Distributed User Authentication in Wireless LANs

Dmitriy Kuptsov, Andrey Khurri, and Andrei Gurtov
Helsinki Institute for Information Technology
Helsinki University of Technology
{dmitriy.kuptsov, andrey.khurri, gurtov}@hiit.fi

***Abstract*—An increasing number of mobile devices, including smartphones, use WLAN for accessing the Internet. Existing WLAN authentication mechanisms are either disruptive, such as presenting a captive web page prompting for password, or unreliable, enabling a malicious user to attack a part of operator's infrastructure. In this paper, we present a distributed authentication architecture for WLAN users providing instant network access without manual interactions. It supports terminal mobility across WLAN access points with the Host Identity Protocol (HIP), at the same time protecting the operator's infrastructure from external attacks. User data sent over a wireless link is protected by the IPsec ESP protocol. We present our architecture design and implementation experience on two OpenWrt WLAN access points, followed by measurement results of the working prototype. The system is being deployed into pilot use in the city-wide panOULU WLAN.**

I. Introduction

An increasing number of laptop and smartphone users utilize WLANs for Internet access at work, home, and public places. Unfortunately, authentication mechanisms in WLANs remain cumbersome, unreliable and disruptive to users. Typically, users are required to re-enter their login and password periodically through a captive web page, or manually configure the WPA keys or 802.1X settings. Owners of open WLANs risk to fall under police investigation in case of network misuse.

Several trends make the situation harder with time. Some emerging devices, such as smart key chains, are being equipped with WLAN capability, although missing a screen to display and enter login information. Furthermore, recent advances in breaking WPA encryption [1] and 802.1X [2], render WLAN users vulnerable and call far robust IP-layer encryption over the wireless link.

Several potential solutions to the above problem have been proposed (e.g., [17], [4]). However, none of them achieves all of the following properties: 1) disruption-free user authentication 2) protection of the operator's infrastructure from external attacks 3) host mobility and multihoming 4) IPsec encryption over the wireless link.

The Host Identity Protocol (HIP) is a new security and mobility protocol standardized by the IETF [21], [22], [13], [20], [19], [24], [23], [25]. HIP uses IPsec ESP encryption of data and a version of Diffie-Hellman protocol to exchange public keys of two hosts.

In this paper, we propose a HIP-based distributed WLAN authentication architecture. It combines an operator-specific HIP proxy with a HIP-aware firewall running on each of the operator's WLAN access points (APs). The architecture satisfies all of the requirements listed above. Mobile users can instantly gain WLAN network access and be authenticated at any physical location within the operator's network. User data over the wireless link is encrypted and the operator's infrastructure is protected from external attacks.

To test our architecture, we have ported HIPL (HIP for Linux) implementation to run on the OpenWrt WLAN APs. In this paper, we present our architecture design, experimental experience and analysis of the performance results. Performance evaluation suggests that a two-level approach consisting of a single HIP proxy server and a distributed HIP firewall is appropriate, given limited hardware resources of the most WLAN APs. The architecture is being deployed in panOULU [3], a public WLAN in the city of Oulu, Finland.

The rest of the paper is organized as follows. In Section II, we give a short summary of HIP. Section III shows the motivation behind our research and outlines the related work. Section IV presents our distributed WLAN authentication architecture. Section V describes the port of the HIPL implementation to the OpenWrt platform and an experimental testbed used for our measurements. Section VI contains selected measurement results of CPU and memory utilization on two different WLAN AP models, as well as packet filtering time on a HIP-aware firewall. We discuss future work in Section VII, whereas Section VIII concludes the paper with a summary of key findings.

II. Background on Host Identity Protocol

The existing Internet architecture designed for stationary hosts nowadays faces many non-trivial challenges with the growing amount of mobile terminals. Currently, there are two namespaces used globally by the Internet hosts and applications, domain names and IP addresses. IP addresses serve the dual role in the Internet being both end host identifiers and topological locators. This principle does not allow hosts to change their location without breaking ongoing transport connections bound to IP addresses.

[1]Researchers Crack WPA Wi-Fi Encryption, http://it.slashdot.org/article.pl?sid=08/11/06/1546245

[2]802.1X vulnerabilities, http://www.microsoft.com/technet/community/columns/secmgmt/sm0805.mspx

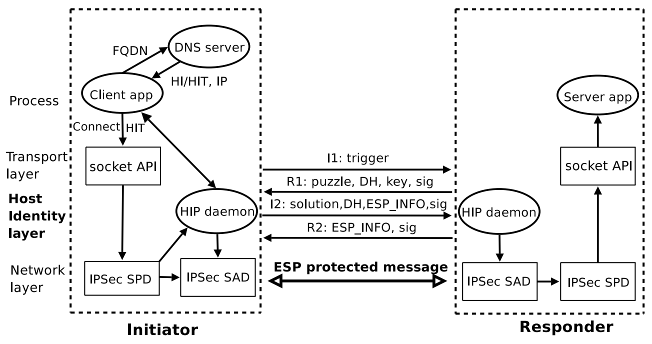

Fig. 1. HIP architecture.

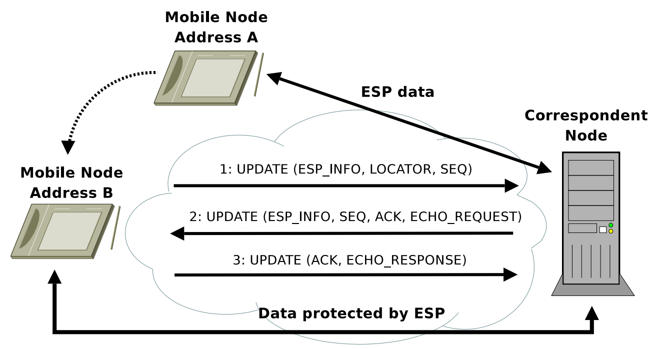
Mobile Node Address A
ESP data
Correspondent Node
Mobile Node Address B
1: UPDATE (ESP_INFO, LOCATOR, SEQ)
2: UPDATE (ESP_INFO, SEQ, ACK, ECHO_REQUEST)
3: UPDATE (ACK, ECHO_RESPONSE)
Data protected by ESP

Fig. 2. HIP mobility update.

A. HIP Architecture

The Host Identity Protocol (HIP) [21], [6] was proposed to overcome the problem of using IP addresses for host identification and packet routing. The idea behind HIP is based on decoupling the network layer from the higher layers in the protocol stack architecture (see Figure 1). HIP defines a new global name space, the Host Identity name space, thereby splitting the double meaning of IP addresses. When HIP is used, upper layers do not anymore rely on IP addresses as host names. Instead, Host Identities are used by a transport protocol for establishing connections. IP addresses at the same time act purely as locators for routing packets towards the destination. For compatibility with IPv6 legacy applications, Host Identity is represented by a 128-bit long hash, the Host Identity Tag (HIT).

HIP offers several benefits including end-to-end security, resistance to CPU and memory exhausting denial-of-service (DoS) attacks, NAT traversal, mobility and multihoming support.

B. HIP Base Exchange

To start communicating over HIP, two hosts have to establish a HIP association. This process is known as the HIP Base Exchange (BEX) [22] and it consists of four messages between the initiator and the responder. After BEX is successfully completed, both hosts are confident that private keys corresponding to Host Identifiers (public keys) are indeed possessed by their peers. Another purpose of the HIP base exchange is to create a pair of IPsec Encapsulated Security Payload (ESP) Security Associations (SAs), one for each direction. All subsequent traffic between communicating parties is protected by IPsec. A new IPsec ESP mode, Bound End-to-end Tunnel (BEET) [25], is used in HIP. The main advantage of BEET mode is low overhead in contrast to the regular tunnel mode.

Figure 1 illustrates the overall HIP architecture including the BEX. The initiator can retrieve the HI/HIT of the responder from a DNS directory [24] by sending a FQDN in a DNS query. Instead of resolving the FQDN to an IP address, the DNS server replies with an HI (FQDN→HI). The transport layer creates a packet with the HI as the destination identifier. During the next step, HI is mapped to an IP address by the HIP daemon on the Host Identity layer. Finally, the packet is processed on the network layer and delivered to the responder. As a result, the conventional 5-tuple socket becomes {protocol, source HI, source port, destination HI, destination port}.

C. Mobility and Multihoming

Since neither transport layer connections nor security associations (SAs) created after the HIP base exchange are bound to IP addresses, a mobile client can change its IP address (i.e., upon moving, due to a DHCP lease or an IPv6 router advertisement) and continue transmitting ESP-protected packets to its peer. HIP supports such mobility events by implementing an end-to-end three-way signaling mechanism [23] between communicating nodes (see Figure 2).

The purpose of the first UPDATE packet is to notify the peer of a new IP address and an ESP information associated with this address. The corresponding parameters are called LOCATOR and ESP_INFO. The message also contains a SEQ parameter (a sequence number of the packet) and is therefore protected against possible losses by retransmissions. Upon receiving the UPDATE message, the peer host must validate it, update any local HI↔IP mappings and assure that the mobile client is accessible via the new link. This is accomplished by sending the second UPDATE packet back to the mobile host at its new IP address containing an echo request along with the ESP_INFO of the peer. Finally, the mobile client is expected to acknowledge the message from its peer and return the content of the echo message. When the peer host gets this response, the new IP address of the client is marked as verified and the update procedure is completed [23].

A multihomed host may have multiple IP addresses associated with a network interface or even several interfaces attached to different access points. HIP provides an opportunity for a multihomed host to inform its peers about available addresses using UPDATE messages described above. The peer hosts update the appropriate HI↔IP bindings and verify each of the IP addresses of the host by sending echo requests and waiting for correct replies. HIP multihoming uses the same mechanisms as mobility for updating the peer with a current set of IP addresses of the host [23].

III. Motivation and Related Work

A traditional access model in an open wireless network (such as presented in Figure 3) essentially lacks security

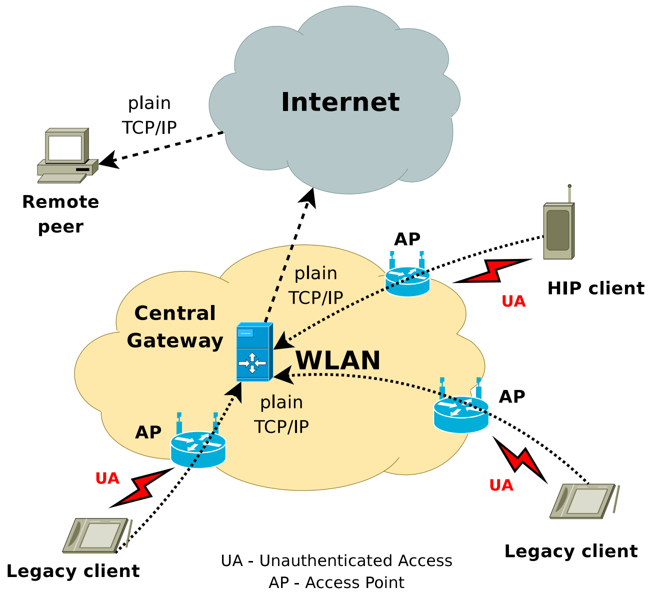

Fig. 3. Open network access model.

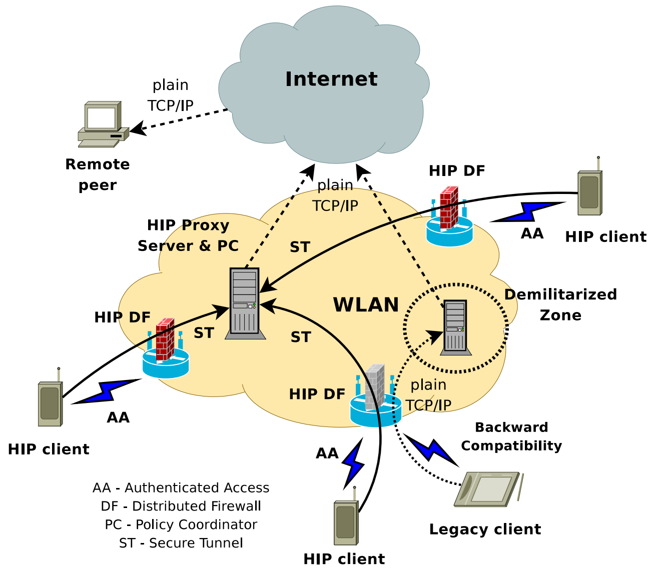

Fig. 4. Distributed authentication model.

features. Such networks usually have no mechanisms for user access control, protection of data integrity and confidentiality.

The core of the problem is that anyone can gain access to the network without providing and validating their identity. This allows an attacker to perform illegal actions and potentially cause damage to the network infrastructure without being caught. On the other hand, publicly available WLANs usually use no encryption, hence all the traffic transmitted over the air can be easily sniffed, analyzed and used for malicious purposes. To eliminate such risks, we need reliable data protection and access control mechanisms. Below we describe several research projects that are relevant to our proposal.

An architecture for secure and mobile Wi-Fi sharing is proposed by the PISA project [9], [7]. PISA architecture eliminates well-known security risks and attacks in open wireless networks. PISA also provides a solution for client authentication and mobility, which is similar to what we propose in this article. On the other hand, PISA focuses on serving a slightly different role in the Internet, namely implementing secure access control for global Wi-Fi sharing communities. Similar work [5] considers distributed authentication, authorization and accounting (AAA) in community Wi-Fi networks. It focuses on building trust chains between the communicating entities using certificates and certificate authorities (CA). Later valid certificates serve for authenticating the clients, as well as for identifying and validating the decentralized AAA servers.

The concept of a distributed firewall is not new but existed for a long time. Studies [27] and [12] discuss the advantages of a distributed firewall over centralized firewall in changing network topologies. The papers describe key points in implementing a distributed firewall, including a mechanism to enforce network security policy through a policy language. Additionally, a distribution scheme and an authentication technique for network entities participating in the policy enforcement process are presented.

Source address validation is another topic related to our work. Source address validation architecture (SAVA) [31], [30], [29] addresses the problem of source address spoofing on different levels from a local subnetwork to autonomous systems.

IV. Distributed Authentication Architecture

In this section we present our approach for automatic WLANs authentication according to design goals stated in Section I.

A. Architectural Components and Principles

The general view of our architecture is shown in Figure 4. We propose to utilize HIP as a backbone that supports client mobility and multihoming in addition to WLAN authentication. A HIP-enabled mobile client establishes a secure association with a central HIP proxy installed on a default gateway in the core network. User data is then protected by the ESP secure tunnel. HIP and IPsec associations are updated when the client moves to another location within the network. Another role of the central HIP proxy is to enable connections from mobile HIP clients to remote servers in the Internet that do not understand HIP. In a simple scenario, we deployed an HTTP/HIP proxy, so that HIP clients can connect to non-HIP Web servers and securely browse WWW pages.

To solve the authentication problem, we introduce a set of interconnected HIP-aware firewalls, the *distributed firewall* (see Figure 4). The main purpose of a HIP-aware firewall is to filter traffic based on a predefined list of allowed Host Identity Tags (HITs) that authenticate clients to the operator. Additionally, the firewall can perform a digital signature check (as, for instance, in PISA project [9]).

Checking the signatures provides a higher authentication level but involves cryptographic operations. The overhead can negatively affect the overall data throughput of the firewall and can reduce the number of clients served with reasonable Quality of Service (QoS).

We believe that HIT-based filtering can be sufficient for a WLAN network when a client connects to the Internet through the central HIP proxy. Even if an attacker is able to generate a valid HIT, it would fail to complete the HIP base exchange after receiving an R1 packet (due to lack of knowledge of a private key). After a HIP association is established, ESP traffic is filtered using SPI values stored from the HIP base exchange.

However, consider a scenario when a mobile client wants to communicate with another mobile client in the same WLAN network. In this case, an attacker has chances to replay the HIP control and ESP packets (sent between two mobile hosts) and establish a communication with another attacker within the same network, thus using the network resources on behalf of the legitimate clients. To eliminate such risks we suggest to use an extension for client authentication and authorization at the middleboxes proposed by Heer et al [8]. In our architecture, the Wi-Fi APs would play the role of the middleboxes that can authenticate HIP and ESP packets transmitted between two mobile clients in the same WLAN.

B. Synchronization of Firewalls

Distributed HIP firewalls are placed on the edge of the wireless network and perform packet filtering based on the predefined access control list (ACL). In other words, traffic from a registered HIT can successfully pass through the firewall and flow to the core network. In such architecture all participating Wi-Fi APs (HIP-aware firewalls) should maintain a fresh copy of the rules, so that a newly registered customer can pass authentication successfully anywhere within WLAN network coverage.

Synchronization of access control lists should be efficient. The lists should be updated frequently without flooding the network with signaling traffic. In this paper we are not proposing any specific protocol for synchronization but offer a general architecture overview and make a few suggestions on synchronizing the ACLs between the firewalls. In particular, we consider the following two approaches.

- First, a firewall can query the central policy coordinator server on-demand (when no matching rule is found locally) or at fixed intervals (this approach will cause a higher network load). A request will update the list of allowed HITs. In this case, the firewall needs to store the complete ACL locally.
- Second, the AP firewalls can form a peer-to-peer network (for instance, using a DHT). Each AP would store a partial list of allowed HITs and perform on-demand queries to the overlay if the matching rule is not found locally.

For the first approach, we assume that a centralized policy coordinator is present in the network (in Figure 4 it is placed on the gateway). Such policy coordinator holds the current list of allowed HITs (or a user registration database). A simple ACL synchronization protocol is exemplified in Algorithm 1.

Algorithm 1 ACL synchronization algorithm.

```
Require: $certificate \neq NULL$
Require: $address_{server} \neq NULL$
  if $authenticate(certificate, address_{server}) = TRUE$ then
    $updateACL(address_{server})$
    $resetStatsForNewEntries()$
    $sortACLbyStats()$
  else
    $reportError()$
  end if
```

To synchronize the ACL, an AP is required to authenticate itself to the central server with a certificate, or using other available mechanisms. Upon successful registration, the AP merges the ACL with the new updates.

C. Rule Management

We observed that packet filtering time on a Wi-Fi AP firewall depends on the position of the appropriate rule in the ACL. Matching of the packet takes $\Theta(n)$ time in the worst and $O(1)$ in the best case, where n is the number of rules in the ACL. Based on the initial experimental results (see Section VI-A), it becomes obvious that some rule management techniques should be employed to achieve better filtering performance. A simple strategy to sort the rules in the ACL may involve collecting per-packet statistics, and then trying first the most frequent rules.

An alternative solution can be a hash table that guarantees $O(1)$ search time. However, this will constrain the flexibility of the rules and restrict the search criteria to only one key, e.g., the source HIT. Such approach might successfully work in a simple setup, but more flexible systems would require a more comprehensive algorithm. The hash table approach might be infeasible if the number of valid remote servers the user can access is limited. In this case, each rule in the ACL needs to have a destination HIT. However, a hash table cannot provide filtering based on multiple criteria.

Yet another approach for matching the rules is using tries and ternary trees. We do not bound our architecture to any particular method for managing the rules. We have suggested a few different approaches and left their detailed evaluation for future research.

D. Service Subscription

Prior to a first-time connection to an operator's WLAN, a client in our architecture is supposed to perform a registration or to subscribe to the service in a secure way. The purpose of such registration is to authenticate the user to the operator and store the mapping between the user identity and HIT in a registration database, which is synchronized among all firewalls. In practice, there might be several alternative ways to accomplish this procedure, including registration in person at an office by providing an identity card or in the Internet using a banking authentication service [18].

E. Compatibility with Legacy Clients

A large-scale deployment of our architecture can require a transition phase, when not all mobile clients will understand HIP. In this section, we consider two possible approaches to provide backward compatibility for legacy clients in early deployment stages. Both approaches require support of legacy and HIP-enabled clients in the WLAN APs.

In the first approach, we can run a HIP proxy on the WLAN access points. This proxy would provide support for non-HIP (legacy) clients by translating plain TCP/IP to HIP and ESP traffic. In this case, a WLAN access point would need to perform cryptographic operations for the HIP base exchange and IPsec encryption (between the AP and the central network gateway). Our measurement results in Section VI-B indicate that this approach is inefficient due to its computational overhead for a resource-constrained WLAN AP. It limits the serving capacity of WLAN APs and the scalability of the whole architecture.

A more rational approach to deal with legacy clients is to perform a simple port switching on the Wi-Fi APs and thus decrease the load on the infrastructure. As the use of the HIP proxy on the APs does not deliver any additional benefits other than supporting legacy clients, it makes sense to replace it with a port switching technique.

An example of such setup is illustrated in Figure 4. An AP routes the traffic from a HIP-enabled client towards the central HIP gateway establishing a secure tunnel and filtering HIP and ESP packets on the HIP-aware AP firewall. At the same time, a legacy mobile client is routed by the same AP to a demilitarized network zone and can be served with lower QoS (depending on the network policies).

V. Experimental Testbed

This section presents our experimental testbed. It starts with a description of the HIP on Linux (HIPL) [1] porting process and elaborates challenges that we confronted during migration to the OpenWrt platform. Next, we show the components of our network setup and introduce the status of architecture deployment in the panOULU WLAN.

A. Porting HIPL to OpenWrt

We ported the HIPL implementation to two access point models, La Fonera FON2100 and Gateworks Avila GW2348-4, both running the OpenWrt Linux distribution.

Porting of the HIPL implementation to the OpenWrt platform was not a straightforward process and required efforts with both access point models. Among the problems we faced were various memory management issues, missing dependencies, and hardware constraints. We have chosen OpenWrt as a reference Linux distribution because of its wide range of supported hardware platforms. Fortunately, OpenWrt is known for its good support of La Fonera and Avila boards. Another consideration was a large and growing community of developers that work on the OpenWrt project.

During the HIPL software migration to OpenWrt we detected a few critical bugs that were hard to locate and eliminate. Besides that, we rewrote parts of HIPL code completely to avoid library dependencies. For instance, reimplementation of list data structures was required to remove *glib* library dependencies. Interestingly, HIPL code running on ARM processor (Avila platform) required static typecasting to a character pointer type for memory copy operations. Otherwise, we were getting ambiguous results; as an example, we have observed that copying of *in6_addr* structure would copy correctly only 96 bits and fill the rest of the structure with zeroes.

As a summary, porting of the current HIPL implementation to other architectures supported by the OpenWrt platform should be feasible, but researchers can encounter problems related to a specific platform. Since HIPL is not included in the OpenWrt distributions, it should be compiled with an OpenWrt SDK and the HIPL patches that are publicly available.

B. Experimental Setup

Our network setup (see Figure 4) comprised a set of Wi-Fi access points running a HIP-based distributed firewall. The first AP model we used for our experiments was La Fonera FON2100 that has 16 MB of RAM, 8 MB of Flash memory, and a 32-bit MIPS processor running at 183 MHz. The second model, Gateworks Avila GW2348-4, is much more powerful than the previous one, combining on a small board 128 MB of RAM, 32 MB of Flash, and a 533-MHz Intel CPU.

Another component of the implemented architecture was a central HIP proxy server, a desktop-like PC, that acted as the main gateway for the whole WLAN network connecting it to the rest of the Internet. The testbed also included a remote peer and a number of mobile clients ranging from a Nokia 810 Internet Tablet and a Symbian S60 smartphone to a mini-laptop ASUS Eee PC 901. There were both HIP-aware and non-HIP hosts among these mobile devices. The clients used several publicly available HIP implementations, including HIPL and OpenHIP [2]. All components of our experimental testbed for distributed user authentication in a wireless network are illustrated in Figure 4.

C. Considerations for Deployment

Our system works in a way that HIP-enabled users establish an association with the central HIP proxy server by performing a HIP base exchange. Each packet sent from a mobile client is filtered by the distributed firewall running on the HIP Wi-Fi APs based on the source and destination HITs. Such scenario provides multiple benefits including strong user authentication to the WLAN network and HIP terminal mobility. In addition, all transmitted data is protected by IPsec ESP encryption.

On the other hand, if there is a need for incremental architecture deployment in a large public WLAN network, such as panOULU, our architecture can be easily extended to provide backward compatibility for legacy clients. This can be realized by a simple port switching technique on the Wi-Fi APs as described in Section IV-E.

Ideally, we recommend to deploy our complete architecture at once so that each WLAN user becomes HIP-aware and is able to authenticate itself to the network. We believe that universal client authentication would significantly reduce the probability of a network abuse. However, we understand that in practical situations for big operator's environments incremental deployment is more feasible. In such cases, the operator may provide backward compatibility for a certain transition period to give time to legacy clients to install HIP software on their terminals.

D. Deployment Status in panOULU

Deployment of our authentication architecture in panOULU, a city-wide WLAN in Finland, is in its initial phase. We have installed an HTTP/HIP proxy on the main network gateway. The proxy allows HIP mobile clients to establish HIP associations with the central gateway. It authenticates clients to the network, provides terminal mobility and encryption of user data over an unprotected wireless link. Our preliminary tests showed that a mobile HIP user can successfully connect to panOULU network, secure browsing sessions by constructing an IPsec tunnel with the central HIP proxy server and keep the sessions ongoing while moving within the network.

As a next step, we added a La Fonera FON2100 access point to panOULU with running the HIP firewall and proxy extensions. Initial experimentation indicated that limited hardware resources of this AP model are stressed by the HIP proxy component that, in turn, is not able to serve many clients simultaneously. In the future, based on performance comparison of different Wi-Fi AP models, we plan to choose the most suitable hardware and continue deployment of our authentication architecture in panOULU network.

VI. Performance Evaluation

This section presents the results of our measurements on two Wi-Fi access point models with different hardware resources, La Fonera FON2100 and Gateworks Avila GW2348-4. The results have been measured in two modes with each access point running as a HIP proxy and a HIP firewall.

HIP involves public-key cryptography and IPsec encryption that can easily stress lightweight resources of Wi-Fi APs. One of our objectives was to evaluate the impact of such heavy operations on performance of our authentication architecture. Our previous work [16], [15], which studied HIP performance on Linux-based Nokia Internet Tablets and Symbian smartphones, served as a good starting point for performance evaluation in this article. In addition, HIP was evaluated on stationary Internet hosts with conventional PC-like resources [11], [10], [14], [26].

We have made an interesting observation that 100% CPU utilization does not necessarily indicate a performance issue for a particular device. Rather, this can be interpreted as utilization of all available resources by running applications when the system allocates maximum capacity to them. We have noticed that when all applications are in the *idle* state, CPU utilization is about 1%. However, upon invoking a resource-demanding application, the system will release all available CPU cycles to it. On the other hand, with multiple applications running in parallel, the Linux scheduler guarantees no starvation for any task by allocating the time slices fairly.

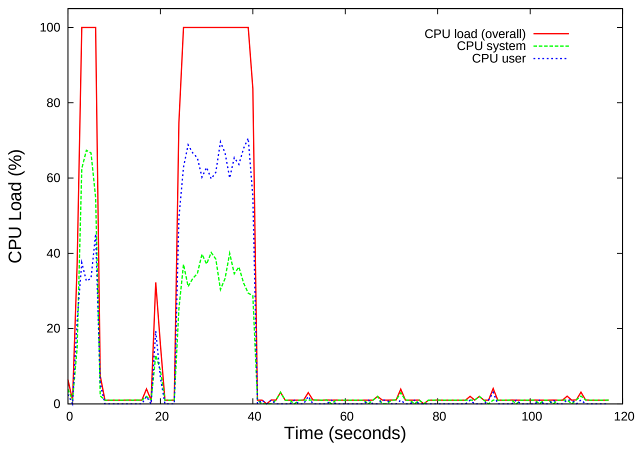

Fig. 5. Fonera CPU load in a firewall mode.

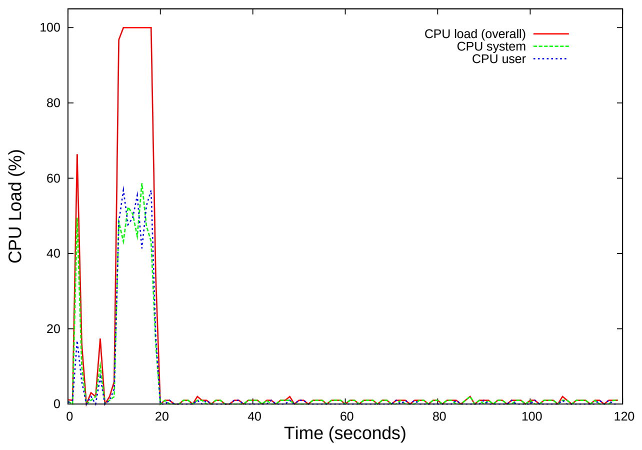

Fig. 6. Avila CPU load in a firewall mode.

A. Firewall Mode

This section contains an analysis of our measurement results on Fonera and Avila access points running in the HIP firewall mode.

A HIP-enabled firewall on a Wi-Fi AP does not require running the HIP daemon, unless the HIP daemon itself is used for user registration or similar tasks. In our experiments, we ran only the HIP-enabled firewall as it is the crucial component in our architecture.

Figures 5 and 6 illustrate a remote file copying over SSH and HIP during traffic filtering on the Wi-Fi access points in the middle. As figures show, the Avila board notably outperforms the Fonera AP in time required to complete this operation. This result indicates that faster AP hardware provides sufficient performance of filtering operations in a distributed firewall.

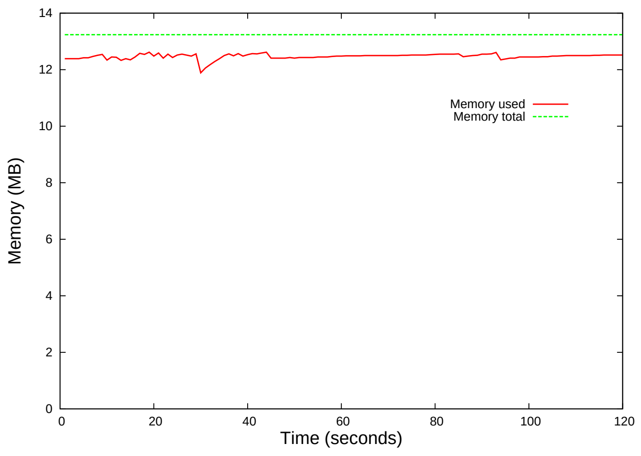

Fig. 7. Fonera memory load in a firewall mode.

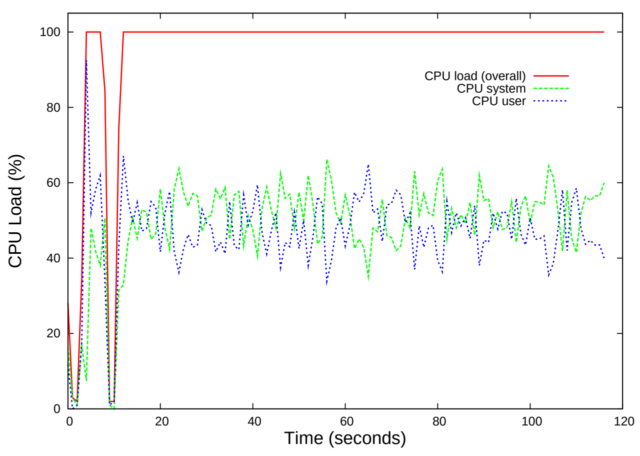

Fig. 8. Fonera CPU load in a proxy mode.

TABLE I
AVERAGE PACKET FILTERING TIME ON AVILA (SECONDS).

Ruleset Size →		2	5000
Packet Type	HIP control	0.002	7.297
	ESP data	0.005	0.005

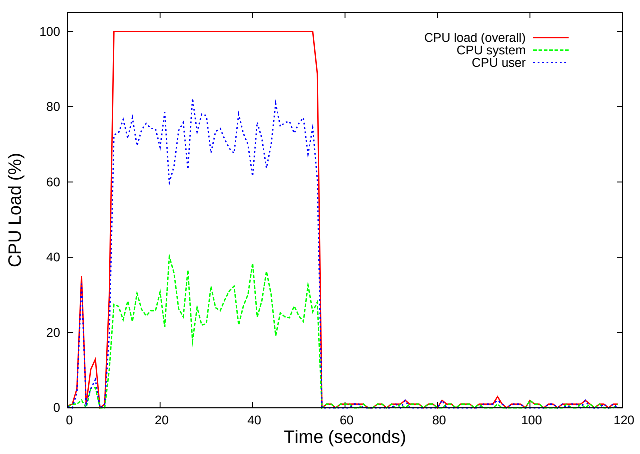

Fig. 9. Avila CPU load in a proxy mode.

Our results on memory utilization in the firewall mode ensure a good level of performance on both AP models. We found that although only 1 MB of RAM is available after the firewall startup on Fonera (see Figure 7), it is sufficient to sustain two-three mobile clients. With Avila board, the situation is better as only 21 MB of total 128 MB of RAM is used on the average. Thus, the amount of RAM in the access point does not make a significant impact on the firewall performance.

In our next experiment we measured average packet filtering time on the Avila board. We found that performance of the firewall is dependent on the number of firewall rules and their ordering (see Table I). We implemented two different methods. In the first one, the number of rules was two (one for each direction) whereas in the second setup we had 4998 mismatching rules and only last two were matching. In our architecture, each firewall that joins the network and performs user authentication has to store a local copy of at least one rule for each user, unless a protocol that stores the rules in a distributed way is used. Therefore, if the network has n registered users, the size of the rule set will be at least n. Hence, the larger is the set, the longer it takes to authenticate the user in the current implementation. We plan to study rule sorting and matching in future work.

B. Proxy Mode

This section presents and analyzes measurement results with Fonera and Avila access points running in the HIP proxy mode.

Figures 8 and 9 depict the CPU utilization during the bulk copy of a file over SSH. A mobile client is HIP-unaware, and hence the proxy performs packet translation. These plots do not give us the CPU use per a particular application. Instead, our scenario suggests 100% of CPU utilization in the user mode allocated to the HIP daemon and proxy. For both experiments the setup was identical except the underlying hardware the proxy was running on.

As can be seen from the figures, the copying task is completed on Avila board within 55 seconds. Fonera, on the contrary, due to limited resources is not able to complete the operation even after 120 seconds. Since the operations of packet translation require a lot of overhead (such as memory copying, database lookup, encryption operations, etc.), the throughput of the network is influenced by the amount of available CPU cycles. In general, our measurements show that the Avila board offers a substantial increase in throughput in comparison to the Fonera access point.

We compared TCP data throughput of the channel between a mobile client and the central network gateway with the Avila board in the middle running in the firewall and proxy modes. The results show that a HIP-aware firewall does not seriously affect the data rate, but a HIP proxy on the Avila AP reduces the throughput (13.10 versus 4.38 Mbps respectively).

The memory use becomes an issue when the HIP daemon

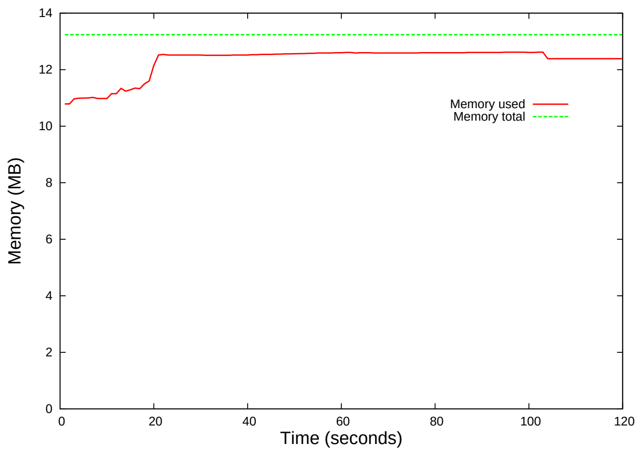

Fig. 10. Fonera memory load in a proxy mode.

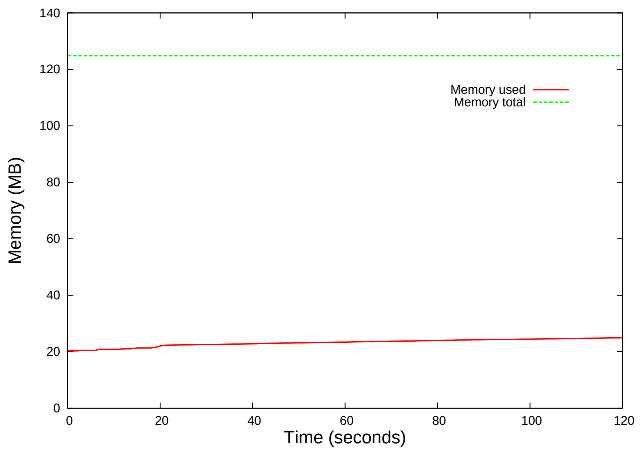

Fig. 11. Avila memory load in a proxy mode.

is running in a proxy mode. In contrast to a firewall mode, 1 MB of memory available on Fonera after the HIP daemon and proxy have been invoked is not sufficient (see Figure 10). Due to the absence of swapping partition on the OpenWrt platform, lack of the RAM memory makes the Fonera access point completely unresponsive with the only option of hard reset to bring it back.

In contrast, on the Avila board with plenty of available RAM (see Figure 11), the HIP proxy can sustain several connections without problems. However, when every packet is served in a FIFO manner, per-packet processing time affects the overall system throughput.

C. Mode Selection

The analysis of the measurement results allows us to give the following recommendations on deploying the architecture proposed in this article:

- For areas with small rate of connections, it is sufficient to have low cost devices such as La Fonera FON2100 to authenticate the users using HIT-based filtering on the distributed firewalls (i.e., running a HIP-aware firewall only).
- Since running only the HIP firewall does not require a lot of resources, one may consider using the existing Wi-Fi access points but only modifying the software that is pre-installed on these devices.
- In cases when support for both plain unauthenticated and HIP authenticated traffic is needed (i.e., support for both legacy and HIP clients during the network transition state), more powerful devices such as Gateworks Avila GW2348-4 are required. However, even on powerful Wi-Fi APs we recommend replacing a HIP proxy with another technique (e.g., forwarding packets to a demilitarized zone) to provide compatibility with legacy clients.

VII. Future Work

We envision several interesting directions for future work including deployment and analysis of the distributed firewall architecture in a large-scale network; building source address validation architecture [30], [28] with the help of HIP; developing a wireless-aware version of HIP. The source address validation architecture based on HIP (SAVAH) can be considered as an extension of the architecture described in this paper. SAVAH allows to prevent several network attacks including source address spoofing and identity forging. In future, we might need to port SAVAH and HIPL to new AP hardware.

VIII. Conclusions

In this paper, we proposed a two-level distributed authentication architecture for wireless networks. Mobile hosts are using the Host Identity Protocol (HIP) to connect to the legacy Internet hosts through operator's WLAN. The system includes an operator-specific proxy server and a distributed firewall running directly on WLAN APs. We have implemented the system by reflashing the firmware of two different AP models with Linux-based OpenWRT distribution.

Performance measurement results of HIP proxy and firewall running on OpenWRT WLAN access points have supported the motivation behind the two-level architecture. The hardware capabilities of WLAN APs are sufficient to run a HIP firewall performing a simple verification of user traffic based on HITs. This prevents a malicious user from attacking the operator's internal infrastructure. Resource-intensive operations, such as serving as a HIP proxy and a target of the HIP base exchange, are given to a powerful server running in the fixed operator's network. The proxy enables a mobile user to benefit from HIP properties such as IPsec encryption, mobility and multihoming, and IP cross-family support.

The proposed architecture is being deployed in a city-wide WLAN network in northern Finland (panOULU). We also plan to use it internally within our research group as a part of the network pilot with Nokia Internet Tablets. Our future plans include integrating the system with Source Address Validation Architecture (SAVA) being developed by the IETF, deployment and performance evaluation of distributed firewall models in large-scale wireless networks.

Acknowledgments

We would like to thank Miika Komu for valuable advice and numerous code tips. Without his guidance it would have been hard to locate and fix those hidden bugs that always occur but are complicated to find. We are also thankful to Dongsu Park for his effort in porting HIPL to OpenWRT Linux distribution.

References

[1] HIPL website. http://infrahip.hiit.fi.

[2] OpenHIP website. http://www.openhip.org.

[3] panOULU network website. http://www.panoulu.net.

[4] K. Brasee, S. K. Makki, and S. Zeadally. A novel distributed authentication framework for single sign-on services. In *SUTC '08: Proc. of the 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pages 52–58, Washington, DC, USA, 2008. IEEE Computer Society.

[5] D. Forsberg. Secure distributed AAA with domain and user reputation. In *Proc. of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007*, pages 1–6. IEEE Computer Society, June 2007.

[6] A. Gurtov. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Wiley and Sons, 2008.

[7] T. Heer, S. Götz, E. Weingärtner, and K. Wehrle. Secure Wi-Fi sharing at global scales. In *Proc. of the 15th International Conference on Telecommunications*, June 2008.

[8] T. Heer, R. Hummen, M. Komu, S. Götz, and K. Wehrle. End-host authentication and authorization for middleboxes based on a cryptographic namespace. In *Proc. of the IEEE International Conference on Communications 2009 (ICC 2009)*, Dresden, Germany, 2009. IEEE. To appear.

[9] T. Heer, S. Li, and K. Wehrle. PISA: P2P Wi-Fi Internet Sharing Architecture. In *Proc. of the 7th International Conference on Peer-to-Peer Computing*, Galway, Ireland, Sept. 2007.

[10] T. R. Henderson. Host mobility for IP networks: A comparison. *IEEE Network*, 17(6):18–26, Nov. 2003.

[11] T. R. Henderson, J. M. Ahrenholz, and J. H. Kim. Experience with the Host Identity Protocol for secure host mobility and multihoming. In *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC'03)*, Mar. 2003.

[12] S. Ioannidis, A. D. Keromytis, S. M. Bellovin, and J. M. Smith. Implementing a distributed firewall. In *CCS '00: Proc. of the 7th ACM Conference on Computer and Communications Security*, pages 190–199, New York, NY, USA, 2000. ACM.

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

[14] P. Jokela, T. Rinta-Aho, T. Jokikyyny, J. Wall, M. Kuparinen, H. Mahkonen, J. Melen, T. Kauppinen, and J. Korhonen. Handover performance with HIP and MIPv6. In *Proc. of the 1st International Symposium on Wireless Communication Systems, ISWCS'04*, Sept. 2004.

[15] A. Khurri, D. Kuptsov, and A. Gurtov. Performance of Host Identity Protocol on Symbian OS. In *Proc. of the IEEE International Conference on Communications 2009 (ICC 2009)*, Dresden, Germany, June 2009. IEEE. To appear.

[16] A. Khurri, E. Vorobyeva, and A. Gurtov. Performance of Host Identity Protocol on lightweight hardware. In *MobiArch '07: Proc. of the 2nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture*, pages 1–8, New York, NY, USA, Aug. 2007. ACM.

[17] J. Korhonen, A. Mäkela, and T. Rinta-aho. HIP based network access protocol in operator network deployments. In *Proc. of the First Ambient Networks Workshop on Mobility, Multiaccess, and Network Management (M2NM'07)*, Oct. 2007.

[18] D. Kuptsov and A. Gurtov. SAVAH: Source address validation with Host Identity Protocol. In *Proc. of the First International ICST Conference on Security and Privacy in Mobile Information and Communication Systems (MobiSec'09)*, June 2009. To appear.

[19] J. Laganier and L. Eggert. Host Identity Protocol (HIP) rendezvous extension. IETF RFC 5204, Mar. 2008.

[20] J. Laganier, T. Koponen, and L. Eggert. Host Identity Protocol (HIP) registration extension. IETF RFC 5203, Apr. 2008.

[21] R. Moskowitz and P. Nikander. Host Identity Protocol architecture. IETF RFC 4423, May 2006.

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

[23] P. Nikander, T. Henderson, C. Vogt, and J. Arkko. End-host mobility and multihoming with the Host Identity Protocol (HIP). IETF RFC 5206, Apr. 2008.

[24] P. Nikander and J. Laganier. Host Identity Protocol (HIP) domain name system (DNS) extension. IETF RFC 5205, Mar. 2008.

[25] P. Nikander and J. Melen. A bound end-to-end tunnel (BEET) mode for ESP: draft-nikander-esp-beet-mode-09, Aug. 2008. Work in progress.

[26] P. Pääkkönen, P. Salmela, R. Aguero, and J. Choque. Performance analysis of HIP-based mobility and triggering. In *Proc. of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'08)*, June 2008.

[27] R. Stepanek. Distributed firewalls. In *Publications in Telecommunication Software and Multimedia*. Helsinki University of Technology, 2001.

[28] J. Wu, J. Bi, X. Li, G. Ren, and K. Xu. A Source Address Validation Architecture (SAVA) testbed and deployment experience. IETF RFC 5210, June 2008.

[29] J. Wu, G. Ren, J. Bi, X. Li, R. Bonica, and M. Williams. Source Address Validation Architecture (SAVA) framework: draft-wu-sava-framework-00.txt, Feb. 2007. Work in progress.

[30] J. Wu, G. Ren, J. Bi, X. Li, and M. Williams. A first-hop source address validation solution for SAVA: draft-wu-sava-solution-firsthop-eap-00, 2007. Work in progress.

[31] J. Wu, G. Ren, and X. Li. Source address validation: Architecture and protocol design. In *Proc. of the IEEE International Conference on Network Protocols*, pages 276–283, Oct. 2007.