Wireless Networks 5 (1999) 373–390

A public-key based secure Mobile IP *

John Zao [a], Stephen Kent [a], Joshua Gahm [b], Gregory Troxel [b], Matthew Condell [b], Pam Helinek [a], Nina Yuan [c] and Isidro Castineyra [b]

[a] *Network Security Department, GTE Internetworking/BBN Technologies Inc., Cambridge, MA, USA*
[b] *Advanced Networking Department, GTE Internetworking/BBN Technologies Inc., Cambridge, MA, USA*
[c] *Ernst & Young LLP, Boston, MA, USA*

The need of scaleable key management support for Mobile IP, especially the route-optimized Mobile IP, is well known. In this paper, we present the design and the implementation of a public key management system that can be used with IETF basic and route optimized Mobile IP. The system, known as the Mobile IP Security (MoIPS) system, was built upon a DNS based X.509 Public Key Infrastructure and the innovation in cross certification and zero-message key generation. The system can supply cryptographic keys for authenticating Mobile IPv.4 location management messages and establishing IPSec tunnels for Mobile IP redirected packets. It can also be used to augment firewall traversal of Mobile IP datagrams. A FreeBSD UNIX implementation of the MoIPS prototype is available for non-commercial uses.

1. Introduction

1.1. Review of Mobile IP protocols

Mobile IP or IP mobility support [23] (abbreviated as MIP) is a protocol for passing IP datagrams between a *Mobile Node* (MN) and its *Corresponding Nodes* (CNs) as the Mobile Node changes it attachment point on the global Internet. The protocol employs network layer agents to capture IP datagrams that are destined to the Mobile Node's *permanent IP address* in its *home network* and redirect these datagrams using IP–IP encapsulation [24] to a temporary IP address, called the *care-of IP address* (COA), that is assigned to the Mobile Node while it is visiting a *foreign network*. The agents in the home network are known as the *Home Agents* (HAs) and the ones in the foreign network are known as the *Foreign Agents* (FAs). Together, these *Mobility Agents* track the movement of Mobile Nodes by exchanging *registration messages* among themselves and the Mobile Nodes. Based on this registration process, a Home Agent may keep track of the locations of the Mobile Nodes under its administration. It also serves as the entry point to the *IP–IP tunnels* that redirect IP datagrams to the Mobile Nodes away from home. The Foreign Agents, however, may or may not be the exit points of these tunnels depending on the nature of care-of addresses. If the care-of address of a Mobile Node is the IP address of a network interface on a Foreign Agent then the care-of address is called a *Foreign Agent care-of address* and the Foreign Agent is a tunnel end-point. On the other hand, if the care-of address is an address assigned temporarily to a Mobile Node by DHCP or PPP then the address is called a *co-located care-of* address while the Foreign Agent serves only as a last-hop router and a registration agent. In addition to these basic Mobile IP tunnels, *reverse tunnels* may be established during the registration process from the Mobile Node care-of address to the Home Agent [21] (again using IP–IP encapsulation) to pass IP datagrams from the Mobile Nodes through the firewalls protecting the foreign network.

A more sophisticated version of Mobile IP, called *route-optimized Mobile IP* [14], was also proposed to the IETF Mobile IP working group. In that protocol, additional messages known as *location binding requests* and *updates* may be exchanged between a Mobile Node and its mobility aware Corresponding Nodes to inform them of the current care-of-address of the Mobile Node. Similarly, the binding update messages may be dispatched to the Foreign Agents visited by the Mobile Node to pass the information. With the knowledge of Mobile Node's current care-of address, the Corresponding Nodes and the previous Foreign Agents may tunnel the IP datagrams destined to the Mobile Node to its current whereabout. These additional tunnels can shorten the transit time of redirected datagrams and thus reduce the number of datagrams dropped due to delivery failure. As a result, they will improve the performance of Mobile IP, especially if it is used to support a connection-oriented protocol such as TCP.

Figure 1 shows the message flow of both basic and route-optimized Mobile IP.

1.2. Security requirements of Mobile IP

While Mobile IP promises uninterrupted IP connectivity for the Mobile Nodes roaming over the Internet, it also increases the risk of causing remote redirection of Internet traffic [4] by introducing bogus registration and binding update messages. Besides, the attachment of a Mobile Node to a foreign network may cause security concerns to both its home network and the visiting network as the Mobile Node, which is not configured and managed by local net-

* This work reported in this paper was sponsored by Defense Advanced Research Project Agency (DARPA) and Air Force Material Command (AFMC) of the Department of Defense under contract number F19628-95-C-0150.

© J.C. Baltzer AG, Science Publishers

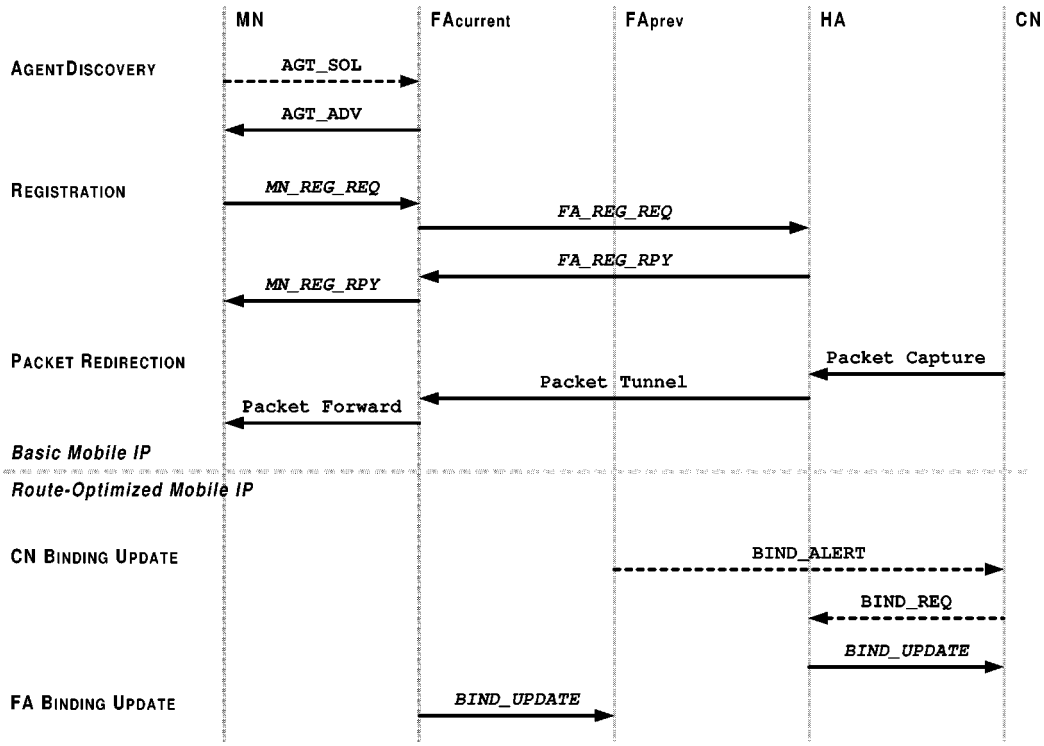

Figure 1. Message flow in Mobile IP protocols.

work administration, may inject and intercept packets in the visiting network, and the home network must receive and process the packets apparently from the Mobile Node as if they are generated by a local node. Consequently, the entire mobile internet, including the Mobile Nodes, the Mobility Agents (Home and Foreign) and the networks accommodating these nodes, must be protected by appropriate security measures.

The ultimate goal of Mobile IP security is to fulfill two general expectations: (1) to allow a Mobile Node to enjoy similar internet connectivity and safety when it visits a foreign network as it resides in its home network, and (2) to protect both the home and the foreign networks from passive and active attacks when the Mobile Node is visiting the foreign network. Throughout the development of Mobile IP, the following security services have been considered for providing the necessary security measures:

- *data integrity*, *data origin authentication* and *anti-replay* protection of Mobile IP registration and location update messages,
- *access control* of the Mobile Nodes when they use resources on the visiting networks,
- *data integrity*, *data origin authentication* and *data confidentiality* protection of IP packet redirecting tunnels,
- *location privacy* of the Mobile Nodes, and
- *anonymity* of the Mobile Nodes.

Among these services, the *first three* are essential to the secure operation of Mobile IP. The *Mobile IP Security (MoIPS) system* discussed in this paper was developed to support these services.

1.3. Organization of the paper

In the remaining *six* sections of this paper, we will discuss the design and implementation of the MoIPS system. Section 2 offers a system overview which explains our approach to provide the three security services (section 2.1) and describes the public-key based MoIPS architecture. Sections 3–5 cover the three components of the system: the DNS-based X.509 public key infrastructure (PKI) in section 3, a lightweight key management scheme for Mobile IP control message authentication in section 4, and the IPSec protection of Mobile IP packet redirecting tunnels in section 5. The implementation of the first MoIPS prototype will be briefly described in section 6 before the conclusions are given in section 7.

2. MoIPS system overview

2.1. Design objectives

The MoIPS system was developed to support three security services that are essential to the safe operation of Mobile IP: (1) authentication of Mobile IP control messages for location update, (2) access control of Mobile Nodes to resources in the foreign networks, and (3) secure tunneling of redirected IP datagrams. In this section, we examine the requirements of these services and explain our approach to provide them.

2.1.1. Authentication of location updates

Among the Mobile IP messages shown in figure 1, the *registration messages* in Basic Mobile IP and the *location binding update messages* in the Route-Optimized Mobile IP (all displayed in bold italics) carry the *location bindings* of Mobile Nodes. These are the associations between the permanent addresses and the current care-of-addresses of the Mobile Nodes. By altering the location bindings in these control messages, creating bogus messages, or replaying pre-recorded messages, an adversary could redirect IP traffic for one node to another node.

In order to frustrate the remote traffic redirection attack mentioned above, registration and binding update messages must be protected with data integrity, origin authentication and anti-replay services. Each of these messages hence includes a 64-bit *identification* tag for detecting replays and one or more *authentication extensions* to provide message integrity and strong authentication using a hashed message authentication code (HMAC) [16,17]. Although the HMAC codes and the anti-replay tags are appropriate means to provide integrity, authentication and anti-replay services, the current Mobile IP lacks a *scaleable key management scheme* for dispatching cryptographic keys needed to support these services. In order to protect the registration messages, keys must be shared *at least* among Mobile Nodes and their Home Agents. In order to protect the binding update messages in the route-optimized Mobile IP, keys must be dispatched among MN-FA, FA-HA and MN-CN pairs.

2.1.2. Access control of Mobile Nodes

For the purposes of network protection, accounting and resource management, it is desirable that the Foreign Agents (in cooperation with the Home Agents) can verify the identity of a Mobile Node before allowing it to complete the Mobile IP registration and establish an attachment point on the visiting networks. The access control procedure should be conducted in two steps: (1) verifying the *identity* of the Mobile Node, and (2) checking the *current status* of the Mobile Node with a relevant authority such as the corresponding Home Agent.

In MoIPS, both the end nodes (Mobile Nodes and Corresponding Nodes) and the Mobility Agents (Foreign Agents and Home Agents) possess X.509 public key certificates issued by hierarchies of certification authorities (CAs). The certificates contain information about identity and network affiliation of these entities as well as the public key parameters necessary for key generation. By exchanging these certificates and challenge-response messages, the end hosts can identify themselves to the Mobility Agents and to one another.

The checking of Mobile Node status, on the other hand, can be conducted implicitly by exchanging *authenticated* registration requests and replies. By forwarding a registration request to a Home Agent, a Foreign Agent indicates that the Mobile Node has successfully passed its scrutiny. By returning a registration reply, the Home Agent informs the Foreign Agent of its approval or rejection of the registration based on the factors such as Foreign Agent identity and affiliation, Mobile Node status, and Home Agent mobility control policy.

2.1.3. Secure tunneling of redirected IP packets

The traffic to/from a Mobile Node, while it is away from its home network, must travel through the visiting network and the public Internet before reaching the destination. Often, the traffic may pass through wireless or other insecure communication media. This form of communication increases significantly the risks of passive eavesdropping and active attacks including packet alteration, insertion or deletion of Mobile IP data traffic. Consequently, the Mobile Node and the Mobility Agents should protect the data traffic with integrity, origin authentication and possibly confidentiality.

In order for the home network to endow the same level of trust and hence provide the same amount of connectivity to a Mobile Node when it roams among foreign networks, the home network will require the traffic to be securely tunneled to/from the Mobile Node (or a trusted agent such as the Foreign Agent connected to the Mobile Node). Similarly, in order for the foreign network to pass traffic for the Mobile Node, the Foreign Agents will require the traffic to be tunneled by an authenticated and trusted Home Agent that manages the Mobile Node. The secure tunnels can be implemented using the *Encapsulating Security Payload* (ESP), *tunneling mode* of *IP security protocols* (IPSec). The protocol will transform each original IP datagram with authentication and encryption mechanisms negotiated by the communicating parties and then encapsulate the datagram within an IPSec header and an external IP header, which specifies the end points of the IPSec tunnel. The MoIPS system provides the service by incorporating an IPSec and an ISAKMP [18] module into the system. Coordinating with the Mobile IP module, these modules impose IPSec protection to selected Mobile IP packet redirection tunnels.

2.2. System architecture

The three security services discussed in the previous section demand the following three kinds of security support:

1. *A scaleable key management infrastructure* capable of generating or dispatching *long-term key parameters* among any pairs of network nodes – without this infrastructure, route-optimized Mobile IP cannot send authenticated binding updates to Corresponding Nodes and establish secure tunnels between MN-CN and FA-HA pairs.
2. *A light-weight key generation algorithm* for supplying the short-term keys needed for authenticating the Mobile IP registration and binding update messages – because Mobile Nodes can only obtain network connectivity after successfully completing Mobile IP registration, it is essential to have a key generation algorithm

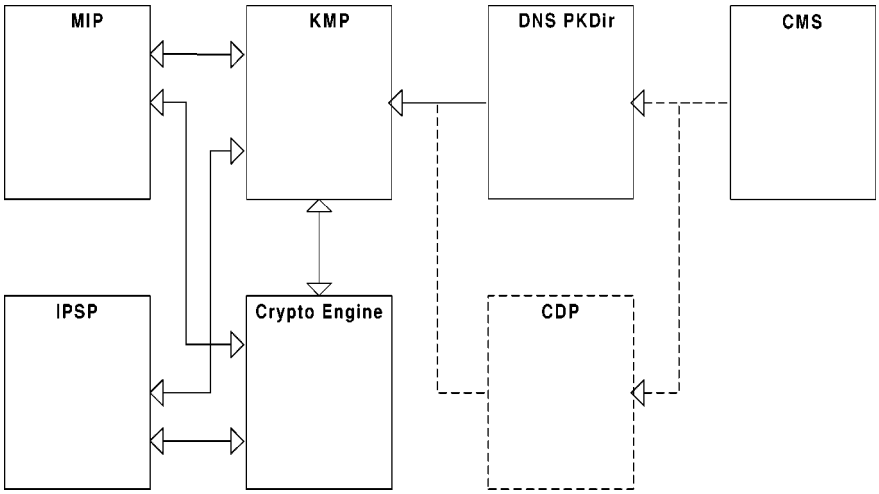

Figure 2. MoIPS system block diagram.

that can supply authentication keys *without* additional message exchange.

3. *Interaction between Mobile IP and IPSec protocols* to enable IPSec protection of selected packet redirection tunnels and ISAKMP negotiation for necessary security associations with minimal disruption of Mobile Node handoffs.

We decided to use public key technology to meet the key management requirements of MoIPS. More specifically, we chose to develop a public key infrastructure (PKI) for managing X.509 v.3 public key certificates and v.2 certificate revocation lists (CRLs) [11] issued to Internet nodes (instead of human subjects). We also chose to use the Internet domain name system (DNS) [20] as the certificate repository. The main reason to use the PKI technology was *scalability*: in order to support global internet mobility, we must have a technology that can establish shared secrets among a large set of nodes spread across multiple Internet domains. A DNS-based PKI has clear advantage over a distributed system of key distribution centers (KDCs), such as a multi-realm Kerberos system for the use of DNS solves the potentially complicated *server discovery problem*, and the use of public key certificates eliminates the need for real-time key dispatches by the KDCs.

Figure 2 shows the functional structure of MoIPS. The Mobile IP and the IPSec modules share their use of a key management module and a cryptographic engine. The key management module generates the short-term keys necessary for the security services while the crypto-engine performs the actual cryptographic processing. Keys and other security parameters are kept in a protected database and passed only to the crypto-engine. Users of security service such as Mobile IP or IPSec make use of *security parameter indices* (SPIs) to refer to the different security settings. The key management module derives the short-term keys from the long-term public keys obtained from the X.509 PKI. In order to obtain the public keys, an X.509 certificate verifier was developed to fetch certificates and CRLs via regular DNS lookup and/or receive them through direct exchanges using the *certificate discovery protocol* (CDP) [3]. The verifier then checks the signatures on the certificates by following the trust relations existing among the hierarchies of certification authorities. The verifier also maintains a cache of verified certificates and CRLs in order to minimize the number of certificate fetch and verification operations.

Wherever feasible, MoIPS makes use of emerging standard cryptographic application programming interfaces (CAPIs) to connect different modules. For example, it uses RSA CryptoKi CAPI as the interface between the users of security service and the crypto-engine; this interface enables MoIPS to use both the RSAREF crypto-library and the Fortezza crypto-token to perform the cryptographic operations. MoIPS also uses PF_Key CAPI to support short-term key and security association management. PF_Key is the standard key management interface between IPSec and ISAKMP in the UNIX environment. Due to the absence of a standard certificate management API, MoIPS developed its own simple interface (known as Cert_API) to connect the certificate verifier with the key management module.

3. DNS X.509 public key infrastructure

We begin our study of MoIPS by examining the public key infrastructure (PKI) that issues certificates to Mobile IP nodes (Mobile Nodes and mobility-aware Corresponding Nodes), and Mobility Agents (Home Agents and Foreign Agents) and also publishes the certificate revocation lists (CRLs) to announce the invalid certificates. In this section, we shall explain our decision of developing a DNS-based X.509 PKI, answer the questions: “who should have a MoIPS certificate” and “what should be the proper subject names”, and then highlight the important fields in the certificate and CRL profiles.

3.1. Reasons for developing a DNS X.509 PKI

We made *two* conscious decisions during the PKI development:

1. We adopted the X.509 v.3 profiles for the certificates and v.2 profiles for the CRLs.
2. We used the Internet Domain Name System (DNS) as the primary mechanism for dispatching the certificates and the CRLs, and only supplemented it occasionally with direct certificate and CRL exchanges between communicating entities.

Following are the tradeoffs we considered before making the decisions.

Advantages of using X.509 certificates and CRLs

There are two reasons to use X.509 PKI to support MoIPS: (1) its hierarchical trust relations among the certification authorities (CAs) can be mapped onto the domain based topology of Mobile IP networks; this makes it a more scaleable and desirable choice than the PGP certificate infrastructure [8]; (2) the X.509 v.3 certificates and v.2 CRLs include many *extension fields* that can be used to carry information relevant to the use of key parameters, the trust relations among the CAs and the management of certificates and CRLs. In particular, following fields enable us to add valuable features to the PKI:

- IssuerAltName enables the establishment of a CA hierarchy independent of the DNS zone structure and thus allows the Mobility Agents and the CAs to be separated from DNS zone servers. In contrast, DNSSec uses the zone servers as the CAs and thus imposes a direct mapping of the DNS hierarchy onto the mobile network topology. In addition, we can use BasicConstraint and NameConstraint to limit the scope of authority of the CAs in issuing certificates.
- CertificatePolicy allows policy information to be included in certificates and thus be used to conduct access control on the Mobile Nodes.
- KeyUsage allows CAs to specify the intended use of key parameters while AuthorityKeyID and SubjectKeyID allow multiple CA signature keys to be distinguished from one another when they are used simultaneously during *key rollovers*. These extensions help to enforce correct use of the public keys.
- By assigning status, required/optional and critical/non-critical, to each certificate and CRL extension, X.509 PKI allows its CAs to issue different kinds of public key certificates with different profiles; for example, the certificates used for Mobile IP control message authentication can be different from those used for IPSec protection.

Advantages of using Domain Name System

The ubiquitous use of DNS over the Internet motivated MoIPS to use it as the certificate/CRL dispatch system. The choice can be justified by the following reasons: (1) MoIPS uses certificates issued to network nodes, all of which should have corresponding entries in the DNS system, (2) the subjects are identified by domain names and/or IP addresses, that are information maintained by DNS, and (3) DNS lookups are often necessary for establishing an end-to-end communication, and thus the certificate fetches can be piggybacked easily onto the regular DNS transactions.

Costs of using a DNS-based X.509 PKI

The use of PKI always entails certain amount of overhead. Our consideration was whether the overhead can be justified when we compare the advantages of using X.509 PKI with those of competing technologies, particularly the Kerberos key distribution centers (KDCs) and the DNSSec public key records.

As mentioned in section 2.2, the advantages of X.509 PKI over Kerberos KDC are obvious. The deployment of KDCs in multiple network domains (as in the case of multi-realm Kerberos system) is complicated by the server discovery problem – a node which intends to communicate with another node in a foreign domain/realm must have a universal mechanism to locate the KDC that serves the foreign domain – besides, the on-line operation of the KDCs are more time-critical and less scaleable than the off-line operation of the CAs.

When comparing X.509 certificates with DNSSec key records, it is worth noting that although the X.509 certificates are larger than the KEY and the SIG records, the certificates carry a lot of valuable information including that which aids Mobile IP access control. Note that the retrieval of KEY and SIG records (like the certificate records) is often conducted in TCP instead of UDP; hence, the effects of different record sizes are reduced. Furthermore, the use of CRLs allows the CAs to issue certificates with long validity periods (which may span several months) and short revocation intervals (which can last less than an hour). In comparison, DNSSec only allows the resource records to be updated daily at fixed time. Obviously, the PKI performs much better in terms of state update and workload amortization. These performance advantages along with the complete independence between the CAs and the zone servers justified the insertion of certificate records into DNS entries.

3.2. Certificate entitlement and policies

All Mobile Nodes, mobility-aware Corresponding Nodes, Home Agents and Foreign Agents that intend to participate in MoIPS must possess X.509 v.3 certificates with the profile specified in section 3.4. These certificates are known as the *MoIPS certificates*, and they perform the following two functions:

1. They supply the Diffie–Helman (DH) public values that are used to generate cryptographic keys for authenti-

cating the Mobile IP control messages and protecting packet redirection tunnels using IPSec protocols.

2. They optionally supply information, including the functional types and the network affiliations of Mobile Nodes and Mobility Agents, which can be used to enforce access control on the Mobile Nodes as they attach themselves onto foreign networks.

The certificates are issued by multiple hierarchies of certification authorities (CAs) (section 3.6) that enforce the MoIPS security policies. The CertificatePolicy extension in these certificates specifies both the functional types and the network affiliations according to the following rules:

- The MoIPS certificates issued to the Mobile Nodes and the mobility aware Corresponding Nodes must specify the *host type*, MN/CN, and the DNS domains that the nodes are affiliated with. The information may be used to exercise rule-based access control policies. A node can be marked as *both* MN and CN since it may allow to play the two roles simultaneously.
- The MoIPS certificates issued to the Mobility Agents must specify the *agent type*, HA/FA, and either the DNS domains or the subnets in CIDR notation that the agents are serving. Again, the information may be used to aid access control. An agent can be marked as *both* a Foreign Agent and a Home Agent if it can provide both services to the corresponding groups of Mobile Nodes.
- A Mobile Node that uses DHCP to obtain a temporary IP address should have a MoIPS certificate marking it as both MN and FA. However, the FA entry in the CertificatePolicy extension must restrict the node to serve only itself.
- A mobile router should have a MoIPS certificate marking it as both MN and HA. The HA role, however, must be restricted to serve only the mobile subnet reachable by the router.
- The minimum security requirements of basic Mobile IP – to authenticate the registration exchanges between the Mobile Nodes and their Home Agents – can be met by issuing MoIPS certificates to those nodes or by establishing shared secrets among them manually. By issuing certificates to the Foreign Agents, MoIPS enables authenticated registration exchanges among the Mobile Nodes, their Home Agents and their visiting Foreign Agents. The Corresponding Nodes only need to have MoIPS certificates if they intend to participate in secure route-optimized Mobile IP.

The MoIPS system mandates the use of MoIPS certificates with CertificatePolicy extension for protecting Mobile IP registration and conducting Mobile Node access control. Nevertheless, it allows the use of other X.509 certificates, that are issued for key encipherment, key agreement and/or authentication services (especially those to be used with IPSec protocols) to support secure packet redirection.

3.3. Subject names

There are two possible choices for the subject name of an Internet node:

1. the *IP address* of the Internet node/interface, and
2. the *canonical domain name* of the node, which is an unambiguous pointer to its DNS entry and the consistent result of reverse DNS lookups given any of the IP address(es) of the node.

The two choices beg for a tradeoff in MoIPS, and their preferences differ slightly depending whether the subject is an Mobile IP node or a certification authority.

For the Mobile Nodes, the Corresponding Nodes, the Foreign Agents and the Home Agents, the subject names in their MoIPS certificates *must* be their *IP addresses* because Mobile IP protocol uses IP addresses to identify the Mobile IP supporting entities. The use of IP addresses has two other *advantages*:

1. It permits issuing of MoIPS certificates to interfaces rather than nodes on the Internet. This allows a multihome node to have a certificate issued to each of its network interfaces as it functions as a Foreign/Home Agents on different interfaces. The configuration is particularly useful when the agent serves multiple subnets or functions as a firewall.
2. It simplifies the verification of NameConstraint as the lowest level CAs may own blocks of IP addresses and issue MoIPS certificates only to the Mobile IP supporting entities with their IP addresses falling within those address ranges.

Nevertheless, the use of IP addresses has two disadvantages:

1. IP addresses tend to have relatively short lifetimes; hence, a certificate must be re-issued whenever there is a change of the IP address of its subject, and a single DNS entry may contain multiple certificates that are issued to different network interfaces to the node.
2. IP addresses do not refer to DNS entries directly; consequently, a *reverse domain name look-up* is needed to convert an IP address to a domain name before searching DNS for a MoIPS certificate.

For the certificate authorities (CAs), their *canonical domain names should* be the preferred subject names because the use of domain names eliminates the need for *reverse DNS lookups*. The only disadvantage of using domain names for the CAs and the IP addresses for the Mobile IP nodes is the use of a heterogeneous naming scheme into the PKI and hence a slight complication of the certificate verification process. It is possible to insert multiple SubjectAltName extensions into a certificate for the purpose of binding both IP address(es) and domain name to the subject. This practice is, however, discouraged in MoIPS because it further complicates certificate verification.

Table 1
Profile of MoIPS certificates.

Fields	Status	Values and remarks
Version	required	= 2 (X.509 v3 Cert)
SerialNumber	required	unique number per CA
SignatureAlgorithm	required	= RSA signature with SHA-1 hash (default)
IssuerName	required	= NULL
Validity	required	certificate valid period
SubjectName	required	= NULL
SubjectPublicKeyInfo	required	DH public value (1024 bit) for end nodes RSA public key (1024 bit) for CAs
AuthorityKeyID	optional/non-critical	SHA-1 hash of CA public key
SubjectKeyID	optional/non-critical (CA only)	SHA-1 hash of subject public key
KeyUsage	required/critical	= 0 × 10 (keyAgreement) for end nodes = 0 × 60 (keyCertSign + cRLSign) for CAs
CertificatePolicy	optional/critical	policy user notice
PolicyMap	*not used*	
SubjectAltName	required/critical	IPv4 address for end nodes canonical domain name for CAs
IssuerAltName	required/critical	canonical domain name of CA
BasicConstraint	required/critical	flag cA = end nodes / CAs pathLenConstraint = max. certificate levels
NameConstraint	required/critical (CA only)	ranges of IP addresses owned by CA
PolicyConstraint	*not used*	
CRLDistributionPoint	optional/non-critical	canonical domain name of distribution point
ExtendKeyUsage	optional/non-critical	= ipsecEndSys + ipsecTunnel + mobileipAuthen
AuthorityInfoAccess	*not used*	

3.4. MoIPS certificate profile

The MoIPS certificates adopt IETF-PKIX X.509 v.3 format as specified in [11]. The certificates *must* be in v.3 format in order to accommodate extension fields that carry additional information. As suggested by the PKIX standard, the fields IssuerUniqueIdentifier, SubjectUniqueIdentifier, PrivateKeyUsagePeriod and SubjectDirectoryAttributes are omitted from the certificates. Table 1 presents a summary of the basic and the extension fields of MoIPS certificates along with their *status*, required/optional and critical/non-critical. In this and the next sections, we review selected fields in the MoIPS certificates and CRLs by collecting them into functional groups and commenting on their intended usage.

3.4.1. Basic fields

Version. The field *must* have the value *two* (2) to indicate that the certificate is in X.509 v.3 format.

Serial Number. This field contains a unique unsigned integer assigned by the issuing CA. The number combined with issuerAltName should uniquely identify the certificate.

Signature Algorithm. The MoIPS system chose to sign their certificates with SHA-1 one-way hash function and RSA encryption algorithm as defined by the OSI Interoperability Workshop. Extra bits can be added as padding according to PKCS#1, section 8.1. The SignatureAlgorithm field specifies the digital signature algorithm[1] used to sign the certificate. Its parameter component is set to NULL.

[1] The actual value of the digital signature is contained as a bit string in the Signature field appended to the certificate along with the same signature algorithm identifier.

Validity. The field has two components, notBefore and notAfter components, which specify the beginning and the end time of the certificate validity period, respectively. The time *may* be encoded in UTCTime or GeneralizedTime, but *must* be Y2K complaint.

3.4.2. Names and Name Constraints

Like other PKIX certificates, the MoIPS certificates use SubjectName and SubjectAltName, IssuerName and IssuerAltName to identify the certificate subjects and the issuing CAs. The CA certificates also use NameConstraint to specify the name space owned by the CAs.

SubjectName and SubjectAltName. The subject or principal of a certificate is identified by its SubjectName field and SubjectAltName extension. SubjectName *must* contain either an X.500 distinguished name or a NULL value. In order to use IP address or domain name as a subject name, we *must* set SubjectName to NULL and place the chosen name in SubjectAltName extension.

IssuerName and IssuerAltName. The CA that issues the certificate is identified by the IssuerName field and the IssuerAltName extension. The IssuerName *must* be set to NULL and the chosen name of the CA *must* be placed in the SubjectAltName extension.

Name Constraints. This extension is used only in CA certificates to specify the subject name space within which the CA is authorized to issue certificates. In the MoIPS cer-

tificate hierarchy, it *should* exist only in the certificates of the lowest level CAs and *must* contain the ranges of IP address corresponding to the Mobile IP networks managed by individual CAs.

3.4.3. Key parameters, usage and identifiers

The certificates carry their public key parameters in the SubjectPublicKeyInfo field. They also use two KeyID fields to identify the signature keys and two KeyUsage fields to specify the proper usage of the key parameters.

Subject Public Key Information. This field contains the public key and identifies the algorithm with which the key is used. In the certificates issued to Mobile IP nodes and agents, the field *should* contain Diffie–Helman public values and the following parameter components:

- prime, p: prime modulus of exponentiation,
- base, g: base of exponentiation,
- privateValueLength: length of private value.

In the certificates issued to the CAs, the field *should* contain RSA public keys and have the parameter component set to `ASN.1 type NULL`.

Authority and Subject Key Identifiers. These two fields are used to distinguish multiple signature keys used by a CA during the key rollover, i.e., an overlapping period of signature key usage in which a new key has been issued but the old key has not yet expired, or when the CA uses multiple signature keys with different algorithms to support different signing policies.

Each signature key used by a CA *must* be identified by the SubjectKeyID extension in a CA certificate. The field *must* be calculated as the SHA-1 hash over the value (excluding tag and length) of the SubjectPublicKeyInfo field in the CA certificate. The same hash value *should* appear in the AuthorityKeyID extension of every certificate signed with that signature key. Hence, one can find the CA certificate and the public key for verifying a certificate issued by the CA by matching the AuthorityKeyID field in the certificate with the SubjectPublicKeyInfo field in the CA certificate.

Key Usage and Extend Key Usage. These two extension fields specify the intended usage and the recommended applications of the public key, respectively. The DH public values in the MoIPS certificates are used to generate short-term keys for message authentication or to establish security associations for IPSec tunneling. Hence, the KeyUsage field of the MoIPS certificates *should* have the keyAgreement bit set. On the other hand, the RSA public keys in the CA certificates are meant to be used for certificate and CRL verification. Hence, the keyCertSign and the cRLSign bits of the KeyUsage field *should* be set in these certificates.

In a MoIPS certificate, the ExtendKeyUsage field may contain the following codes of applications: ipsecUser, ipsecEndSystem, ipsecTunnel and mobileipControl.[2] The field *should* only convey the intended uses of the key parameters without restricting their actual use. Hence, the MoIPS system may use Diffie–Hellman keys contained in the certificates without this extension or with other applications specified in this extension; however, those certificates may not carry access control information in their CertificatePolicy extension.

3.4.4. Certificate policies, policy constraints and mapping

A novelty of MoIPS is its use of CertificatePolicy extension to carry information necessary for Mobile IP access control. Following are brief descriptions of CertificatePolicy, PolicyMap and PolicyConstraint extensions.

Certificate Policies. In the certificates issued to Mobile Nodes, Corresponding Nodes, Home Agents, and Foreign Agents, this extension specifies the *subnet/domain affiliations* and the *host/agent types* of the subjects according to the rules established in section 3.2. Consequently, peer Mobile IP entities may use the contents of this extension to make service and access control decisions.

The extension contains a sequence of PolicyInformation[3] objects, each of which consists of an object identifier and one or more optional qualifiers. Two types of qualifiers known as CPSPointer[4] and UserNotice were defined. MoIPS chose to use the UserNotice policy qualifiers to encode the access control information. Only the NoticeRef field in the qualifier is used: its Organization component contains either a *CIDR address* or a *domain name*, and the NoticeNumber component gives a sequence of integer codes. The NoticeNumber codes *should* specify the *host type* (MN/CN) and/or the *agent type* (HA/FA) of the subject while the Organization component specifies the subnet or domain to which the host or the agent belong. Because a Mobile IP entity may assume multiple roles, the extension may contain several policy qualifiers, each defining a specific role.

Policy Map. This extension is used with cross certification to establish bindings between certificate policies that can be considered equivalent along different certification paths. Although cross certification occurs within MoIPS CA hierarchies, this extension is *not* needed because the CAs use a homogenous policy encoding scheme.

Policy Constraints. This extension, existing only in CA certificates, is used to enable/inhibit policy mapping and require specific policy identifiers to exist in every certificate along the verification paths. The extension is *not* needed in MoIPS because policy mapping is disabled and policy identifiers are expected to exist only in the certificates for end nodes.

[2] The ASN.1 encoding of mobileipControl object is yet to be determined.

[3] The ASN.1 encodings of PolicyInformation, NoticeRef and NoticeNumber objects for specifying the subnet/domain affiliation and the host/agent type of the Mobile IP entities are yet to be assigned.

[4] CPS is the abbreviation of *certification practice statement.*

Table 2
Profile of MoIPS Certificate Revocation Lists (CRLs).

Fields	Status	Remarks
Version	required	= 1 (X.509 v2 CRL)
Signature	required	RSA (default)
IssuerName	required	= NULL
ThisUpdate	required	GMT date and time of this CRL issue
NextUpdate	required	GMT date and time of next CRL issue
Revoked Certificates		Sequence of revoked certificates each with CRL Entry Extensions
UserCertificate	required	serial number of revoked certificate
RevocationDate	required	GMT date and time of revocation
ReasonCode	optional/non-critical	reason for certificate revocation
HoldInstructionCode	*not used*	
InvalidityDate	*not used*	
CRL Extensions		Extensions common to all revoked certificates
AuthorityKeyIdentifier	optional/non-critical	SHA-1 hash of CA public key
IssuerAltName	required/critical	canonical domain name of CA
CRLNumber	optional/non-critical	CRL serial number
IssueDistributionPoint	optional/non-critical	canonical domain name of CRL distribution point
Delta CRL Indicator	optional/critical	> 0 if this is a delta-CRL

3.5. MoIPS CRL profile

The certificate revocation lists (CRLs) issued by the CAs in MoIPS PKI adopt IETF-PKIX X.509 v.2 format [11]. Table 2 contains a summary of the basic and extensions fields in the CRLs. Since many of the fields are identical to the corresponding ones in MoIPS certificates (section 3.4), this section describes only selected fields unique to the CRLs.

Last Update and Next Update. Similar to the notBefore and notAfter components of the Validity field in certificates, these two fields may contain either UTCTime or GeneralizedTime in Greenwich Mean Time. The ThisUpdate field specifies the issuing time of this CRL, and the NextUpdate field specifies the time by which the next CRL will be issued. Note that the next CRL *may* be issued *before* the time specified in NextUpdate.

3.5.1. Revoked certificates

Each revoked certificate object in the CRL consists of *three* components: UserCertificate (serial number), RevocationDate and zero or more CRL *entry extension(s).*

User Certificate. This field contains the SerialNumber of a revoked certificate. The value of this field combined with IssuerName or IssuerAltName, which *must* be the same in the certificates and their CRLs, *should* uniquely identify the revoked certificate.

Revocation Date. This field specifies the date and time the certificate was revoked. It *should* be in the same format as the other Time fields.

CRL Entry Extension. None of the three entry extensions of X.509 v.2 CRLs, ReasonCode, HoldInstructionCode and InvalidityDate, are used in MoIPS because the Mobile IP nodes and agents are *not* expected to have the ability to interpret different causes and consequences of certificate revocation.

3.5.2. CRL extensions

CRL Number. Similar to the SerialNumber field in the MoIPS certificates, this extension contains a monotonically increasing unsigned integer that uniquely identifies each CRL issued by a specific CA.

Delta CRL Indicator. This extension indicates that the CRL is a delta-CRL. A delta-CRL is issued when the key of a Mobile IP node or agent is compromised and immediate remedial actions are needed. A delta-CRL is always issued along with a complete CRL. The delta-CRL will be pushed down to the relevant nodes using certificate discovery protocols (CDPs).

3.6. Certificate hierarchy

The MoIPS certificate hierarchy takes the form of a multiple-tree structure (figure 3). Each tree has a top-level certification authority (TLCA) at the root, zero or more layers of middle-level CAs (MLCAs), and a layer of low-level CAs. Each low-level CA owns one or more contiguous blocks of IP addresses, and is responsible for issuing MoIPS certificates to the Mobile IP nodes and agents with their IP addresses falling in the ranges. Different low-level CAs may be dedicated to issue certificates to Mobile Nodes, Corresponding Nodes, and Mobility Agents according to different security policies (shown as dashed arrows in figure 3).

TLCAs and MLCAs in different trees may be linked by *cross certificates*. These cross certificates establish the verification paths between leave certificates in different trees.

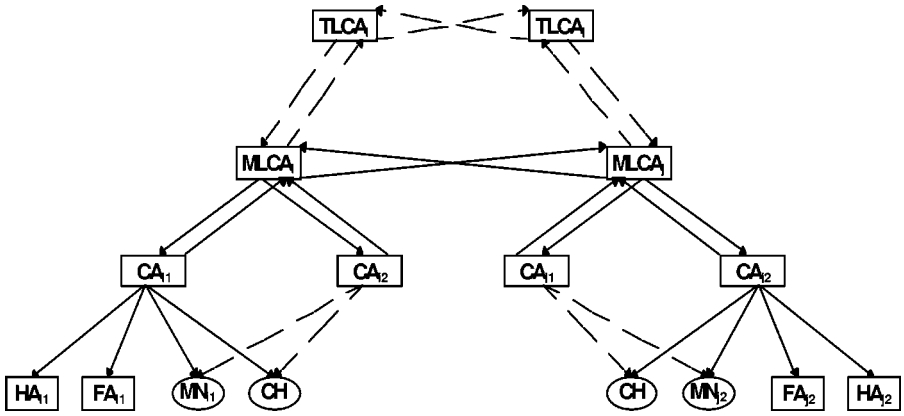

Figure 3. CA hierarchy fo MoIPS certificates.

During initialization, every Mobile IP node and agent is loaded with a *self-signed certificate*[5] of its CA. When one of these entities wants to retrieve the public key of another entity, it *should* retrieve the certificates along the verification path between its CA and the target entity. By using the public key contained in the self-signed certificate of its CA, the entity can verify all the certificates along the path and retrieve the public key of the target entity.

3.7. DNS based certificate dispatch

The MoIPS certificates are stored in new types of resource records in both DNS and DNSSec systems: type X509CCRL in DNS and type CERT in DNSSec [6].

In the CA certificates, the canonical domain name (as SubjectName) *should* refer to their DNS entries, which also serve as the certificate distribution points.

In the MoIPS certificates, IP addresses of Mobile IP nodes and agents are used as the subject names. These certificates *should* be stored in the DNS entries under the canonical domain names of these entities. These names can be discovered by a reverse DNS look-up using the IP addresses in the subject names. If an entity has multiple IP addresses then its DNS entry may contain multiple certificate resource records, each maintaining a MoIPS certificate issued to one of its IP addresses.

The MoIPS CRLs are stored in the DNS entries specified by the CAs. X.509 v.3 certificates allow their CRLs to be stored in parts at distribution points specified by the CRLDistributionPoint extension. The DNS entries of network administrative workstations may be the ideal sites for maintaining MoIPS CRLs. In the prototype, the CRLs are kept in the DNS entries of their issuing CAs for simplicity sake.

There are soft limits on sizes and offsets of resource records in a DNS entry. To honor those limits, X509CCRL and CERT records should be stored at the end of a DNS entry without pointer reference. The resource records may be *compressed* if the total size of a DNS entry exceeds 64 K bytes or individual certificates/CRLs are larger than 500 bytes, which is the maximum payload size of a UDP-based DNS lookup. Otherwise, the certificates or the CRLs must be fetched using TCP sessions.

3.8. Direct certificate exchanges

The X.509 certificates and CRLs can also be sent to the requesting entity using IPSec certificate exchange protocol (CDP) [3] because the authenticating entities may engage in real-time communication. However, DNS lookup is preferred because it provides a global distribution and caching mechanism.

Direct exchange of certificates and CRLs may be used to cope with particular incidents of *certificate revocation*. In order to shorten the wait time between the act of revocation and the publication of the next CRL, MoIPS PKI may dispatch a delta-CRL through the Mobility Agents to the Mobile Nodes.

4. Protection of control messages

As stated in section 2.1.1, basic and route optimized Mobile IP protocols need to use short-term symmetric keys to authenticate their registration and location update messages. In this section, we present a key generation algorithm that can derive the necessary keys from Diffie–Helman (DH) public values carried in the certificates issued to Mobile IP nodes and agents. The algorithm does not require additional information exchanges besides fetching the certificates of corresponding entities, and was thus known as the "zero-message" key generation algorithm. The key generation operation can be separate from production and verification of the authentication tags of Mobile IP control messages. Also, the generated keys should not and need not be revealed to the Mobile IP module.

[5] A *self-signed certificate* of a CA is a X.509 certificate signed by the CA using its own private key to bind its subject name with public key parameters and other relevant attributes. The certificate serves well as an end point of a verification path because (1) it does not refer to another signing authority, and (2) it provides integrity protection (but not authentication) of the name-key binding.

4.1. Design goals

The key generation algorithm was designed to satisfy the following *five* requirements:

1. *Usable by all Mobile IP nodes and agents* – unlike manual key installation, which can only establish symmetric keys between limited pairs of Mobile Nodes (MNs) and their Home Agents (HAs), this algorithm can be used to establish shared symmetric keys between any MN–CN, MN–FA and MN–HA pairs.
2. *No modification of Mobile IP message formats* – besides certificate fetches, the algorithm does *not* require additional communication between the authenticating entities nor any modification to the format of Mobile IP registration and binding update messages.
3. *No use of encryption operations* – unlike SKIP [2], the algorithm is free of encryption operation, and hence is not subjected to export restriction.
4. *Strong protection of master keys* – the algorithm was designed to make the discovery of DH symmetric secrets based on the knowledge of generated keys, DH public values and/or replay protection nonce as difficult as random guesses.
5. *Weak correlation with other Diffie–Helman based key generation* – the algorithm was designed to be different from other key generation algorithms using the Diffie–Helman key agreement technique, esp. those incorporated into Transport Layer Security (TLS) [5] and Internet Key Exchange (IKE) [9] protocols. The difference in algorithm design ensures weak correlation among the keys generated by these algorithms.

4.2. Underlying technology

In order to fulfil the above design requirements, the algorithm employs the following three security techniques.

Diffie–Helman secret sharing algorithm

The algorithm chose to use the Diffie–Helman key agreement algorithm to generate the long-term shared secrets because of the computation simplicity of discrete exponentiation and the ability of the algorithm to limit the damages caused by the compromise of private keys. The algorithm also relies on X.509 v.3 certificates to distribute the DH public values.

Timestamp or nonce replay protection

In order to eliminate the need of passing transient values between the authenticating entities, the algorithm uses the *replay protection identification numbers* in the Mobile IP control (registration and binding update) messages as the transient values for key generation. The replay protection identification number is suitable for this purpose owing to two reasons:

1. The identification number (either as a 64-bit timestamp or as a pair of 32-bit pseudo-random nonces) is designed to be different for every control message sent from a Mobile IP entity and have a very low probability of repetition. Its transient and non-repetitive nature makes it suitable to be the changing argument for key generation.
2. Mobile IP protocol mandates that a control message must be discarded if the identification number in the message was found to repeat the identification of a previous message within a set period. This policy further reduces the damage that would be caused by reusing the short-term keys generated by this algorithm.

HMAC one-way hashing function

The algorithm employs the HMAC-MD5 one-way hashing function [16] as the generation function of pseudo-random keys. The HMAC function was designed originally for keyed message authentication, but we adopted it for this purpose because, like MD5, it offers good randomization of output patterns with efficient software implementation and better than MD5, it separates the key as a distinct argument and provides stronger protection against its discovery.

4.3. Key generation algorithm

The algorithm generates short-term keys for two authenticating parties that share a DH symmetric secret by feeding a folded version of the DH secret as the "key" and a finite repetition of the replay protection identification number as the "message" into a HMAC function. The output of the HMAC function is then used to authenticate a Mobile IP control message by feeding the message and the control message again into a HMAC function.

The algorithm can be divided into three steps: computation of long-term master key, preparation of transient values, and production of short-term keys.

Master keys

The algorithm begins by computing the symmetric secret $S_{i,j}$ based on the Diffie–Helman private values i, j and the public values $g^i \bmod p$, $g^j \bmod p$ possessed by the two authenticating entities:

$$S_{ij} = \left(g^i\right)^j \bmod p = \left(g^j\right)^i \bmod p.$$

The long symmetric secret S_{ij} is then "folded" by the following operation to produce the *long-term master key* K_{ij}:

$$K_{ij} = \overset{M}{\oplus} [S_{ij}]_{Lk} \quad \text{with } M = \left\lceil \frac{L(S_{ij})}{Lk} \right\rceil.$$

The folding begins with the breaking down of S_{ij} (starting from its lowest order bits) into fragments of length Lk equal to that of the short-term keys to be generated. In case the last fragment is shorter than Lk then a fixed pattern of $55_{16} = 01010101_2$ will be padded repeatedly beyond the

highest bit. After the fragmentation, a series of *exclusive OR operations* are performed iteratively to the fragments in ascending order starting with the one with lowest order bits. The long-term master key K_{ij} is yielded as the final result of the operations.

Transient values

A 512-bit transient value T_n is prepared by *eight* repeated concatenation of the 64-bit *replay protection identification number* R_n embedded in the Mobile IP control messages:

$$T_n = \overset{8}{|}\, R_n.$$

The purpose of the repeated concatenation is to increase the length as well as the number of changing bits in the transient value to be fed into the HMAC function. This step is particularly important if R_n is derived from a timestamp with many slow changing bits. Nevertheless, the concatenation does *not* increase the total number of transient values and hence the total number of short-term keys which can be generated. If the replay protection numbers are 64 bits in length then a total of 2^{64} different keys can be generated for each pair of communicating parties that share a DH secret.

Short-term keys

Once the long-term master key K_{ij} and the transient value T_n are prepared, they are fed into the HMAC function for generating the short-term key K_{auth}. The *default* HMAC function, expressed below, uses MD5 as the basic, one-way hash function:

$$\begin{aligned} K_{\text{auth}} &= \text{HMAC}(K_{ij}, T_n) \\ &= \text{MD5}\big(K_{ij} \oplus P_1 \mid \text{MD5}(K_{ij} \oplus P_2 \mid T_n)\big), \end{aligned}$$

where

$$P_1 = \overset{48}{|}\, 36_{16} \quad \text{and} \quad P_2 = \overset{48}{|}\, 48_{16}$$

are two constant paddings *Xored* with K_{ij}. If more protection is desired, the MD5 function used in the expression can be replaced by SHA-1 function, which is more suitable for pseudo-random number generation. Then,

$$P_1 = \overset{64}{|}\, 36_{16} \quad \text{and} \quad P_2 = \overset{64}{|}\, 5C_{16}$$

will be the two constant paddings. Note that the values of $K_{ij} \oplus P_1$ and $K_{ij} \oplus P_2$ can be pre-computed as suggested in [17].

5. IPSec protection of packet redirection

Another function of the MoIPS system is to offer IPSec data integrity, origin authentication and data confidentiality services to the IP datagrams redirected by Mobile IP. When implemented on selected packet tunnels, these security services enable the Mobile Nodes to enjoy the same network connectivity (with possible performance degradation) and communication privacy as when they were attached to their home networks. These services also augment the firewall traversal guidelines proposed by Montenegro and Gupta [22] to pass redirected datagrams through the firewalls defending Mobile Node's home and visiting foreign networks.

The protection *should* be provided by a combined implementation of IPSec and Mobile IP protocols. Such an implementation allows a single IP–IP encapsulation to be used for both IPSec protection and Mobile IP packet redirection (except the case of MN–HA tunneling). This approach enables FAs and HAs to behave as IPSec supporting security gateways. Separate implementations of Mobile IP and IPSec protocols following IPSec "bump-in-the-stack" or "bump-in-the-wire" approaches will introduce extra IP encapsulations.

5.1. Use of MIP-IPSec tunnels

Due to different options existing in Mobile IP (particularly, the use of *reverse tunneling* and the choice between *co-located or foreign-agent care-of addresses*), IP tunnels can be established between different pairings of Mobile Nodes and Mobility (Home and Foreign) Agents using either full or minimal IP–IP encapsulations. Any of these tunnels can be protected by IPSec protocol. Table 3 lists the possible tunnels.

Among possible IPSec tunnels, the MN–CN pairs are end-to-end tunnels that may exist regardless of Mobile IP. We recommend to use them whenever end-to-end security is needed. The remaining *three* pairs of tunnels, HA–FA, MN–HA and MN–FA, are created primarily for Mobile IP

Table 3
IPSec tunnels to be used with Mobile IP*.

	∼C ∼R	C ∼R	∼C R	C R
CH → HA HA → CH				
HA → FA FA → HA	√		√ √	
FA → MN MN → FA	√		√ √E	
HA → MN MN → HA	√	√	√ √	√ √
CH → FA FA → CH				
CH → MN MN → CH	+	+	+	+

*Table 3 notations: C and ∼C mark the use of co-located or Foreign Agent bounded Care-of Address (COA); R and ∼R mark the use or not use of reverse tunneling; √ marks the packet redirecting tunnels; + marks the end-to-end tunnels between communicating hosts; E marks the tunnel existing only when the MN–FA encapsulation is used in reverse tunneling; finally, the dark shade marks an un-intended use of reverse tunneling flag to select an MN–HA tunnel with co-located COA while reverse tunneling is used primarily with FA-COA.

packet redirection. Their uses are studied in the following paragraphs.

FA–HA tunnels

The MIP-IPSec tunnel going from a Home Agent to a Foreign Agent (and from a Foreign Agent to a Home Agent if reverse tunnel and FA-COA are used) are the easiest ones to establish. They can be implemented by adding IPSec protection to the Mobile IP tunnels.

These tunnels provide a *virtual private network* (VPN) connection between the home network and the foreign network visited by the Mobile Node. The most notable value of using the FA–HA tunnels is perhaps its use in firewall traversal. By configuring the firewalls in the foreign networks as Foreign Agents and setting up the FA–HA tunnels, we created authenticated communication paths through the firewalls. With the knowledge of Mobile Nodes, the FA/firewalls can easily screen the packets.

MN–HA tunnels

The MN–HA IPSec tunnels are the most useful ones as they provide a secure communication channel between a Mobile Node and its home network. Data integrity and origin authentication prevent active attacks while data confidentiality prevent passive eavesdropping by adversaries in the foreign network and/or the open Internet. These are the necessary tunnels that enable a Mobile Node to obtain the same connectivity as it has at home.

The MN–HA tunnels are more expensive to establish. Since they are not a part of the packet redirection mechanism, they must be built separately using the procedure described in section 5.2.

MN–FA tunnels

The MN–FA IPSec tunnels can be used in two ways if there is no link-layer mechanism providing the services: (1) data confidentiality for the Mobile Node over the foreign network, and (2) data origin authentication of MN–FA exchange. However, the MN–FA tunnels exist only if the Mobile Node chooses to use a Foreign Agent Care-of Address and re-encapsulate the IP datagrams. Hence, these tunnels are also expensive to build and should be replaced by MN–CN or MN–HA tunnels whenever possible.

Tunnel use in firewall traversal

The MIP-IPSec tunnels do *not* offer a complete solution to the firewall traversal problem encountered by packets redirected by Mobile IP. In particular, they cannot hide unknown source or destination addresses caused by the use of private network addresses. Nevertheless, they allow the firewalls and the Mobility Agents to work together in the following two ways:

1. A firewall may permit MIP-IPSec tunnels to pass through and terminate on Mobility Agents lying in the network under its protection.
2. A Mobility Agent may function as a firewall by operating as both an end-point of MIP-IPSec tunnels and a packet filtering node. This composite function is particularly important in protecting the home network from the intrusions launched from the foreign network and vice versa.

However, this network architecture carries certain assumptions:

- Both Foreign Agents and Home Agents can function as *IPSec supporting security gateways* capable of performing *encryption/decryption* and *packet filtering*.
- In a foreign network, the Foreign Agents *should* be the firewalls *closest* to the Mobile Nodes. Other firewalls on the network *should* pass IPSec protected packets to and from the Foreign Agents. Reverse tunneling must be used if INGRES source filtering is employed by the firewalls.
- The Home Agents *should* also function as the *innermost* firewall guarding the home network. Again, other firewalls on the network *should* pass IPSec protected packets to and from the Home Agents.

5.2. Establishment of MIP-IPSec tunnels

The MIP-IPSec tunnels are established in two steps: (1) the mutual agreement among Mobile Nodes, Foreign Agents and Home Agents on the selection of tunnels, and (2) the negotiation of security associations using ISAKMP.

5.2.1. IPSec tunnel selection

The selection process aims at arriving at an agreement among Mobile Nodes, Foreign Agents and Home Agents on the tunnels to be protected with IPSec.

To reduce communication overhead, the exchanges will be carried as extensions to the Mobile IP control messages. The *IPSec Tunnel Selection* extension may be added to Agent Solicitation, Agent Advertisement and Registration Request messages. Figure 4 shows the formats of these extensions. Each of these extensions contains several one-bit fields indicating whether IPSec protocols should be used to protect specific packet redirection tunnels. A total of *six* tunnels may be protected using IPSec with a pair going in opposite directions between MN–HA, MN–FA and FA–HA. The tunnels are marked by F or R: F (means *forward*) always marks the tunnels going from the Home Agents towards the Mobile Nodes via the Foreign Agents and R (means *reverse*) marks the ones going in the opposite direction.

Figure 5 displays the order of message exchange for conducting the tunnel selection. The sequence follows the basic steps of a Mobile IP registration with decisions made at both Mobile Nodes and Home Agents.

A Mobile Node *should* select the IPSec tunnels between itself and the Foreign Agent based on the exchange of Agent Solicitation and Agent Advertisement messages. It *should*

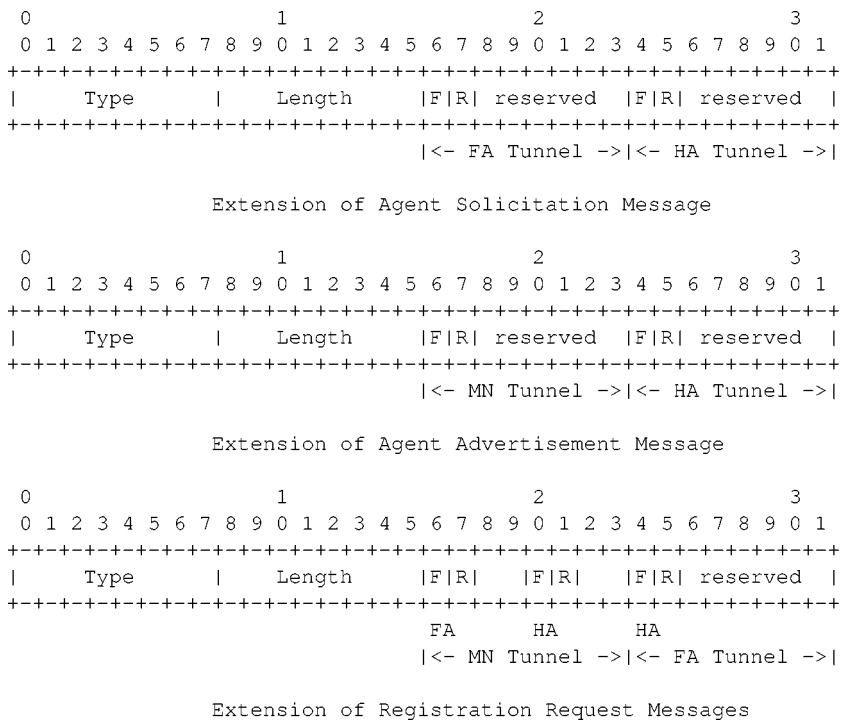

Figure 4. IPSec tunnel selection extension of Mobile IP registration messages.

also select the MN–HA tunnels according to its security policy.

The Mobile Node includes a Tunnel Selection extension in its Registration Request message to indicate its tunnel choices. Upon reception, the Foreign Agent *must* compare its own choices of IPSec tunnel with the choices appearing in the extension. The Foreign Agent *must* send a Registration Reply message containing a TunnelSelectionConflict error code to the Mobile Node if it disagrees with the choices. Otherwise, the Foreign Agent *must* forward the Registration Request to the Home Agent.

After receiving the Registration Request, the Home Agent checks the tunnel choices against its security policy and decides whether to reject any of the choices. It then sends the Registration Reply which contains a code indicating approval or rejection of the tunnel choices.

In the Registration Reply message, two types of codes are used to explain the rejection of tunnel choices. The TunnelSelectionUnsupported code indicates a difference between the tunnel choices and Mobile IP care-of address and reverse tunnel modes. The TunnelSelectionConflict code indicates a mismatch between the tunnel choices and the security policies of the Home Agents or the Foreign Agents.

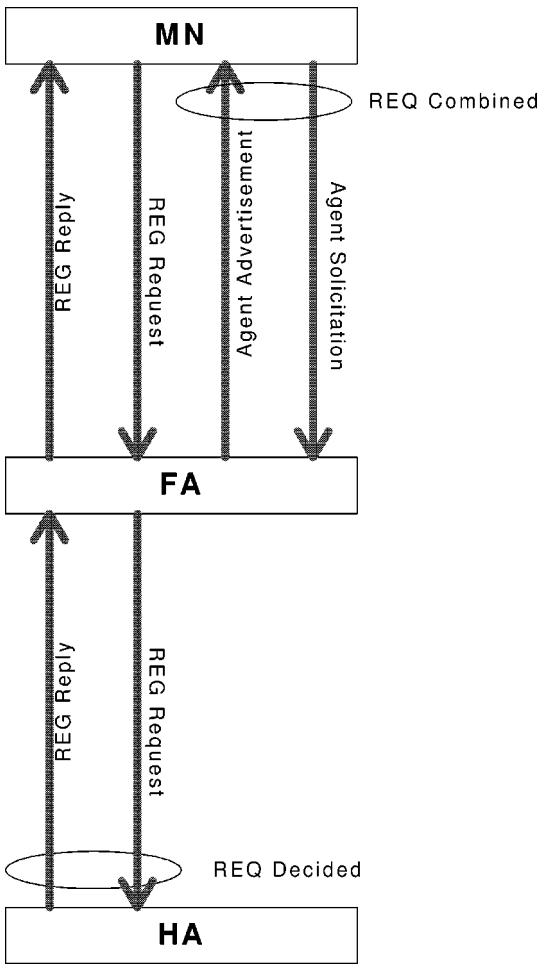

Figure 5. Message exchanges for IPSec tunnel selection.

5.2.2. Security association negotiation

The tunnel selection extension only specifies the use of IPSec, but they do not choose AH/ESP protocols, the security services, or the security mechanisms to be used. All these negotiation of security protocols and mechanisms *should* be conducted by the security association and key management protocol (ISAKMP) based on a proper domain

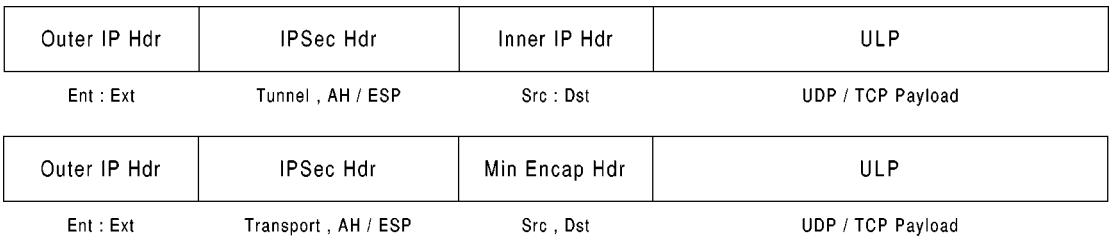

Figure 6. MIP-IPSec encapsulation formats.

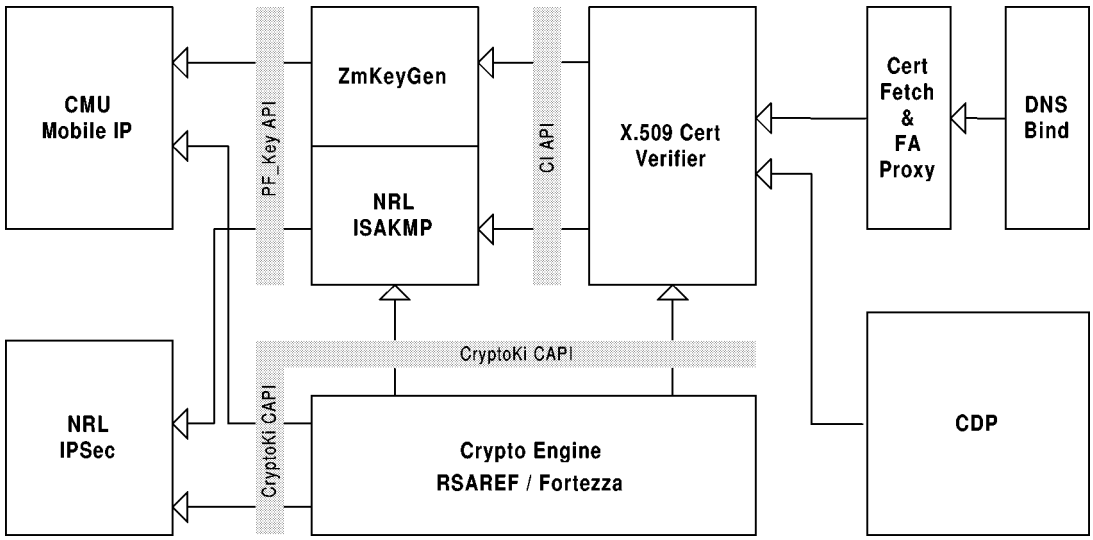

Figure 7. Block diagram of first MoIPS prototype.

of interpretation (DOI). Currently, we use IP-DOI [26] to specify the ISAKMP encoding for Mobile IP.

The use of ISAKMP inevitably complicates the packet redirection process because the negotiation of security associations may fail *after* a successful Mobile IP registration. The failure of an ISAKMP negotiation does not block the packet redirecting tunnels, but it does mean the absence of IPSec protection. Error messages *must* be generated and logged in these cases.

5.3. Encapsulation of MIP-IPSec packets

Both Mobile IP packet redirection and IPSec protection in MN–FA and FA–HA tunnels *should* be implemented by a single IP–IP encapsulation. In the case of MN–HA tunneling, the IPSec tunnel *must* be established within Mobile IP tunnels. Thus, a pair of IP and IPSec headers must be inserted between the outermost and the original IP headers.

The IP encapsulation (except the inner ones for the MN–HA tunnels) can be implemented using either full IP–IP encapsulation [24] or minimal IP–IP encapsulation [25]. The formats of the two encapsulations are shown in figure 6. Note that IPSec tunnel mode can *not* be used with minimal encapsulation because the encapsulation scheme replaces the inner IP header with a special header.

6. Prototype implementation

The MoIPS project team delivered the first prototype in August 1997. The prototype is built upon FreeBSD UNIX v.2.2.1 and marked by the following features:

- the capability of obtaining x.509 certificates and CRLs from DNS as X509CCRL resource records[6] (via an FA proxy in the case of Mobile Nodes), certificate discovery protocol (CDP) and pre-installed files,
- the capability of verifying x.509 certificates and CRLs (including cross certificates between CAs) by following the multi-tree hierarchy shown in figure 3,
- the ability of authenticating the registration messages of IETF Mobile IP using symmetric keys produced by the zero-message key generation algorithm,
- the integration of MN–CN IPSec tunnels (in transport mode) with Mobile IP packet redirection – the use of only end-to-end IPSec with Mobile IP tunnels avoids temporarily the need of IPSec tunnel selection and special DOI.

Figure 7 displays the block diagram of the prototype. The following paragraphs provide brief description of individual modules.

[6] Fetching of CERT resource records from DNSSec was *unavailable* at the time of prototype implementation and thus was *not* implemented.

```
u_char *MakeMasterKey(u_char *public,unsigned publen,
                      u_char *private, unsigned privlen,
                      u_char *p, unsigned plen,
                      unsigned st_key_len);

u_char *MakeShortTermKey(int alg,
                         u_char *master_key,
                         u_char *ident);
```

Figure 8. Interface functions to zero-message key generation module.

Mobile IP module

The MoIPS prototype was built upon an IETF RFC2002 compliant Mobile IP implementation developed by Prof. David Johnson's team in Carnegie-Melon University [13]. The version currently used was v.1.0.2 on FreeBSD v.2.2.1. The Mobile IP implementation was slightly modified in order to integrate with the zero-message key generation module.

Zero-message key generation module

This module supplies the short-term symmetric keys necessary for authenticating Mobile IP registration and binding update messages. It implements the "zero-message" key generation algorithm (section 4.3) and offers two interface functions (figure 8). The function `MakeMasterKey` creates a master key of length `st_key_len` from a pair of Diffie–Halmen public and private keys, and the function `MakeShortTermKey` derives a short-term authentication key from `master_key` and the eight byte replay protection identifier `ident`.

IP security and ISAKMP key management modules

The IPSec module is a FreeBSD port of NRL IPSec [v.α3] implementation from Portland State University Secure Mobile Networking Team [19]. This module also uses the `ipkey` utility in NRL IPSec for manual key input.

The ISAKMP module is a FreeBSD port of Cisco's ISAKMP/OAKLEY [v.α5] implementation. It uses a cryptographic library from Cylink Inc. and can conduct key negotiation only using manually inserted DSS keys. In the final release of the prototype, the ISAKMP module will be interfaced to the *certificate verifier* (CV) module so that it can obtain DSS keys from X.509 certificates stored in DNS entries.

X.509 certificate verifier

The certificate verifier (CV) consists of a UNIX process daemon and an interface library. ISAKMP or ZmKeyGen modules are the clients of CV. They can obtain public keys, certificate fields or even complete certificates from CV by using the function calls in the interface library.

Whenever possible, the CV daemon responds to these requests based on information stored in its internal certificate database. If the requested information is not available in the database then the daemon will take the following steps in order: (1) obtains the certificates using the Certificate Fetcher, (2) verifies the certificates (also obtaining and verifying the CA certificates if necessary), and (3) caches the verified certificates in an internal database. The chains of certificates verified by CV are always ended with self-signed certificates.

Since the module has been developed some time ago, it can only parse an essential subset of extensions that were specified in an older version of the X.509 profile [12]:

- for v.3 certificates: KeyUsage, SubjectAltName, IssuerAltName, BasicConstraint, NameConstraint,
- for v.2 CRL: IssuerAltName, CRLNumber.

The verification procedure for each certificate is conducted in the following steps: (1) check signature, (2) check dates, (3) check CRL, and (4) check KeyUsage extension.

When a certificate is verified, CV does not check for the required extensions as part of the verification process. Instead, when a certificate or key is requested by an application, the CV uses the required extensions array to identify which extensions are currently required by the application, and checks the certificate against the array. If the certificate does not have the required extensions, a `MissingExtensions` error code is returned, but the certificate is still considered valid.

Cryptographic engine

The MoIPS prototype uses RSAREF as the default cryptographic library, but chooses to interface with the library via the RSA PKCS#11 CryptoKi CAPI. This design decision allows the prototype to be compatible with other cryptographic processing support such as the Fortezza hardware tokens.

The CryptoKi is a low-level session-oriented CAPI, which was thoroughly documented in [27,31]. In order to hide some tedious low-level function calls from its clients, the MoIPS prototype implemented three "wrapper" functions [10], `CR_Initialize`, `CRHMAC`, `CR_ImportKey`. Function `CR_Initialize` was written so that the complicated multi-step initialization of the cryptographic module can be accomplished in one function call made at the start of MoIPS application. Function `CR_HMAC` was written to compute a keyed one-way hash using the new HMAC algorithm, which was not available in CryptoKi, and `CR_ImportKey` was written to import a key generated outside of the CryptoKi library.

Currently, Mobile IP, CV and ZmKeyGen modules all use the CryptoKi CAPI. IPSec and ISAKMP modules, on the other hand, uses built-in cryptographic functions.

Certificate fetcher and foreign agent proxy

The DNS Certificate Fetcher connects CV to the DNS daemon either directly or indirectly via the Foreign-Agent proxy. In both cases, the two modules work together to provide CV with X509CCRL resource records extracted from DNS entries.

The FA/DNS proxy was designed to solve a "chicken-and-egg" problem arising at a secure Mobile IP registration process. In order to authenticate the registration messages, the Mobile Node, the Home Agent and probably the Foreign Agents *must* establish shared cryptographic keys. However, without pre-arrangement, the key sharing can only be accomplished via exchanges of certificates, which in turn must use network connectivity the registration process aims at establishing. The proxy mechanism provides a work-around for this problem. It permits the Mobile Node to access certificate and CRL information stored in the DNS *before* it attempts to register. The proxy consists of a simple client/server model and request/ response protocol in which the Foreign Agent may receives requests to fetch certificates and CRLs for the Mobile Node, and provide this service before the completion of Mobile IP registration.

Certificate discovery protocol executive

The CV module can use DNS or CDP to fetch certificates and CRLs. In MoIPS, CDP was used to receive CRLs as they are pushed to the nodes in urgent cases.

7. Conclusions

In this paper, we describe a public key management architecture which can satisfy the security requirements of Mobile IP by authenticating Mobile IP control messages and protecting packet redirection with IPSec protocols. Its first prototype was completed in August 1997, and was tested later on a testbed consisting of one Home Agent, five Foreign Agents (three on BBN Cambridge Campus and two in the developer's homes) and five wireless Mobile Nodes. Both authenticated registration and end-to-end IPSec tunneling has been successfully demonstrated and is currently being used.

The system may have many promising applications including scaleable implementations of secure route-optimized Mobile IP and IPSec supported virtual private networking of Mobile IP traffic. The project shall be supplemented with future work on fast and hierarchical location management and efficient management of security associations based on security policies of network domains.

References

[1] R. Atkinson, Security architecture for the Internet protocol, RFC1825, IETF Network Working Group (August 1995).

[2] A. Aziz, Simple Key-management for Internet Protocol (SKIP), <draft-ietf-ipsec-skip-06>, IETF IP Security Working Group (November 1995).

[3] A. Aziz, T. Markson and H. Prafullchandra, Certificate discovery protocol, <draft-ietf-ipsec-skip-06>, IETF IPSec Working Group (November 1995).

[4] S.M. Bellovin, Security problems in TCP/IP protocol suite, ACM Computer Communications Review 19(2) (March 1989).

[5] T. Dierks and C. Allen, The TLS protocol version 1.0, RFC2246, IETF Network Working Group (January 1999).

[6] D.E. Eastlake III and O. Gudmundsson, Storing certificates in the domain name system, <draft-ietf-dnssec-certs-01>, IETF DNS Security Working Group (November 1997).

[7] D.E. Eastlake III and C.W. Kaufman, Domain name system security extensions, <draft-ietf-dnssec-secext-06>, IETF DNS Security Working Group (October 1995).

[8] S. Garfinkel, *PGP: Pretty Good Privacy* (O'Reilly and Associates, 1995).

[9] D. Harkins and D. Carrel, The Internet Key Exchange (IKE), RFC2409, IETF Network Working Group (November 1998).

[10] P. Helinek, N. Yuan, M. Condell and J. Zao, Security architecture for global host mobility, Quarterly Technical Report #6, BBN (February 1997).

[11] R. Housley, W. Ford, W. Polk and D. Solo, Internet public key infrastructure, Part I: X.509 certificate and CRL profile, <draft-ietf-pkix-ipki-part1-06>, IETF PKIX Working Group (October 1997).

[12] R. Housley, W. Ford and D. Solo, Internet public key infrastructure, Part I: X.509 certificate and CRL profile, <draft-ietf-pkix-ipki-part1-02>, IETF PKIX Working Group (June 1996).

[13] D.B. Johnson, The CMU monarch project, http://www.monarch.cs.cmu.edu/

[14] D.B. Johnson and C. Perkins, Route optimization in MIP, <draft-ietf-mobileip-optim-03>, IETF Mobile IP Working Group (November 1995).

[15] P. Karn and W.A. Simpson, Photuris session key management protocol, <draft-ietf-ipsec-photuris-08>, IETF IP Security Working Group (November 1995).

[16] H. Krawczyk, M. Bellare and R. Canetti, HMAC-MD5: Keyed-MD5 for message authentication, <draft-ietf-ipsec-hmac-md5-03>, IETF IP Security Working Group (March 1996).

[17] H. Krawczyk, M. Bellare and R. Canetti, HMAC-SHA-1: Keyed-SHA-1 for message authentication, <draft-ietf-ipsec-hmac-sha1-03>, IETF IP Security Working Group (March 1996).

[18] D. Maughan, M. Schertler, M. Schneider and J. Turner, Internet Security Association & Key Management Protocol (ISAKMP), <draft-ietf-ipsec-isakmp-07>, IPSec Working Group (February 1997).

[19] J. McHugh and J. Binkley, The Portland State University Secure Mobile Networking Project, http://www.cs.pdx.edu/research/smn.

[20] P.V. Mockapetris, Domain Names: Concepts and facilities, RFC1034 (November 1987).

[21] G. Montenegro, Reverse tunneling for Mobile IP, <draft-ietf-mobileip-tunnel-reverse-02>, IETF Mobile IP Working Group (March 1997).

[22] G. Montenegro and V. Gupta, Firewall support for Mobile IP, <draft-montenegro-firewall-sup-03>, IETF Mobile IP Working Group (January 1998).

[23] C. Perkins, ed., IP mobility support, RFC2002, proposed standard, IETF Mobile IP Working Group (October 1996).

[24] C. Perkins, IP Encapsulation within IP, RFC2003, proposed standard, IETF Mobile IP Working Group (October 1996).

[25] C. Perkins, Minimum encapsulation within IP, RFC2004, proposed standard, IETF Mobile IP Working Group (October 1996).

[26] D. Piper, The Internet IP security domain interpretation for ISAKMP, <draft-ietf-ipsec-ipsec-doi-06>, IPSec Working Group (November 1997).

[27] Public Key Cryptographic Standard No. 11 – Cryptoki. V.1.0, RSA Laboratories (April 1995).

[28] J.K. Zao and M. Condell, Use of IPSec in Mobile IP, <draft-ietf-mobileip-use-01>, IETF Mobile IP Working Group (November 1997).
[29] J. Zao, J. Gahm and M. Condell, Security architecture for global host mobility, Quarterly Technical Report #5, BBN (October 1996).
[30] J. Zao and S. Kent, New key generation algorithm for Mobile IP control message authentication, MoIPS Quarterly Technical Report #3, Sect. 4, BBN Corp. (April 1996).
[31] N. Ziring and D.E. Peele, Programming with Cryptoki, an object-oriented approach, NSA Technical Report (January 1996).

John K. Zao is a Senior Scientist in the Network Security Department at BBN and a member of the Security Practice Center of GTE Internetworking. He is the principal investigator of two DARPA-funded projects, "Security Architecture for Global Host Mobility" (MoIPS) and "Policy Based Security Management" (PBSM). Before coming to the United States, he worked in Canada for six years in communication and security system design. Dr. Zao obtained his Ph.D. degree in computer science from Harvard University in 1995. His current research interests include security policy management and mobile internet security.

Stephen T. Kent is the Chief Scientist in information security of BBN and the Chief Technology Officer of GTE Internetworking CyberTrust Solutions Inc. Dr. Kent served as a member of the Internet Architecture Board (1983–1994), and chairs the Privacy and Security Research Group of the Internet Research Task Force (1985–), both under the auspices of the Internet Society. He has chaired the Privacy Enhanced Mail (PEM) working group and is now co-chairing the Public Key Infrastructure (PKIX) working group, both of the Internet Engineering Task Force. He also served on the Presidential SKIPJACK Review Panel (1993–1994) and on the National Research Council Secure Systems Study Committee. Dr. Kent works with government and commercial programs and consults on system design issues. He has acted as system architect in the design and development of several network security systems for the Department of Defense and served as the principal investigator on many network security research and development projects. He has also lectured extensively in the United States, Europe and Australia on computer communication security. His current work includes the design and development of X.509 based public-key certification infrastructures, IP security protocols and internet routing infrastructure security.

Joshua Gahm. Photograph and biography not available at time of publication.

Gregory D. Troxel received his S.B., S.M., E.E. and Ph.D. degrees from the Massachusetts Institute of Technology. He is a Senior Scientist at BBN Technologies, where he is the technical leader of a project exploring mobility, security, QoS, and routing for airborne routers. Dr. Troxel's research interests include clock synchronization, security and mobility.

Matthew Condell received his Bachelor of Science and Master of Engineering degrees from the Massachusetts Institute of Technology in 1996. He has worked on Secure Mobile IP and Policy Based Security Management projects at BBN Technologies.

Pam Helinek. Photograph and biography not available at time of publication.

Nina H. Yuan is a Senior Consultant in the New England Area Information Security and Electronic Commerce Services practice at Ernst & Young LLP. Prior to joining Ernst & Young in 1998, she was a Staff Engineer in the BBN Technologies Network Security Department (currently, GTE Internetworking CyberTrust Solutions). She received her Bachelor's degree in applied mathematics from Harvard University in 1994 and will receive her Master's degree in computer science from Harvard in 1999. Her research interests include IP security, public-key infrastructure technology, operating systems security and distributed systems security.

Isidro Castineyra. Photograph and biography not available at time of publication.