Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2011, Article ID 437868, 14 pages
doi:10.1155/2011/437868

Research Article

HOP: Achieving Efficient Anonymity in MANETs by Combining HIP, OLSR, and Pseudonyms

Javier Campos, Carlos T. Calafate, Marga Nácher, Pietro Manzoni, and Juan-Carlos Cano

DISCA, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain

Correspondence should be addressed to Carlos T. Calafate, calafate@disca.upv.es

Received 19 May 2010; Revised 31 July 2010; Accepted 1 September 2010

Academic Editor: Damien Sauveron

Offering secure and anonymous communications in mobile ad hoc networking environments is essential to achieve confidence and privacy, thus promoting widespread adoption of this kind of networks. In addition, some minimum performance levels must be achieved for any solution to be practical and become widely adopted. In this paper, we propose and implement HOP, a novel solution based on cryptographic Host Identity Protocol (HIP) that offers security and user-level anonymity in MANET environments while maintaining good performance levels. In particular, we introduce enhancements to the authentication process to achieve Host Identity Tag (HIT) relationship anonymity, along with source/destination HIT anonymity when combined with multihoming. Afterward we detail how we integrate our improved version of HIP with the OLSR routing protocol to achieve efficient support for pseudonyms. We implemented our proposal in an experimental testbed, and the results obtained show that performance levels achieved are quite good, and that the integration with OLSR is achieved with a low overhead.

1. Introduction

In the last decade, mobile ad hoc networks (MANETs) have been one of the most challenging fields of research. Several new types of attacks specific to MANETs have been detected and thoroughly analyzed in the literature [1, 2], evidencing that securing these networks efficiently is of utmost importance.

Communication anonymity, a field of research tightly related to security, has recently received much attention from the research community. Despite most proposals have focused on wired environments [3–5], mobile ad hoc networks have also received much attention [6–9]. Nácher et al. [10] present a comprehensive survey of anonymous communications in mobile ad hoc networks, which emphasizes on the vast amount of literature in this field. More recently we can find several proposals addressing privacy and security in the context of single-path routing [11–15], and even some anonymous multipath routing proposals [16, 17].

The communication anonymity research field encompasses various topics [18] such as sender anonymity, recipient anonymity, relationship anonymity, and localization anonymity. In this work we are mainly interested in relationship anonymity, that is, in avoiding that MANET participants are able to identify the two communicating parties involved, for example, in a VoIP call, thereby compromising security.

Typically networked hosts are identified by their IP and MAC addresses; however, these can be easily altered by users. The problem of IP/MAC address forgery can be solved through a unique cryptographic identifier that remains unaltered throughout time, and that must be completely independent from the network identifiers. In earlier anonymous protocols for MANETs, the authentication issue is either not considered [6] or a Third Trust Authority is required to exchange a set of preestablished parameters [7]. A ring signature scheme is used in more recent proposals [8, 9] where an authenticated key agreement protocol based on a group-oriented signature is used to verify that a message has been signed by one of the group members without actually knowing whom. Hence, this scheme requires that nodes have *a priori* knowledge of some trusted nodes in the network; this means that other nodes, besides the two endpoints, must also implement the protocol. An integrated solution, known as HIP (Host Identity Protocol)

[19], has been proposed to support authentication, security, and mobility but specifically in the scope of the Internet infrastructure.

In this paper, we propose HOP, a novel solution that improves the functionality of HIP to achieve authentication and relationship anonymity in MANET environments. In our proposal we provide full integration with the OLSR routing protocol [20] to adopt layer-3 pseudonyms in an efficient manner. Our solution is compatible with the original HIP mechanisms and is integrated with the IPsec protocol [21], providing a secure communications environment. The proposed solution does not require the rest of the nodes in the network to implement the HIP protocol or our modified version of OLSR. Thus, only the sender and the recipient have to implement HOP, thereby reducing resource consumption on intermediate nodes, which is considered a critical issue in MANETs. We implemented and evaluated the efficiency of our proposal in a real testbed, showing that the impact on performance is minimal.

To the best of our knowledge, our proposal is pioneer in this area since no other authors have proposed anonymity mechanisms for MANETs based on improvements to the HIP protocol. Additionally, we implemented and validated our proposal in a real testbed, including a performance evaluation to verify that the proposed solution does not compromise performance.

The structure of this paper is the following. In Section 2, we refer to some related works on MANET anonymity, focusing mostly on the performance assessment of the different proposals. In Section 3, we make a brief introduction to the Host Identity Protocol. Section 4 describes our proposal to extend HIP in order to achieve anonymity in MANET environments. The adversarial model is described in Section 5. Section 6 presents HOP, a solution integrating the anonymous HIP solution with the OLSR protocol to support a higher degree of anonymity through the adoption of pseudonyms. It is followed by a security analysis in Section 7. Implementation details are addressed in Section 8. Experimental testbed results are then presented in Section 9. Finally, Section 10 concludes the paper.

2. Related Works

Although the field of MANET anonymity has received significant attention from the research community in the past decade, most of the proposals found in the literature suffer from the following two drawbacks: (i) the performance of the different protocols has remained mostly untackled, and (ii) no actual implementations and validation of the different protocols in a real testbed have been made. For instance, focusing on the two solutions more frequently cited—MASK and ANODR—the MASK protocol [7] has only been validated through simplistic and overly optimistic simulation tests. In the case of ANODR [6], limited simulation tests with only minimal amounts of traffic are offered. More recent solutions suffer from similar drawbacks, being that several do not include any performance tests [12–14].

Concerning the performance evaluation of anonymity solutions for MANETs, we find that some recent works have specifically addressed the performance of anonymous protocols for MANETs. Liu et al. [22] present a comprehensive survey and performance evaluation of different anonymous routing schemes, focusing on the existing tradeoffs between the performance and the degree of protection. Through simulation they show that the processing delay associated with public key cryptography based protocols causes performance to degrade significantly. Another study by Nácher et al. [23] has shown that, for anonymous routing protocols like MASK and ANODR, anonymity is obtained at the expense of reducing performance values down to inefficient levels for both TCP and UDP protocols. In particular, they found that ANODR's throughput ranges from 10 to 100 Kilobits per second, which is a really bad performance, while for the MASK protocol these values range from 100 to 500 Kilobits per second, which is still considered quite poor. With respect to UDP traffic, they found that excessive delay values impede the use of applications with real-time requirements, being the packet loss ratio also considered quite high. Dong et al. present ARMR [16] an anonymous routing protocol in which multiple routes and fake routes are established; authors use simulations to assess the route request efficiency of their solution compared to MASK and AODV. Defrawy and Tsudik [11] evaluate the performance of PRISM and ALARM, two protocols of their own design, in terms of percentage of nodes exposed and routing overhead. Chen and Wu [17] do something similar, relying on simulation to evaluate both a single and a multipath routing protocol of their own design in terms of message compromising probability and routing overhead. Notice that most of these works miss important metrics such as data traffic delay or packet loss ratio.

In this paper, we offer a solution to provide anonymity in MANETs that is pioneer in the sense that it includes the proposal description along with the implementation details and performance assessment tests made in a MANET testbed using real devices.

3. The Host Identity Protocol

The Host Identity Protocol (HIP) [19] was introduced by IETF's HIP working group [24]. It was designed to make host identification independent from the points of attachment (IP addresses). Such solution, among other benefits, is able to solve the problem of tracking mobile hosts.

One of the main concepts introduced by HIP is the Host Identity Tag (HIT). A HIT consists of a 128-bit identifier assigned to a specific machine. HITs, differently from IP addresses, are always permanent. This means that a host can have several IP addresses that change frequently throughout time without causing hosts to break transport layer connections between them since HITs are used to identify socket connections instead of IP addresses. Therefore, the use of HITs requires introducing a new layer between the routing and transport layers to achieve the desired independence between them.

One of the main advantages of HITs is their close bonds with asymmetric cryptography. In fact, HIT generation consists of obtaining a straightforward 128-bit hash of a Host Identifier (HI), which consists of a public key generated through an asymmetric encryption algorithm. In particular, HIP authors propose using the SIGn-and-MAc (SIGMA) algorithm [25].

HIP was designed to operate in the scope of the Internet by extending the DNS functionality to incorporate Host Identifiers (HIs) and Host Identity Tags (HITs). Through such a service it is easy for a user to discover and maintain both HI/HIT and IP addresses for a particular host or domain.

With HIP, the setup of a secure channel between two hosts relies on a authenticated four-way handshake based on the SIGMA algorithm. Such four-way handshake includes a *Request* message from initiator to responder (I_1), a *Challenge* message sent back to the initiator (R_1), a *Response/Authentication* message sent from initiator to the responder (I_2), and, finally, an *Authentication* message that is sent back to the Initiator (R_2). During this message interchange a session key as well as a pair of IPsec ESP Security Associations are created (for more details on IPsec please refer to [21]).

In the scope of mobile ad hoc networks, HIP suffers from two main problems: (i) there is usually no access to a DNS server for HI/HIT and IP retrieval, and (ii) it is quite easy to track connections and their endpoints. Thus, a solution specific for MANET is required when attempting to operate in these environments.

4. Adapting HIP to Offer Anonymity in MANET Environments

In this section, we present *Anonymous HIP* (A-HIP), an improvement of the original HIP implementation offering security, relationship anonymity, and limited sender/recipient anonymity to MANET communications. The basic assumption for our approach is that, for communication to take place, end-points must be aware of each other's HIT and the respective public key (HI). This requires a previous exchange of HIs/HITs between peers through a trustworthy mechanism. This exchange could be achieved using currently available technology by embedding HIs/HITs on *digital Business Cards* and then use Bluetooth's *Business Card Exchange* function [26] (part of the *Object Push Profile*) to make them available to trusted parties. Another option is to rely on mobile telephony messagery for this task. However, how hosts gain knowledge of each other's HI/HIT is outside the scope of this paper.

Our A-HIP solution adopts the concept of multihoming, allowing each endpoint to use a different IP address for each destination. Through such technique, both sender and recipient anonymity is achieved with respect to the rest of users. In addition, we extend the standard four-way handshake defined by HIP, allowing to translate HITs into an IPv4 or IPv6 address anonymously, and without requiring any DNS service.

Packet type	HIT_{src}	HIT_{dst}	Data

Figure 1: Structure of a HIP message.

4.1. Making HIP Messages Anonymous. Before detailing the proposed HIP message adaptations to achieve anonymity, we must first introduce some definitions. Let HIT_{src} and HIT_{dst} denote the source and destination HIT identifiers, and let κ^i_{Pub} and κ^i_{Pri} be the public and private keys associated to a certain HIT_i, respectively. We define the encryption of message m using key κ as

$$m^* = E(m, \kappa) \tag{1}$$

and the message decryption using the complementary key as

$$m = D(m^*, \kappa'). \tag{2}$$

With HIP, when two stations wish to exchange session setup information, the messages exchanged must comply with the format defined in the RFC for HIP [19]. The basic structure of these messages is shown in Figure 1.

Since the *Packet type*, HIT_{src} and HIT_{dst} fields are unencrypted, all participants of a MANET are able to identify the two communication endpoints. Thus, in our scheme, we propose encrypting all the fields in HIP messages. The participants involved in a HIP session will have to use the public key of the destination—κ^{dst}_{Pub}—to encrypt a given HIP message m_{HIP}:

$$m^*_{HIP} = E\left(m_{HIP}, \kappa^{dst}_{Pub}\right). \tag{3}$$

Message m^*_{HIP} may then be propagated through the MANET without other stations being aware of the communication endpoints, as desired. Nevertheless, all the stations in the path must try to decrypt the message using their private key(s) to determine whether they are the destination.

4.2. A-HIP Message Exchanges. Figure 2 illustrates the modified message exchanging scheme proposed. Before proceeding we should remark that, in the scope of HIP, the Initiator and Responder concepts are introduced to refer to both endpoints, and they remain unaltered throughout the HIP session. Therefore, they are unrelated to the source and destination concepts, which maintain their usual meaning and thus alternate in time, as shown in this figure (see, e.g., messages I'_1 and R'_1).

Our proposed scheme works as follows. Initially, a HIT discovery packet is generated (message I'_1), being flooded to all nodes. Flooding is mandatory in a MANET environment since the initiator has no knowledge about the responder's IP. Hosts receiving the message will use their private key (κ^i_{Pri}) to try to obtain the original m_{HIP} message. Only if the appropriate key ($\kappa^{Resp.}_{Pri}$) is applied will the original message be retrieved:

$$m_{HIP} = D\left(m^*_{HIP}, \kappa^{Resp}_{Pri}\right), \tag{4}$$

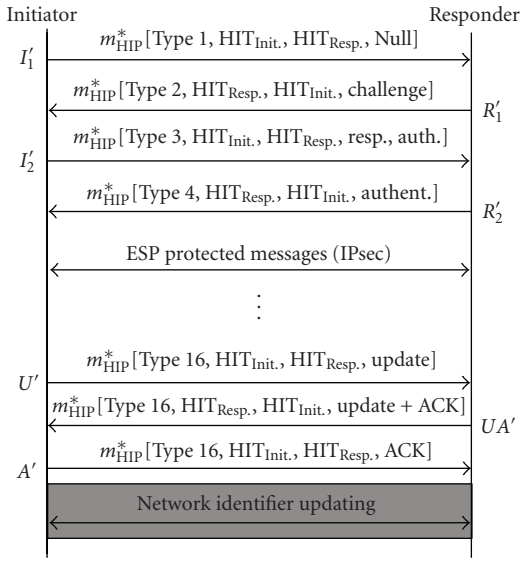

Figure 2: Modified message exchanging between initiator and responder based on HIP.

making the HIT discovery process fully confidential by avoiding HIT traceability.

The responder, aware of the initiator's IP address based on the initial request, sends a challenge message (R'_1) back to the initiator via unicast, allowing the initiator to associate the responder's HIT to its IP address. If the responder does not want to communicate with the initiator, it may opt to block the authentication process by not replying at all.

A new message interchange between initiator and responder takes place afterward, during which a Diffie-Hellman authenticated key exchange is used to create a session key, allowing to establish a pair of IPsec ESP Security Associations (SAs) between the initiator and the responder, as defined by the standard HIP. If, later, either the initiator or the responder wish to change their network identifiers, they must then proceed to update the connection using encapsulated HIP UPDATE packets (represented as U', UA', and A' in Figure 2); since this is part of the standard HIP, please refer to [19] for more details.

Long connection disruptions, which may be due to node mobility, require restarting the base exchange to update the existing security associations (SAs). In such case, a new HIT discovery packet (message I'_1) must be again flooded throughout the MANET, as described before.

5. Adversarial Model

The anonymity improvement presented above avoids that any intermediate host is able to read the HIT source and destination fields included in HIP packet headers since the whole packet is encrypted. Additionally, the trusted exchange of HI/HIT pairs assures that the cryptographic identity of a user is only known by trusted users. However, our adversary model considers the possibility that an attacker may become a trusted user by both parties, and is also able to establish connections towards both hosts, as illustrated in Figure 3.

This figure shows three nodes tagged as A, B, and C. Intermediate node C is the attacker, and our adversarial model contemplates the possibility that it is able to participate in the MANET as a regular user, as well as capture, modify or inject information into the network.

Suppose that the adversary (C) has previously established a HIP association with both A and B. As a consequence, C is able to store the relationship between the identities of both A and B and their current IP addresses in an IP-to-HIT mapping table. Thus, when node A sends and encrypted packet to B, C will be able to trace the source and destination IP addresses. By consulting its IP-to-HIT mapping table, it will then be able to deduce which are the HIT identities associated with the encrypted packet, thus compromising anonymity.

Our goal when facing this type of adversary is to prevent it from guessing the associations IP-to-HIT associations of the different nodes. Below we describe a novel solution that achieves this goal by avoiding that attackers having gained the trust of all parties can easily determine the two endpoints of any connection. To achieve this, a strategy based on the use of perconnection pseudonyms is recommended [27]. However, since in this case we require layer-3 pseudonyms, any solution designed to be used in MANET environments requires full or at least partial integration with the MANET routing protocol used. In the next section, we propose such a solution integrating A-HIP with the OLSR protocol.

6. HOP: Combining A-HIP, OLSR, and Pseudonyms

In this section, we present HOP, our proposal to achieve enhanced anonymity protection in MANET environments. HIP combines the A-HIP protocol with the use of pseudonyms, which are efficiently integrated with the OLSR protocol.

A pseudonym is a new identifier that is used instead of the original identifier to improve anonymity. In particular, we propose using multiple IP addresses per station (one per destination) to achieve a higher degree of anonymity when communicating. This way, when two nodes wish to establish a secure connection, each will select a free IP address from its IP address pool that is used as a pseudonym for that connection.

Figure 4 illustrates the proposed solution. When node A establishes a HIP association with node C, it will pick one of its multiple IP addresses available (e.g., IP_D). Such pseudonym will be used as IP source address for the first packet (I_1) in the establishment of a HIP association. When this packet arrives to C, it will also pick one of its pseudonyms for the reply. The IP addresses used as pseudonyms will be maintained for all the communications between A and C.

We can also observe how anonymity is now preserved: for the A-C HIP association, A is using IP_D as its pseudonym; for the B-C HIP association, B is using IP_F as its pseudonym. When an association between A and B is established, both

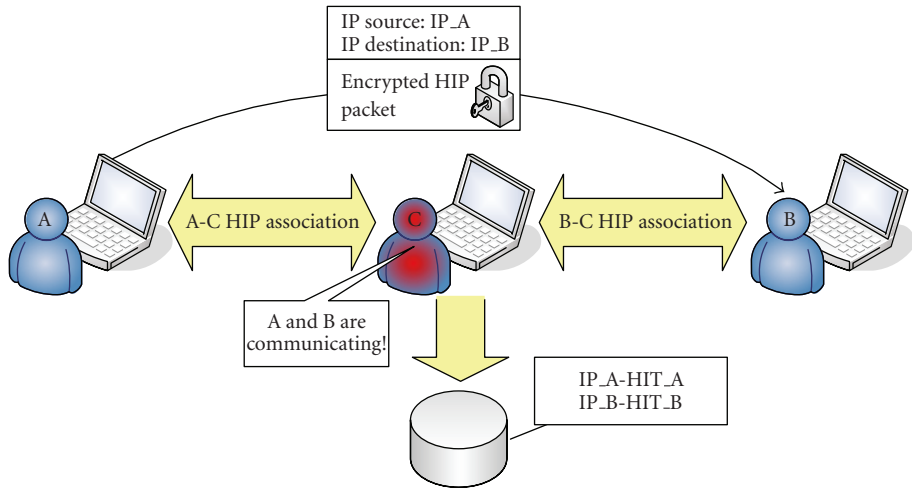

Figure 3: Anonymity loss with A-HIP.

nodes avoid using the previous pseudonyms. In particular, A and B use IP_E and IP_G, respectively. Thus, although the encrypted HIP packets for this connection may be captured by node C, it is now unable to trace the HIT identities used in that connection.

Thus, thanks to our solution, it becomes quite difficult for a malicious node to identify the two endpoints of a connection since session initiation is made anonymous by: (i) encrypting all packets (both HIP-related and data), and (ii) avoiding static mappings between HI/HIT and IP addresses.

6.1. Efficient Integration of HIP Pseudonyms and OLSR. Deploying the aforementioned solution in a real MANET environment requires solving some technical issues. First of all, we must prevent that the different network nodes become aware of the different pseudonyms employed by each node. To achieve this, each IP address must be seen by all modes as if it was a unique IP address associated with a real node. We accomplish this task through an efficient integration with the OLSR routing protocol [20]. In particular, we propose modifying the OLSR routing protocol so that the pseudonyms used by a terminal are announced to other terminals as if they were real IP addresses. This way, when a node wants to communicate with another one using its pseudonym, it will merely send a packet to that regular IP address, which is then handled by the OLSR protocol to make sure it arrives to the correct terminal. Upon arrival the packets are then processed internally without any further transmission.

Focusing again on Figure 4, the pseudonyms used by node A (D and E) and node B (F and G) would be announced to other terminals along with the real IP addresses (IP_A, IP_B, IP_C).

An optimal integration of A-HIP with the OLSR daemon requires, among other things, simulating the arrival of control messages coming from the fake neighbors (pseudonym IPs). For example, the OLSR daemon at node A must simulate the reception of control messages coming from fake nodes D and E. These messages will be of both HELLO and TC types, and shall contain all the necessary information to simulate the topology shown in the figure, where node A is selected and Multipoint Relay (MPR) of nodes D and E, so that A is responsible for the propagation of messages coming from this fake neighbors. This way the remaining nodes in the network are unable to distinguish real nodes from fake ones.

In our proposal, the OLSR daemon shall also offer automatic selection of pseudonyms, which means that the daemon itself will be responsible for picking a free IP address range to be used as pseudonyms, which simplifies the users' tasks.

One of the advantages of our solution is that we maintain compatibility with the original OLSR protocol, meaning that nodes using the modified version of OLSR will benefit from anonymity even when most of the nodes in the MANET use the original OLSR version.

6.2. Prevention of ARP-Based Attacks. To support different IP addresses in a single node we configure the network interface card to have multiple addresses. This way, data associated with that IP address is processed by the TCP/IP protocol stack seamlessly. However, an attacker could attempt to detect such configuration to cause disclosure of pseudonyms and compromise anonymity.

Figure 5 illustrates the attack process, whereby the attacker (C) attempts to determine which IP addresses are used by A as pseudonyms. With this purpose it broadcasts an ARP Request asking for IP address A, which is received by all nodes at 1 hop; the reply is used to determine the MAC address associated with that particular IP. Afterward it generates ARP Requests for all the pseudonyms it wants to check, including the known pseudonym (in this case IP address D) and other pseudonyms detected earlier. In case an ARP Reply with a same MAC address is used for different pseudonyms, the attacker would find that all addresses are

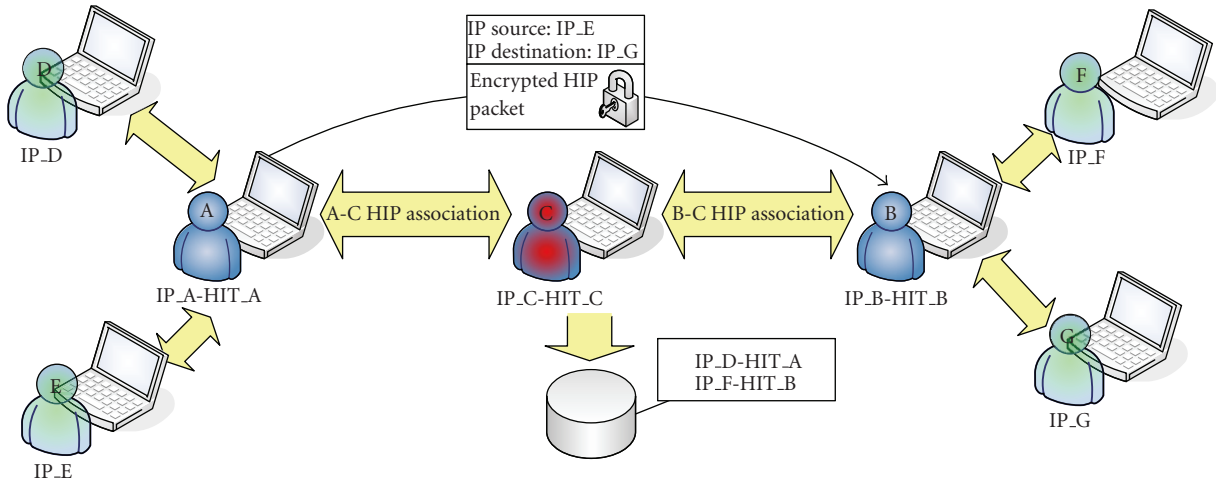

Figure 4: Example of the use of pseudonyms in the scope of HOP.

related to a single node, and thus the anonymity provided by HOP would be compromised.

To avoid the attack described above, our proposal includes a solution that consists in configuring the system's firewall to filter all ARP queries except those for the main IP address, which must be enabled for normal operation. Thus, the main IP address of a node can never be used in the scope of HIP-based communications, being only pseudonym IPs allowed for thus purpose.

7. Security Analysis

The proposed HOP solution is able to address most of the security and anonymity concerns when communicating in a MANET environment. Issues such as confidentiality, message integrity, data origin authentication, connectionless integrity, and protection against replay attacks are provided by the IPsec framework, which works in combination with HOP.

HOP attempts to offer relationship anonymity at the HIP protocol level (between the network and the transport layer), which was our current aim. This was achieved by creating a false topology that can mixes real and pseudonym nodes, which prevents an attacker from determining which IP addresses are being used by each terminal.

Since each terminal is able to adopt multiple identities, the attacker is forced to track the entire path of a packet in order to be able to determine which are the two communicating endpoints. Thus, the actions required for an attacker to get a chance at breaking anonymity are (i) the attacker has to initiate secure connections towards all known users (whose HITs are cached) and simultaneously try to obtain the geographic locations of such users; (ii) the attacker must promiscuously listen to the on-going traffic in the network and locate geographically the sources and destinations of that traffic, which possibly requires being able to gain awareness of all the transmission events that take place in a network; (iii) the attacker will compare the geographic positions of known users (whose HIT is known) against the geographic positions of sources and destinations of traffic being traced, and attempt to guess the IP-to-HIT mapping table, thus breaking anonymity.

Obviously, such an attack is quite complex to undertake, especially when geographical discrimination of users is complex due to their proximity or the presence of obstacles (e.g., indoor scenarios). Moreover, the use of directional antennas by users can also complicate severely the promiscuous listening of all traffic sources.

8. Implementation Details

In this section, we offer details about the different software elements that we had to adapt or improve in order to develop the HOP solution proposed in the previous section. (Our implementation is freely available upon request.) Our target platform is a Linux/Unix system, and our goal is to develop a fully functional testbed to validate our proposal and assess its effectiveness and performance.

8.1. A-HIP Implementation. The first step in our endeavors was to enhance an existing implementation of HIP [28] for Linux/Unix systems in order to implement A-HIP.

In Figure 6 we show a block diagram that illustrates the different elements that conform the HIP service for the reference implementation, along with the interaction with other software components. We highlight in the figure those modules that required enhancements to implement our proposal: *session startup* and *I/O HIP packets*.

Regarding the session startup module, the changes proposed focused on encrypting all HIP messages using the responder's public key to provide anonymity. Concerning the Input/Output module, changes mainly focused on modifying the target of I_1 messages at the IP layer for them to be broadcasted and relayed by intermediate HIP agents. This way they are able to reach the intended message recipient in a multihop network scenario. The HIP daemon (hipd) is composed by the former two modules, together with the session management module that is dedicated to handling

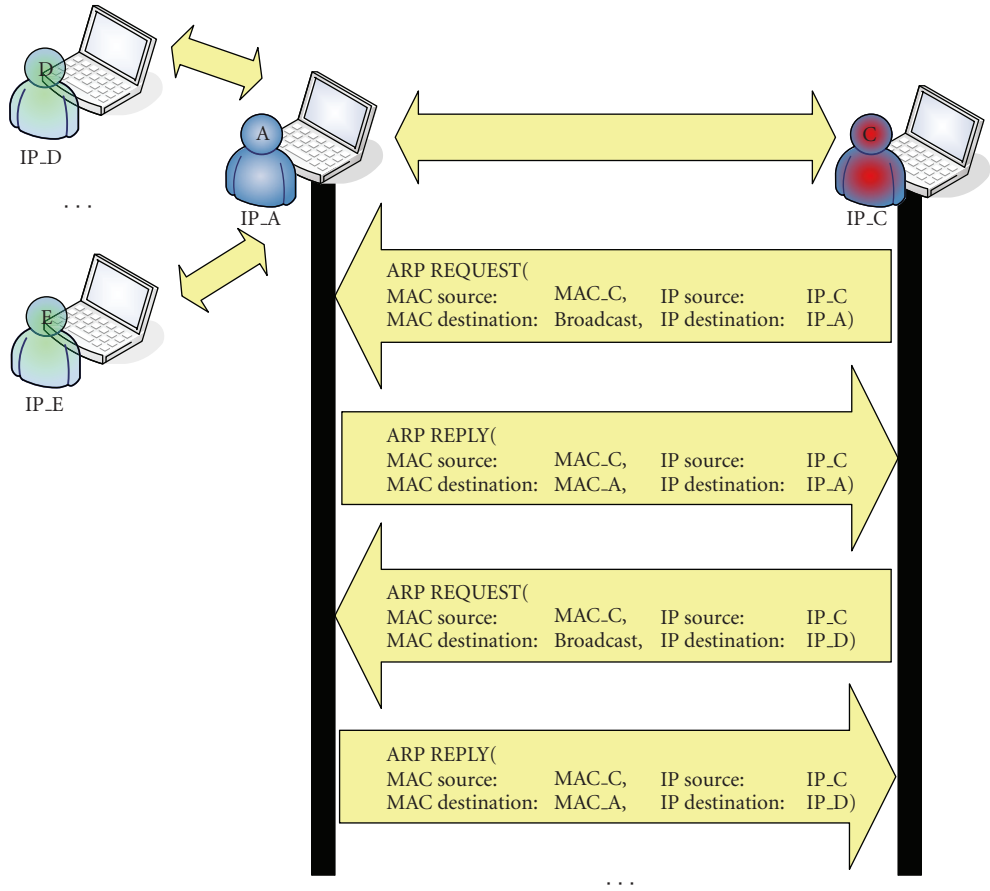

Figure 5: Attack to HOP attempting to disclose hidden pseudonyms through ARP queries.

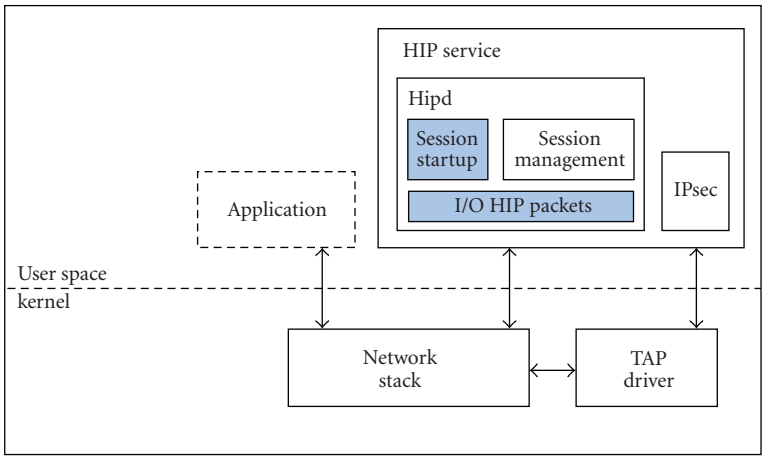

Figure 6: Integration of the HIP service with other Linux software components.

updates and terminating sessions. The HIP service as a whole combines the HIP daemon with the IPsec module to provide the full set of services required.

At the kernel level, a TAP driver communicates with the TCP/IP network stack to allow the HIP service to detect and temporarily block new connections, until a secure channel is established.

Applications attempting to anonymously contact another MANET user can identify the destination by relying on notation <*domain.hip*>. All DNS resolutions in the *.hip* domain are intercepted and handled internally by the HIP service, which must provide a mapping to a specific IP address. In our case the modified HIP session startup phase begins, allowing the establishment of a temporary

mapping between the recipient's HIT and one of its current IP addresses. Data packets are afterward transferred securely using IPsec technology.

8.2. Optimal Dissemination of I_1' Messages Using OLSR. OLSR [20] is a proactive routing protocol for MANETs. As such, it creates and continually maintains routes to all stations participating in the MANET. Route maintenance is achieved by propagating *Topology Control* (TC) messages throughout the MANET using Multipoint Relay (MPR) nodes. These nodes are a subset of the stations participating in the MANET which conforms a minimum spanning tree (MST). Thus, message propagation through MPR rebroadcasting offers high efficiency with little cost in terms of imposed traffic overhead.

The basic A-HIP solution requires the HIP software itself to be in charge of propagating I_1' messages, achieving what is usually called an *application level broadcast* scheme. As an alternative to this solution, we propose an efficient integration of our proposal with OLSR which is achieved by configuring the latter to forward broadcasted I_1' messages. This strategy allows to optimize the flooding process by taking advantage of the MST defined by the different MPR nodes, thus reducing the number of broadcast transmission events in the network as we show in Section 9. Concerning nonbroadcasted messages (R_1', I_2', R_2'), these will be forwarded with minimum latency since OLSR provides paths between all sender and recipient pairs with no delay by constantly maintaining routes.

Table 1 presents detailed information about the packet overhead introduced at session startup. We discriminate the number of transmissions associated with each startup message for the sake of clarity.

In this table, MST_i is the minimum spanning tree defined by the different MPRs as seen by node i, HC refers to the number of hops between sender and recipient, and N refers to the total number of nodes.

With respect to OLSR, a single broadcast flooding is required for the first message. The proposed optimization reduces the total amount of packets transmitted by limiting rebroadcast events to MPR nodes ($N - \text{MST}_i$ packets less). For high node densities in the MANET, this optimization becomes quite relevant compared to the default solution. Taking a typical example where we have a MANET with 50 nodes, with an average hop count of 4 and an average MST size of 10, the high degree of integration with OLSR achieved by our HOP solution allows reducing the number of transmissions from 62 to 22 (a reduction of 65%). Also, notice that HOP allows reaching a near-minimum number of transmission events, which in our example would be of 16.

8.3. Efficient OLSR Support for Pseudonyms. Supporting the proposed HOP solution requires extending the OLSR daemon to make it pseudonym aware. The proposed strategy consisted in simulating the reception of OLSR messages coming from the fake neighbors in the following manner: each node using pseudonyms will be receiving fabricated messages of both HELLO and TC types coming from each of the fake neighbors used as pseudonyms. In particular, we simulate the reception of HELLO messages coming from a pseudonym IP every 5 seconds. These messages will inform about the existence of a symmetric link between the fake node and the real node. Additionally, we simulate that the fake node picks the real node as its MPR, meaning that the latter should propagate any TC messages generated by the former one.

Similarly to HELLO messages, we simulate a TC message reception event coming from the different pseudonym IPs every 5 seconds, which are marked for forwarding. Another optimization that we adopt for our solution is the grouping of several TC messages separated by short intervals of time in a same OLSR packet. Such approach reduces the control overhead in the MANET, and thus the number of times a node has to compete for access to the shared medium.

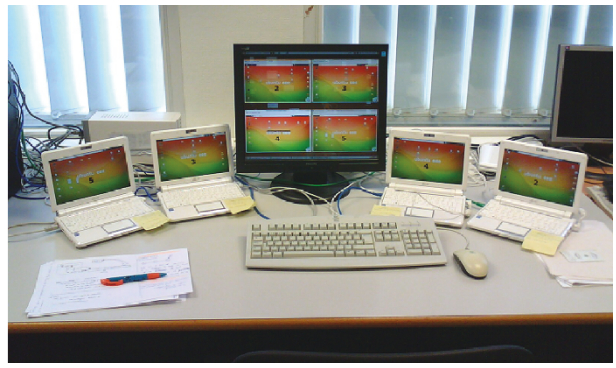

Figure 7: Snapshot of the testbed used for performance evaluation tests.

9. Experimental Results

In this section, we evaluate the performance of our proposal, focusing both on the efficiency of the OLSR integration process and on the overall system performance.

To accomplish this task we set up a small testbed composed of four Asus EeePC 901 netbooks and a desktop PC (see Figure 7). We configured their IEEE 802.11g integrated wireless cards (Ralink RT 2860 chipset [29]) in the ad hoc mode, and we fixed the data rate at 54 Mbit/s. All the terminals involved in the testbed had a GNU/Linux operating system installed, kernel version 2.6.24. The Ralink wireless card drivers used were version 1.7.0.0.

Concerning the OLSR protocol, we used the version made available by the *olsr.org* team [30].

Using the iptables tool [31], we enforced a chain topology that allowed us to assess performance at different hop counts between source and destination. Notice that, in this setup, a manual preloading of the source's cache with the IP address of the destination was required to allow the default HIP implementation contact stations at more than one hop away. Obviously, such approach is not required for HOP.

Table 1: Packet overhead introduced at session startup.

Mode of operation	Number of Tx per message				Total number of Tx events
	I_1'	R_1'	I_2'	R_2'	
A-HIP	N	HC	HC	HC	$N + 3 \times HC$
HOP	MST_i	HC	HC	HC	$MST_i + 3 \times HC$
Ideal solution	HC	HC	HC	HC	$4 \times HC$

9.1. OLSR Control Overhead. In this section, we analyze the control overhead imposed by the OLSR protocol, as well as the additional overhead introduced by the use of pseudonyms in our HOP solution.

According to the OLSR specification, each node should generate HELLO messages every 2 seconds and TC messages every 5 seconds, that is, about 0.7 messages per second. Additionally, some of the nodes should propagate TC messages received from their neighbors if they were chosen to be their MPRs. With the HOP solution, each node should additionally forward one TC message per each pseudonym used.

Below we study the total number of OLSR packets generated, as well as their sizes. The results are obtained from a 60-second period, and after the network topology converged to a steady state. The scenarios analyzed are the following:

(i) Scenario #1: Standard OLSR,

(ii) Scenario #2: HOP using 2 pseudonyms per node,

(iii) Scenario #3: HOP using 3 pseudonyms per node.

Figure 8 shows the number of messages for the three different scenarios under analysis. Notice that, in all scenarios, the sum of the HELLO and TC messages generated is always less than the total number of OLSR messages generated. For the standard OLSR solution (scenario #1) this occurs because all messages queued within a short time interval are encapsulated in a same OLSR packet. For scenarios #2 and #3 the number of TC messages generated is much higher due to the presence of pseudonyms. However, HOP synchronizes the generation of the TC messages belonging to the different pseudonyms, thus allowing them to be encapsulated within a same OLSR packet. This explains why, despite the significant increase in terms of TC messages, the actual number of OLSR messages increases only slightly.

We now focus in detail on the sizes of the messages generated. According to the OLSR RFC, the header size for both TC and HELLO messages is of 12 bytes. The body of TC messages varies according to the number of neighbors of each particular node, occupying 4 bytes (minimum) plus 8 bytes per neighbor. In the case of HELLO messages, the size depends not only on the number of neighbors, but also on the neighbor characteristics (e.g., link symmetry, MPR selection). Thus the number of bytes ranges from 8 to 12 bytes per neighbor, in addition to other data. Figure 9 shows how the sizes of the TC and HELLO messages as we increase the number of neighbors (real or pseudonyms) in the network. For HELLO messages we include the best- and worst-case sizes. For our purposes, we find that message sizes will be generally small, being that several can fit into a same packet, as desired.

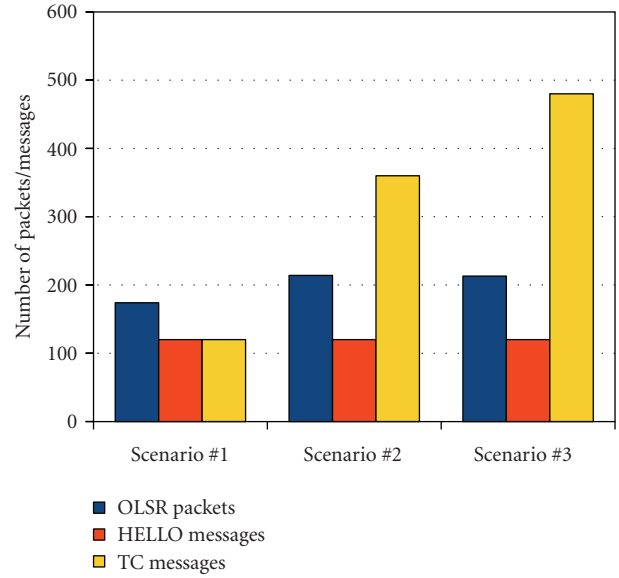

Figure 8: Details about the number of HELLO and TC messages created in the three different scenarios, along with the actual number of OLSR packets injected into the network.

Table 2: Data rate and network utilization for OLSR traffic in different scenarios.

	Mean data rate	Mean network utilization
Scenario #1	2.2 kbit/s	0.010%
Scenario #2	4.6 kbit/s	0.021%
Scenario #3	5.5 kbit/s	0.026%

To conclude our analysis of the control overhead associated with OLSR and HIP, we have calculated both the mean data rate generated by all sources and the mean network utilization for our 4-hop chain network when assuming a channel capacity of about 20 Mbit/s. These results, which are summarized in Table 2, confirm that the OLSR network utilization is insignificant, and that the pseudonyms introduced by the HOP solution do not represent a meaningful traffic increase. In fact network utilization is maintained below 0.03% in all cases.

9.2. Overall Performance Evaluation. We now proceed to evaluate the performance of the HOP proposal. First we will assess the increase in terms of session startup times when comparing HOP to the default HIP solution (without

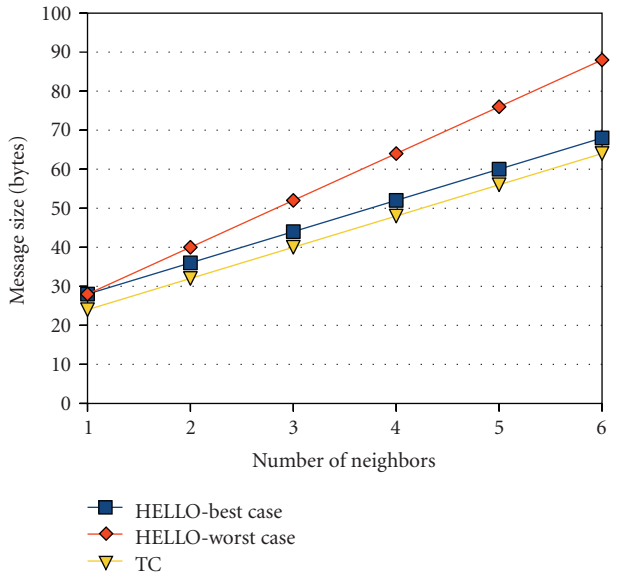

Figure 9: HELLO and TC message sizes when varying the number of neighbors.

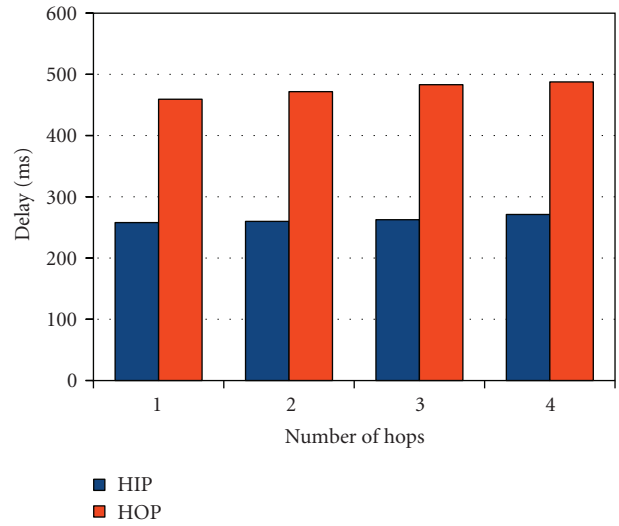

Figure 10: Session startup times at different hop counts for the default HIP and the proposed A-HIP solution.

Table 3: Encryption and decryption times for the different HOP packet types.

HIP Packet type (size)	Encryption time (ms)	Decryption time (ms)
I_1' (40 bytes)	0.56	1.31
R_1' (608 bytes)	3.7	74.8
I_2' (659 bytes)	4.6	96.0
R_2' (216 bytes)	1.2	47.8
Close/Close ack (208 bytes)	1.2–1.9	22.1

anonymity). Afterward we analyze the average end-to-end delay and throughput comparing HOP to an insecure solution (No HIP), which is used as a reference to determine the security/performance tradeoff.

The session startup time is the time it takes to create a HIP association, and it is measured as the time elapsed from the moment the user requests the association until packet R_2 is received and fully processed. The establishment of a HIP session requires four packets to be interchanged: I_1, R_1, I_2, and R_2, which involves both a computation time and a network delay. In our HOP solution all HIP packets are encrypted, meaning that these times will be added to the session startup time for the original version.

In Table 3 we summarize the time required to encrypt and decrypt the different packet types used by HOP. We include session close packets also for the sake of completeness. The overhead differences in time are mainly due to the packet size differences. As can be seen, I_1' packets are the smallest, I_2' packets being the largest (because they carry, among other things, a puzzle to be solved by the session initiator).

We now proceed to show the global impact of these encryption and decryption processes in the proposed testbed. For all the charts shown in this section, we can state that the maximum error for all the values represented (mean values) is below 5%, with a degree of confidence of 95%. Figure 10 shows the overhead introduced by HOP compared to the default HIP implementation as we increase the hop count between the source and destination terminals. We can see that the extra encryption effort required to offer anonymity introduces an additional delay between 200 and 220 ms. Since this overhead is limited to the initial exchange, we consider that the tradeoff achieved is reasonable, the startup time being within acceptable bounds from a user perspective. Also notice that the number of hops does not negatively affect HOP, the increase being minimal as for the default HIP case.

In the experiments that follow we compare HOP against an insecure solution. We do not include the results for the *Default HIP* case since the performance values obtained do not differ from the ones achieved with HOP.

Figure 11 shows the mean round-trip time (RTT) delay for different payload sizes for both HOP and *No HIP* (insecure) solutions. We find that the additional processing required by HOP causes the RTT to increase between 0.4 and 1.2 ms, being on average of 0.7 ms. Notice that the relative impact of this increase tends to disappear as the number of hops increases—from 97% (1 hop) down to 23% (4 hops)—since the additional overhead imposed is independent of the number of hops.

In terms of throughput, Figure 12 shows that the maximum throughput achievable with the proposed solution is of about 12.5 Mbit/s. This upper limit is inherent to the CPU-bounded characteristics of the encryption processes, and can only be improved by using faster CPUs or specialized encryption hardware (notice that we are using low-cost EEEPCs). As we increase the number of hops we observe that the performance drop caused by using encryption is quite

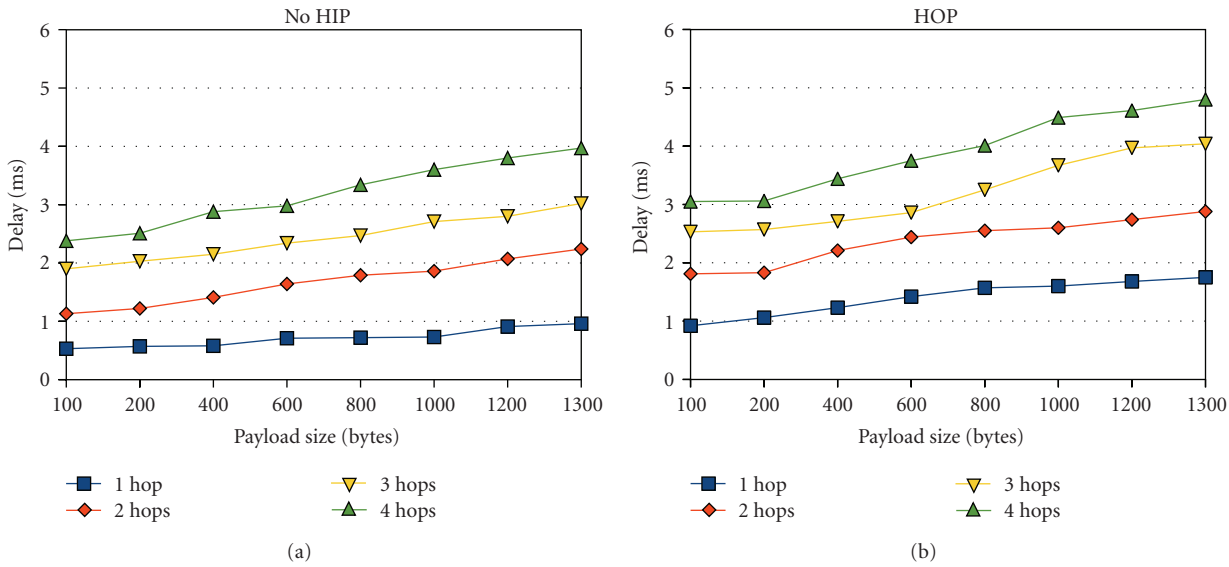

Figure 11: Round-trip time delay for different payload sizes in an insecure mode (a) and using HOP (b).

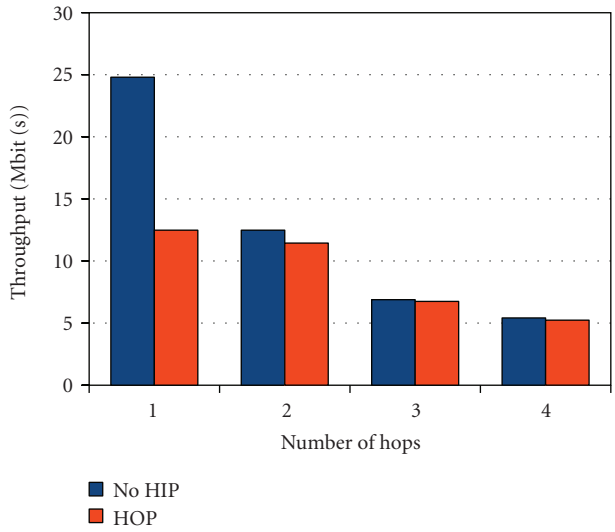

Figure 12: Throughput at different hop counts in an insecure mode (No HIP) and using A-HIP.

limited, achieving throughput values close to those without encryption. Therefore, in a typical MANET environment, the impact on throughput will not be relevant, and it will certainly not compromise the transmission efficiency.

9.3. Assessing the Impact of Mobility on Performance. As described above, we validated and assessed the performance of HOP using a real testbed. However, our tests were limited to only a few nodes and no mobility. In this section, we assess the performance of HOP in mobile environments and with a higher number of nodes. To achieve this goal we relied on the ns-2 [32] simulation tool. In particular we picked a scenario sized 10001000 meters, where 60 nodes are constantly moving towards random destinations at a speed of 5 m/s. The wireless interfaces transmit at 54 Mbit/s using IEEE 802.11g technology, and the radio range is of 250 meters. The routing protocol used is OLSR tuned with an HELLO interval of 2 seconds and a TC interval of 5 seconds. Concerning traffic, we vary the number of source-destination pairs, where each transmits CBR/UDP traffic at a rate of 50 packets per second, using a packet payload size of 512 bytes. We tuned the behavior of the system so that the HIP-related packet interchanges and the encryption delays introduced by HOP and IPsec were taken into account, being similar to the values obtained in the real tests presented before. Also, each participating node adopts three different pseudonyms.

The results obtained are shown in Figures 13 and 14. Notice that, for the delay and routing overhead results presented in the shown charts, we have a 90% confidence that the mean is within ±10% of the values represented. For the packet delivery ratio we have a 95% confidence that the mean is within ±5% of the values represented.

In terms of packet delivery ratio, Figure 13(a) shows that HOP introduces a small decay in performance. This is expected due to the higher overall load in the network in terms of control traffic (see below), which increases channel contention and subsequently the number of collisions. Compared to the testbed experiments presented in the previous section using a static scenario, we find that mobility has a significant impact on the packet delivery ratio for both solutions tested since both rely on OLSR for route discovery and route maintenance tasks. In terms of delay, Figure 13(b) shows that the increase is moderate and remains within strict bounds; remember that the highest delays take place at connection setup, and so all the traffic that follows merely experiences the delay introduced by IPsec's symmetric encryption process, which is typically less than 1 millisecond.

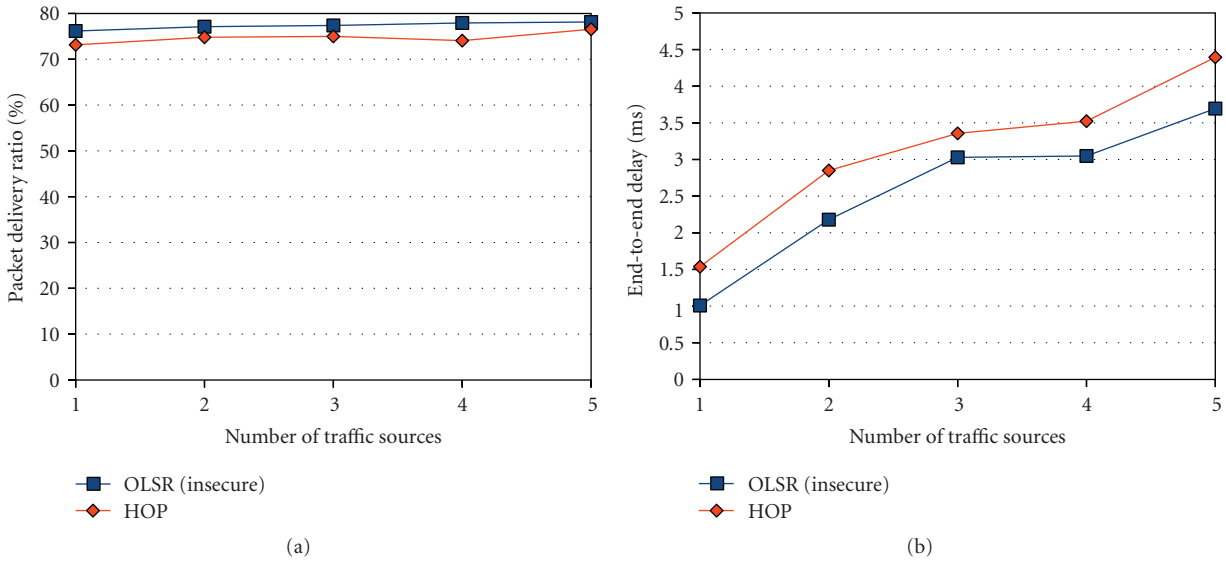

Figure 13: Packet delivery ratio (a) and end-to-end delay values (b) for OLSR and the HOP solution when increasing the aggregated load in the system.

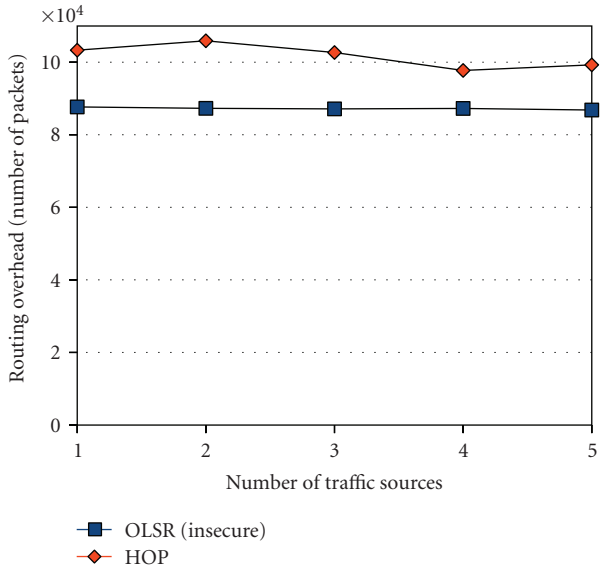

Figure 14: Routing overhead for OLSR and the HOP solution when increasing the aggregated load in the system.

In terms of routing overhead we find that, as expected, HOP increases the amount of control packets injected into the network (see Figure 14). Compared to OLSR, which maintains a stable routing overhead value due to the proactive nature of the protocol, the overhead associated with HOP decays slightly for higher load values due to losses.

Overall, we find that the impact of mobility on performance affects equally an insecure MANET based on OLSR and an enhanced solution based on HOP, the performance of the latter always being slightly lower although comparable to the former one. Thus, we consider that the simulation-based analysis of HOP sustains the conclusions drawn before in our testbed experiments, showing that the security and anonymity strategy adopted by HOP does not provoke any significant performance degradation to the system, the tradeoff being achieved between performance and security/anonymity a quite reasonable one.

According to different studies [22, 23], these performance values are better than those achieved by other MANET anonymity solutions. In particular, Marga et al. [23] show that delay values for the ANODR protocol [6] vary from 0.3 to 1.5 seconds whereas for the MASK protocol [7] most delay values are between 1 and 2 seconds, both significantly higher than the values achieved by our solution. Liu et al. [22, 23] further show that, compared to a insecure solution (best case in terms of performance), most approaches found in the literature suffer from excessive delays (often one order of magnitude higher than an insecure solution) and a delivery ratio up to 20% lower compared to an insecure solution.

10. Conclusions

In this paper, we proposed a novel solution to provide private and untraceable communication between MANET peers. Compared to previous proposals, we were pioneer at actually implementing and testing our solution in a real testbed.

We relied on the concept of HITs to offer user discovery and end-to-end encryption of data through full integration with HIP and IPsec technology. One of the main benefits of our proposal is of being lightweight and easily implementable in real-life operating systems, as demonstrated in the paper.

To achieve a high degree of anonymity, our HOP proposal combines pseudonyms with an anonymous version of the HIP protocol (A-HIP) that we also developed. Additionally, the use of pseudonyms is efficiently integrated with the OLSR protocol to maximize performance.

We implemented the proposed solutions and evaluated them in our testbed. Experimentally we show that the integration with OLSR is quite efficient, being that pseudonyms do not impose a significant overhead to the network. We also find that the encryption process required for anonymity during session startup time introduces an additional delay between 200 and 220 ms, the total time always being below 500 ms. In a MANET environment, such initial delay is not considered restrictive.

In terms of delay and throughput, HOP offers the same performance as the original HIP implementation. Compared to an insecure solution, delay and throughput values merely experience a very slight increase. The only exception was detected for one-hop distances, where the maximum data encryption rate was limited to 12 Mbit/s for the hardware used in the experiments.

Overall, we presented a solution providing secure and anonymous communications in MANET environments that was validated in a real testbed, and that achieved good performance levels. We also used simulation to determine the effectiveness of the proposal in the presence of mobility, showing that the performance of our anonymity solution remains similar to that of an insecure solution when adopting OLSR as the routing protocol.

Acknowledgments

This work was partially supported by the *Ministerio de Educación y Ciencia*, Spain, under Grant TIN2008-06441-C02-01, and by the Generalitat Valenciana under Grant GV/2009/010.

References

[1] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey on attacks and countermeasures in mobile ad hoc networks," in *Wireless/Mobile Network Security*, Springer, New York, NY, USA, 2006.

[2] Y.-C. Hu and A. Perrig, "A survey of secure wireless ad hoc routing," *IEEE Security and Privacy*, vol. 2, no. 3, pp. 28–39, 2004.

[3] D. Chaum, "The dining cryptographers problem: unconditional sender and recipient untraceability," *Journal of Cryptology*, vol. 1, no. 1, pp. 65–75, 1988.

[4] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[5] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[6] J. Kong and X. Hong, "ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '03)*, pp. 291–302, New York, NY, USA, June 2003.

[7] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "MASK: anonymous on-demand routing in mobile ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 9, pp. 2376–2385, 2006.

[8] X. Lin, R. Lu, H. Zhu, P. -H. Ho, X. Shen, and Z. Cao, "ASRPAKE: an anonymous secure routing protocol with authenticated key exchange for wireless ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 1247–1253, 2007.

[9] J. H. Paik, B. H. Kim, and D. H. Lee, "A3RP: anonymous and authenticated ad hoc routing protocol," in *Proceedings of the 2nd IEEE International Conference on Information Security and Assurance (ISA '08)*, pp. 67–72, April 2008.

[10] M. Nácher, C. T. Calafate, J. C. Cano, and P. Manzoni, "An overview of anonymous communications in mobile ad hoc networks," *Wireless Communications and Mobile Computing*. In press.

[11] K. El Defrawy and G. Tsudik, "Prism: privacy-friendly routing in suspicious manets (and vanets)," in *Proceedings of the IEEE International Conference on Network Protocols*, Orlando, Fla, USA, October 2008.

[12] E. H. J. Kumari and A. Kannammal, "Privacy and security on anonymous routing protocols in manet," in *Proceedings of the 2nd International Conference on Computer and Electrical Engineering*, Dubai, UAE, December 2009.

[13] S. Seys and B. Preneel, "ARM: anonymous routing protocol for mobile ad hoc networks," *International Journal of Wireless and Mobile Computing*, vol. 3, no. 3, pp. 145–155, 2009.

[14] X. Li, H. Li, J. Ma, and W. Zhang, "An efficient anonymous routing protocol for mobile ad hoc networks," in *Proceedings of the 5th International Conference on Information Assurance and Security (IAS '09)*, vol. 2, pp. 287–290, Xi'an, China, August 2009.

[15] J. Ren, Y. Li, and T. Li, "Spm: source privacy for mobile ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, Article ID 534712, 10 pages, 2010.

[16] Y. Dong, T. W. Chim, V. O. K. Li, S. M. Yiu, and C. K. Hui, "ARMR: anonymous routing protocol with multiple routes for communications in mobile ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1536–1550, 2009.

[17] S. Chen and M. Wu, "Anonymous multipath routing protocol based on secret sharing in mobile ad hoc networks," in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '10)*, vol. 1, pp. 582–585, Changsha, China, March 2010.

[18] A. Pfitzmann and M. Hansen, "Anonymity, unobservability, and pseudonymity—a proposal for terminology," in *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*, pp. 1–9, 2000.

[19] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol," RFC 5201, April 2008.

[20] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR). Request for Comments 3626,MANET Working Group," Work in progress, October 2003, http://www.ietf.org/rfc/rfc3626.txt.

[21] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," IETF RFC 4301, December 2005.

[22] J. Liu, J. Kongt, X. Hong, and M. Gerla, "Performance evaluation of anonymous routing protocols in MANETs," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '06)*, pp. 646–651, New Orleans, La, USA, April 2006.

[23] M. Nácher, C. T. Calafate, J. C. Cano, and P. Manzoni, "Anonymous routing protocols: impact on performancein MANETs," in *Proceedings of the IEEE International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS '09)*, London, UK, September 2009.

[24] Internet Engineering Task Force, "Host identity protocol working group charter," http://www.ietf.org/html.charters/hip-charter.html.
[25] H. Krawczyk, "SIGMA: the 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in theIKE protocols," in *Proceedings of the International Conference on Cryptology (CRYPTO '03)*, Springer LNCS Advances in Cryptography, pp. 400–425, Santa Barbara, Calif, USA, August 2003.
[26] IEEE 802.15.1(tm) IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs(tm)), 2002.
[27] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.
[28] OpenHIP, http://downloads.sourceforge.net/openhip/hip-0.5.tgz..
[29] Ralink Technology Corporation, January 2009, http://www.ralinktech.com/.
[30] A. Tonnesen et al., "Olsrd: an ad hoc wireless mesh routing daemon," http://www.olsr.org.
[31] The netfilter.org iptables project, January 2009, http://www.netfilter.org/.
[32] K. Fall and K. Varadhan, "ns notes and documents," The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2000.