IEICE TRANS. INF. & SYST., VOL.E96–D, NO.2 FEBRUARY 2013

PAPER Special Section on The Internet Architectures, Protocols, and Applications for Diversified Futures

Design and Implementation of Security for HIMALIS Architecture of Future Networks

Ved P. KAFLE[†a)], **Ruidong LI**[†], **Daisuke INOUE**[†], ***and*** **Hiroaki HARAI**[†], ***Members***

SUMMARY For flexibility in supporting mobility and multihoming in edge networks and scalability of the backbone routing system, future Internet is expected to be based on the concept of ID/locator split. Heterogeneity Inclusion and Mobility Adaptation through Locator ID Separation (HIMALIS) has been designed as a generic future network architecture based on ID/locator split concept. It can natively support mobility, multihoming, scalable backbone routing and heterogeneous protocols in the network layer of the new generation network or future Internet. However, HIMALIS still lacks security functions to protect itself from various attacks during the procedures of storing, updating, and retrieving of ID/locator mappings, such as impersonation attacks. Therefore, in this paper, we address the issues of security functions design and implementation for the HIMALIS architecture. We present an integrated security scheme consisting of mapping registration and retrieval security, network access security, communication session security, and mobility security. Through the proposed scheme, the hostname to ID and locator mapping records can be securely stored and updated in two types of name registries, domain name registry and host name registry. Meanwhile, the mapping records retrieved securely from these registries are utilized for securing the network access process, communication sessions, and mobility management functions. The proposed scheme provides comprehensive protection of both control and data packets as well as the network infrastructure through an effective combination of asymmetric and symmetric cryptographic functions.

key words: *ID/locator split architecture, security, new generation network, future network*

1. Introduction

To overcome the problems caused by the dual roles of IP addresses as host IDs and locators, various approaches to introducing the ID/locator split concept into network architectures have been recently discussed [1]–[7]. ID/locator split architectures use two distinct sets of values for host IDs and locators, whose mappings are stored in some servers or registries. Locator/ID Separation Protocol (LISP) [1] is introducing ID/locator split into edge routers to reduce the BGP routing table growth rates and route update frequencies in the backbone Internet. Host Identity Protocol (HIP) [2] applies ID/locator split in the host protocol stack to make session establishment and mobility functions secured. Similarly, Shim6 [3] applies ID/locator split to enable hosts to support multihoming. While each of the above protocols tries to address a specific issue (viz. routing scalability, secured mobility, and multihoming) of the current Internet, HIMALIS (Heterogeneity Inclusion and Mobility Adaptation through Locator ID Separation) [4] proposes a generic architecture, based on the ID/locator split concept, that can natively support mobility, multihoming, scalable backbone routing and heterogeneous protocols in the network layer.

These proposals require retrieving ID/locator mapping records to find corresponding locators for the given names or IDs before forwarding packets into the network. They need to update the mapping records when hosts change their locators (due to mobility) or add new locators (due to multihoming or renumbering). The Domain Name System (DNS) is not suitable for the fast updates of this type of dynamic mapping records because of the existence of multiple copies of cached records in the global system of DNS servers [5]. It would be impossible to quickly update all the copies when the record changes. Therefore, besides DNS servers, additional servers are needed for storing the ID/locator mapping records of dynamic hosts. For this purpose, ALT [6] in LISP, rendezvous servers in HIP, and hostname registries in HIMALIS, for instance, have been proposed in the literature. These proposals, however, lack inbuilt security functions for collectively protecting the update and retrieval procedures. For example, although HIP uses certificates and public keys to establish secured sessions between hosts, it does not cover the security of ID/locator mapping servers or registries. DNS Security (DNSSEC) [8] provides integrity protection of records retrieved from a DNS server. However, the DNS structure itself is not favorable for frequently updating the records in the server although it is favorable for faster and efficient retrieval of static records. Recently, LISP-security [15] has been published as an Internet-draft. However, it describes only security issues related with securely retrieving of ID/locator mapping records. So, LISP still lacks mechanisms for securing an update of ID/locator mapping records and securing data packets when they are encapsulated with a LISP header and tunneled through the transit network. Similarly, HIP uses IPsec [16] for securing data packets in the network layer, but this approach would not be applicable in heterogeneous networks where the end-to-end network protocol is not the same, i.e., when the edge and transit network use different network layer protocols such as IPv4, IPv6, or future non-IP protocols.

To secure the HIMALIS architecture, not only in the data plane but also in the control plane, this paper proposes an integrated security scheme. The scheme exploits both asymmetric (i.e., using public/private keys) and symmetric (i.e., using shared keys) cryptographic functions to authen-

Manuscript received June 5, 2012.
Manuscript revised October 1, 2012.

[†]The authors are with National Institute of Information and Communications Technology (NICT), Koganei-shi, 184–8795 Japan.
a) E-mail: kafle@nict.go.jp

DOI: 10.1587/transinf.E96.D.226

ticate hosts and protect the integrity of signaling messages and data packets. Thus, the scheme protects the network from impersonation attacks and man-in-the-middle attacks. It securely stores and updates hostnames, IDs, locators, and public keys in the domain name registries (DNR) and host name registries (HNR). It specifies the security of the mapping record retrieval procedure and utilizes the mapping record to secure the network access process, communication sessions, and mobility management functions. For network access security, hosts are authenticated by the network by verifying if they possess the same IDs and public keys as stored in the HNR. After the authentication, an access key is created for authentication and integrity protection of subsequent messages exchanged by the host with the network. Similarly, two hosts use the mapping records to authenticate each other and create a session key to secure their sessions. The mobile host uses its access key and session keys to secure mobility signaling messages.

The contribution of this paper is as follows. This paper presents the integrated security scheme of the HIMALIS architecture for securing not only the ID/locator mapping storage, retrieval and update functions, but also the data sessions and mobility functions. Other ID/locator split-based related works do not have all these features together. They need add on functions, which are not yet fully stated. In contrast, our security scheme is fully integrated as a fundamental component of HIMALIS architecture to enable it to optimally support secure and mobile network services in the future heterogeneous and dynamic networking environment where hosts frequently change their locations or network access points. Since the integrated scheme shares the same security functions (such as encryption, key generation, and identity verification modules) and context (such as IDs, access keys, session keys, and public keys) to secure both the signaling and data plane functions, it is more efficient than conventional approaches where distinct security mechanisms are added randomly to secure different functions without considering their global optimization.

This paper is a modified and extended version of our previous papers [12]. As the new content, it includes the descriptions of attacker model and threat analysis as well as elaborated explanations of hostname registration security, mobility security, and the implementation layout. The other sections have also been heavily revised. The remainder of this paper is organized as follows. Section 2 briefly describes the attacker model and threat analysis. Section 3 presents HIMALIS architectural components involved in the proposed security scheme. Section 4 describes the ID/locator mapping registration and retrieval security. Section 5 describes the network access security. Section 6 presents communication session security and Sect. 7 describes the security for mobility. Section 8 outlines the implementation and Sect. 9 discusses the feasibility and scalability issues. Section 10 concludes the paper.

2. Attacker Model and Threat Analysis

The network entities may comply with the designed protocol or deviate from it. To make the HIMALIS architecture trustworthy, we first analyze an attacker model applicable to the security-related problems of the architecture. Similar to Dolev-Yao model [13], we assume that an attacker can initiate a conversation with any node, receive any message passing through the network, and read, modify, block, replay, or insert any message in the network. It is also assumed that the cryptographic algorithms are unbreakable.

This attacker model is thus based on the assumption that an attacker can initiate a mapping procedure and receive, modify, block, replay, and insert any mapping request, mapping reply, and mapping update packets in the network. Based on this attacker model, the possible attacks on the HIMALIS architectural components can be summarized as follows.

The attackers can perform an impersonation attack by sending false requests for ID/locator mapping registration or mapping update to the mapping registries and carry out denial of service (DoS) attacks to the registries. The attacker can falsely announce other entities' IDs and locators as his own IDs and locators for a bad purpose of hijacking others' sessions. The attacker can also impersonate the DNR and HNR registries and send wrong ID/locator mappings in the hostname resolution response message to the legitimate querying hosts. This can lead to various types of attacks, such as session hijacking, DoS, and phishing.

The attacker can stand in the middle between two hosts or between a host and a mapping registry to carry out man-in-the-middle attacks. It can send a modified or fake ID/locator mapping to one of the hosts in a hostname resolution message or in a communication initialization message. It can also send a modified ID/locator mapping in a hostname registration/update message to the registry. For example, the attacker can modify a hostname resolution response packet by replacing the correct ID/locator mapping with a wrong one and can divert subsequent data sessions to a target victim host. This can lead to a DoS attack by depleting resources of the victim host. Similarly, by a replay attack, the attacker can replay old or stall ID/locator mapping update packets originated from a mobile host and poison the ID/locator mapping record stored in the HNR.

3. HIMALIS Network Architecture

To inhibit the attacks described in the previous section, we propose an integrated security scheme. Figure 1 shows the HIMALIS architecture with security-related nodes. It consists of the edge networks, the global transit network, and the logical control network. As the name indicates, the global transit network that consists of high speed routers and links interconnects the edge networks. The security scheme presented in this paper is associated with the edge network and the logical control network nodes.

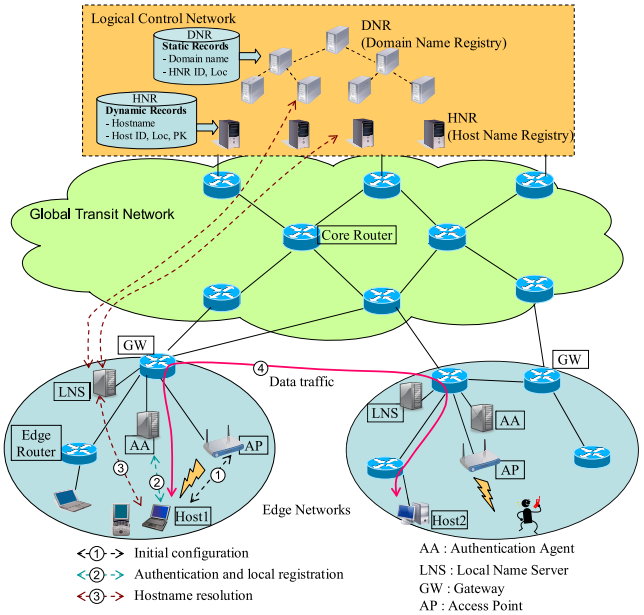

Fig. 1 HIMALIS network architectural components.

3.1 Edge Network Security Components

The edge network consists of the following entities: gateway (GW), local name server (LNS), authentication agent (AA), and hosts. The AA, LNS, and GW collectively enforce security for network access by hosts. They belong to the same trust domain and possess pairwise shared secret keys, which are used for securing messages exchanged between them. The AA located in the home edge network helps the local hosts, which are being connected for the first time to their home edge network, to register their records in the HNR. The AA located in the other edge network authenticates the visiting hosts. Namely, the AA receives an authentication request from the visiting host and verifies the host authenticity by comparing the host provided info with the host's record stored in the HNR. It generates access IDs and access keys and assigns them to hosts as well as announces them securely in the edge network. The AA may also contact an authentication server for verifying the host's credentials (not shown in figure). The LNS stores all authenticated local hosts' (i.e., hosts currently located in the LNS's edge network) hostnames, IDs, local locators (LLoc) and public keys (PK) in the Host Table. The Host Table is used to resolve a local host's hostname into the ID, LLoc, and PK when another local host located in the same edge network wishes to communicate with the former host. The LNS also provides the local hosts with remote hosts' hostnames to IDs, global locators (GLocs) and PKs mapping records by performing hostname resolution. The GW has an ID Table to store all authenticated local hosts' IDs and LLocs as well as the remote hosts' IDs and GLocs, with whom the local hosts are communicating. The GW uses the ID Table to perform layer 3 or L3 protocol (or locators) translation in packet headers when the edge network is using a different L3 protocol (or private locator space) than the transit network's L3 protocol (or global locator space).

The host possesses its hostname, ID and PK, and obtains its LLoc and GLoc from the edge network. The hostname has two parts, local host name and global domain name, which are connected together by "#" symbol. An example of hostname is *myhost#mydomain.com*, where *myhost* is the local part and *mydomain.com* is the global domain name. The hostnames are globally unique. The details of the hostname and ID configuration methods are given in [5]. When the hostname and ID are configured, the public/private key pair is also generated. When the host successfully authenticates itself with the edge network, it obtains its LLoc from the locator space used by the edge network's L3 protocol. The host also obtains its GLoc, which is actually the GW's GLoc assigned by the transit network from the global locator space used by the global L3 protocol. The host maintains a Security Association Table (SAT) containing its own and remote host's IDs, LLocs, GLocs, and PKs, as well as the session related security parameters such as shared keys, which are negotiated during a session establishment between the hosts.

In addition to the GW, LNS, and AA, the edge network also contains another functional entity that would provide the host with initial setup parameters such as the GW's, AA's and LNS's IDs and LLocs. Dynamic Host Configuration Protocol (DHCP) or router advertisements can be used for this purpose. DHCP has been chosen for our implementation.

3.2 Logical Control Network Security Components

The logical control network contains the hostname resolution system, consisting of the DNR and the HNR, to store and provide hostnames to IDs, GLocs and PKs mapping records. The DNR stores static mappings between domain names and HNRs' IDs, GLocs, PKs, etc., while the HNR stores dynamic mappings between hostnames and IDs, GLocs, and PKs of the hosts. Since HNRs are fixed server nodes whose locators do not change often, the DNR records are mostly static. The HNR records, on the other hand, are dynamic as the hosts' GLocs need to be updated frequently to support mobility. For each domain name there is at least one authoritative HNR, which stores the records of all hosts that share the same domain name in their hostnames. Since the DNS structure is favorable for the faster and scalable retrieval of static mapping records, DNRs are organized in a DNS-like hierarchical structure, where multiple cached copies of the records exist. On the other hand, since HNRs' records need to be updated frequently, HNRs have a flat structure, where only one copy of the records would exist. New security functions are proposed for securing the DNR and HNR records.

The proposed security scheme makes the HNRs trustable anchor points so that the HNR records (i.e., host IDs, GLocs, PKs stored in the HNR) can be used by hosts

Table 1 Notations.

Notations	Descriptions
$E_{PK(A)}(M)$	Message M encrypted by public key *PK* of node A using an asymmetric cryptographic function
$E_{SK(A-B)}(M)$	Message M encrypted by shared key *SK* of nodes A and B using a symmetric cryptographic function
$E_{regKey}(M)$	Message M encrypted by registration key *regKey* using a symmetric cryptographic function
$HMAC_K$	Hash-based message authentication code generated by using key *K*
SIG_A	Signature of node A by using its private key
$X/Y/Z_A$	X, Y, Z of node A; where X, Y, Z indicate hostname, ID, GLoc, PK, etc.
$M \parallel HMAC_K$	$HMAC_K$ appended to message M
$M \parallel SIG_A$	SIG_A appended to message M

and networks in securing the network access process, communication sessions, and mobility. To make the HNRs trustable, their PKs (along with IDs and GLocs) are stored in the DNR records, which are authenticated by using a chain of certificates from the root DNR to the DNR storing the records. By retrieving the DNR record, along with the certificate chain, and using the root DNR's PK, it can be easily verified that the HNR as well as its records are trustable. Similarly, there exists a security context between the HNR and the AA of the home edge network whose local hosts have their IDs, GLocs, and PKs stored in the HNR as the HNR records.

3.3 Functional Components and Notations

The proposed security scheme is mainly composed of four security components: mapping registration and retrieval security, network access security, communication session security, and mobility security. These functional components correspond to the security protections of different signaling procedures. When a host wants to access a network service, the mapping registration and retrieval security can be used to provide security protections to the host for the registration of its own ID/locator mapping in the HNR or for the retrieval of ID/locator mapping of a peer host from the DNR and HNR. After secure registration, the authenticity and accessibility of the host is assured by the network access security function, by which host's identity is verified and its information is securely registered in the edge network entities: the AA, LNS, and GW. If the host intends to start a communication session with a peer host, the communication session security procedure is used to enable the host to authenticate the peer host and protect data packets exchanged between them. At the same time, the mobility security function secures the handover process when the host moves from one edge network to another. In all these procedures, both asymmetric and symmetric cryptographic functions are leveraged to avoid the attacks described in the previous section.

While describing the procedures of the proposed security scheme, notations shown in Table 1 are used.

4. Mapping Registration and Retrieval Security

In this section, we describe the security functional components involved in the mapping registration and retrieval procedures. It enables secure registration and update of ID/locator mapping records, as well as secure retrieval of the records from the DNRs and HNRs.

4.1 DNR and HNR Records Registration Security

The DNR records are created and updated by the HNRs by sending HNR registration messages containing the HNRs' IDs, GLocs and PKs. The messages are integrity protected by a shared key of the HNR and the DNR.

Similarly, the HNR records are created by hosts with the help of the AA when they attach to their home edge network for the first time. Figure 2 shows the steps involved in hostname registration in the HNR. In the figure, arrows indicate the message flow from one entity to the other and values inside the square brackets indicate the parameters included in the message. The messages are integrity protected and authenticated by including hash-based message authentication codes (HMAC). As shown in the figure, when a host is booted up first time, it configures its local hostname and PK. It then obtains its local locator and network parameters such as the IDs and LLocs of the AA, LNS, and GW as well as the domain name and ID prefix of the home network using a host configuration protocol such as DHCP. The host uses the domain name to configure its global hostname and generates its ID by hashing the global hostname (to 64 bits) and attaching the ID prefix, scope, and version fields values [5]. The host then sends a host registration message containing its hostname, ID, PK, and registration ID (*regID*) to the AA. The message is protected by an HMAC computed by using the registration key (*regKey*). The AA verifies the *regID* and *regKey* by consulting the authentication server function that holds the list of *regIDs* and *regKeys* (not shown in the figure). After the verification, the AA forwards the host registration message to the HNR. The message is protected with an HMAC computed by using the shared key $SK_{(AA\text{-}HNR)}$ of the AA and the HNR. The HNR checks if the hostname is unique. If the hostname is not unique, i.e., some other host already possesses the same local hostname and the hostname already exists in the HNR record, the HNR replies to the AA without creating a new record. Otherwise, the HNR assigns a shared key $SK_{(Host\text{-}HNR)}$ to the host and creates a record for the host. After registration, the HNR replies the AA with a host registration response message, containing $SK_{(Host\text{-}HNR)}$ and its lifetime, both of which are encrypted by using $SK_{(AA\text{-}HNR)}$. The AA assigns an *accessID* and *accessKey* to the host and sends these values along with $SK_{(Host\text{-}HNR)}$, all of which are encrypted by *regKey*, to the host in the host registration response message. The message is protected by an HMAC computed over the message and *regKey*.

Thus in the hostname registration procedure, *regKey*,

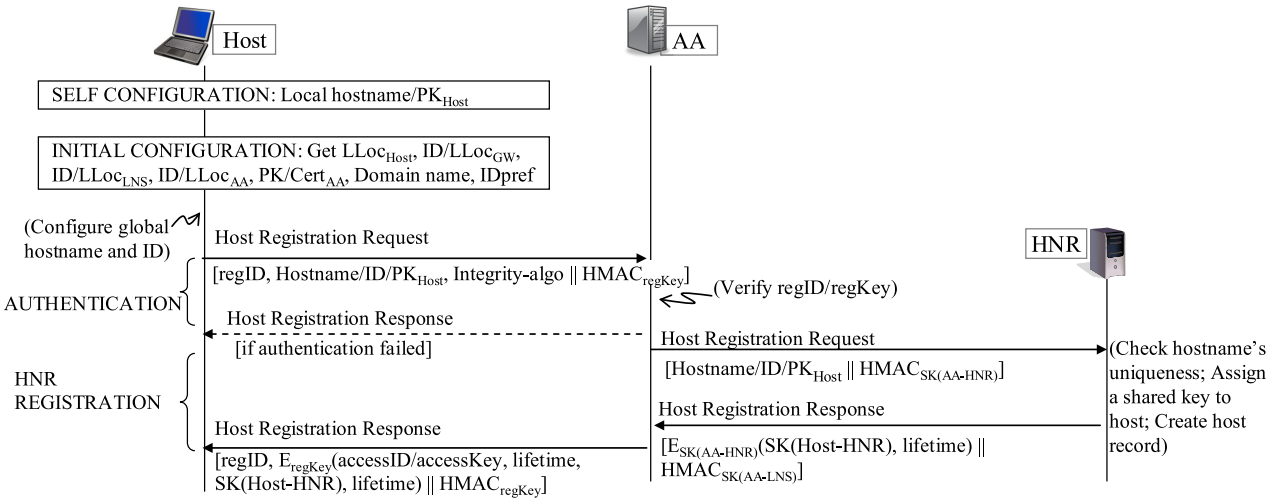

Fig. 2 Hostname registration in HNR. The signaling sequence shown by the dotted line occurs only when the host authentication fails and further steps of the HNR registration cannot take place.

HMAC, and $SK_{(AA\text{-}LNS)}$ are utilized to achieve the protection of the system from an impersonation attack and a man-in-the-middle attack. Meanwhile, replay attack can simply be inhibited by appending a timestamp or sequence number to the packets, and thus we will not mention again about replay attack inhibition in the security design of the other procedures.

4.2 DNR and HNR Records Retrieval Security

The HNR records are retrieved by executing a hostname resolution procedure that comprises a sequence of requests (or queries) and responses (or answers) as shown in Fig. 3. The hostname registration procedure is initiated by the host (called source host or SH) that wants to know about another host's (called target host or TH) ID, locator, and PK for initiating a session with the latter. For hostname resolution, the source host initially knows only the target host's hostname. The source host sends a hostname resolution request containing the target hostname and receives back a response containing the target host's ID, one or more locators, and PK from the HNR. The messages are integrity protected and authenticated by using HMACs or signatures (SIG).

As shown in Fig. 3, the source host sends the LNS a hostname resolution request, which is authenticated and integrity protected by an HMAC computed using the access key (*accessKey*) the source host has obtained from the edge network during the network access procedure as described in the next section. The LNS first checks its Host Table to find if the target host is located in the same edge network. If no record is found in the Host Table, it resolves the hostname by first sending a query to the DNR and then another query to the HNR. From the DNR, the HNR's ID, GLoc, and PK are obtained, while from the HNR the target host's ID, GLoc and PK are obtained. The DNR and HNR responses also include signatures and certificates, which are used to protect the authenticity and integrity of the message

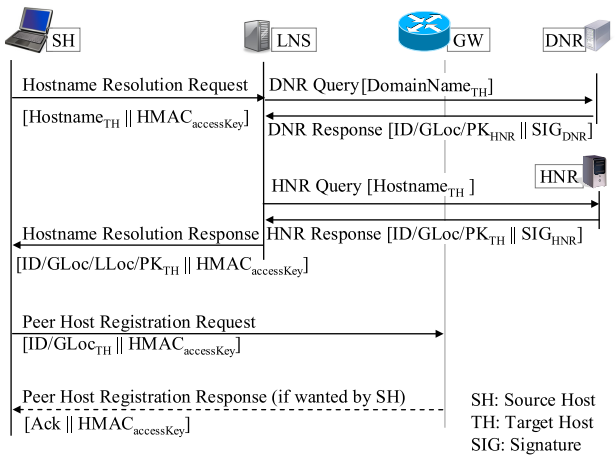

Fig. 3 ID/locator mapping retrieval from registries. The signaling sequence shown by the dotted line is optional.

content, i.e., the DNR and HNR records. The DNR records (which are static) are cached by other DNR or the LNS, but the HNR records (which are dynamic) are not.

The LNS forwards the target host's ID, GLoc (there may be many GLocs if the target host is multihomed) or LLoc (only if the target host is located in the same edge network), and PK to the source host in the hostname resolution response message, which is integrity protected by the HMAC. In case the target host is located outside the source host's local network, its LLoc seen by the source host would be the LLoc of a source GW. The source host stores the target host's info in its SAT. It then registers the target host's ID and GLocs (in case of multiple GLocs, the source host also assigns a priority value to each GLoc based on a destination locator selection algorithm) in the source GW (in case it has many GWs, one of them is selected on the basis of a source GLoc or GW selection algorithm) by sending a host registration message, authenticated and integrity protected by an HMAC computed using the access key. The

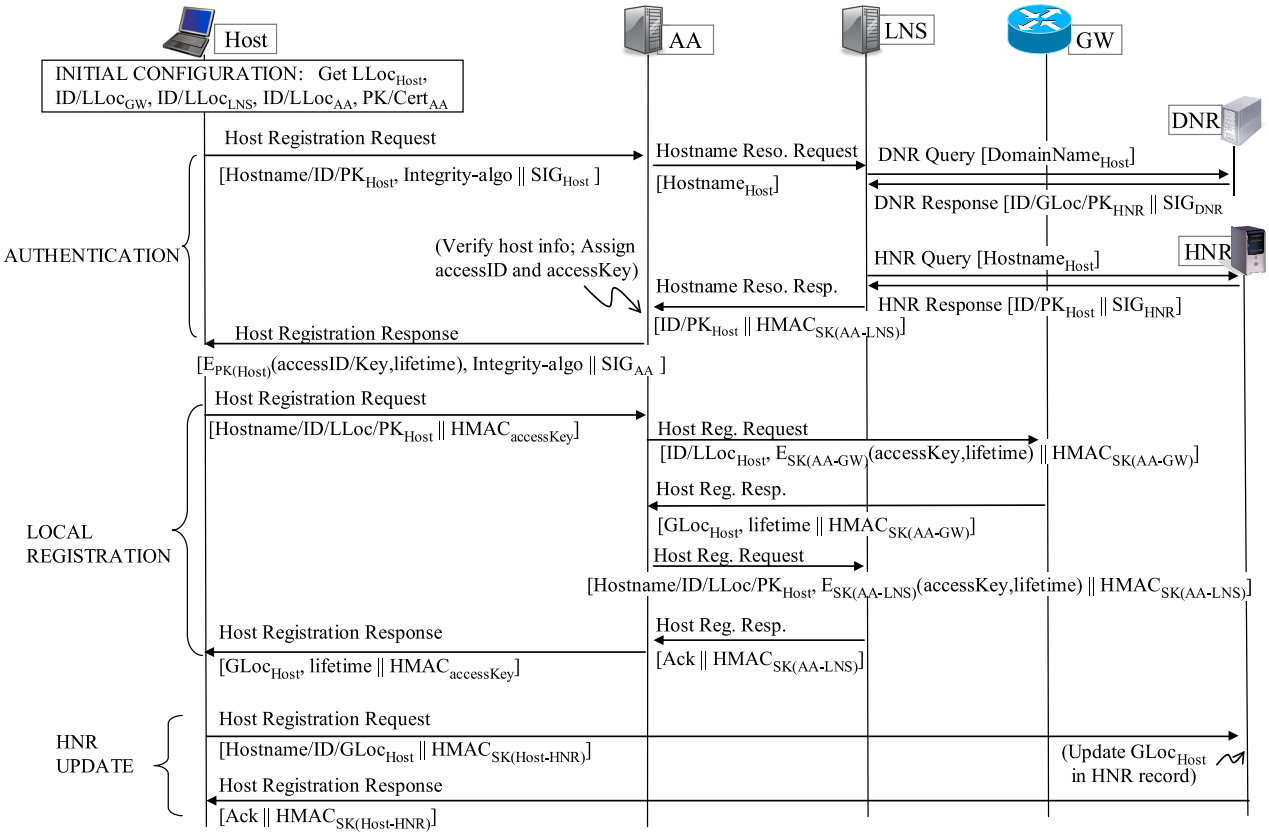

Fig. 4 Network access procedure.

GW adds the target host's ID and GLocs to its ID Table and then sends a response (i.e., an ack) to the source host if the latter had expressed its interest in the request message.

Alternatively, the LNS can register the target host's ID and GLoc in the source GW, if the source host had preferred this by setting a flag bit in the hostname resolution request message sent to the LNS. The message is integrity protected by an HMAC computed by using the shared key $SK_{(LNS\text{-}GW)}$ of the LNS and GW.

Thus, the ID/locator retrieval security is achieved through the use of HMAC computed with *accessKey*, SIG_{HNR} and SIG_{DNR}, which enable the whole procedure to become robust against impersonation and man-in-the-middle attacks.

5. Network Access Security

Network access security functions are for authenticating hosts and securely registering their hostnames, IDs, LLocs, and PKs in the AA, LNS, and GW, and updating the HNR records. For authenticating a host, the edge network uses the host's record securely retrieved from the HNR. As shown in Fig. 4, the network access procedure consists of four steps: initial configuration, authentication, local registration, and HNR update. Since the AA, LNS, and GW belong to the same trust domain, they possess pairwise shared keys, $SK_{(AA\text{-}GW)}$, $SK_{(AA\text{-}LNS)}$, and $SK_{(LNS\text{-}GW)}$, which are used for authentication and integrity protection of messages exchanged between any two of them. Thus, these shared keys, together with the hosts' PKs and *accessKeys*, protect the access procedure against impersonation and man-in-the-middle attacks.

When the host detects that an edge network is available in its surrounding, it obtains the initial configuration parameters such as its own LLoc as well as IDs and LLocs of the AA, LNS, and GW, and the PK and certificate of the AA either by executing DHCP or through router advertisements. After getting these parameters, the host enters into the authentication phase by sending a host registration request to the AA. The request message includes the hostname, ID, PK and one or more integrity-check algorithms that the host wants to use to compute HMACs for integrity protection of subsequent messages exchanged with the AA, GW, and LNS. The host registration request message is authenticated by the host signature, which also protects the AA from impersonation attacks. After verifying the host authenticity (by getting the host record from the HNR through the hostname resolution procedure as explained earlier), the AA selects an appropriate integrity-check algorithm, which may be HMAC-SHA1 [11]. The AA also assigns an access ID (*accessID*) and an *accessKey* (which has a definite lifetime) and replies the host with a host registration response, containing the *accessKey* encrypted by the host's PK and the whole message signed by the AA. *accessKey* is used to secure subsequent control messages the host exchanges with the edge network entities, i.e., the AA, LNS, and GW. When

the lifetime of *accessKey* is about to expire, the host gets the lifetime extended from the AA by sending a lifetime extension request message (not shown in figure).

The host then enters into the local registration phase by sending another host registration request to the AA to have its hostname, ID, LLoc and PK registered in the LNS and GW. The message is integrity protected by an HMAC computed using *accessKey*. The AA registers the host info (including *accessKey*) in the GW and LNS by sending host registration messages protected by HMACs computed using the keys shared with the GW and the LNS, respectively. During this registration the GW also assigns a GLoc (with an associated lifetime) to the host and stores the host info in its ID Table. Similarly, the LNS also stores the host info in the Host Table. After finishing the GW and LNS registrations, the AA responds back to the host with a host registration message containing the GLoc and integrity protected by an HMAC computing using *accessKey*. This completes the local registration phase and the host moves on to the HNR update phase. The host updates its ID/GLoc mapping in the HNR by sending a host registration message. The message is authenticated and integrity protected by the shared key, $SK_{(Host\text{-}HNR)}$, which the host had obtained when it registered its info in the HNR for the first time. Completion of the HNR update completes the network access procedure. Now the host is ready for communication with other hosts.

6. Communication Session Security

Communication sessions are secured by authenticating the peer hosts and protecting data packets exchanged between them by including some authentication data to inhibit impersonation and man-in-the-middle attacks. This scheme uses the HNR records to authenticate the hosts. After authentication, a security context is established in both ends, which is used to compute the authentication data for ensuring the packets' origin authentication and integrity protection.

6.1 Authentication and Security Context Establishment

When an application wants to use security function of the identity layer, the source host (SH) and target host (TH) establish security contexts (including a session key and integrity-check algorithm) by exchanging four control packets from the identity layer as shown in Fig. 5, after having finished the TH's hostname resolution. The SH sends a communication initialization request message containing the SH's hostname, ID, GLoc and PK, TH's ID and GLoc, and a list of integrity-check algorithms that the SH supports, and signed by the SH. The TH validates the SH's authenticity by retrieving the SH's ID and PK from the HNR through a hostname resolution, and comparing the HNR record with those values supplied by the SH. If verified, the TH does a peer host registration in the target GW (T-GW) to store the SH's ID and GLoc in the ID Table. The TH then selects an integrity-check algorithm from the list supplied by the SH. It stores in a temporary buffer the SH's ID, GLoc, PK, LLoc

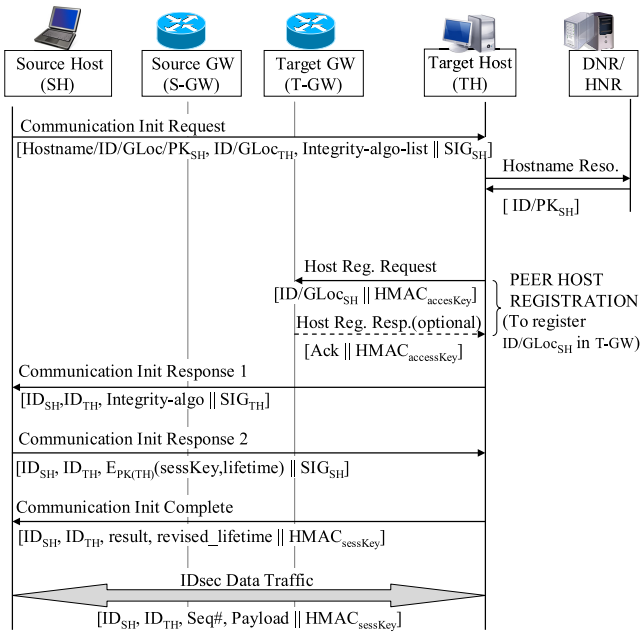

Fig. 5 Communication session establishment procedure. The signaling sequence shown by the dotted line is optional.

(i.e., the LLoc of T-GW) and the selected integrity-check algorithm. Then it configures a communication initialization response 1 message containing both the TH and SH's IDs and the integrity-check algorithm. The message, which is protected by the TH's signature, is then sent to the SH.

The SH generates a session key (*sessKey*) which would be used to authenticate and protect integrity of packets exchanged between the hosts. A lifetime is also associated with the session key. It then stores the TH's ID and PK, and the session key in a temporary buffer. After that it configures a communication initialization response 2 message including the session key encrypted by the TH's PK. The message is signed by the SH's private key and sent to the TH.

The TH copies the temporary buffer containing the SH's ID, GLoc and PK to the SAT. It also adds the session key received in the communication initialization response 2 message to the SAT. It then configures a communication init complete message by including the result of session initiation process. The message also includes an HMAC computed using the session key. The communication initialization complete message is sent to the SH.

The SH checks the result parameter to know if the TH has accepted the connection request. The TH might have also changed the value of session key's lifetime from what the SH had proposed in communication initialization response 2 message. The SH copies the temporary buffer containing the session security parameters to its SAT.

6.2 Data Packet Security

After completing the communication initiation by confirming the session key, the SH starts sending data packets whose ID header would contain the source and destination IDs, the

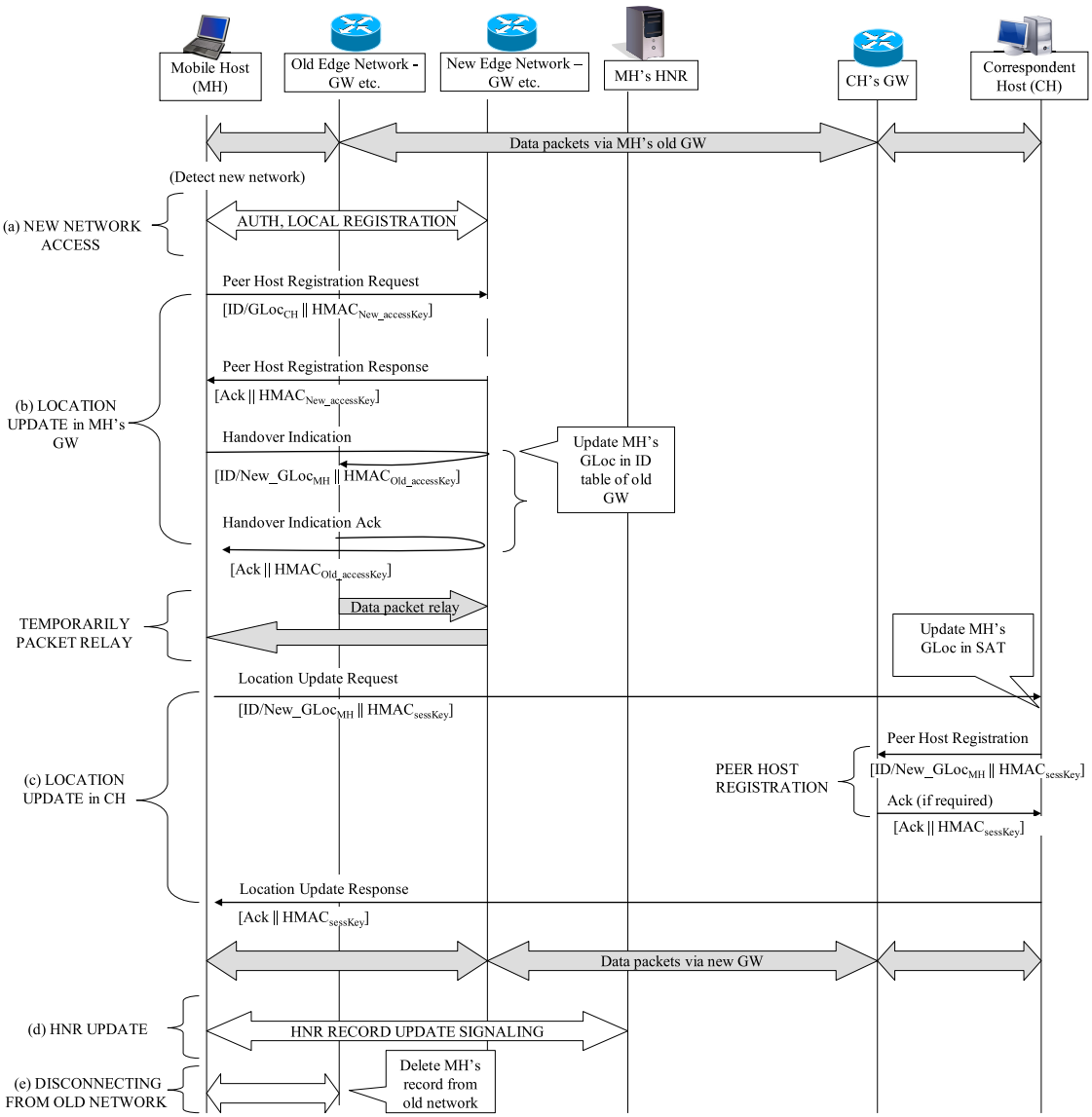

Fig. 6 Handover procedure.

packet sequence number and authentication data, which is an HMAC computed over the ID layer protocol data unit (PDU) and the session key by using the integrity-check algorithm negotiated during the communication init phase. The session key can also be utilized to encrypt the payload for achieving confidentiality if the need arises. On receiving these packets, the TH performs the packet's integrity check and origin authentication by referring to its SAT for the relevant security context.

7. Mobility Security

The network access security and communication session security functions can be leveraged to secure the mobility process. That is, the security contexts established in the host and network during the network access process and in the peer hosts during the session initialization process can be used to secure mobility management functions when the host moves from one edge network to another. In the handover procedure shown in Fig. 6, the mobile host (MH) performs the following signaling functions: (a) new network access to get new LLoc and GLoc from the new edge network; (b) updating the old GW with the MH's new GLoc so that the old GW can temporarily relay packets to the new GW during handover, (c) updating the peer or correspondent host (CH) and its GW so that the packets will be forwarded to the new location; (d) updating MH's HNR with the MH's new GLoc; and (e) disconnecting gracefully from the old edge network by deleting the MH related entries from the ID table of the old GW, AA and LNS. Functions (a), (b), (d), and (e) are related with the network access security, while (c) can be performed securely by using the security context established for the communication session.

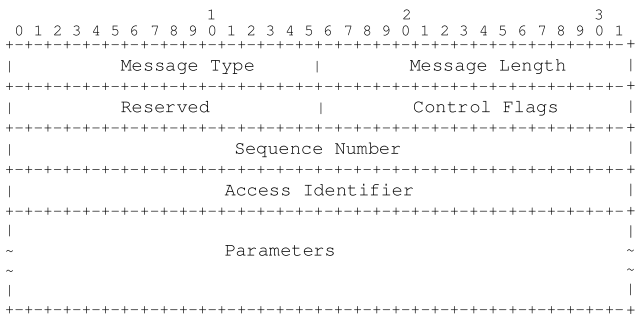

Fig. 7 Host registration message format.

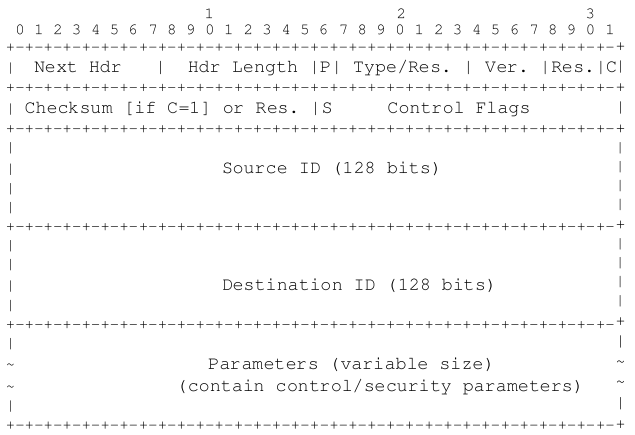

Fig. 8 ID header format.

8. Implementation

We have been implementing the proposed scheme on Linux. The DNR and HNR records are stored in the similar manner as DNSSEC [8] records. On top of that, we defined some new resource records such as HID and PK to store the host ID and public key, respectively. The DNR records are created by DNS UPDATE [9] using a TSIG (transactional signature) [10] for protecting the message integrity. Both the DNR and HNR queries (and responses) issued (and received) by the LNS are structured in formats similar to a DNSSEC query (and response). Additionally, we define a new common format for the host registration and hostname resolution messages exchanged between the host and edge network. These messages are exchanged from the application layer. Similarly, we also define an ID layer header format, which is used to carry communication initialization signaling messages, and, subsequently, authentication data for protecting packets. The following subsections describe these message formats.

8.1 Host Registration Control Message Format

The header of the host registration and hostname resolution control messages exchanged by the host with the access network entries or with the HNR is shown in Fig. 7. The basic header is 16 Bytes long and composed of six fields: Type (2 Bytes), Length (2 Bytes), Reserved (2 Bytes), Control Flags (2 Bytes), Sequence Number (4 Bytes) and Access Identifier (4 Bytes). Message Type specifies the type of message, e.g., 1 for host registration, 2 for peer host registration, and 3 for hostname resolution. Message Length specifies the length of the message including both the basic header and parameters. Control Flags specify additional meanings associated with the message. The following flag bits are defined: R (1 for Request/ 0 for Response), N (New registration), U (Update registration), T (Terminate registration), G (Registration in GW), S (Registration in LNS), H (Registration in HNR), A (Ack needed), and P (Peer host registration). Sequence Number specifies the sequence number of message and Access Identifier specifies the host's access identifier that the host acquires from the edge network at the time of network access. The sequence number and access identifier collectively identify the message uniquely and help in avoiding replay attacks.

The parameters field carries the message content in the type, length, value (TLV) format, i.e., Type (2 Bytes), Length (2 Bytes) and Value (variable size). Parameters such as the host ID, locator, PK or other information are carried by this field.

8.2 ID Header Format

The identity layer header format is shown in Fig. 8. The basic header, which is 40 Bytes long, has the following fields: Next Header (1 byte) specifies the protocol number of the next header in the ID layer's PDU payload (usually transport protocol number); if no payload exists, Next Header = NO_NXT_HDR (59). Header Length (1 byte) specifies the length of the ID header including the variable length parameters field; P (1 bit) specifies if the packet contains any upper layer payload; Type (7 bits) specifies the ID layer packet type (pType), i.e., types of control messages carried in the parameters field (e.g., pType = 1 for communication init request, 2 for communication init response 1, 3 for communication init response 2, 4 for communication init complete, and 0x81 for authentication data to protect the payload); Version (4 bits) specifies the ID layer protocol version, which is 1; C (1 bit) specifies if the ID header has a control message carried in the parameters field; Checksum (2 Bytes) specifies the checksum covering the ID basic header and parameters; Control Flags (2 Bytes) specify additional control information such as the nature of IDs and their mapping with locators (persistent/temporal, anonymous, etc.). Flag S is set to 1 to indicate that security functions implemented in the identity layer have been used. The source and destination ID fields (128 bits each) carry the packet's source and destination host IDs. The parameters field follows the basic header. It is of variable size and contains one or more parameters in the TLV format. It is used to carry security and control parameters for communication initialization, mobility management, and data packet transport.

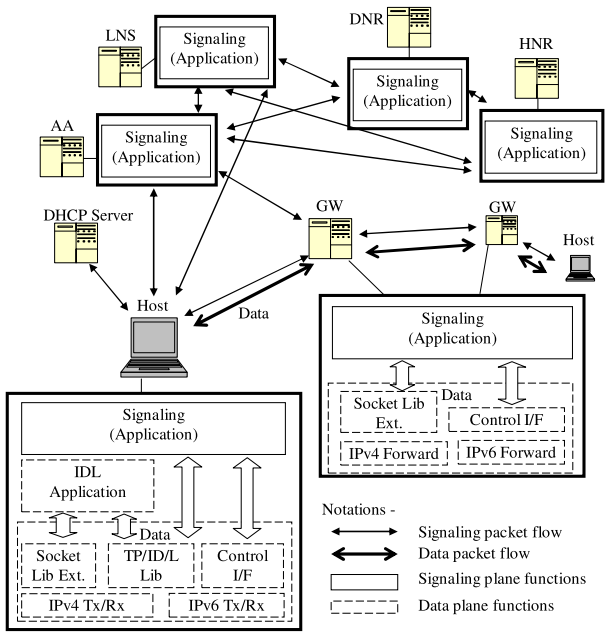

Fig. 9 Implementation layout.

8.3 Implementation layout

Figure 9 shows the functional block diagram of the implemented components. We have implemented new functions and extended existing functions both in the user space and kernel space. In the user space, new signaling plane functions are implemented while in the kernel space the ID/locator split-based data plane functions are implemented.

In the signaling plane, hostname registration and resolution, network access, communication initialization, and mobility functions are implemented. The signaling plane is spread over all nodes in the edge network and the logical control network. In order to enable the signaling plane in the user space to exchange information with the data plane in the kernel space, a control interface has been implemented using *ioctl* system call. The control interface is used to pass on signaling parameters, e.g., to add ID/locator mappings to the ID table in the kernel space.

We have been making the signaling plane flexible so that it can run its functions with or without a transport protocol. For example, except the DHCP which is running over UDP, the other signaling protocols may not need a transport protocol. They can send signaling messages directly in the ID header optional parameter fields via a raw socket.

In the data plane, new functions have been implemented in the kernel space by extending the existing socket library and introducing a new TP/ID/L[†] library. The new library enables TCP to work independently of the underlying IP protocol version or IP addresses. A new protocol family `PF_IDL` has been introduced in the socket API. Application data belonging to this protocol family are processed by TP/ID/L library. The system calls such as `bind()`, `connect()` and `accept()` that traditionally use IP addresses as a parameter have been extended to use IDs instead. In order to exchange data packets over the ID/locator split-based network stack, ID/locator split (in short, IDL)-supporting applications *Echo*, *IDperf*, and *VLC media player* have been developed. These applications create a socket of the protocol family `PF_IDL` and bind the source and destination IDs to the socket.

Thus, the signaling plane exchanges control messages for incorporating security in the network access process, hostname registration and resolution, and data sessions. The security contexts established in hosts and network nodes such as GWs are utilized to secure subsequent signaling messages (e.g., for mobility) or to secure data packets transmitted in the data plane. Making the signaling functions and IDL applications independent of the underlying IP protocol enables the HIMALIS architecture to support communication between hosts that may be using different protocols, such as IPv4 and IPv6, in the network layer. It also enables mobile hosts to perform smooth handovers across the networks that have heterogeneous network layer protocols.

9. Merits, Feasibility and Scalability

In this section, we discuss the merits, feasibility, and scalability of the HIMALIS architecture and its security mechanism.

The unique feature of the architecture is that it has been designed to support secure communications in heterogeneous and mobile networking environments. To our knowledge, none of ID/locator split-based related works has considered security for both heterogeneous and mobile networking environments. For example, HIP [2] intends to provide mutual authentication through the exchange of identities and public keys, and secures data packets by using IPsec. It does not cover security for communications across heterogeneous networks such as IPv4 and IPv6 networks. HIMALIS overcomes these limitations. It enables a host located in an IPv4 network to securely communicate with another host located in the IPv6 network. At the same time, it enables hosts to securely perform seamless handovers from the IPv4 network to the IPv6 network, and vice versa. Moreover, HIMALIS security features are intrinsic to the architecture, i.e., they are incorporated as an optimally inbuilt component of the architecture. The integrated security scheme shares the same security functions (such as encryption, key generation, and identity verification modules) and context (such as IDs, access keys, session keys, and public keys) to secure both the signaling and data plane functions. So, it is more efficient than other conventional approaches where distinct security mechanisms are added separately to secure different functions without considering their global optimization.

We have been implementing the HIMALIS security mechanism on Linux. So far we have completed implement-

[†]Detail of kernel implementation, not including the security components, is given in [14].

ing data plane functions and the implementation of signaling plane functions are progressing now, which are expected to complete by 2013 March. The preliminary results that have been reported in [14] indicate that the HIMALIS performance in the data plane is comparable with the conversional TCP/IP's performance although the HIMALIS hosts have to perform ID/locator mapping for mobility and multihoming support and the GWs have to perform network layer protocol translation for heterogeneity support. Namely, the HIMALIS architecture, which has additional benefits of supporting intrinsic security, mobility and heterogeneity features, has the potential to scale in the same manner as the current Internet.

Since the implementation of the signaling plane functions that include the security features for name resolution and mutual authentication is still progressing, we are unable to show the performance data. However, we can qualitatively explain the feasibility and scalability of the HIMALIS signaling plane as follows. To keep the HIMALIS's name resolution process as scalable as the current Internet's, we have designed it in such as way that it provides both hosts and their GWs with the ID/locator mapping records through the name resolution process executed at the beginning of a session. Unlike in LISP [1], no additional signaling or time is needed at the GWs to lookup for the ID/locator mapping records. This would reduce the burden on the GWs because they do not need to carry out the ID/locator mapping lookup and related security functions.

Similarly, to make the name resolution mechanism secured and scalable in both retrieving and updating dynamic ID/locator mappings, we have proposed the two-layered name resolution architecture consisting of DNRs and HNRs. DNRs are designed to favor the scalable retrieval of static records about HNRs, and the HNRs are designed to favor the faster update of dynamic records about hosts. The proposed integrated security scheme uses HNR's static records provided by the DNRs to secure the retrieval of dynamic hosts' ID/locator mappings from the HNR. Moreover, the IDs, public keys, etc. retrieved from HNRs are used for multiple purposes, e.g., mutual authentication of a host and an edge network when the host attaches with the edge network, mutual authentication of two hosts when they initiate a data session, and securing their sessions.

When we finish implementing all components of the HIMALIS security mechanism, we will evaluate their effectiveness in an at-scale experimental facility consisting of NICT's JGN-X [17] and StarBED [18]. We implement the HIMALIS architectural components, mainly the GWs, DNRs, HNRs and hosts, in JGN-X nodes located in different places from Hokkaido to Kyushu and measure the time taken and signaling overhead incurred by the ID/locator mapping retrieval process. We will compare these metrics with those of DNSSEC. Similarly, to assess the performance in an emulation environment consisting of a large number of nodes, we will use StarBED. We will measure the effectiveness of the proposed scheme in terms of delays and signaling overheads when a large number (about one thousand) of hosts issue name resolution requests. We will also study these parameters when many hosts (say, about 500) move to access a new edge network simultaneously and update their ID/locator mappings in the HNR. We also measure the session setup delay, i.e., the time duration starting from the instance a host issues a name resolution request to get the ID, locator and public key of a target host to the instance it starts sending data packets to the target host. Based on these results, we will continuously improve the proposed security mechanism by optimizing the implemented functions as well as by adding new functions.

10. Conclusion

We presented the integrated security scheme for the hostname registration and resolution, network access process, communication sessions and mobility signaling of the ID/locator split-based HIMALIS architecture. The proposed scheme is fully integrated as the intrinsic component of HIMALIS architecture to enable it to optimally support secure and mobile network services in the future heterogeneous and dynamic networking environment where hosts change locations frequently due to mobility. This scheme protects the network from impersonation, man-in-the-middle, and replay attacks. By avoiding impersonation attacks, this scheme protects the network from DoS attacks as well. In future work, will refine the scheme further on the basis of the experimental results.

Acknowledgments

The authors are grateful to Masugi Inoue, Hajime Tazaki, and Kenji Fujikawa for their valuable input to this research.

References

[1] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID separation protocol (LISP)," IETF Internet-Draft, http://www.ietf.org/id/draft-ietf-lisp-23.txt, May 2012.
[2] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host identity protocol," RFC 5201, April 2008.
[3] E. Nordmark and M. Bagnulo, "Shim6: Level 3 multihoming shim protocol for IPv6," RFC 5533, IETF, June 2009.
[4] V.P. Kafle and M. Inoue, "HIMALIS: Heterogeneity inclusion and mobility adaptation through locator ID separation in new generation network," IEICE Trans. Commun., vol.E93-B, no.3, pp.478–489, March 2010.
[5] V.P. Kafle, H. Otsuki, and M. Inoue, "An ID/locator split architecture for future networks," IEEE Commun. Mag., vol.48, no.2, pp.138–144, Feb. 2010.
[6] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis, "LISP alternate topology (LISP+ALT)," IETF Internet-Draft, http://www.ietf.org/id/draft-ietf-lisp-alt-.txt, IETF, Dec. 2011.
[7] ITU-T Recommendation Y.2015, "General requirements of ID/locator separation in NGN," 2009.
[8] R. Arends, A. Austein, M. Larson, D. Massey, and S. Rose, "Resource records for the DNS security extensions," RFC 4034, IETF, March 2005.
[9] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, "Dynamic updates in the domain name system (DNS UPDATE)," RFC 2136, IETF, April 1997.

[10] P. Vixie, O. Gudmundsson, D. Eastlake, and B. Wellington, "Secret key transaction authentication for DNS (TSIG)," RFC 2845, IETF, May 2000.
[11] C. Madson and R. Glenn, "The use of HMAC-SHA-1-96 within ESP and AH," RFC 2404, IETF, Nov. 1998.
[12] V.P. Kafle, R. Li, D. Inoue, and H. Harai, "An integrated security scheme for ID/locator split architecture of future network," Proc. FutureNet Workshop (held with IEEE ICC'12), June 2012.
[13] D. Dolev and A. Yao, "On the security of public key protocols," IEEE Trans. Inf. Theory, vol.29, no.2, pp.198–208, 1983.
[14] V.P. Kafle, H. Tazaki, T. Tomuro, Y. Kobari, and H. Harai, "Prototype implementation and evaluation of ID/locator-split protocol stack," IEICE Technical Report, NS2012-26, May 2012.
[15] F. Maino, V. Ermagan, A. Cabellos, D. Saucez, and O. Bonaventure, "LISP-security (LISP-SEC)," IETF Internet-Draft, draft-ietf-lisp-sec-03, Sept. 2012.
[16] R. Atkinson, "Security architecture for the Internet protocol," RFC 4301, IETF, Dec. 2005.
[17] New Generation Network Testbed JGN-X, http://www.jgn.nict.go.jp/index.html
[18] StarBED Project, http://www.starbed.org/

Ved P. Kafle received the B.E. degree in electronics and electrical communications from Punjab Engineering College (now PEC University of Technology), Chandigarh, India, the M.S. degree in computer science and engineering from Seoul National University, South Korea, and the Ph.D. in informatics from the Graduate University for Advanced Studies, Japan. Dr. Kafle is a senior researcher at NICT, where he is involved in the design, implementation, evaluation, and optimization of algorithms, protocols, and architectures of new generation networks or the future Internet. In particular, his current research interests include new naming and addressing schemes, ID/locator separation architectures, name or ID resolution mechanisms, integration of heterogeneous network layer protocols, integration of resource-constrained sensor networks into the Internet for ubiquitous sensing and computing, distributed mobility management, and privacy, security and trust in communication networks. He has been awarded with the ITU Association of Japan Award in 2009 for his active contributions to the standardization of Next Generation Network architectures. He also received the best paper award (second prize) at the ITU-T Kaleidoscope event on Innovations for Digital Inclusion, 2009.

Ruidong Li received a bachelor in engineering from the Department of Information Science & Electronic Engineering, Zhejiang University, China in 2001. He received a masters and doctorate of engineering from the University of Tsukuba in 2005 and 2008, respectively, both in computer science. Since 2008, Dr. Li has been working as a researcher at NICT. He is currently investigating security architectures for the new generation networks. Dr. Li's research interests include new generation network architecture, network security, mobility, and modeling and performance evaluation.

Daisuke Inoue received his B.E. and M.E. degrees in electrical and computer engineering and Ph.D. degree in engineering from Yokohama National University in 1998, 2000 and 2003, respectively. He joined the Communications Research Laboratory (CRL), Japan, in 2003. The CRL was relaunched as the National Institute of Information and Communications Technology (NICT) in 2004, where he is the director of Cybersecurity Laboratory in Network Security Research Institute. His research interests include security and privacy technologies in wired and wireless networks, incident analysis and response technologies based on network monitoring and malware analysis. He received the best paper award at the 2002 Symposium on Cryptography and Information Security (SCIS 2002), and the commendation for science and technology by the minister of MEXT, Japan, in 2009.

Hiroaki Harai received M.E. and Ph.D. degrees in Information and Computer Sciences from Osaka University, Japan in 1998. He is currently a Director at National Institute of Information and Communications Technology (NICT), Tokyo, Japan, where he is leading Network Architecture Lab. for Optical and New-Generation Networks. He is concurrently a Visiting Associate Professor of The University of Electro-Communications, Tokyo, Japan. His current research topic is design and development of new generation network architecture. Dr. Harai was elected to Outstanding Young Researcher in the 3rd IEEE ComSoc Asia-Pacific Young Researcher Award, 2007. He received the 2009 Young Researcher Award from the Ministry of Education, Culture, Sports, Science and Technology. He is a member of IEEE.