APPLYING HOST IDENTITY PROTOCOL TO TACTICAL NETWORKS

Mikko Sarela
Theoretical Computer Science
Helsinki University of Technology
Espoo, Finland

and

Pekka Nikander
Nomadiclab
Ericsson Research IP Networks
Jorvas, Finland

ABSTRACT

In this paper, we describe the current status of the Host Identity Protocol and discuss how it could be applied to tactical networks, including mobile ad hoc networks. The Host Identity Protocol (HIP) is a protocol proposal at the IETF for separating the end-point identifier and locator nature of IP addresses. It introduces a new name space, consisting of public cryptographic keys, and uses these keys to identify hosts. All applications deal with the public keys instead of IP addresses; with a backward compatibility layer, most current applications will continue to work unchanged. A new layer in the kernel dynamically maps the public keys in outgoing packets into IP addresses, and vice versa for incoming packets.

INTRODUCTION

The term "tactical network" generally refers to a communications network employed in a military setting. There is increasing interest in using Internet-based protocols as the foundation for future tactical networks. While there are certain cost benefits to this approach (equipment choices, lower training and operations costs), the generally available standard Internet protocols may not satisfy the communications requirements of tactical networks in terms of security, mobility, and protocol performance.

One concept common among Internet users is the notion that their computer is identifiable by an IP address. This has certainly been true for most users connected by wires, via a single interface, to the network. However, the situation becomes more complicated when a device has more than one network interfaces. In a mobile setting with possibly spotty radio performance, it may be increasingly common for devices to use more than one interface, to improve network availability. Moreover, when a device moves around, it typically needs to obtain a new IP address to conform to the locally-available address prefix, since IP addresses are hierarchical and aggregated by the prefix. Once devices have more than one IP address, and once IP addresses become dynamic, it becomes increasingly hard and less secure to rely on the assumptions that IP addresses have a static, one-to-one mapping with a particular computer.

In this paper, we describe how the Host Identity Protocol (HIP) [7], a new architecture and protocol for IP-based networks, may improve the situation for IP-based tactical networks that are faced with these types of mobility and multi-homing scenarios. In general, we suggest that additional layers of abstraction between the network layer and application layer can allow hosts to better adapt to changing networking conditions. In this paper, we only concentrate on a few aspects, namely mobility, multi-access, and security, leaving considerations such as congestion control and transitory connectivity for future work.

The rest of this paper is organized as follows. First, in the next Section, we briefly describe the problem at hand. The following four Sections briefly describe the HIP architecture and based exchange, HIP based mobility and multi-homing, HIP based access control and untraceability, and bridging IP addressing realms with HIP. In the last two sections, we suggest how HIP could be applied to tactical networks, and provide some conclusions.

TACTICAL AD HOC NETWORKS

In general, NATO requirements suggest that tactical networks should be

- designed for joint combined operations at the battle field,
- easy to install and maintain within different network scenarios, and
- backward connected to legacy WAN systems. [4]

Tactical networks consist of a combination of semi-static, slowly moving, and rapidly moving devices. There is a desire to secure the networks to prevent eavesdropping, and typically multiple independent levels of security are provided. There are also some conflicting desires on host identification. On one hand, there is a desire to be able to identify computers in the network in a manner that cannot be spoofed, for the purposes of access controls and traffic prioritization. On the other hand, there is a desire to prevent eavesdroppers from discerning the whereabouts of the important nodes. Therefore, the system must employ strong identity authentication in combination with obfuscation techniques.

Problems in current practice

There are several problems in current commercial Internet technologies that need to be resolved. First, in the current systems there is a strong tendency to use IP addresses as endpoint identifiers, and make authorization decisions based on the IP addresses of the peers. This clearly breaks down in both mobile environment, and in multihoming environment (which is increasingly of interest to tactical hosts who want path diversity), and is basically difficult to deal with from a preplanning or provisioning standpoint, because one cannot perform dynamic address allocation.

Second, home-agent based solutions to mobility, such as Mobile IP [11] and Mobile IPv6 [5], are fragile. In fact, the return routability test required by the commercial Mobile IP route optimization solutions brings this fragility to route optimisation, as the home agent needs to be reachable at least time to time. A more direct authentication of hosts for mobility purposes is desired.

Third, while it is desirable to allow IP address based access control in order to support current system, it would be desirable to provide access control based on strong cryptography. Preferably, such a system not only allows access control of hosts or servers, but also access control as to who is even allowed to have a packet floating around on a particular network segment.

Fourth, many of the current security protocols open a direct venue for CPU exhaustion denial-of-service attacks by sending in garbage.

Finally, there is the desire to limit the possibilities for traffic analysis even by legitimate parties. Information about the current IP addresses (and therefore the location) of important units should not be visible to parties that are not involved in direct communication with them.

HOST IDENTITY PROTOCOL (HIP)

The Host Identity Protocol (HIP) [6, 7] separates location and identity by defining a new *Host Identity* namespace between the transport and internetworking (IP) layers. Figure 1 provides a comparison between the current and HIP architectures. In the current architecture IP addresses represent both location (for routing) and identity along with port numbers through sockets (for processes).

The new HIP architecture is depicted on the right side of the figure 1. The transport layer sockets are now

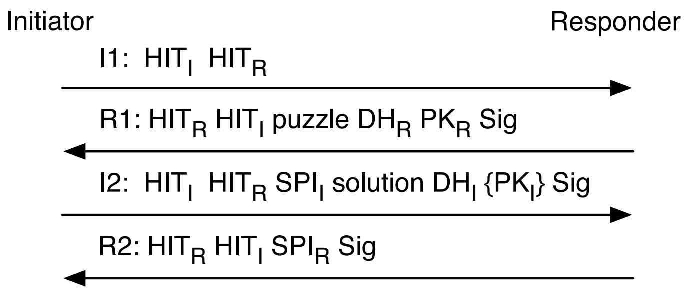

Figure 2: HIP base exchange

named with separate host identities, which the Host Identity layer translates to one or more IPv4 or IPv6 addresses. This binding between Host Identities and IP addresses is simultaneously dynamic and one-to-many, providing for mobility and multihoming, respectively. Both of these features make IP level traffic analysis protection easier to achieve.

Each host generates one or more public/private key pairs to provide identities for itself. The public keys act as *Host* or *End-Point Identifiers*. A host can prove that it corresponds to the Host Identity by signing some data with the (non-disclosed) private key. All other parties can use the Host Identity (a public key) to authenticate the host.

A *Host Identity Tag* (HIT) is a 128-bit representation for a host identifier. It is created by taking a cryptographic hash of the public key. There are two advantages of using a hash over using the public key as such. First, its fixed length makes protocol coding easier. Second, it presents a consistent format for protocols, independent of the public key technology.

The introduction of new cryptographical end-point identifiers clarifies the role of IP addresses. When HIP is used, IP addresses become pure topological labels, naming locations in the Internet. An end-point may change its IP address without breaking connections. Thus, the relationship between location names and identifiers becomes dynamic.

HIP base exchange

The Host Identity Protocol (HIP) [6] consists of a two-round-trip, end-to-end Diffie-Hellman key exchange protocol (called base exchange), a mobility management protocol, and some additional messages. The purpose of the HIP base exchange is to create assurance that the peers indeed possess the private key corresponding their host identifiers. Additionally, the exchange creates a pair of IPSec Encapsulated Security Payload (ESP) security associations (SAs), one in each direction.

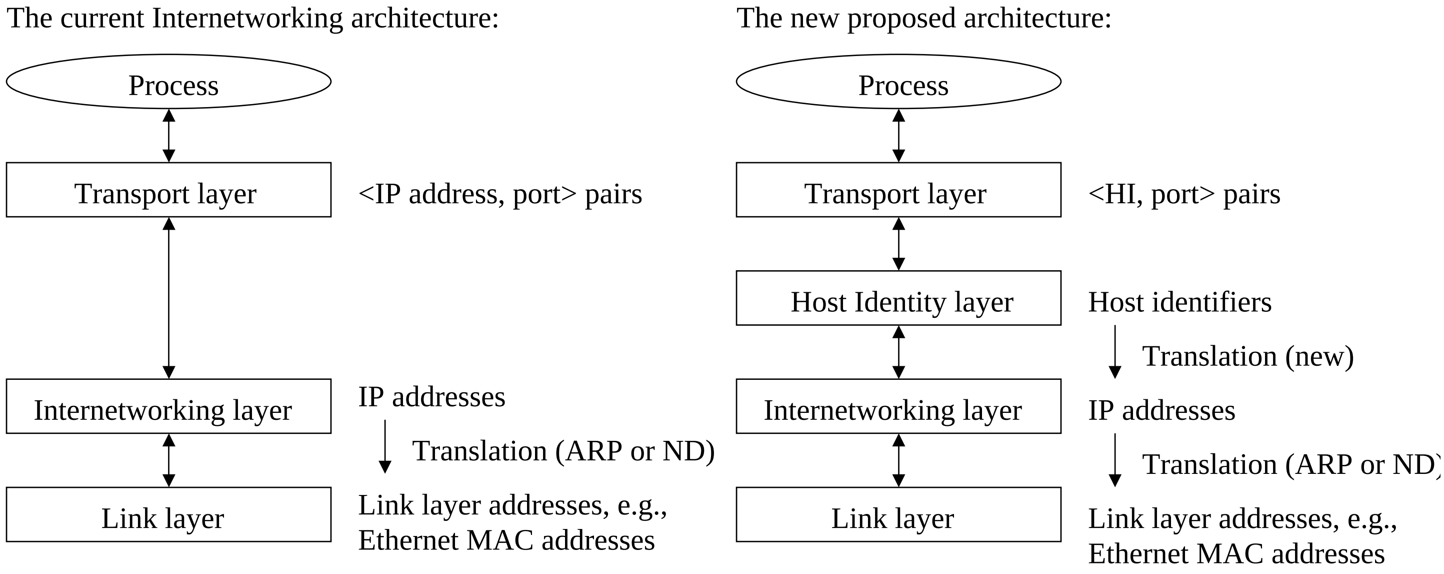

Figure 1: The current Internetworking and the proposed new architectures

The base exchange consists of messages I1, R1, I2 and R2. The HIP base exchange is illustrated in Figure 2. Each HIP message consists of fixed fields, including the HITs of an initiator and a responder, followed by a number of variable length parameters. The first packet, I1, contains only the fixed header, i.e., the HITs. If the initiator does not know the responder's HIT, it may leave that field empty. If so, the reponder is free to select among any of its identities.

When the responder receives an I1 packet, it selects a suitable R1 packet from a pool of precomputed messages. As DoS resistance has been one of the main design goals in HIP, the responder maintains a pool of pre-computed and signed R1 packets, allowing it to pick a pre-computed message instead of constructing one. To facilitate this, the initiator's HIT is not included in the R1 signature.

The R1 message contains a puzzle that the initiator has to solve. The same message also initiates the Diffie-Hellman exchange. It contains the responder's host identity public key, together with the Diffie-Hellman public key and other Diffie-Hellman parameters. From the traffic analysis point of view, it is important to notice that the responder is not able to form the session key before the I2 packet arrives. Therefore, the responder's host identity public key is currently transmitted in clear.

Upon receiving R1, the initiator solves the puzzle, computes a session key, and sends I2. I2 includes the puzzle solution, Diffie-Hellman parameters, SPI, and the initiator's host identity public key. The host identity public key is encrypted using the session key.

The responder verifies that the puzzle is correctly solved, creates the session key, authenticates the initiator, and creates session state. The final message, R2, contains the responder's SPI and a signature. The signature allows the initiator to complete the authentication procedure.

The HIP negotiation results in the parties having an authenticated Diffie-Hellman secret, $\mathrm{KEY}_{\mathrm{DH}}$. The HITs and the Diffie-Hellman secret are used to generate key-material in the following way:

$$\begin{aligned}
\mathrm{KEY}_1 &= \mathrm{SHA1}(\mathrm{KEY}_{\mathrm{DH}}|\mathrm{HIT}_{\mathrm{smaller}}|\mathrm{HIT}_{\mathrm{larger}}|1)\\
\mathrm{KEY}_2 &= \mathrm{SHA1}(\mathrm{KEY}_{\mathrm{DH}}|\mathrm{KEY}_1|2)\\
\mathrm{KEY}_n &= \mathrm{SHA1}(\mathrm{KEY}_{\mathrm{DH}}|\mathrm{KEY}_{n-1}|n)\\
\mathrm{KM} &= \mathrm{KEY}_1|\mathrm{KEY}_2|...|\mathrm{KEY}_n
\end{aligned}$$

The actual keys, used in encryption and integrity protection, are derivered serially from this key-material. It is important to notice that both of the peers must know both the HITs and the shared Diffie-Hellman secret before they become able to encrypt or decrypt anything. Since the HITs are sent as plain text in the base exchange messages, this is not a problem in the current HIP protocol. However, in [15] it is shown how to *blind* the HITs; see also the Section on Untraceability. The blinding could play an essential role in traffic analysis protection.

New semantics for IPSec

It is important to notice that HIP does not change the IP or IPsec packet structure. However, it modifies the details of packet handling within the end-nodes. On the other hand, at the logical level, the new name space imposes changes to the *logical* packet structure. That is, each packet must logically include both the end-point identifiers and IP addresses of the sender and recipient. However, when IPsec is used, the Security Parameter Index (SPI) values can be used as *indices for end-point identifiers*, resulting in packets that are syntactically identical to those used today.

Since the packets are integrity protected with ESP, the recipient is always able to verify that a received packet was sent by the peer, no matter what the source and destination addresses are. Thus, by binding the IPsec security associations to public keys instead of IP addresses, the destination address becomes purely routing information. Only during the base exchange, when the hosts have not authenticated each other, and during re-addressing, does the source address play a substantial role. Once the peer hosts have secure bindings between the public keys and IP addresses, the source address is no more needed by the hosts, and its function reduces to carrying information about the topological path the packet has taken [2].

MULTI-ADDRESSING AND MOBILITY

Once the HIP base exchange has been completed and the security associations are in place, the end-points can inform their peers about the interfaces they have and the current IP addresses assigned to the interfaces. This is useful, when a host has either multiple addresses, or when a host has moved into a new location and received a new IP address. The mechanism is defined in the HIP re-addressing protocol [9]. The protocol proposal consists of Re-address (REA) and New SPI (NES) packets.

With a REA packet, the mobile node informs its peer about its IP addresses. The peer optionally responses with a NES packet, containing a new SPI, that is used to verify that the mobile node is indeed in the claimed location. The third message, ESP to the new SPI, acts as a response to the NES. The purpose of the NES/ESP message pair is to prevent legitimate mobile nodes from inducing flooding attacks. The NES/ESP check is optional based on the level of mutual trust in the network.

The REA packet contains information about interfaces and corresponding IP addresses. It includes a signature. The optional NES packet is used to implement a reachability test procedure for each IP address (corresponding to the Return Routability (RR) test in Mobile IPv6). Each end-point has complete freedom to select which interfaces and IP addresses to announce to the peer. All that the peer needs to know is that the announcing end-point is indeed reachable through the claimed IP addresses. Note that the above approach allows hosts to move around, change IP addresses, and have multiple active IP addresses, without inhibiting the ability of peer hosts to authenticate to whom they are talking.

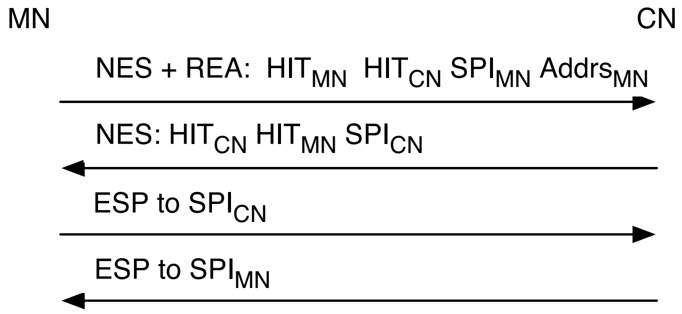

Figure 3: HIP mobility exchange

HOST IDENTIFIERS, ACCESS CONTROL, AND UNTRACEABILITY

Host Identities are not present in every data packet, and the data packets are merely integrity protected, not authenticated. It is likely too burdensome from a computational standpoint to sign every data packet. Nevertheless, there exists a mechanism whereby firewalls and other devices that perform access control can authenticate data flows and regulate which flows (and thereby which hosts) are allowed to access a particular network segment.

The key to this approach is to construct HIP-aware firewalls that observe the HIP base exchange and re-addressing exchanges. These HIP exchanges have been explicitly designed to allow firewalls and other middleboxes to observe the required fields. These firewalls can authenticate the (signed) HIP control packets, and then observe which IP addresses and SPIs the protocol negotiates to include. Thereafter, the addresses and SPIs serve as a proxy for the HITs in the subsequent data packets. This approach is much more flexible than relying on IP addresses for access controls, as is typically done, although since the HIT name space is flat, there is no opportunity to aggregate hosts behind a single prefix.

In [15], Ylitalo et. al. introduce a technique called *BLIND* where the real identity of HIP hosts can be completely hidden from eavesdroppers while still retaining the identity authentication properties of the protocol. The idea is based on using temporary, obfuscated Host Identity Tags (HITs) in the place of the permanent, well known ones. Since the goal is to make the temporary HITs non-sensible to eavesdroppers, any nodes that need to be able to detect the real identity of the communicating nodes must be *preconfigured* with the identities of the potential peer hosts. While this may be a problem in a commercial open network, this is typically not a problem for firewalls or end-nodes in a military setting, where the identities must be preconfigured anyway. The temporary HITs can be changed into different ones whenever a host moves, making tracking virtually impossible.

The basic idea in [15] is to replace the real HITs with a hash of the real HIT and a random nonce. The resulting temporary HIT and the nonce are carried in the initial protocol messages. All nodes that have the real HIT in their possession can find it by a simple iterative search, while nodes that do not possess the real HIT face a computationally impossible problem. For nodes configured with a large number of potential HITs, the initial packets can carry a hint, thereby reducing the required search time.

In summary, when the BLIND approach is used, it is possible to achieve the conflicting goals of strong, cryptographic identity authentication while protecting the identities from eavesdropping outsiders.

HOST IDENTIFIERS, NAT, AND EPHEMERAL IP ADDRESSES

The purpose of network address translation (NAT) is to bridge different IP addressing domains. The most common need for NAT is the use of private IP address space (because of a shortage of IPv4 addresses), but there are also other motivations, like address stability. Basically, any NAT approach makes it possible for a middle box to change the IP addresses of in a packet without breaking end-to-end communications. In standard NAT today, the transport layer identifiers, i.e., ⟨IP-address, port⟩ pairs, are used as static identifiers. However, this is problematic because the transport level identifiers ⟨IP address, port⟩ and network layer addresses ⟨IP address⟩ are smeared together.

When location names and host identifiers are separated, as is done in HIP, the new global name space can be used for static transport layer identifiers. As a result, there are several advantages for using Host Identity name space with NAT. First, a NAT device can easily identify connections using the Host Identities. This means that it becomes possible to inititate connections through a NAT device in both directions[1]. Second, the introduction of a name space allows IP address changes even between IPv4 and IPv6, because higher level protocols use Host Identities rather than IP addresses.

A HIP enabled NAT device translates IP addresses, using the HITs as identifiers for the connection state. However, the HITs are not present in the regular traffic packets between two HIP hosts. Instead, the IPsec SPI is used as an index to the NAT state. If it uniquely identifies the state, as can be fairly easily arranged it may take the place of HITs for handling regular data packets. However, since there may be several HITs behing a single public IP address, the NAT device must learn the SPI values during the initial HIP base exchange, or during mobility signaling. Using 32-bit SPI values instead of 16-bit port numbers also increases the number of connections that can be maintained using a single IP address.

In [14], Ylitalo et. al. propose a new NAT concept called SPINAT: SPI multiplexed NAT. It works in the same way as a regular NAT-PT but uses SPI numbers instead of port numbers. A SPINAT device learns the SPIs and HITs by inspecting HIP base exchange and/or HIP mobility signaling. It can do this securely as there are signatures present in the packets. If a given SPI value is already in use, the SPINAT device may securely replace it with a unique one. Alternatively, if it has multiple public IP addresses, it can assign conflicting SPIs on different public IP addresses, and use the ⟨address, SPI⟩ pair as an index to the translation state.

The SPINAT technique does not require any tunneling headers. The advantage in packet size compared to the current Mobile IP based solution is substantial.

If we compare the HIP based NAT mechanism to routing, there are some similarities. A HIP based NAT device changes IP addresses while using the upper layer state as an index, just like a router changes link layer addresses using the IP address as an index. The difference is in how the state is created: in the case of IP layer routing, the forwarding state is created as a result of running routing protocols, while in HIP "routing" the state is created by inspecting HIP control packets.

HIP IN TACTICAL ENVIRONMENTS

To utilize HIP in tactical environments, we propose an approach based on the following principles.

- Utilize HITs as host identifiers, allowing usage of current IP address based access control mechanisms with strong security controls. To prevent location tracking, combine this with the BLIND approach [15].

- Use a public key infrastructure (PKI) for identities that can divide participants of the network into different trustworthiness classes (for example, our own troops of different kind, allies, and neutrals who need to access different parts of our network). Such a PKI must support fast revocation, must be decentralized, and must tolerate network partition. While leaving the design of such a PKI for future

[1]This requires that the NAT device is able to map the HIT to a private IP address. This is likely to be the typical case when HIP is used with NAT.

work, we envision that it could be based on a partitioning tolerant Distributed Hash Table (DHT) design.

- Use HIT based IPv6-like ad hoc routing in small networks and within a single cluster, solving the ad hoc network addressing and Duplicate Address Detection (DAD) problems. In larger and more stable networks traditional IPv4 and IPv6 addressing and routing can be used.

- Use the SPINAT approach [14] to pass packets between addressing domains. In this context, an addressing domain may be an ad hoc network (using HITs as addresses), a cluster in a larger ad hoc network, or any other independently managed network. This allows HIT based ad hoc domains and more traditional IP address based domains to be combined.

- Use the signalling delegation approach by Nikander et. al. [8] to reduce mobility signalling within an addressing domain.

While the details of the approach need more work, especially in the PKI area, the foundation appears to be solid. Using HITs as host identifiers has been shown to work [10]. Using HITs instead of IP addresses in an ad hoc network is straightforward as the typical ad hoc routing protocols assume pre-defined, unstructured, stable address space [12]. The SPINAT approach is very similar to the IPNL approach [3] by Francis et. al. while using ESP for tunneling and HIP for soft state management in the middle boxes. Finally, the signalling delegation approach [8] is a straightforward application of the more generic trust management approaches, including SDSI/SPKI and KeyNote2 [1].

The approach is demonstrably less fragile than Mobile IP [10]. In particular, no fixed home agents are needed. To facilitate fast movement and to solve the simultaneous movement problem, a Forwarding agent can be used to keep track the current IP addresses of a mobile host [10]. As discussed recently at the 59th IETF meeting [13], basically any node can act as a forwarding agent for other nodes that it has a connection with. This can act as a starting point for designing a robust rendezvous infrastructure that works well even under network partitioning and intermittend connectivity.

CONCLUSIONS

The Host Identity Protocol (HIP) is a promising new protocol proposal currently under discussion at the IETF. Additionally, a number of research projects are considering HIP as an architectural component. There are currently five publicly known implementations of the HIP base protocol, three of which are distributed under open source or compatible licenses (OSS); see Table 1.

Table 1: HIP implementations

Boeing Phantom Works	Linux	
Ericsson Research Nomadiclab	FreeBSD	OSS
Helsinki University of Technology	Linux	OSS
Indranet technologies	Python	OSS
Sun Research Grenoble	Solaris	

In summary, it can be seen that HIP can solve the problems identified above, namely:

- host identifiers can be used with strong security guarantees instead of IP addresses, thereby allowing IP addresses to change over time without disrupting communications;

- notifying a peer of an IP address change due to mobility can be done directly with no communications through a home network;

- HIP allows firewalls to cryptographically authenticate which hosts have packets on a given network segment;

- HIP has been designed to minimize vectors for denial-of-service attacks;

- since intermediary routers and firewalls can change the IP addresses, tracking IP addresses brings relatively little benefit to an eavesdropper. The focus is moved to end-point-identifiers, such as public keys and HITs; and

- extensions to HIP should allow hosts to protect their identities from eavesdroppers while still authenticating themselves to each other.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Mats Naslund and Satu Virtanen, and especially to Thomas R. Henderson, for their constructive comments and suggestions on various versions of this paper.

References

[1] T. Aura. *Distributed access-rights management with delegation certificates*, pages 211–235. Number 1603 in LNCS. Springer, 1999.

[2] C. Candolin and P. Nikander. IPv6 Source Addresses Considered Harmful. In *Proc. NordSec 2001*, Nov. 2001. Sixth Nordoc Workshop on Secure IT Systems, Lyngby, Denmark.

[3] P. Francis. Ipnl: A NAT-extended Internet architecture. In *Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 69–80. ACM Press, New York, NY, USA, 2001.

[4] ISSC NATO Open Systems Working Group. *Section 3.1.1. General Requirements*, chapter 3. Dec. 2003.

[5] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. Internet Draft, work in progress, June 2003.

[6] R. Moskowitz and P. Nikander. Host Identity Protocol. Internet Draft, work in progress, June 2003.

[7] R. Moskowitz and P. Nikander. Host Identity Protocol Architecture. Internet Draft, work in progress, May 2003.

[8] P. Nikander and J. Arkko. Delegation of signalling rights. In *Security Protocols*, LNCS, 2003. Cambridge Security Protocols Workshop, April 2002.

[9] P. Nikander and J. Arkko. End-Host Mobility, Multi-Homing and NAT Travelsal with Host Identity Protocol. Internet Draft, work in progress, May 2003.

[10] P. Nikander, J. Ylitalo, and J. Wall. Integrating Security, Mobility, and Multi-Homing in a HIP Way. In *Proc. Network and Distributed Systems Security Symposium*, Feb. 2003. NDSS'03, San Diego, CA, USA.

[11] C. Perkins. IP Mobility Support. RFC 2002, 1996.

[12] C. Perkins. *Ad Hoc Networking*. Addison-Wesley, 2000.

[13] T. Shepard. Some thoughts on hip rendezvous. In *Proceedings of the 59th IETF Meeting*, Mar. 2003.

[14] J. Ylitalo and P. Nikander. Spinat: Spi multiplexed nat for secure and efficient mobility. Unpublished manuscript.

[15] J. Ylitalo and P. Nikander. Blind: A complete identity protection framework for end-points. In *Security Protocols*, LNCS, 2004. Cambridge Security Protocols Workshop, April 2004.