography that MBs can verify. The MB can, therefore, map each payload packet to the corresponding HIP association, and thus, detect injected and modified packets. Yet, signature verification in PLA is CPU-intensive and requires specialized hardware to achieve reasonable throughput.

Lower-layer security protocols can secure parts of the communication path, thus preventing packet injection, modification, and forgery on the protected path segments. However, point-to-point link-layer authentication requires additional security protocols and can only protect the path partially.

IV. Packet-level Authorization

We briefly present packet-level authorization as another step towards a stronger binding between the control and payload channel. Packet-level authorization prevents unauthorized senders from injecting packets into a payload stream but does not provide integrity protection for the packets. It maps payload packets to control channels efficiently and performs per-packet processing based on this mapping.

A. Authorization Tokens

To provide per-packet authorization, we attach a cryptographic authentication token to each IPsec ESP packet. This authentication token certifies that the legitimate sender generated the packet. Authentication tokens were first used for reducing the cost of link-state routing by Hauser et al. [10]. We use the well-know technique of hash chains first proposed by Lamport [11] for generating the tokens. For each outbound IPsec SA, each end-host iteratively generates a sequence of hashes using a cryptographic one-way hash function H. The first element of the hash chain h_0 is chosen randomly whereas all other elements h_i are computed by hashing the previous element: $H(h_{i-1}) = h_i$. The last element h_n of the hash chain

TABLE I
HI VERIFICATION TIME: TWO LOW-COST AND ONE PC MB. (IN MS)

	RSA HI			DSA HI		
	AR2315	BR5365	PC	AR2315	BR5365	PC
768 bits	3.6	3.7	0.2	38.9	43.8	2.1
1024 bits	5.4	5.9	0.3	58.6	69.8	2.8
1536 bits	10.4	10.2	0.6	120.0	122.2	6.0

TABLE II
MB THROUGHPUT (IN MBIT/S) AND STANDARD DEVIATION.

	No authorization		Packet-level-authorization	
	AR2315	PC	AR2315	PC
TCP	6.6 (0.03)	86.1 (0.07)	5.9 (0.02)	84.4 (0.01)
UDP	9.8 (0.03)	91.8 (0.05)	8.8 (0.04)	90.6 (0.04)

is called anchor. Each peer attaches this anchor to the HIP handshake and uses attribute binding to express that the anchor is related to the HIP association. When sending packets, the end-host discloses the elements of the chain in reverse order of their creation, attaching a fresh element to each packet.

During the HIP handshake, MBs on the communication path read the hash-chain anchors from the signed HIP packets. For consecutive IPsec packets, the MB checks whether the packet contains a *fresh* element that is a part of the corresponding hash chain. The MB checks this by hashing the most recently disclosed element h_j and verifying that $H(h_j)^n = h_i$ for $1 \leq n \leq w$. The exponent n signifies n repeated iterative applications of the hash function, h_i is a previously verified hash chain element and w is the verification window size. This window limits the number of hash computations that an MB will perform to find the successor element in cases of missing intermediate elements (e.g., elements used in lost packets). Limiting w is a precaution against DoS attacks targeting the per-packet hash computations on the MB. An element h_j is considered fresh when it has not been used in a previous IPsec packet and no successor h_k with $k < j$ has been used before. A fresh hash-chain element indicates that the packet a) was generated by the legitimate sender and b) is not a replay. Based on these indications, an MB can decide how to process a packet (e.g. drop or forward with higher or lower priority).

As effect of using this IPsec extension, attackers cannot generate valid packets by themselves because each packet must be accompanied by a hash chain element that only the sender knows prior to its disclosure. Thus, an attacker can only modify existing, but not create new valid packets. Hence, the maximum damage regarding to wasted or misused bandwidth is bounded by the sending rate of legitimate packets. Moreover, authorization tokens significantly complicate attacks for *off-the-path attackers* because they need to learn the current hash chain element before they can attempt to send a forged packet. Specifically, they need to be able to receive the authentic packet and deliver the forged packet to the MB before the authentic packet arrives. Otherwise, the forged packet is not considered fresh. In effect, this prevents attacks in which the attacker uses the same medium as the potential victim (e.g. the same local wired or wireless subnetwork). Note that, this extension cannot prevent misuse from attackers that are located on the communication path because these can still alter the contents of the legitimate packets.

Hash-chains – by nature of their design – have a finite length, which requires to replace a chain with a fresh one before it depletes. We use the authenticated HIP control channel for securely signaling a new anchor to the MB. Long hash chains are preferable because the bandwidth of the payload channel can be high. This might deplete short chains quickly and require new anchors to be exchanged frequently. Due to the notably low CPU demand of hash functions, even low-scale mobile devices can instantly generate long chains (e.g., a Nokia N810 Internet Tablet, 400 MHz ARM CPU can generate a chain of 10.000 elements within 30 ms).

Burst drops of packets cause gaps in the sequence of authentication tokens that may exceed the size of the verification window, causing MBs to drop the packets. In such cases, the hosts and the MBs need to re-synchronize by exchanging new hash chain anchors with a HIP update. To further mitigate the negative effect of burst drops, senders can use several hash chains in parallel. This linearly reduces the computational costs for the verification of token-gaps at a linear increase of the storage space requirements at the sender and at the MBs.

To maintain backward compatibility to network infrastructure elements that inspect and process the IPsec headers [12], [13], we abstain from modifying the basic IP and IPsec header structures. Therefore, we append the token to the IPsec payload field. The receiver removes the token prior to further IPsec processing.

B. Performance

In this section, we briefly present basic performance results of the proposed extensions to show the feasibility of our approach. We first focus on the HIP MB authentication extension, which we implemented for the HIP for Linux (HIPL) implementation. The key factor in efficiently verifying HIs is the processing time of the PK-related verification procedures at the MBs. We evaluate the performance for MBs on two specialized commodity routers, The "La Fonera" wireless router with a 180 MHz Atheros AR2315 32-bit MIPS CPU and the Netgear WGT634U with a 200 MHz Broadcom 5365 MIPS-32 based CPU. We also evaluate with an AMD PC (AMD Athlon CPU at 1.3 GHz) that represents the class of dedicated multi-purpose MB network components (e.g. a firewall in a small company). Table I shows that consumer-grade routers can verify 169 RSA and 14 DSA HIs per second with common key lengths of 1024 bits. Taking into account that the expected number of clients opening a connection simultaneously is expected to be quite low in consumer scenarios, the performance of the commodity router hardware is sufficient for supporting our extension. For 1024-bit keys, the PC MB can perform about 3300 RSA and 350 DSA verifications per second which suffices even large scenarios.

The key factors for evaluating the performance impact of packet-level authorization are latency and throughput. We implemented our IPsec extension for the end-hosts and the MBs based on the HIPL user-space firewall. Note that the user-space IPsec processing suffers from additional context switches and a higher per-packet processing cost. We evaluated the measurements with the AR2315 consumer wireless router and the PC firewall that processed the authentication token in each forwarded packet. The end-hosts that were used for stress-testing the MBs are equipped with 3-GHz CPUs to avoid the end-systems becoming the performance bottleneck. Table II shows that the impact of the packet authorization is notably low. UDP and TCP throughput differs only 1% from the throughput without authorization tokens for the PC MB. For the tightly resource-constrained consumer-grade router, throughput decreases by 10% for UDP and 11% for TCP, leaving the low-cost router with sufficient resources for most ADSL-line speeds even when using the user-space firewall. The latency in our local network only increased by 0.02 ms from 0.62 ms to 0.65 ms for the PC MB and by 0.2 ms from 2.4 ms to 2.6 ms for the consumer router.

V. Related Work

Martin et al. [3] proposed to use path-coupled signaling for explicit configuration of forwarding entities, such as firewalls and NATs. Their approach uses the notion of probable trust based on non-cryptographic network- and transport-layer identifiers (e.g. address ranges, port numbers, etc.). This approach has the advantage that it does not require public-key authentication or persistent security associations between neighboring network entities. However, applicability of the approach is limited to cases that do not require provable identities. The Resource reSerVation Protocol (RSVP) [14] is a protocol for QoS reservation on on-path MBs. To avoid resource misuse, it employs a hop-by-hop authentication and integrity protection scheme based on certificates and pair-wise keys between adjacent routers. However, the initial trust bootstrapping, key-management, and router-coordination introduce a considerable management overhead. A common approach for MBs to authenticate end-hosts is to use an authentication server. Popular protocols following this scheme are RADIUS [15] and DIAMETER [16]. These approaches solve the particular problem in an infrastructure-based scenario. However, this scenario requires separate authentication protocols and hardware. In general, it does not follow the concept of path-coupled signaling.

The HIP registration extension [17] defines how HIP hosts can register to a network service. This mechanism could also be used to register to an MB and to exchange information with it. However, it requires MB detection and explicit registration with an MB using a separate HIP handshake. Although this procedure makes the full services (i.e. AAA) available to the MB and allows the MB act as an end system by being the explicit endpoint of the HIP association, it introduces considerable cryptographic overhead and protocol complexity. Especially cascaded MBs require several detection and registration steps that slow down connection establishment and mobility signaling. In [18] we proposed the Peer-to-Peer Wi-Fi Internet Sharing Architecture (PISA). The approach discussed in this paper forms the basis for [18] which only briefly touches some of the aspects explained in detail in this work.

VI. Conclusion

In this work, we show how on-path middleboxes (MBs) can use cryptographic host identities as a basis for AAA-related services. With such a cryptographic namespace, MBs can rely on secure first-level host identities rather than on secondary host and protocol information that is implicit and not cryptographically secure. In particular, we address an impersonation attack targeting HIP-aware MBs and mitigate it by active participation of the MB in the HIP handshake. We introduce cryptographic authentication tokens for IPsec to strengthen the binding between the HIP control channel and the IPsec payload channel. The proposed extensions integrate well with the HIP and IPsec standards, have a notably low computational overhead, and provide counter-measures against memory- and CPU-exhausting DoS attacks. With the presented extensions, on-path MBs can use the benefits of the cryptographic HIP namespace and achieve a higher level of security.

References

[1] H. Krawczyk, "Sigma: The 'sign-and-mac' approach to authenticated diffie-hellman and its use in the ike-protocols." in *CRYPTO*, 2003.

[2] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol," RFC 5201 (Experimental), Apr. 2008.

[3] M. Martin, M. Brunner, M. Stiemerling, and A. Fessi, "Path-coupled signaling for NAT/firewall traversal," *High Performance Switching and Routing, 2005. HPSR. 2005 Workshop on*, 2005.

[4] T. Henderson, P. Nikander, and M. Komu, "Using the Host Identity Protocol with Legacy Applications," RFC 5338 (Experimental), Sep. 2008.

[5] P. Jokela, R. Moskowitz, and P. Nikander, "Using the Encapsulating Security Payload (ESP) Transport Format with the Host Identity Protocol (HIP)," RFC 5202 (Experimental), Apr. 2008.

[6] T. Heer, M. Komu, and K. Wehrle, "End-Host Authentication for HIP Middleboxes," Internet Engineering Task Force, Internet-Draft draft-heer-hip-midauth-02, Feb. 2009, work in progress.

[7] T. Heer, S. Varjonen, and H. Wirtz, "Service announcements and Service Classification for HIP," Internet Engineering Task Force, Internet-Draft draft-heer-hip-service-00, Feb. 2009, work in progress.

[8] T. Aura, P. Nikander, and J. Leiwo, "DoS-Resistant Authentication with Client Puzzles," in *Revised Papers from the 8th International Workshop on Security Protocols*. London, UK: Springer-Verlag, 2001.

[9] C. Candolin, J. Lundberg, and H. Kari, "Packet level authentication in military networks," *Proceedings of the 6th Australian Information Warfare & IT Security Conference*, 2005.

[10] R. Hauser, A. Przygienda, and G. Tsudik, "Reducing the Cost of Security in Link State Routing," *NDSS '97*, 1997.

[11] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, no. 11, 1981.

[12] L. Berger and T. O'Malley, "RSVP Extensions for IPSEC Data Flows," RFC 2207 (Proposed Standard), Sep. 1997.

[13] J. Ylitalo, P. Salmela, and H. Tschofenig, "SPINAT: Integrating IPsec into Overlay Routing," *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)-Volume 00*, 2005.

[14] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205 (Proposed Standard), Sep. 1997.

[15] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865 (Draft Standard), Jun. 2000.

[16] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol," RFC 3588 (Proposed Standard), Sep. 2003.

[17] J. Laganier, T. Koponen, and L. Eggert, "Host Identity Protocol (HIP) Registration Extension," RFC 5203 (Experimental), Apr. 2008.

[18] T. Heer, S. Götz, E. Weingärtner, and K. Wehrle, "Secure wi-fi sharing on global scales," in *Proc. of 15th International Conference on Telecommunication (ICT)*. St. Petersburg, Russian Federation: IEEE, 2008.