Security In Mobile IP

Jian Hui Wang

A Major Report

In

The Department

Of

Computer Science

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Computer Science

Concordia University

Montreal, Quebec, Canada

April 2003

Abstract

Security In Mobile IP

Jian Hui Wang

With the rapid development of wireless networks, mobile computing has become a reality. Mobile IP provides a framework for transparent mobility based on the existing IP protocol, which allows the mobile user to use the same IP address to communicate with others, even when the mobile user changes its point of attachment to the network. However, Mobile IP is subject to a variety of security threats. Some originate from Internet Protocol (IP) vulnerabilities; some are rooted in the Mobile IP protocol itself. IPSec is an IP-layer security solution for network traffic, which provides extensive services for confidentiality, authentication, and integrity of data communication. Mobile IP data communication can be protected by integration with IPSec. This report describes the security issues in Mobile IP, what IPSec is, and how and why IPSec can solve Mobile IP security problems. This report also describes the details of SecMIP, a prototype we have created to demonstrate the integration of IPSec with Mobile IP.

Acknowledgements

I would like to thank Professor Lata Narayanan for the great support and valuable advice that I had during my work. Without her help, I couldn't have finished this report so well. I am also grateful to Dr. Atwood for his important suggestions for my report. I am also thankful to Halina and all my friends, for their excellent support for my studies.

Table of Contents

List of Figures

1 Introduction

In recent years, an increasing variety of wireless devices such as PDAs, cellular phones and handheld devices have been developed rapidly and wireless networks are becoming as ubiquitous as wired networks [6]. Cell phones offer users freedom of movement. Personal Digital Assistants (PDA) allow individuals to access e-mail anywhere. Some technologies even offer Global Positioning System (GPS) capabilities that can pinpoint the location of the device anywhere in the world [7]. Wireless technologies promise to offer even more features and functions in the next few years. One of the main advantages of wireless technologies is mobility, which means a mobile user can always connect to the network when he moves. This causes a problem for the current network protocols since the protocols are IP-based and use network-prefix routing [1]: when a computer changes its point of attachment, its communication cannot be maintained without requiring a new IP address and establishing a new TCP connection. To solve this problem, the Mobile IP Working Group in the Internet Engineering Task Force (IETF) proposed the Mobile IP (MIP) protocol [14] to support the mobile user.

Mobile IP is built on the IP protocol without major changes for current Internet infrastructure. Since Mobile IP is the network-layer solution for IP mobility [6], it will suffer the same security problems as IP, as well as wireless-specific problems. Besides that, there also exist specific concerns related to the Mobile IP protocol itself. The problem of securing Mobile IP has been generating more interest with increasing demand for Mobile IP. This report addresses security problems and solutions for Mobile IP. We begin with an examination of current network security vulnerabilities, wireless security threats and special issues for Mobile IP. Then, we give an overview of current

technologies for network security such as cryptography and authentication, which are fundamental building blocks of any security solution. Later, we describe an IP layer security solution: The IPSec protocol, and discuss how it ensures Mobile IP security. In addition, we briefly present the design and implementation of the SecMIP simulation, which is a prototype showing the integration of IPSec with Mobile IP.

To talk about network security issues, we have to know the goals of information security. According to [26], the goals of information security are *confidentiality*, *integrity* and *availability*. Generally speaking, confidentiality protects data so that it is not disclosed in an unauthorized fashion. Integrity protects data against unauthorized modifications to data. Availability protects data from unauthorized attempts to withhold information or computer resources.

Tracing the history of the Internet, we find that the Internet is inherently insecure. The Internet began as ARPANET in 1969, a project funded by the Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense [18]. The Internet was therefore designed to share information among the ARPA researchers. So, it was designed for openness and flexibility, not for security. As a result, the Internet protocol cannot satisfy the three goals of information security due to its originally open nature. When the Internet is exposed to millions of interconnected computers, it is inevitably subject to security threats such as eavesdropping, denial of service, address impersonation, session hijacking, etc. In order to defend against these threats, security technologies and products such as cryptography, authentication and firewalls are used to provide secure communication services.

Wireless technologies have become increasingly popular in our everyday business and personal lives. The major difference between wireless networks and wired networks is the transmission medium. Wireless networks normally use radio frequencies to transmit data rather than physical connections such as cable in wired networks. Less wiring often means more portability and flexibility, increased productivity, and lower installation costs [7]. However, several risks are inherent. Firstly, all the threats that exist in wired networks also apply to wireless networks. Secondly, there exist more risks for the underlying communication medium. Since the airwaves are openly exposed to intruders, the loss of confidentiality and integrity and the threat of denial of service (DoS) attacks are a greater risk in wireless communications. In addition, the lack of efficient cryptography technologies, security management tools and key management exacerbates wireless network security problems.

Mobile IP is an extended IP protocol for mobility. Normally, it is deployed in a wireless network environment. This means that Mobile IP inherits all the vulnerabilities of wired and wireless networks [6,15]. Moreover, Mobile IP also introduces new vulnerabilities due to the protocol design. For example, lack of strong authentication and encryption makes the registration process the weakest link for the intruder. With the prevalence of Mobile IP, we need to seek a way to secure Mobile IP.

Since Mobile IP can be viewed as a network- layer mobility approach, the natural and useful solution is to preserve confidentiality, integrity and availability on the same layer. IPSec seems to be the answer. IPSec framework, put forward by IETF IPSec working group [1], is an IP-layer solution to ensure private communication over IP networks [7]. It consists of three protocols [10]: Authentication Header (AH) [8], Encapsulation

Security Payload (ESP) [9], and Internet Key Exchange (IKE) [4]. The three protocols interoperate with Security Association (SA) [10]. IPSec provides the following protections: *confidentiality, connectionless integrity, data origin authentication, replay protection,* and *traffic analysis protection* [10]. Confidentiality ensures that others cannot read the information in the message. Connectionless integrity guarantees that a received message has not changed from the original message. Data origin authentication guarantees that the received message was sent by the originator and not by a person masquerading as the originator. Replay protection provides assurance that the same message is not delivered multiple times, and that messages are not out of order when delivered. Traffic analysis protection provides assurance that an eavesdropper cannot determine who is communicating or the frequency or volume of communications. IPSec accomplishes the task of routing the messages via an encrypted tunnel; this is accomplished by inserting ESP and AH header immediately after the IP header in each message [10,7]. There exist several ways to integrate with the IPSec and Mobile IP [6].

The SecMIP simulation described in this report establishes a prototype that integrates the basic components in IPSec with Mobile IP basic operations. For IPSec, we provide basic protocols whose core is Security Association (SA). In particular, we implemented a simplified ESP protocol and IKE protocol. For the ESP protocol, cryptographic algorithms such as DES and authentication algorithms such as HMAC-MD5 are applied. For the IKE protocol, the Diffie-Hellman algorithm is utilized. For Mobile IP, the entities Home Agent (HA), Foreign Agent (FA), Mobile Node (MN) and Correspondent Node (CN) are simulated. Besides, basic scenarios of Agent Discovery, Registration and Tunneling are implemented. We also implement an Attack Node (AN) to perform three

typical attacks: Denial of Service, Replay Attack and Session Hijacking. SecMIP demonstrates the different results for the same attack with IPSec and without IPSec. In SecMIP, the places that IPSec is applied to Mobile IP are the registration process and the data tunneling.

The rest of the report will be organized in the following way: Chapter 2 discusses Internet vulnerabilities, secure technologies and some specific security issues related to wireless networks. Chapter 3 describes the basic domain knowledge of Mobile IP and its specific vulnerabilities. Chapter 4 briefly introduces the IPSec protocols including some details on AH, ESP and IKE and explains four ways to integrate the IPSec with Mobile IP. Chapter 5 gives the details of the design and implementation of the SecMIP simulation. We conclude with some discussion of the limitations of the IPSec solution in Chapter 6.

2 Network Security

In this section, we discuss the vulnerabilities of the Internet, current available security technologies and special security concerns in wireless networks.

2.1 Internet Vulnerabilities

The Internet is known to begin as ARPANET [2,18], which was developed so that ARPA researchers could share information easily. So the original design of the Internet protocol is for openness and flexibility, not for security [2]. The assumption that "everyone is a friend" works when the internet is just a small academic network, but gives rise to innumerable threats and difficulties to maintain information confidentiality, integrity and availability when the Internet becomes a giant which interconnects millions of computers after 3 decades of evolution. In general, the common threats to the Internet are classified into several kinds: packet sniffing, denial of service, address impersonation and session hijacking [2,26,22].

- **Packet Sniffing**

A packet sniffer is a wiretap device that plugs into computer networks and eavesdrops on network traffic. It captures the binary data passing through the network and decodes this data into a human readable form [23]. Packet sniffing can be used both in good and malevolent ways. It can help the network administrator maintain the network traffic and do some protocol analysis. It also can be used to steal private information such as the user accounts and user passwords. Obviously, information confidentiality is compromised if IP packets are caught in clear text across untrusted networks. Unfortunately, many services on the Internet send data in plain text. For example, POP

mail and SMTP (for sending mail) send data in clear text by default. The same applies to FTP, Telnet, News clients, ICQ, MSN and AOL Instant messenger [26]. In addition, the packet sniffer can replay the sniffed packet to the dedicated computer at the same connection, which is known as a *replay attack.*

- **Denial of Service**

The goal of denial-of-service attacks is not to gain unauthorized access to machines or data, but to prevent legitimate users of a service from using it [2], thus compromising availability. Traditional denial of service attacks are done by exploiting a buffer overflow [3], which deliberately consumes a scarce or limited resource, or exploits a system bug that makes the system no longer functional. Recently, a new form of denial of service attack has been developed called Distributed Denial of Service (DDoS) attack [26]. A Distributed Denial of Service attack uses multiple machines operating together to "flood" useless packets to a network or site. The nature of the attacks causes so much extra network traffic that it is difficult for legitimate traffic to reach a site while blocking the forged attacking packets. In addition, DDoS are much harder to stop and track down since they are generated from a vast number of computers [26].

- **Address Impersonation**

Every computer on the network has an IP address associated with it that uniquely identifies it [22]. In an IP packet, two pieces of information that must always be in clear text are the source and destination IP addresses. Otherwise, intermediate gateways and routers on the Internet could not forward packets properly. It means that everyone, including the attacker, can legally get the addresses. To make matters worse, a computer might have several IP addresses due to different network adapters. Privileged users are

allowed to set the source IP address in the packet to any desired value. Since the IP address is the identity in IP layer and no authentication is provided for these network addresses, impersonating an IP address seems not too difficult with some packet modifying tools [2,3].

- **Session Hijacking**

If network traffic between two nodes flows in the clear and if you know the protocol that the nodes are using, you can disable one of the nodes and impersonate that node using IP address impersonation. A simple hijacking procedure will look like this: the two-endpoint nodes of a communication session send traffic in the clear. The hijacker node's location is such that all traffic between these two nodes must flow through a node that it controls. It sniffs packets and the intercepted packets can be easily altered, discarded, or substituted. Then the hijacker node can take over the whole communication session and do what it wants [2,3,8].

2.2 Security Technologies

A variety of security technologies has been developed to ensure the confidentiality, integrity, and availability of information. The classical technologies are cryptography, authentication, key management, firewalls, and auditing. These technologies are quite mature after decades of development. Current information security solutions use them as building blocks to construct security protocols. The following section will give a brief overview of these technologies [3,22,23,26].

- **Cryptography**

Cryptography secures information by protecting its confidentiality and then keeps the communication private.

Encryption is the process of translating information from its original form (called *plaintext*) into an encoded, incomprehensible form (called *ciphertext*). The ciphertext cannot be decoded without the appropriate knowledge (key). Decryption is the reverse process of encryption: that of taking ciphertext and translating it back into plaintext. Any type of data may be encrypted, including digitized images and sounds.

There are two kinds of cryptosystems: symmetric and asymmetric. Symmetric cryptosystems use the same secret key to encrypt and decrypt the message. For example, Data Encryption Standard (DES) [22], a widespread cryptography protocol, is a symmetric cryptosystem. In asymmetric cryptosystems, a public key encrypts the message and a private key decrypts it. Asymmetric cryptosystems are also called public key cryptosystems. RSA [22] is a well-known public key cryptosystem. Generally speaking, since symmetric cryptosystems have the problem of secret key distribution, asymmetric systems seem more reliable, but are less efficient methods. Both are often combined to perform cryptographic operations.

- **Authentication**

Authentication is the process to assure message integrity, that is, to assure that data have not been modified in transit or storage. Network authentication must authenticate the sender identity, data integrity and receiver identity [26,22]. Authentication methods are typically based on something you know, something you have, or something you are [3]. The classical authentication algorithms include digital signatures for sender and

receiver identity [19], MD5 [22], and SHA [22]. For the application level, the system login, and Kerberos [22] third party authentication are normal authentication usages. Authentication is usually integrated into cryptography protocols to provide data integrity.

- **Key Management**

Key management is the whole process of handling cryptographic keys. It includes generating keys, distributing them, protecting them, and eventually destroying them. Keys must be changed frequently. Key authenticity and usually secrecy must be assured. Expired keys must be destroyed [22].

For example, Key Management Standard (KMS), put forward by the NISC[22], allows either a two-level or three-level key hierarchy. The three-level hierarchy consists of data keys (KDs), key encrypting keys (KKs) at the optional level, and highest-level key encrypting keys (KKMs). Data keys are used to encrypt or authenticate data and also to authenticate messages used in key distribution protocols. The KKs are distributed over the network. The KKMs are generated, distributed and entered manually according to guidelines of the standard [22].

The best-known key exchange algorithm is the Diffie-Hellman key agreement algorithm (also called exponential key agreement). It allows two users to exchange a secret key over an insecure medium without any prior secret.

- **Firewall**

A firewall is a collection of systems to enforce the access control policy between two communication networks [26]. A simple firewall may filter the unauthorized incoming packets which fail to meet the security criteria. A sophisticated one may provide proxy services, authenticate service requests, verify data packets and replay the right packet to

the appropriate service host. Since the firewall usually stands in the frontline of a local network, its configuration must be carefully designed [2,26].

- **Auditing**

Auditing is a trusted mechanism to log system activities. It has been recognized as an important technology for information security. Nowadays, the traditional auditing mechanism has evolved to the intrusion detection system. An intrusion detection system helps to construct attacker scenarios and predict possible attacks by analyzing suspicious activities and patterns of events [3,23].

2.3 Special Issues in Wireless Networks

There are many advantages in using wireless networks. Users are provided always-on network connectivity in wireless networks. Networks are easily set up, augmented and deployed without installing and moving wires. However, along with the convenience, wireless networks are subject to vulnerabilities besides those we mentioned in Section 2.1.

Radio frequencies are used to transmit data in wireless networks. Compared to the cable in wired networks, radio frequencies are more susceptible to be eavesdropped, jammed, taken over, and to be attacked in any other way affecting wired networks [21].

Unlike wired networks, the communication medium in wireless network cannot be physically secured or protected. In addition, wireless networks do not include network and security management tools like those in the wired network infrastructure. Tools that address airwave security, authentication, and user rights are absent from wireless networks [15].

Lack of key management protocols is another limitation for current wireless networks. For example, the security options in the current IEEE wireless local area network protocol IEEE 802.11 do not scale appropriately in a large infrastructure network for the access control of key management databases [15]. Moreover, lack of inter-access point protocol (IAPP) [20] further compounds key management issues.

2.4 Summary

In sum, the Internet faces many vulnerabilities resulting from its open nature. Wireless networks are more likely to be exposed to such attacks because of their transmission medium. Besides that, wireless networks have their own specific security problems. Though there exist some security technologies, current wireless network protocols have a lack of encryption to preserve privacy, lack of cryptographic authentication to identify the source of information and lack of cryptographic checksums to preserve the integrity of data. In order to have a secure foundation for the network application regardless of whether it is wired or wireless, a secure internetworking protocol has to be developed.

3 Mobile IP

Traditionally, IP routing is based on the IP address, which uniquely identities a node's point of attachment to the Internet [5]. When a node moves and enters a new network, it has to change its IP address and reconstruct a new TCP connection. If a communication with this moving node exists at that time, the communication has to be interrupted until it gets the new IP address of the moving node. Obviously, interrupted communication is unpleasant. To solve this mobility problem, a working group within Internet Engineering Task Force (IETF) proposed a solution, which is the *Mobile IP Protocol.* In this section, we give an overview of Mobile IP as well as security problems specific to Mobile IP.

3.1 Mobile IP Overview

Mobile IP is the network layer solution designed to solve the mobility problem. It is built on the IP protocol and allows a mobile node to send and receive packets over the Internet using its home address regardless of its point of attachment.

The idea behind Mobile IP is analogous to postal service delivery: whenever you move to a new location, you ask your home post office to forward your mail to your new address via the local post office there. Thus, a mobile node first leaves its home network and connects to a foreign network. An agent on the home network then intercepts packets sent to the mobile node and forwards them to an agent on the foreign network. This agent then delivers packets locally to the mobile node visiting that network [14].

A mobile node thus has two IP addresses: a fixed *home address* and a temporary *care-of-address* that changes at each new point of attachment. With these two addresses,

a portable computing device can be moved from one network to another without changing its IP home address and without losing existing connections [21,14]. In addition, in Mobile IP, all the needed reconnections or redirection between the home address and the care-of-address occur automatically and transparently [16]. The care-of-address and the Mobile IP operations will be described in great detail in the following sections.

There are two versions of Mobile IP –IPv4 and IPv6. IPv4 Mobility support is built on current IP structure, which is an IP extension. In IPv6, mobility support is an integral part of the specification [17]. In the following text, Mobile IP for IPv4 is presented.

3.2 Mobile IP Entities

Mobile IP implements the following core functional entities: *Mobile Node*, *Home Agent*, *Foreign Agent* and *Correspondent Node* [21,14,16].

1) Mobile Node (MN): A moving node, which can change its point of attachment from one network to another. It keeps ongoing communications without interruptions by using its home address.
2) Home Agent (HA): A support node on the mobile node's home network. It keeps track of the mobile node location (care-of-address), advertises its reachability, intercepts and tunnels packets destined for the mobile node.
3) Foreign Agent (FA): A support node on the foreign network. It provides the care-of-address to the mobile node, assists the mobile node in informing the home agent of its care-of-address, acts as a default router for the packets generated by the mobile node, decapsulates and delivers packets to the mobile node.

4) Correspondent Node (CN): A node that sends or receives a packet to or from a mobile node; the correspondent node may be another mobile node or a non-mobile Internet node.

Each mobile node has a home address as its permanent address. A care-of-address is a temporary IP address of a mobile node that identifies a mobile node's current point of attachment to the Internet and makes it possible to connect from a different location without changing its home address. When a mobile node is away from its home network, it is assigned a care-of address. This may be a foreign agent care-of address, which is static IP address of a foreign agent on a visited network, or a collocated care-of address, which is a temporary IP address assigned to the mobile node. A collocated care-of address must be acquired by some assignment procedure, such as the Dynamic Host Configuration Protocol (DHCP), the Point-to Point Protocol's IP control protocol (PPP), or manual configuration.

3.3 Mobile Operations

Basically, the operations in Mobile IP can be divided into three phases: Agent Discovery, Registration, and Tunneling [5,14,16, 21].

3.3.1 Agent Discovery

In the Agent Discovery phase, the home agent and foreign agent advertise their availability to any attached links via periodically broadcasting a special message called Agent Advertisement. A mobile node listens to these Agent Advertisements and examines its network prefix to determine whether it is connected to its home link or a

foreign link. If the mobile node stays in its home network, it works like any other stationary node without Mobile IP functionality. If the mobile node is connected to a foreign link, it acquires a care-of address. If the care-of-address is the foreign agent care-of-address, the foreign agent will be the endpoint of the tunnelled message. The foreign agent performs the encapsulation, decapusulation and delivers packets to the mobile node. If the care-of-address is the collocated care-of-address, the mobile node itself will perform the encapsulation and decapusulation for tunneling. If the mobile node returns to the home agent after registration elsewhere, it needs to deregister to its home link.

Besides listening to the broadcast agent advertisement, a mobile node may optionally solicit an agent advertisement message from any locally attached mobility agents through an agent solicitation message. Figure 3.1 shows the agent discovery process.

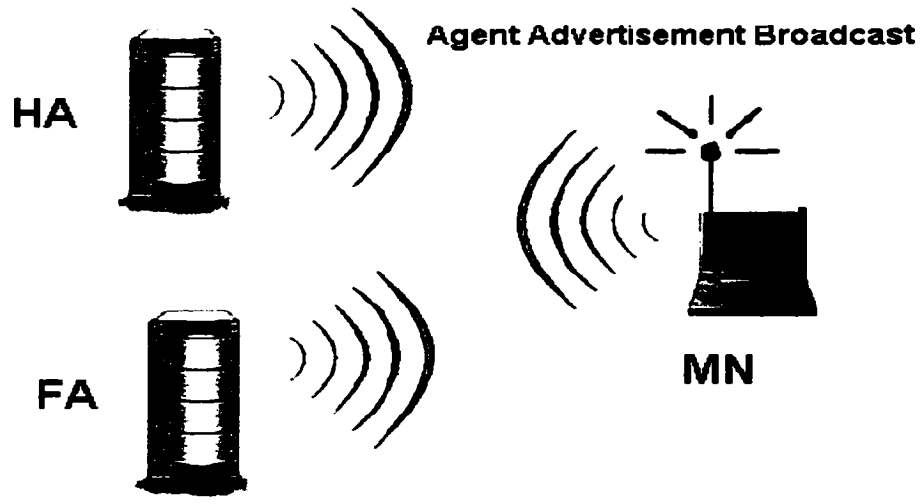

Figure 3.1. Agent Discovery

3.3.2 Registration

Once a mobile node gets the care-of-address, its home agent has to know where it is. The registration phase begins when the mobile node sends a *registration request* with care-of-address information to the home agent, assisted by the foreign agent. In some cases, the mobile node also may register directly with the home agent. The registration request should contain three parameters: home address of the mobile node, care-of-address of the mobile node and the registration lifetime. When the home agent receives the registration request, it may authenticate it. If the request is approved, the home agent updates its route table, associating the home address of the mobile node with its care-of-address. The home agent will also maintain the association until the registration lifetime expires. Then, the home agent sends a *registration reply* to the mobile node via the foreign agent or directly. Though Mobile IP provides some authentication methods, such as timestamp and identification field, for the registration request and registration reply, the registration phase is still the weakest link in the Mobile IP protocol. Figure 3.2 shows the registration process.

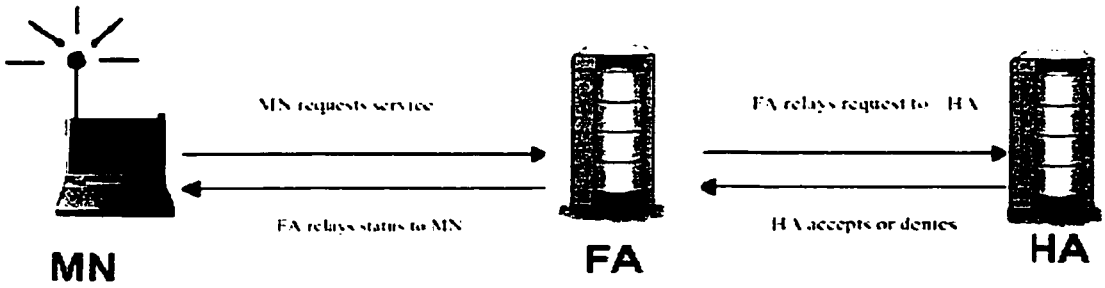

Figure 3.2. Registration

3.3.3 Tunneling

After a mobile node successfully registers its care-of-address to its home agent, a tunnel to the care-of-address is constructed. The tunneling to the care-of-address is accomplished by using encapsulation mechanisms. All mobility agents, i.e., home agents and foreign agents, must be able to use a default encapsulation mechanism – the IP within IP protocol. In IP within IP, the entire original IP header is preserved as the first part of payload of the packet. By using this protocol, the source of the tunnel, i.e., the home agent, inserts an IP tunnel header, in front of the header of any original IP packet addressed to the mobile node's home address. The destination of this tunnel is the mobile node's care-of-address. This is called encapsulation. By removing the tunnel header, the original packet can be recovered. This process is called decapsulation, which could be done by the foreign agent or the mobile node.

Message delivery to the mobile node will work like this: all data packets sent to the mobile node's home address are intercepted by its home agent, encapsulated by the home agent, tunnelled by the home agent to the mobile node's care-of address, received at the tunnel endpoint (either at a foreign agent or at the mobile node itself), decapsulated by the tunnel endpoint and finally delivered to the mobile node. For the data packets sent by the mobile node, they are generally delivered to their destination using standard IP routing mechanisms, and the foreign agent serves as a router for all packets generated by a visiting mobile node. Figure 3.3 shows the tunneling process.

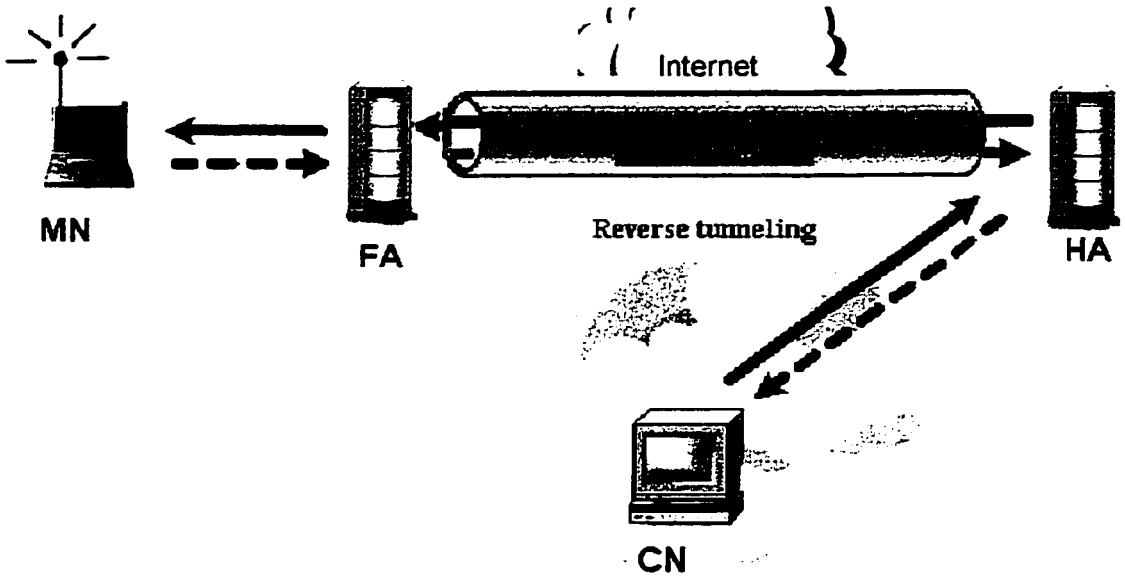

Figure 3.3. Tunnelling Data

3.4 Security Issues in Mobile IP

Since Mobile IP builds upon the IP protocol without major changes to the current Internet infrastructure, it has the same vulnerabilities inherited from the Internet open model. Besides that, Mobile IP has its own security leaks such as lacking strong protection for the registration operation. In particular, it is especially weak for the following attacks [6,8]:

- **Passive eavesdropping**

Since there is no cryptography available in Mobile IP traffic, it is possible for the attacker to listen to the communication with some packet sniffer programs.

- **Replay attack**

By eavesdropping, an attacker can store a valid registration sequence and reuse it for later diverting mobile node traffic.

- **Denial of service**

Since the sensitive IP addresses of the home agent and mobile node are visible in registration request, the attacker can overflow malicious packets to these nodes.

- **Session hijacking**

Since there is no authentication protection for the tunnelled data, the attacker can take over the whole communication session after successful registration.

- **Data compromising**

Since encryption and authentication are missing, tunnelled data are susceptible to be modified.

- **Tunnel spoofing**

The tunnel to the foreign network may be used to hide malicious IP packets and get them past the firewall.

Since registration request and registration reply are key parts of the Mobile IP, Mobile IP has some basic security solutions in mind. Mobile IP requires authentication for the registration message between the mobile node and the home agent. Additionally, Mobile IP uses timestamp and identification field to protect against attacks in registration requests [21]. But these protections are not enough for the large-scale Mobile IP deployment [21], for example, there is no protection for the data packets being routed in tunnels in Mobile IP.

4 IPSEC

In this section, we give an overview of IPSec, including IPSec components and their interoperation.

4.1 IPSec Overview

The IPSec protocol suite is used to provide privacy and authentication services at the IP layer. It provides a set of security algorithms plus a general framework that allows a pair of communicating entities to secure the traffic [10]. It consists of three protocols [10]: Authentication Header (AH) [8], Encapsulation Security Payload (ESP) [9], and Internet Key Exchange (IKE) [4]. The three protocols interoperate with Security Association (SA) [10]. IPSec provides assurance for confidentiality, connectionless integrity, data origin authentication, replay protection and traffic analysis protection [10].

4.1.1 Components

The IPSec protocol suites consists of three parts:

- Authentication Header (AH) – This header provides integrity and authenticity of the data, including the invariant fields in the outer IP header. It does not provide confidentiality protection [8].
- Encapsulating Security Payload (ESP) – This header provides for confidentiality, integrity, and authenticity of the data. If ESP is used to validate data integrity, it does not include the invariant fields in the IP header [9].
- Internet Key Exchange (IKE) – This protocol is a key agreement protocol, which provides a powerful, flexible negotiation on authentication algorithms, encryption

algorithms, the keys to use, the lifetime of the keys using a special message called Security Association (SA). Figure 4.1 shows the structure of IPSec [4].

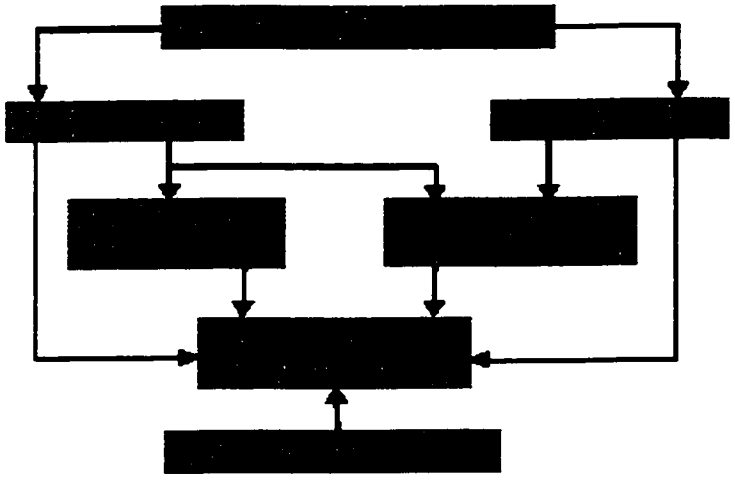

Figure 4.1. IPSec Architecture

4.1.2 Security Association (SA)

An SA is a one-way directional agreement between two communicating parties that specify a set of policies and keys to protect the future communications between them. The concept of Security Association is fundamental to IPSec. Both AH and ESP make use of SAs and a major function of IKE is the establishment and maintenance of SAs. The SA is uniquely identified by the security parameter index (SPI), the destination IP address and the security protocol (AH or ESP) identifier. An SA could be established between end users, security gateways, end users and security gateways. Figure 4.2 shows an example of a table of SAs identified by their SPI [10].

Security Parameter Index	Authent. algorithm	Authent. key	Replay protection	Encryption algorithm	Encryption key
01234567	e.g., Keyed MD5	(a secret key)	timestamp		
89ABCDEF				e.g., RSA	(public/ private key)

Figure 4.2. Example of SA Table

An SA specifies the following policies under IPSec [10]:

- The mode of authentication algorithm used in the AH and the keys of the authentication algorithm.
- The ESP encryption algorithm mode and keys.
- The presence (or absence) and size of any cryptographic synchronization to be used in the encryption algorithm.
- How to authenticate communications (using what protocols, what encrypting algorithm and what key).
- How to make communication private (again, what algorithm and what key).
- How often those keys are to be changed.
- The authentication algorithm, mode and transform for use in ESP plus the keys to be used by that algorithm.
- The key lifetimes.
- The lifetime of the SA itself.
- The SA source address.

In IPSec, there also exist two databases to manage the SAs: Security Policy Database (SPD) and Security Association Database (SAD). The SPD specifies the types and orders of SAs and provides the entries to the SAD. It is consulted during the processing of all traffic including inbound, outbound and security and key management traffic. The SAD

specifies the parameters associated with one SA. For outbound processing, entries are pointed to by entries in the SPD. For inbound processing, each entry in the SAD is indexed by a destination IP address, IPSec protocol types, and SPI [10].

IPSec protocols interoperate in the following way: If two systems need IPSec protection for their communications, they first negotiate SAs for the security parameters with IKE. After that, when a system sends a packet that requires IPSec protection, it looks up the SA in its database (first the SPD and then SAD), applies the specified processing which may be AH, ESP or both, and then inserts the SPI from the SA into the IPSec header. When the IPSec peer receives the packet, it looks up the SA in its database (first SPD and then SAD) by destination address and SPI and then processes the packet as required.

4.1.3 Modes

There are two modes available in IPSec: transport mode and tunnel mode [9,8,10].

In transport mode, only the IP payload is encrypted and the original IP header is left intact. The advantage is that just a few bits are added into each packet. The disadvantage is that the source and destination IP addresses are visible across the public network, which might be the targets of the attacks.

In the tunnel mode, the entire original IP packet is encrypted and it becomes the payload in a new IP packet. The advantage is that the operations are simple and the source and destination addresses are invisible to avoid session hijacking attacks.

The tunnel mode is more often deployed than the transport mode since the network structure, the operating system and any applications on the PCs, servers, and hosts do not have to be modified.

4.1.4 Technologies

IPSec combines several different security technologies into a complete system to provide confidentiality, integrity, and authenticity. In particular, IPSec uses:

- Diffie-Hellman key exchange for generating key material between peers on a public network.
- Public key cryptography for signing the Diffie-Hellman exchanges to guarantee the identity of the two parties.
- Bulk encryption algorithms, such as DES, IDEA, Blowfish, and RC4. The most commonly used of them is DES.
- Keyed hash algorithms, such as HMAC [22], combined with traditional one-way hash algorithms such as MD5 or SHA for providing packet authentication.
- Digital certificates signed by a certificate authority to act as digital ID cards.

4.2 AH

The Authentication Header (AH) is used to provide integrity, authentication and anti-replay protection for IP packets including part of the IP header, but it does not provide confidentiality [8].

The suitable algorithms for AH are the keyed Message Authentication Codes (MACs) based on symmetric encryption algorithms and one-way hash function (e.g. MD5, SHA1) [8]. AH uses a keyed-hash function more often rather than digital signatures, because digital signature technology is too slow and would greatly reduce network throughput [19]. Figure 4.3 shows the AH format [8]:

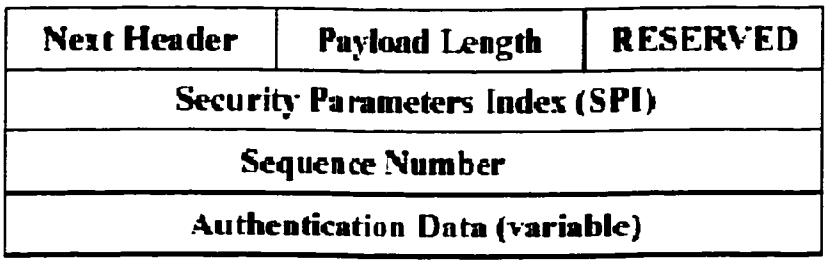

Figure 4.3. AH Format

- The Next Header is an 8-bit field that identifies the type of the next payload after the Authentication Header.
- Payload Length is an 8-bit field that specifies the length of AH in 32-bit words.
- Reserved is a 16-bit field reserved for future use.
- Security Parameters Index (SPI) is an arbitrary 32-bit value that uniquely identifies the Security Association (SA) for this figure.
- Sequence Number is an unsigned 32-bit field that contains a counter value that is defined for replay-protection purposes.
- Authentication Data (variable) is a variable length field that contains the Integrity Check Value (ICV) for the packet.

In transport mode, AH is inserted after the IP header and before the header of an upper layer protocol such as TCP, UDP, ICMP, etc., as well as before any other IPSec headers that have already been inserted. In tunnel mode, the "inner" IP header carries the ultimate source and destination addresses, while an "outer" IP header may contain distinct IP addresses, or addresses of security gateways [8].

4.3 ESP

The Encapsulating Security Payload (ESP) header is used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality [9]. ESP doesn't provide authentication for the IP header. ESP supports symmetric encryption algorithms such as DES-CBC [22], and integrity algorithms such as HMAC-SHA1 [22]. ESP may be applied alone, in combination with the IP Authentication Header (AH), or in a nested fashion, e.g., through the use of tunnel mode. The set of services provided depends on the options selected at the time of Security Association establishment and on the placement of the implementation [9]. Figure 4.4 shows the ESP format [9].

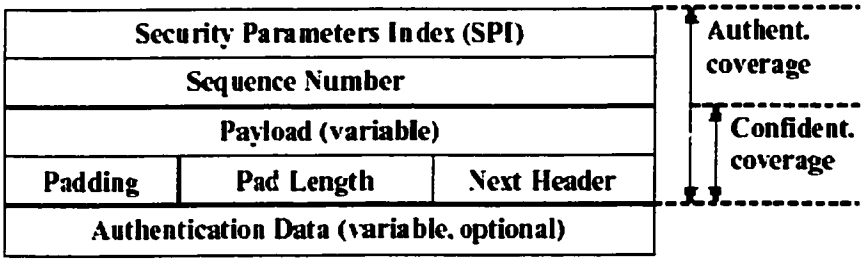

Figure 4.4. ESP Format

- Security Parameters Index (SPI) is an arbitrary 32-bit value that uniquely identifies the Security Association (SA) for this data diagram.
- Sequence Number Field is an unsigned 32-bit field that contains a counter value that is defined for replay-protection purposes.
- Payload (variable) is a variable-length field containing data described by the Next Header field.

- Padding is an optional field defined to ensure that the Authentication Data field is aligned on a 4-byte boundary.
- Pad Length indicates the number of pad bytes immediately preceding it.
- Next Header is an 8-bit field that identifies the type of data contained in the Payload Data field, e.g., an extension header in IPV6 or an upper layer protocol identifier.

The ESP header is inserted after the IP header and before the upper layer protocol header in transport mode or before an encapsulated IP header in tunnel mode.

4.4 IKE

Internet Key Exchange (IKE) is a key agreement protocol, whose purpose is to negotiate and provide authenticated keying material for security associations in a protected manner [K1]. It is a hybrid protocol using part of Oakley [13] and part of SKEME [11] in conjunction with ISAKMP [12] to obtain authenticated keying material for use with ISAKMP and for other security associations [4].

- Phases

There are two phases in IKE. Phase1 is dedicated to establishing a secure channel for doing IKE. In this phase, the SAs for IKE primer secure channel are negotiated. The primer secure channel is the secure channel to exchange the security parameters for doing secure data communication. There are two modes available in this phase-Main mode and Aggressive mode. Phase2 is dedicated to establish non-IKE SAs for IPSec AH and ESP services. Only one mode (Quick Mode) is available in this phase [4].

- Modes

Main mode is an instantiation of the IKE identity protect exchange: the first two

messages negotiate policy; the next two exchange Diffie-Hellman public values and ancillary data (identity nonce); and the last two messages authenticate the Diffie-Hellman Exchange. Figures 4.5a, 4.5b and 4.5c show the main mode procedures.

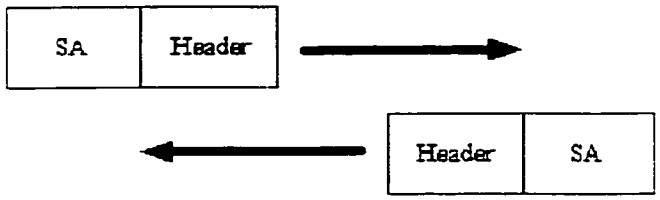

Figure 4.5a. Main Mode Step 1

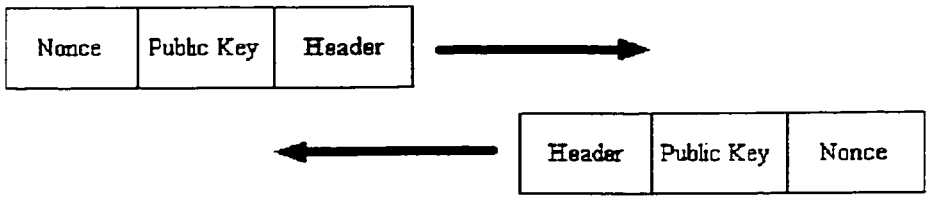

Figure 4.5b. Main Mode Step 2

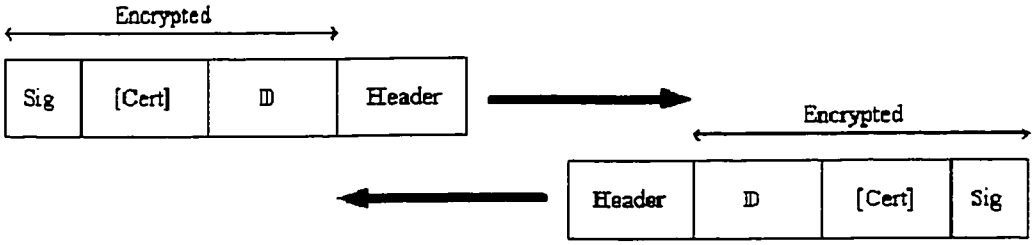

Figure 4.5c. Main Mode Step 3

Similarly, Aggressive Mode is also an instantiation of the IKE identity protect exchange: The first two messages negotiate policy, exchange Diffie-Hellman public

values and ancillary data necessary for the exchange entities. Compared with the Main Mode, Aggressive Mode is faster but no identity protection is provided. Figure 4.6 shows the procedure in Aggressive mode.

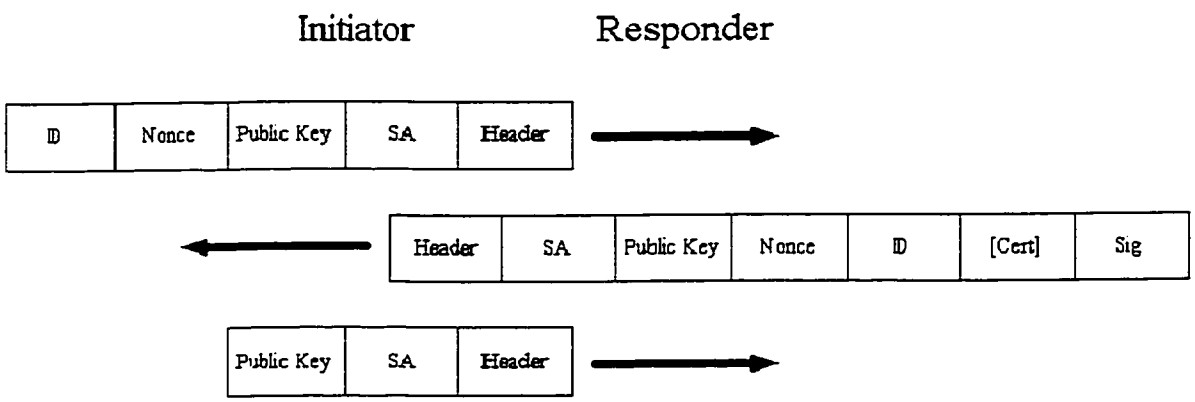

Figure 4.6. Aggressive Mode

After having established an IKE SA, the two communicating parties use Quick Mode to negotiate general IPSec security services or keying material. The IKE SA must protect the information exchanged along with Quick Mode. It means all Quick Mode packets are encrypted and a HASH payload must immediately follow the IP header and a SA payload must immediately follow the HASH. Figure 4.7 shows the procedure in Quick Mode.

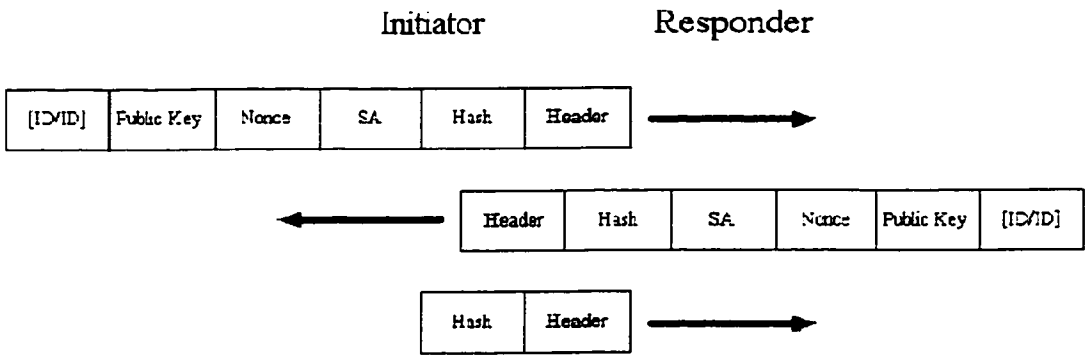

Figure 4.7. Quick Mode

4.5 IPSec and Mobile IP Interoperability

Though Mobile IP has some protection for registration messages, it lacks protection for the Agent Advertisement and tunnelled data, which should be protected as well. Since IPSec is an IP layer security solution, which is at the same layer as Mobile IP in the network model, the use of IPSec protocols in Mobile IP can preserve privacy and integrity for data communication.

Basically, there are four different ways to apply IPSec to Mobile IP [6]:

- Encryption and authentication only outside of private network
- End-to-end encryption and authentication
- End-to-end encryption, intermediate authentication
- Encryption and authentication inside and outside of private network

The first approach is encrypting traffic between the mobile node and the home network firewall. The encryption overhead is only on the public network. The second approach is establishing an IPSec tunnel between the home agent and the mobile node. The home agent and mobile node carry the authentication and encryption overhead. The third alternative is that encryption is done in the mobile node and the agent. The firewall shares a part of SA, does the authentication and forwards the encrypted data packet to the agent. The last approach is every part in the public and private networks has the security association. For example, the mobile node shares a SA with the firewall in the public network and the firewall shares another SA with the home agent in the private network.

As we see, strong cryptography is applied to every data packet in Mobile IP data communication. As a result, passive eavesdropping cannot decrypt the data; the sequence number maintained in the SA and authentication defend against replay attacks. The

authentication in AH or ESP ensure that data packets cannot be accepted if they are changed during transmission, which prevents session hijacking efficiently. The authentication of the IP header in AH prevents address impersonation. The SA lookup for each packet results in quicker dropping of denial of service packets. In sum, IPSec is sufficient to provide confidentiality and integrity to the Mobile IP data packets. For Mobile IP, confidentiality and integrity is the prerequisite for availability.

5 SecMIP Simulation

In this chapter, we describe the system requirements, design and implementation of the SecMIP Simulator.

5.1 System Requirement

The SecMIP simulator requirements include general requirements, hardware requirements and software requirements.

5.2.1 General Requirements

The objective of our SecMIP simulation is to establish a prototype, which integrates the basic components in IPSec with Mobile IP basic operations. For Mobile IP, the entities HA, FA, MN and CN are simulated. Besides, basic scenarios of Agent Discovery, Registration and Tunneling are implemented. For IPSec, the structure of IPSec is established: SA management, ESP processing and IKE negotiation are implemented. The Diffie-Hellman key exchange algorithm is utilized, and cryptography algorithms such as DES are also applied. In order to show the difference between Mobile IP with IPSec and without IPSec, an Attack Node (AN) is implemented. The Attack Node can perform three kinds of attacks: Replay Attack, Denial of Service, and Session Hijacking.

We take the approach that the encryption and authentication happen outside the private network. We make the assumption that communication inside the private network is considered to be safe. We implement IPSec in the registration process and in the data tunneling.

In the SecMIP simulator, we make the following assumptions for simplification:

- The HA and FA internal networks are considered safe.
- The HA and FA work as firewalls, which can perform routing, encryption and authentication.
- The transmission range of the HA and the range of the FA do not overlap.
- Attack nodes are neither in the range of the FA nor in the range of the HA.
- The correspondent node is in the HA.
- The mobile node is connected to the HA at the beginning.
- The HA can identify its mobile node.
- The attack node only attacks the FA and the HA.

From the user's perspective, the SecMIP simulator provides five operations:

1) Move MN between HA and FA;
2) Set HA/FA in secure mode;
3) Send messages from MN to CN;
4) Send messages from CN to MN;
5) Perform attacks.

Figure 5.1 shows the main use case.

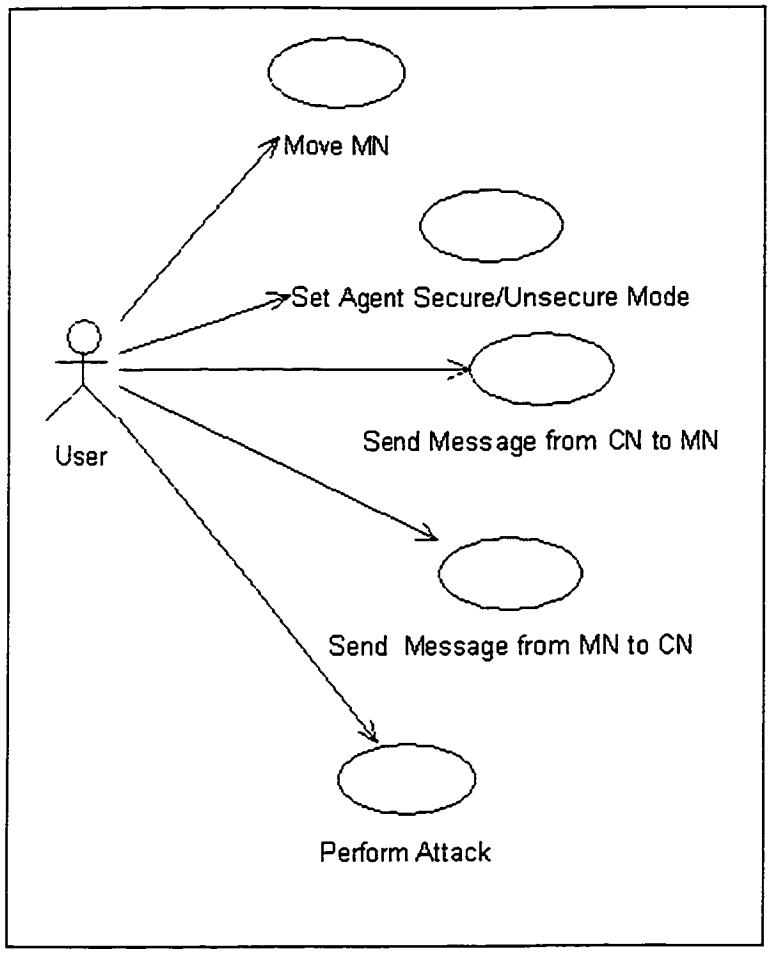

Figure 5.1. Main Use Case

We describe the resulting actions briefly. Detailed use cases are given in Appendix I.

1) Move MN between HA and FA

MN can move between the HA and the FA ranges randomly by the user's clicking. When it moves into the Agent (HA/FA) range, it receives the agent advertisements. If the movement is from the HA to FA, the mobile node sends a registration request to the HA via the FA. If the movement is from the FA to the HA and the MN has been registered by the HA, MN sends a deregistraion request to the HA.

2) Set HA/FA in Secure Mode

If the simulator is in secure mode, a registration request will initiate an IKE security negotiation process and a secure tunnel will be established. All traffic between the HA and FA is processed by ESP according to the security association.

If the simulator is in insecured mode, ESP does not protect the traffic and the plaintext will be transmitted. They will be attacked if the AN performs an attack.

3) Send Messages from CN to MN

When the CN sends a data packet to the MN, the HA intercepts the data packet. Then, the HA checks the MN location. If the MN is currently in the home network, the data packet is delivered to the MN. If the MN has been registered with a COA, the HA will perform MIP processing on the data packet. Then, if the SA is available and in secure mode, the MIP data packet will be subject to the ESP processing and be delivered to the FA. The FA will perform the reverse ESP processing and reverse MIP processing, and recover the original data packet. The original data packet will be sent to the MN.

4) Send Messages from MN to CN

The process of the MN sending a message to the CN is similar to that of the CN sending a message to the MN. The HA and FA perform the reverse processes.

5) Perform Attacks

The AN can perform three kinds of attacks: replay, denial of service and session hijacking. Replay attack is the AN listening to the data packet and sending the used packet to the HA or FA. Denial of service is the AN sending meaningless packets in batch to the HA or FA. In session hijacking, the AN listens to the traffic between the HA and FA, and the AN modifies the traffic between the HA and FA.

In insecure mode, all the three attacks succeed either in fooling or disrupting operations of the HA or FA. In secure mode, the three attacks are blocked.

5.2.2 Hardware Requirements

SecMIP can run on any Windows station. Since the cryptographic and key management algorithms are resource-consuming and time-consuming, the SecMIP simulator must run on at least 3 stations with at least 64M RAM. Minimal node deployments are FA and AN in one station, HA in one station and CN and MN in one station. One node for one station is preferred.

In the SecMIP simulator, the nodes-running procedure is: run AN node-> run FA node-> run HA node-> run MN node-> run CN node.

5.2.3 Software Requirements

We used UML, Java and Windows NT to implement the SecMIP simulator.

- UML is used to design the implementation of SecMIP simulation.
- Java is used to program the implementation of SecMIP simulation. Java Security Package and Java™ Cryptography Extension (JCE) are required. Java 2 SDK version 1.4 is highly recommended.
- Windows NT is used as the development platform.

There are several reasons to use Java as the SecMIP simulator development language. First, Java is a cross-platform programming language. Though we developed the SecMIP simulator in Windows NT, it can be transplanted to LINUX and UNIX. Moreover, Java provides extensive low-level APIs such as socket APIs for network programming, which is an essential part of the MIP scenarios simulation. In addition, the internal packages in

Java such as Java™ Cryptography Extension (JCE) provide a framework and implementation for encryption, key generation and key agreement, and Message Authentication Code (MAC) algorithms, which are used for cryptography, key agreement and authentication.

5.2 Class Diagrams

The following text, package diagram, overall class diagram and inheritance diagram, give the top-level view for the classes developed.

Figure 5.2 shows that the classes of SecMIP are divided into 6 packages: AN, CN, FA, HA, MN and Utils according to the Mobile IP entities and IPSec functionalities. Figure 5.2 shows the implementation packages in the SecMIP simulation.

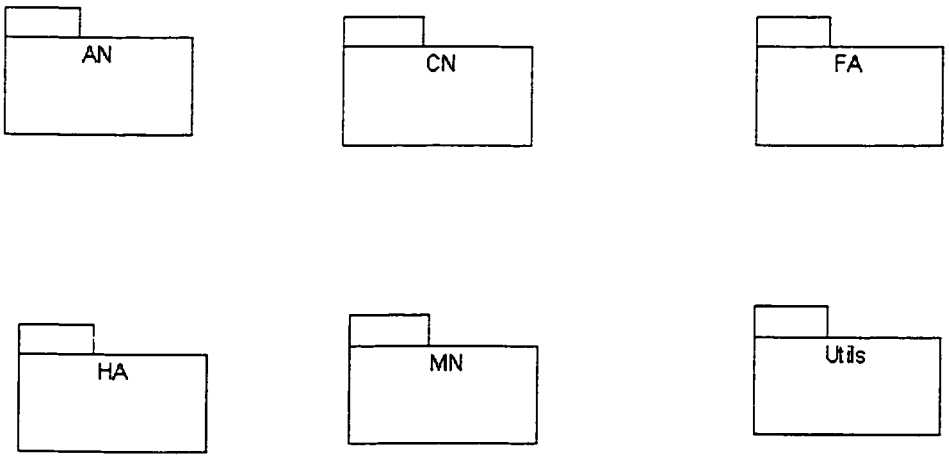

Figure 5.2. Package View

Figure 5.3 describes the relationship between the most important objects in the SecMIP simulation. Detailed class descriptions are given in Appendix II. The objects in SecMIP fall in six major categories: objects in Utils implement the IKE negotiation, ESP

protocol processing, encryption and authentication, auditing; objects that implement the FA behaviors including agent advertisement, sending and receiving data packets, initiating IKE processing and performing the ESP processing; objects that implement the HA behaviors which are similar to the FA plus the registering and deregistering of MN; objects that implement the MN behaviors including agent discovery, registration and deregistration requests and sending and receiving the message; objects that implement the CN behaviors including sending and receiving the message and objects that implement the AN behaviors including performing the attacks.

The SA class implements the IKESA, which will be applied on the IKE security association negotiation, and DATASA is inherited from the SA class, which is applied on the DATA security association negotiation.

The Info class is the parent class to use for logging and auditing message communication. The AgentInfo class is inherited from the Info class, which extends the IKE negotiation functionalities. FAInfo, MNInfo and HAInfo are child classes of AgentInfo which log and audit the FA, MN and HA functionalities respectively.

Since many entities in Mobile IP are a kind of node, so the Node class is the parent class. The Agent class, MN class, CN class and AN class are the children of the Node class. They implement the agent, MN, CN and AN behaviors respectively. In addition, the FA class and HA class extend the Agent class to simulate the FA and HA respectively. Figure 5.4 shows the inheritance relationship between classes.

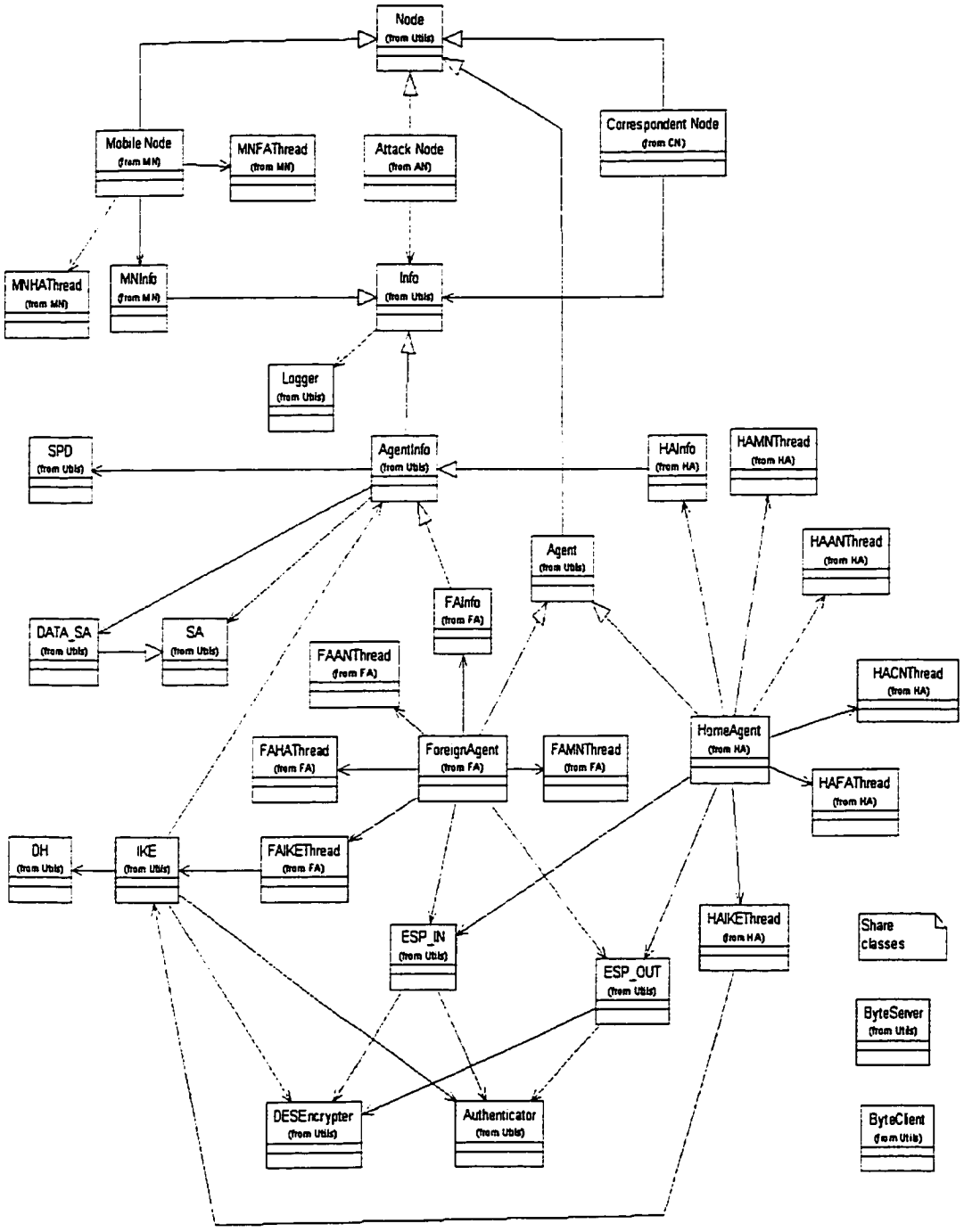

Figure 5.3. Entities Class Diagram

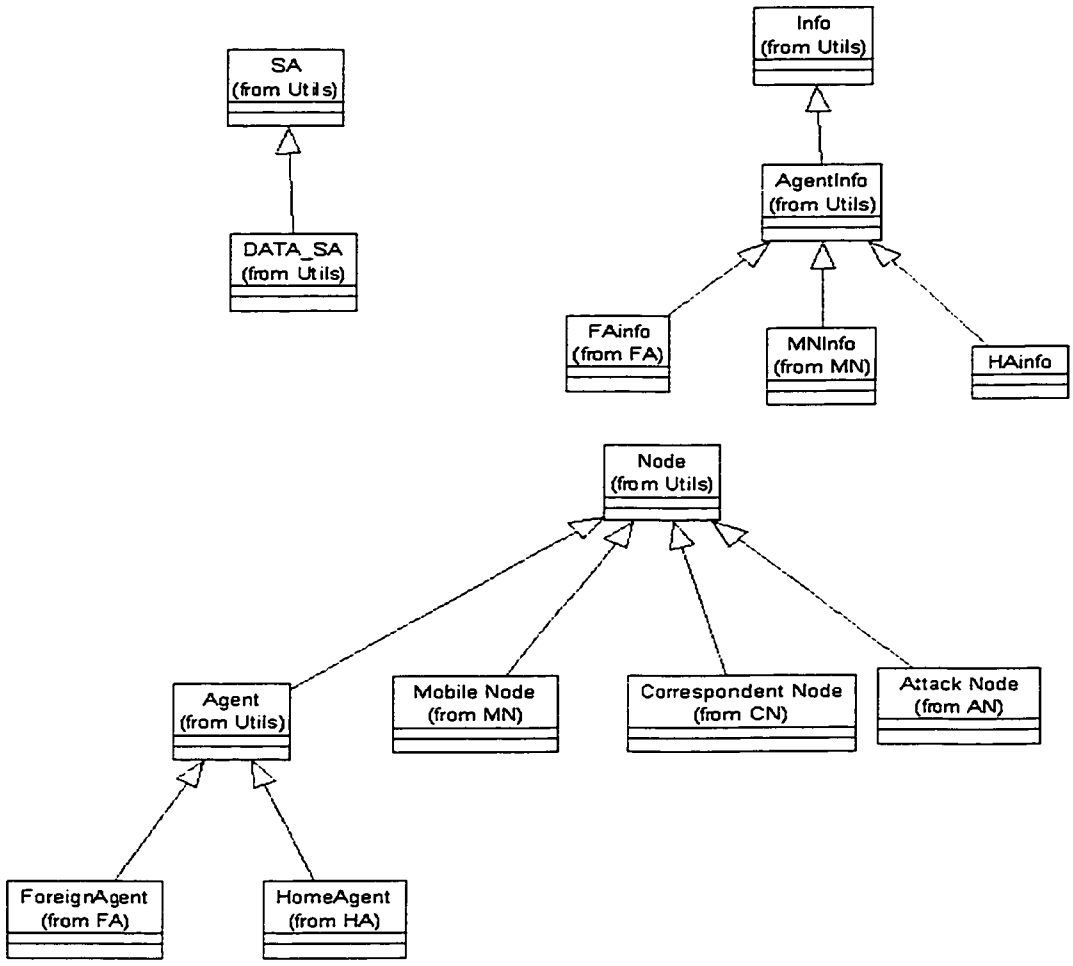

Figure 5.4. Class Inheritance Diagram

5.3 Sequence Diagrams

The following sequence diagrams and descriptions depict the basic scenarios in the SecMIP simulation. The basic scenarios include successful registration in secure mode, deregistration, IKE negotiation, ESP_IN process and ESP_OUT process. Every entity has different threads to listen to the message from the other entities. For example, MNFA thread is the thread of MN entity (package) to listen to the traffic from the FA, and the FAMNthread is the thread of FA entity (package) to listen to the traffic from MN. A

message starting with a capital letter is the encrypted or authenticated or processed result of the message. For example, in figure 5.5, *Registration* is the ciphertext of registration processed by ESP and *registration* is the plaintext. More details of the classes can be found in Appendix B.

5.3.1 Successful Registration in Secure Mode

When the MN moves from the HA to the FA, it gets the care-of-address (COA) and sends a registration request packet to the FA. If the MN does not get the registration reply in 4 minutes, it sends the registration request again until it gets the registration reply. When the FA gets the registration request, it sends the IKE begin signal to the HA and the HA sends back the IKE begin signal to the FA. When the FA gets the HA IKE begin signal, it starts IKE Phase1 for IKESA negotiation and IKE Phase 2 for DATASA negotiation. The HA will react accordingly. When The IKEPhase2 is finished, the HA will signal the FA that the IKE negotiation process is ended. If IKE negotiation is successfully finished, the HA and the FA will initiate their ESP_IN processing for the incoming packets and ESP_OUT processing for outgoing packets. From then on, all traffic between the HA and the FA including registration request and reply will be subject to ESP_IN or ESP_OUT processing. When the HA gets the registration request, it updates the Mobile Node location information and sends the registration reply to the FA. The FA relays the registration reply to the Mobile Node. The MN updates its information. For the insecure mode, the IKE negotiation, ESP_IN and ESP_OUT are omitted. For details of IKE negotiation, refer to Figure 5.7a, Figure 5.7b and Figure 5.7c, the sequence diagrams for IKE negotiation.

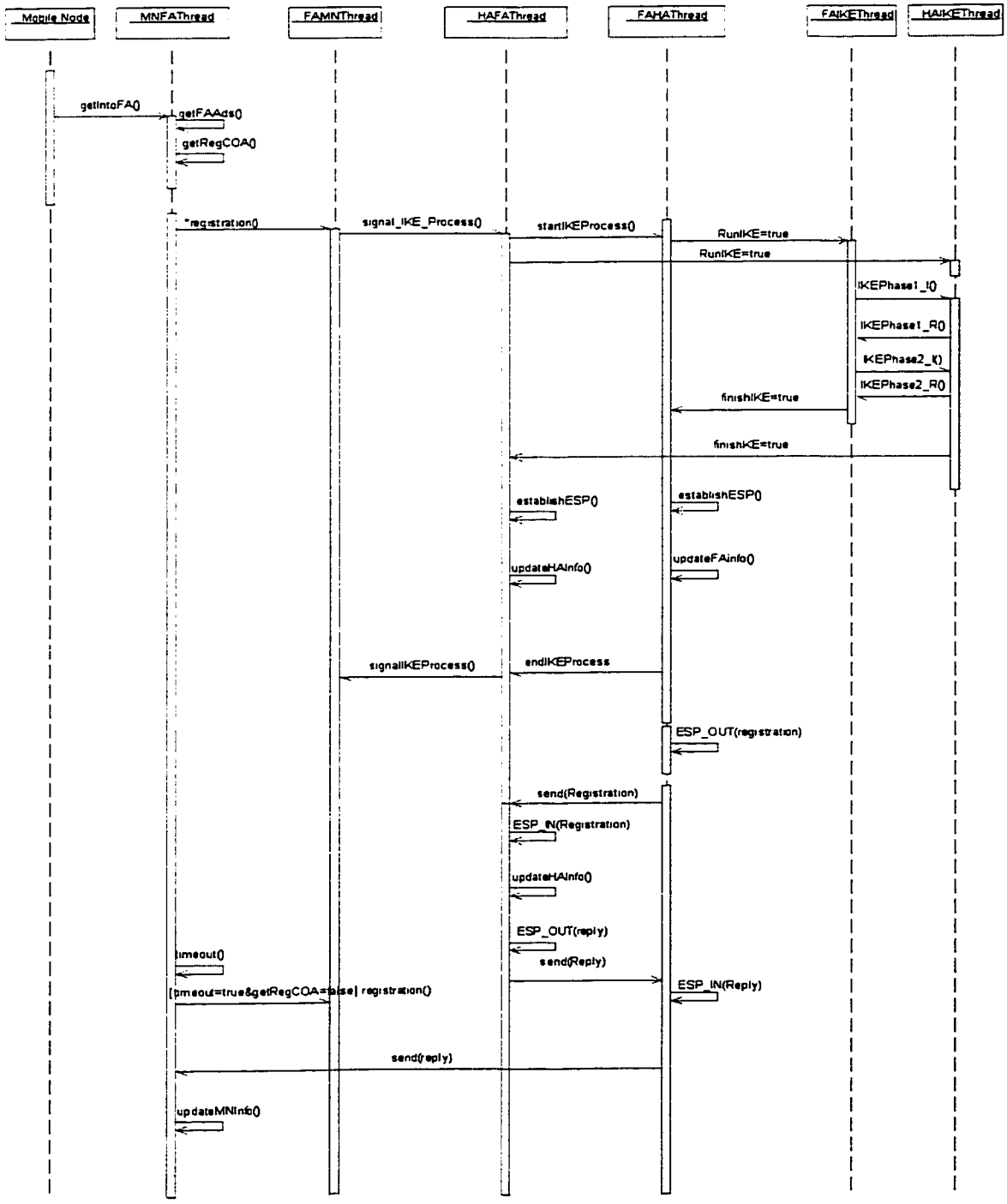

Figure 5.5. Sequence Diagram for Successful Registration in Secure Mode

5.3.2 De-registration

When the MN moves from the FA to HA after successful COA registration, it sends a deregistration request packet to the HA. When the HA gets the MN deregistration request, it updates the MN location information and sends back the deregistration reply to the MN. The MN updates its information. Figure 5.5 shows the sequence diagram for de-registration.

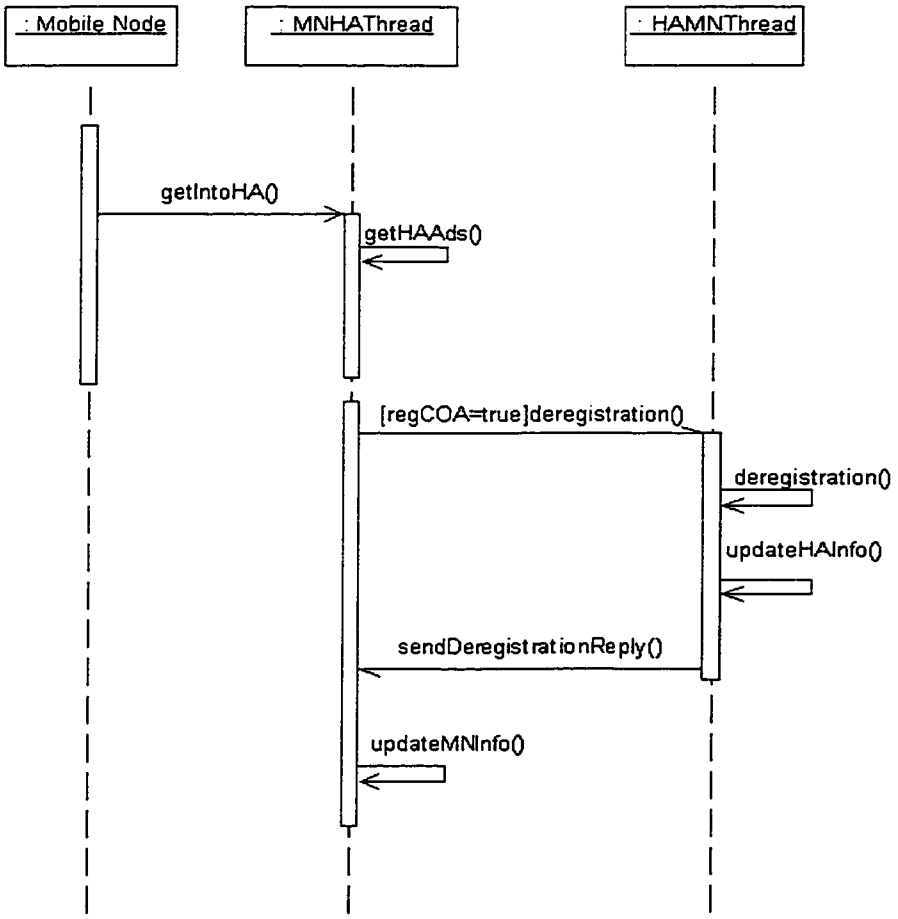

Figure 5.6. Sequence Diagram for Deregistration

5.3.3 IKE Negotiation

First, the FA and HA negotiate the shared secret by the Diffie_Hellman algorithm. This phase is the IKEInit Phase. After that, if the IKE thread of the FA detects that the RunIKE is true, the IKE negotiation begins.

The IKE negotiation process follows the following sequence:

1) If there is no security policy available, a new policy is created. IKE phase1 and phase2 are executed.
2) If the IKESA is expired, IKE phase1 and phase2 are executed.
3) If the IKESA is available but if there is no DATA_SA or DATA_SA is expired, IKE phase2 is executed.
4) If the DATA_SA is available, IKE phases will not execute.

The IKE negotiation returns a Boolean value to indicate the success of the negotiation. As a result, the HA and the FA have the same IKESA and DATASA for the data communication.

The following are the some details of the successful IKE phases.

- IKEInit Phase
    1) An initiator generates a public key and sends it to a responder.
    2) The responder generates its public key by the received initiator's public key and sends it to the initiator.
    3) The initiator and the responder generate the shared secret by the received peer public key and derive the same DES cryptography key from the shared secret.

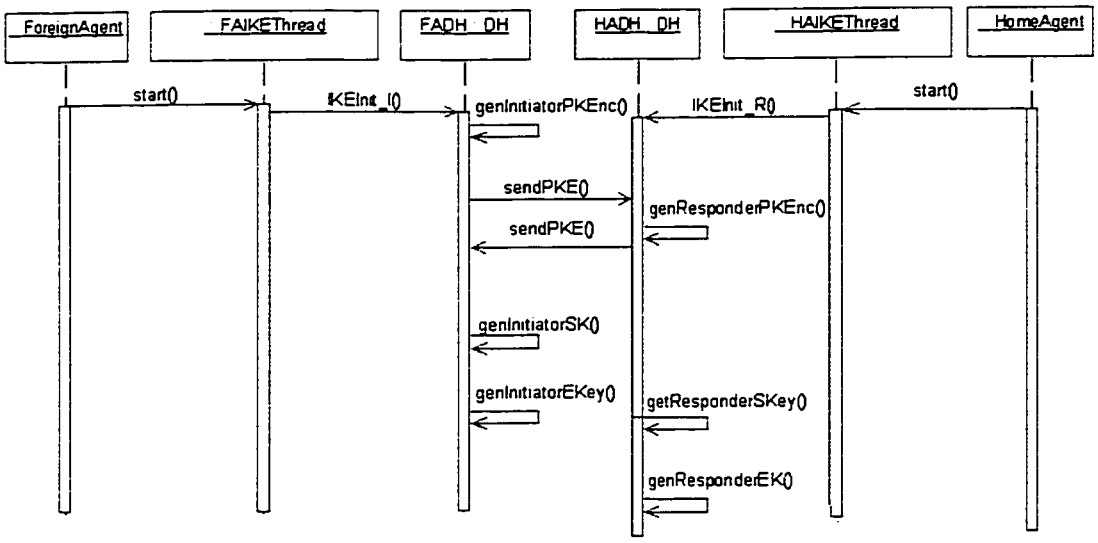

Figure 5.7a. Sequence Diagram for IKEInit Phase

- IKEPhase1

1) The initiator and responder get the DESEncrypter from the same DES key.
2) The initiator initiates the authenticator engine and encrypts the authentication key and sends to the responder.
3) The responder decrypts the encrypted authentication key from initiator and initiates the authentication engine.
4) The initiator and the responder authenticate their own nonce (the identity of agent) and get the nonce checksums.
5) The initiator creates the IKESA packet and appends the initiator nonce to the packet.
6) The initiator encrypts the encapsulated IKESA and sends it to the responder.
7) The responder decrypts the encrypted proposed SA packet from the initiator and extracts the initiator nonce.

8) The responder authenticates the initiator nonce and gets the initiator nonce checksum.

9) The responder encrypts the SA reply packet, which includes the SA, the responder nonce and the initiator nonce checksum and sends it to the initiator. The initiator decrypts the SA reply packet and extracts the initiator nonce checksum and responder nonce from the SA packet.

10) The initiator gets the encrypted SA reply packet from the responder.

11) The extracted initiator nonce checksum from the SA reply packet is compared with the initiator nonce checksum.

12) If the checksums are identical, the initiator authenticates the responder nonce and the responder nonce checksum is encrypted and sent back to the responder.

13) The responder decrypts and compares the received checksum with its own checksum.

14) The initiator and the responder update their SPD table and IKESA table.

15) The initiator and the responder send a synchronization signal to each other.

Figure 5.7b. Sequence Diagram for IKEPhase1

- IKEPhase2

1) The initiator and the responder check their SPD and IKESA.
2) The initiator and the responder initiate their DesEngine and Authentication Engine from the same IKESA.
3) The initiator creates the DATASA packets.
4) The initiator authenticates the DATASA packet and the checksum is appended to the DATASA packet.
5) The initiator encrypts the DATASA packet and sends it to the responder.
6) The responder gets and decrypts the initiator's encrypted DATASA packet.
7) The responder extracts the DATASA and the initiator checksum from DATASA packet.
8) The responder authenticates the DATASA and gets the responder checksum.
9) The responder compares the responder checksum and the initiator checksum.
10) If the checksums are identical, the responder appends the responder checksum to DATASA.
11) The responder encrypts the DATASA packet and sends it back to the initiator.
12) The initiator gets the responder DATASA reply packets and decrypts it.
13) The initiator extracts the checksum from the DATASA reply packet and compares with its DATASA packet checksum.

14) The responder creates the DATASA by the DATASA packet.

15) The initiator and the responder update the SPD table and DATASA table.

16) The initiator and the responder send a synchronization signal to each other.

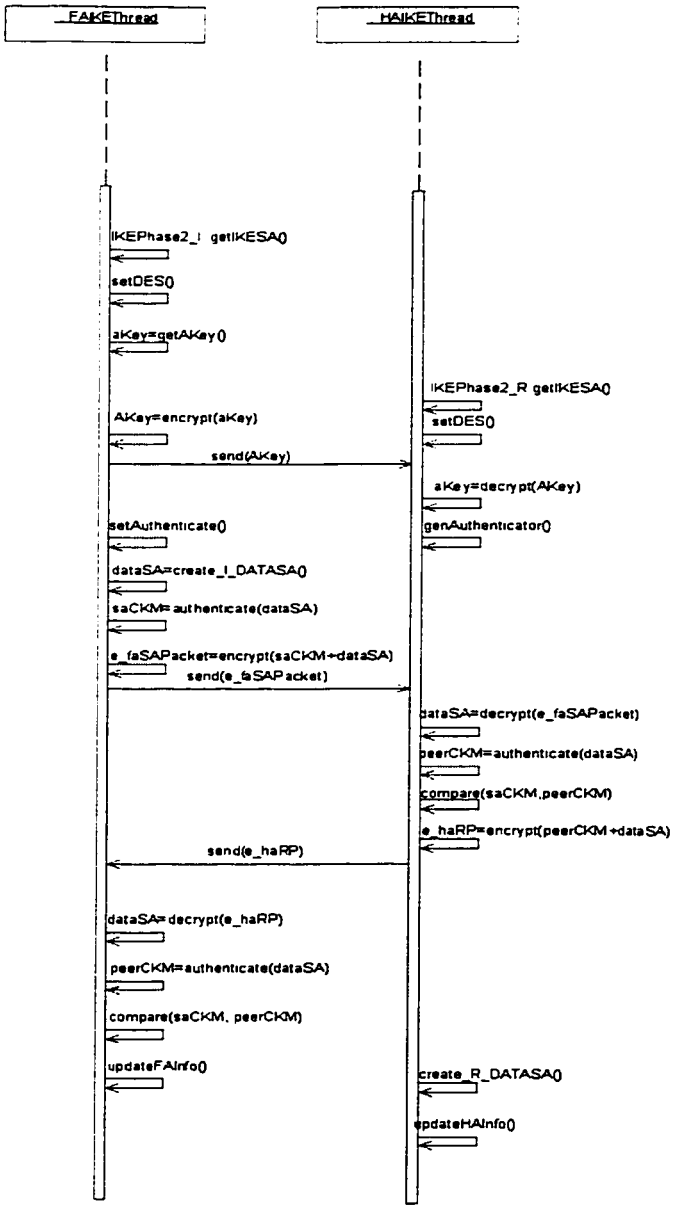

Figure 5.7c. Sequence Diagram for IKEPhase2

5.3.4 ESP Inbound Processing

After the successful IKE negotiation, the ESP_IN class processes the incoming data in the following sequence.

1) Look up the security policy in the SPD table by the source address.
2) Extract the SPI and sequence number from ESP header of the data packet.
3) Look up the DATA_SA by the extracted SPI.
4) Check the DATA_SA expiration.
5) Get the DESEncrypter and Authenticator from the DATA_SA.
6) Compare the sequence number with the expected sequence number in the DATA_SA.
7) Update the sequence number in the DATA_SA.
8) Check for data packet using Authenticator.
9) Process decryption using DESEncrypter.

Any exceptions will result in the data packet being dropped. Figure 5.8 shows the sequence diagram for the ESP_IN processing.

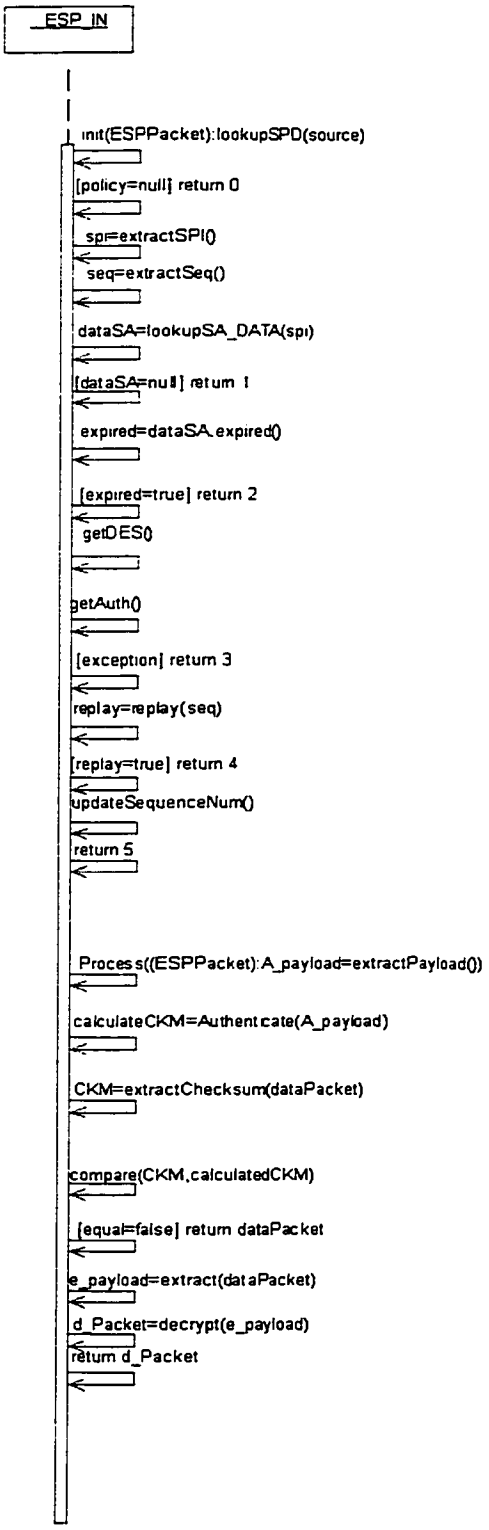

Figure 5.8. Sequence Diagram for ESP Inbound Processing

5.3.5 ESP Outbound Processing

After the successful IKE negotiation, the ESP_OUT class processes the outgoing data packet in the following sequence:

1) Look up the security policy in the SPD table by the source address.
2) Look up DATA_SA by the SPI from the SPD.
3) Check the DATA_SA expiration.
4) Get the DESEncrypter and Authenticator ready if the SA exists.
5) Encrypt the original data packet using DESEncrypter.
6) Encapsulate the DATA_SA SPI and sequence number as the header to the encrypted data packet.
7) Authenticate the encapsulated packet using the Authenticator.
8) Add the checksum to the tail of the encapsulated packet.
9) Update the sequence number.

Any exception will result in ESP protocol processing failure and return the original data packet. Figure 5.9 shows the ESP_OUT processing sequence.

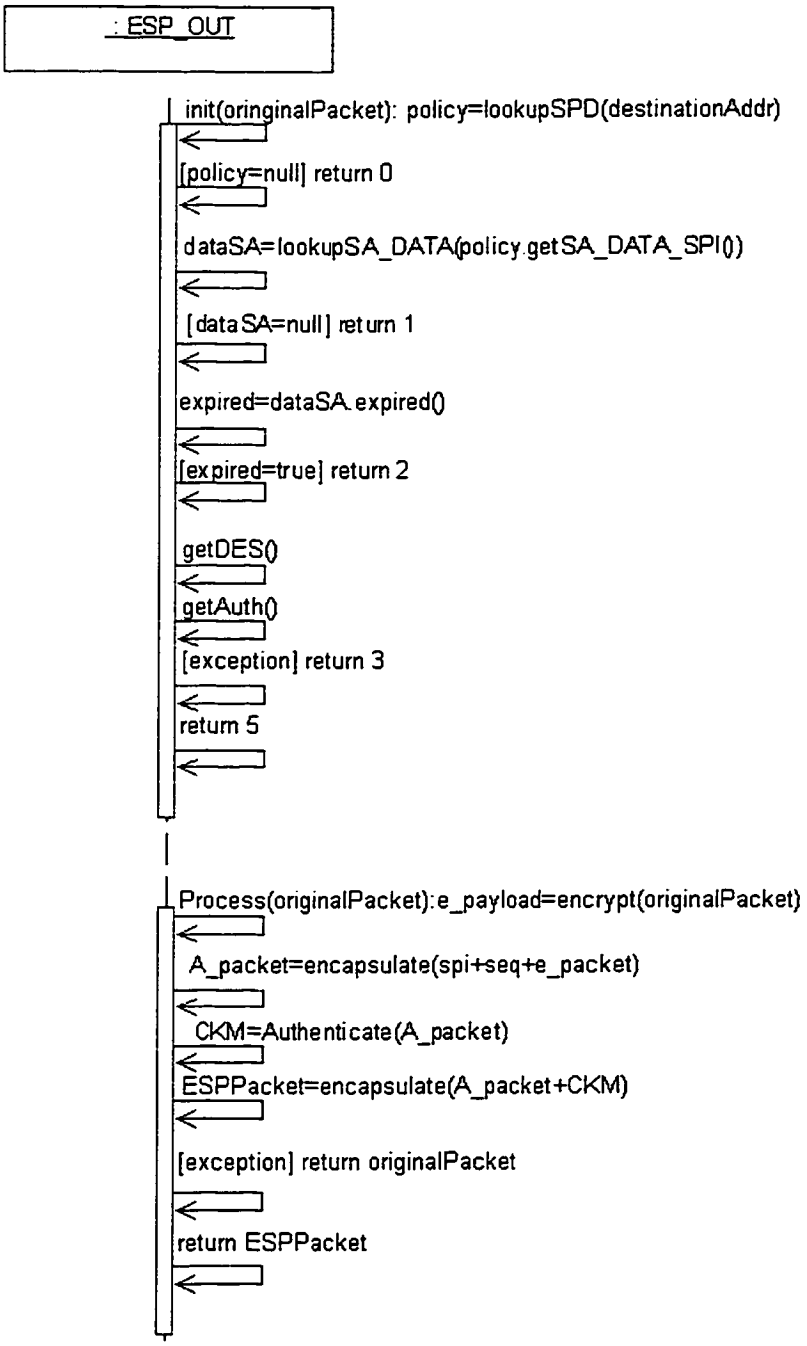

Figure 5.9. Sequence Diagram for ESP Outbound Processing

5.3.6 Sending A Message from CN to MN

When the CN sends a message to the MN, the HA intercepts the packet. The HA checks the location and registration of the MN. If the MN is in the home network, the HA will forward the packet directly to the MN. If the MN is outside the home network and has already registered the COA, the HA will add the MIP header to the CN packet. Otherwise, the HA informs the CN that the MN cannot be reached. Then, if ESP is available between the HA and the FA, the HA will perform the ESP_OUT processing for the packet and send it to the FA. The FA will do the reverse process to recover the original packet. The FA forwards the original packet to the MN. Figure 5.10 shows the sequence diagram for the CN sending a packet to the MN.

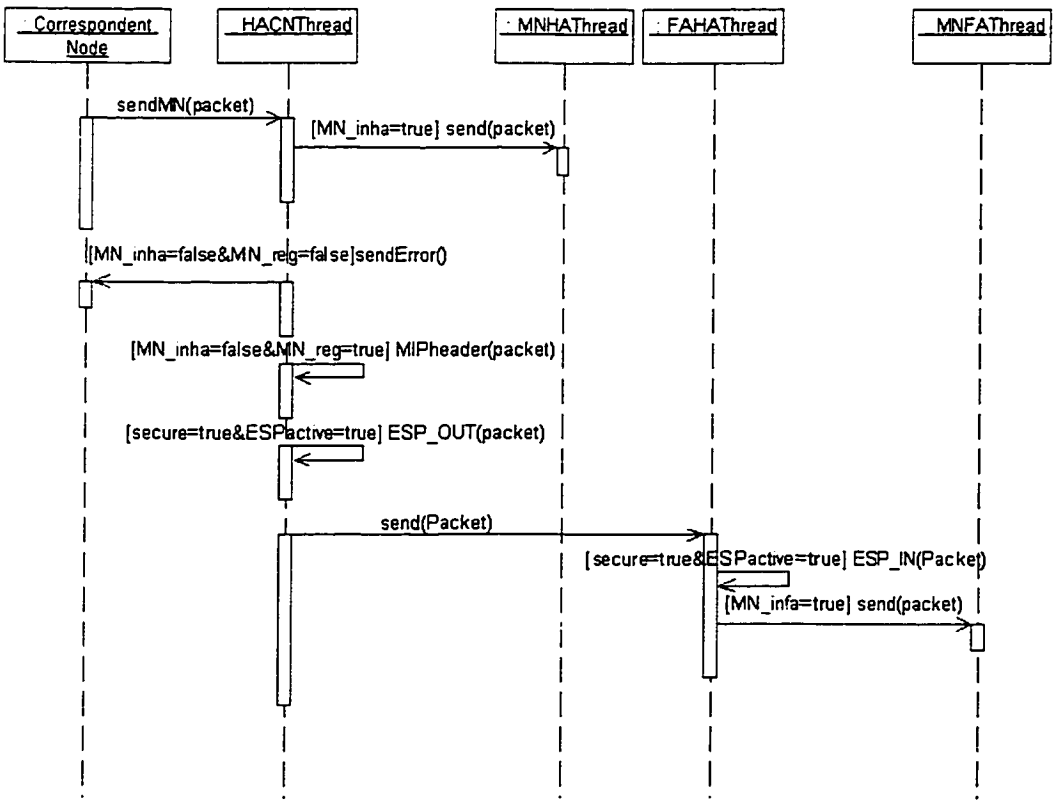

Figure 5.10. Sequence Diagram for CN Sending A Message to MN

5.3.7 Sending A Message from MN to CN

When the MN sends a message to the CN, the MN checks its own location. If the MN is in the home network, it sends the packet to the HA and the HA forwards the packet to the CN. If the MN is in the foreign network and has already registered its COA, it sends the packet to the FA. Otherwise, the MN cannot send anything to the CN. Then, if ESP is available between the HA and the FA, the FA will perform the ESP_OUT processing for the packet and the HA will do the reverse process to recover the original packet. The HA forwards the original packet to the CN. Figure 5.11 shows the sequence diagram for the MN sending a packet to the CN.

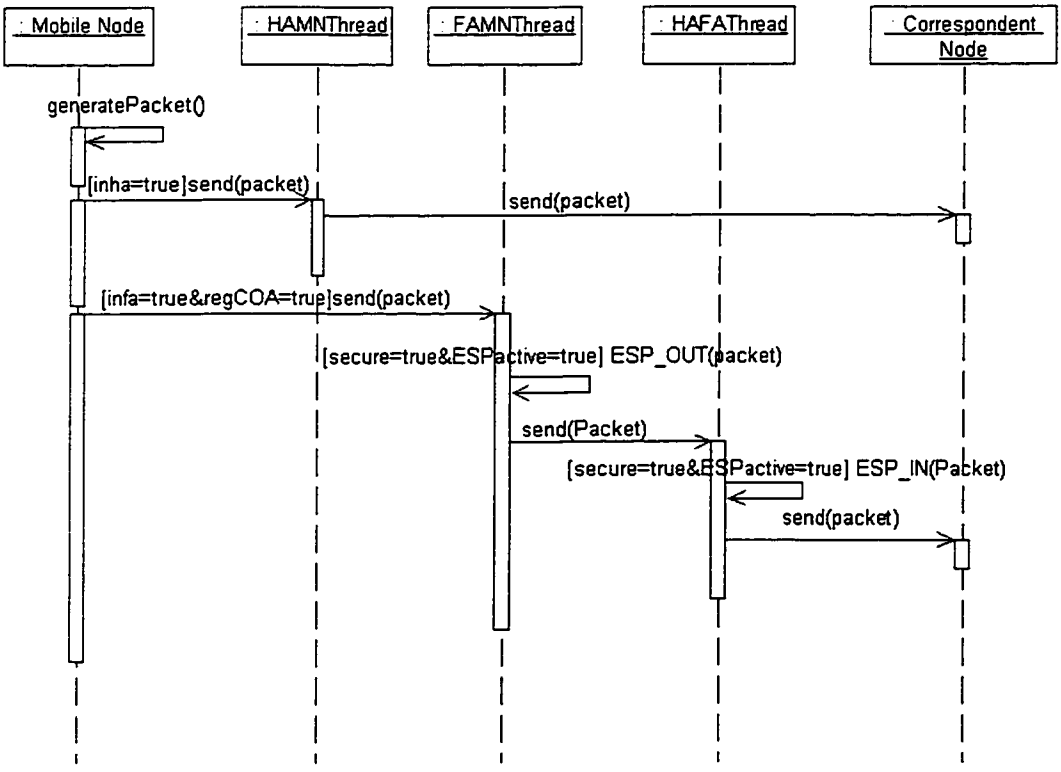

Figure 5.11. Sequence Diagram for MN Sending A Message to CN

5.3.8 Perform Attack

When the AN performs an attack, it generates different types of attack packets for the different attack types: denial of service, replay attack and session hijacking. For details on how attack packets are generated, please refer to the Appendix II. Then the AN chooses a victim (HA/FA) to attack. If the chosen agent is under the secure mode and ESP is available, it will perform ESP_IN processing for the malicious packet, which results in dropping the attack packet. Otherwise, the agent is attacked. For more details, please refer to the result of attacks in Appendix B. Figure 5.12 shows how the HA defends against attacks under the secure/insecure mode. The FA defense is similar to that of the HA.

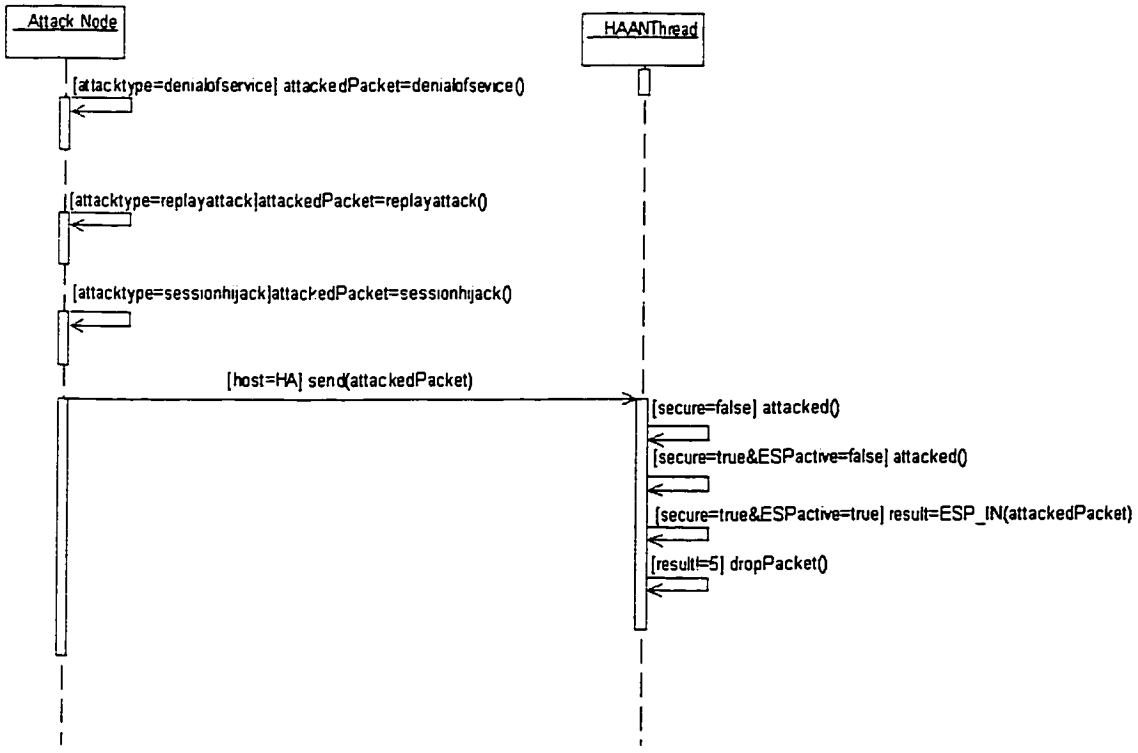

Figure 5.12. Sequence Diagram for AN Attacking HA

5.4 Extensibility and Future Work

In this section, we describe some ways in which the SecMIP simulator may be easily extended. Object-oriented design and implementation methods were used in this project, which will make many extensions easy to implement. For example, there is one FA, one HA, one MN, and one CN, but more of each entity can be easily implemented. More suggestions for future development are given below.

For the SecMIP functionalities, the UDP protocol for data communication is recommended because the UDP protocol is more suitable for the large Mobile IP simulation deployment than the TCP/IP protocol. The reason is that UDP doesn't need to establish and maintain a connection like TCP does. It is faster for sending small amounts of data without connection setup. In addition, the no guarantee for data delivery in UDP is more suited to the Mobile IP scenario.

For the multiple nodes extension, the FA and HA classes need to be investigated further. When multiple AN, CN and MN nodes are implemented, more socket connections should be created and a database implementation should be involved to keep the information on multiple nodes. For example, an FA needs to keep the different COAs for different MNs and an HA keeps track of the different MN registrations to use. In the SecMIP implementation, it is quite straightforward and efficient to use flags to denote the node status. When multiple FA and HA nodes are implemented, the database implementation plays a more important role for information management to facitilate the communication between nodes.

Similar extensions need to be made to the IPSec implementation for the multiple node implementation. It is important to develop database functionalities to support the SPD,

IKESA and DATA_SA for different MN nodes. Currently, the bundles of SPD, IKESA and DATA_SA are implemented by Hashtable.

In addition, the cryptography algorithms and authentication algorithms are limited to DES, RSA, HmacMD5 and HmacSHA1. Other choices for these algorithms such as DESede and Blowfish can be easily provided. An interface for the SA parameters selection can be created to facilitate the usage.

For the SecMIP simulator, only IKE and ESP are implemented for IPSec. In the future, the AH protocol can be extended. The concatenation of AH and ESP protocols can provide more secure performance for data communication.

6 Conclusions

Mobile IP provides a solution for network mobility in the network layer. It also increases the security risks for network traffic. In this report, we first investigated typical Internet vulnerabilities and solutions since the Mobile IP risks partly come from the current open network model. In addition, we also discussed the Mobile IP-specific threats due to lack of strong security options in the protocol. Some security technologies have to be applied to Mobile IP; IPSec is demonstrated to be an adequate technology to provide a security solution for Mobile IP. We also describe a simulation of an integration of IPSec and Mobile IP: SecMIP, which is a preliminary prototype for the IPSec and Mobile IP combination.

IPSec is a set of protocols to provide confidentiality and authentication extensively in the IP layer. Since Mobile IP and IPSec are protocols in the same layer in the network model, the combination of the two is a powerful way to preserve both IP mobility and data security. IPSec for Mobile IP does have some limitations such as no traffic analysis for AH & ESP and no non-repudiation for AH & ESP with proposed algorithms (keyed MD5, DES/CBC). These limitations will be the subjects of future work in Mobile IP security research.

Besides IPSec, there also exist other security architectures and protocols to deal with special security problems of Mobile IP such as the AAA protocol (Authentication, Authorization and Accounting) [6] and the PKI protocol (Public Key Infrastructure) [25]. The AAA protocol provides strong authorization and authentication for network access and PKI protocol provides an infrastructure for the strong key distribution and

management for data communication. These protocols can be integrated with IPSec to provide a more secure environment for Mobile IP in the future.

Currently, the security measures for Mobile discussed in this paper are based on the IPV4. Both Mobile IP and IPSec are extensions to IPV4. Existing network equipment can be enhanced with these features but they cannot be expected to perform optimally. IPv6 is a new version of IP, which is designed to be an evolutionary step from IPv4. For the IPV6 protocol, mobility and IPSec are mandatory headers of data packet [17]. Furthermore, security features in IPv6 have been introduced mainly by way of two dedicated extension headers: the Authentication Header (AH) and the Encrypted Security Payload (ESP). They can be used separately or together. The core set of IPv6 protocols were made by IETF on August 10, 1998 and the IPV6 specification is still undergoing. Security concerns have been taken into account much more in IPV6 than in IPV4. There may yet be unforeseen security holes with the integration of Mobile IP with IPV6.

References

[1] Marc Danzeisen, Torsten Braun, "Secure Mobile IP Communication", Institute of Computer Science and Applied Mathematics, University Bern, 2001.

[2] Marcel Dekker, "Security of the Internet", *The Froehlich/Kent Encyclopedia of Telecommunications vol. 15*, New York, 1997, pp. 231—255.

[3] Terry Escamilla, "Intrusion Detection: network security beyond the firewall", New York, John Wiley, 1998.

[4] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

http://www.ietf.org/rfc/rfc2409.txt, October 2002.

[5] H. Hansen, "IPsec and Mobile-IP in Mobile Ad Hoc Networking", Helsinki University of Technology, April 2000.

http://www.tcm.hut.fi/Opinnot/Tik-110.551/2000/papers/IPsec/index.html, October 2002.

[6] Matthias Hollick, "The Evolution of Mobile IP Towards Security", German National Research Center for Information Technology Institute IPSI, 2000.

[7] Tom Karyginnis, Les Owen, "Wireless Network Security 801.11,Bluetooth and Handheld device", Internet Draft, NIST special publication 800-48,October 2001.

http://csrc.nist.gov/publications/drafts/draft-sp800-48.pdf, October 2002.

[8] S. Kent, "IP Authentication Header", RFC 2402, March 2002.

http://www.ietf.org/internet-drafts/draft-ietf-ipsec-rfc2402bis-00.txt , October 2002.

[9] S. Kent, "IP Encapsulating Security Payload (ESP)", RFC 2406, March 2002.

http://www.ietf.org/internet-drafts/draft-ietf-ipsec-esp-v3-02.txt, October 2002.

[10] S. Kent, Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[11] Hugo Krawczyk, "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", IBM Research Center, 1996.

http://citeseer.nj.nec.com/cache/papers/cs/2735/http:zSzzSzwww.isoc.orgzSzconferences zSzndss96zSzkrawczyk.pdf/krawczyk96skeme.pdf , October 2002.

[12] D. Maughan, M. Schneider, M. Schertler, J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC2408, November 1998.

http://www.ietf.org/rfc/rfc2408.txt, October 2002.

[13] H. Orman, "The OAKLEY Key Determination Protocol", November 1998.

http://www.ietf.org/rfc/rfc2412.txt, October 2002.

[14] C. Perkins, "IP Mobility Support for IPv4", RFC2002, January 2002.

ftp://ftp.ietf.org/rfc/rfc3220.txt, October 2002.

[15] C. Perkins, "Mobile Networking in Internet", *Mobile Networks and Applications*, December 1998, Volume 3 Issue 4.

[16] C. Perkins, "Mobile Networking Through Mobile IP", online tutorial

http://www.computer.org/internet/v2n1/perkins.htm, October 2002.

[17] C. Perkins, "Mobility Support in IPv6", February 2000.

http://www.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-10.txt, October 2002.

[18] Richard D. Pethia, "Internet Security Issues", Pittsburgh, May 25, 2000.

[19] Fred Piper, Simon Blake-Wilson, John Mitchell, "Digital Signatures - Security & Controls", Information Systems Audit and Control Foundation, 1999.

[20] Anna Ren, "Inter-Access Point Protocol (IAPP)" slides, November 1997.

http://www.it.kth.se/~ren/iapp3/iapp3-1.html, October 2002.

[21] James D. Solomon, "Mobile IP, the Internet unplugged, Prentice Hall", 1998.

[22] Rita C. Summers, "Secure computing: threats and safeguards",

New York, McGraw-Hill, 1997.

[23] Fred Simonds, "Network security: data and voice communications"

New York, McGraw-Hill, 1996.

[24] J. Zao, M. Condell, "Use of IPSec in Mobile IP", Internet Draft, November 1997.

ftp://ftp.ietf.org/internet-drafts/draftietf-mobileip-ipsec-use-00.txt, October 2002.

[25] John Zao, Joshua Gahm, Gregory Troxel, Matthew Condell, Pam Helinek, Nina Yuan, Isidro Castineyra, Stephen Kent, "A public-key based secure Mobile IP"

Wireless Networks, Vol. 5, Issue 5, October 1999.

[26] Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman, "Building Internet Firewalls", Beijing, Cambridge, O'Reilly, 2000.

Appendix A Use Cases

1.0 UC0: Main Use Case

The SecMIP provides five operations for the user:

- Move MN between HA and FA
- Set HA/FA in secure/insecure mode
- Send messages from MN to CN
- Send messages from CN to MN
- Perform Attacks

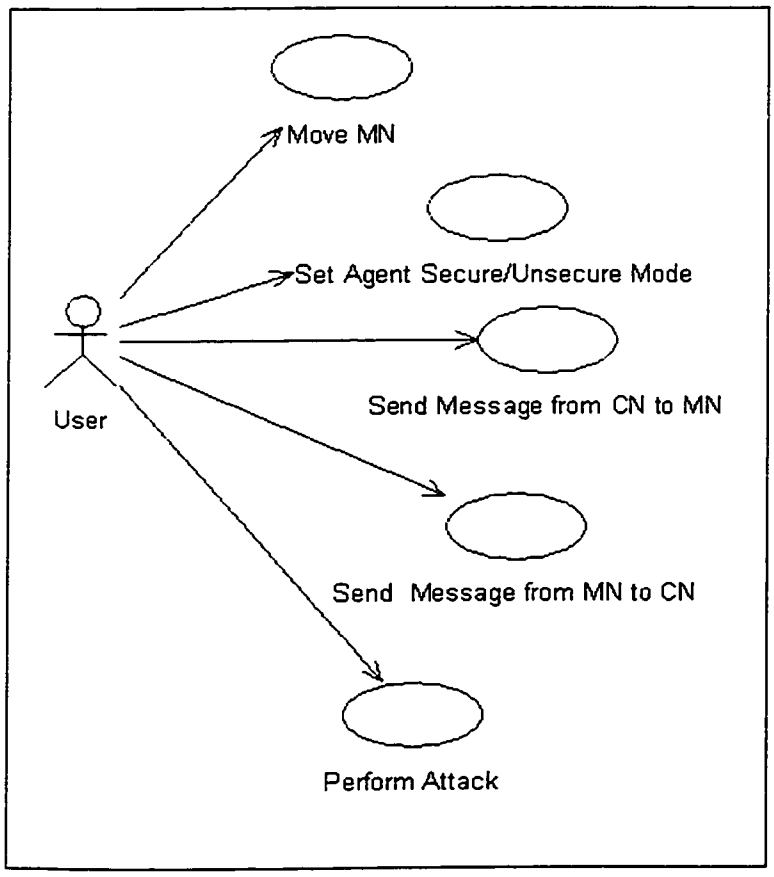

Figure A1. Main Use Case

There is another use case: IKE negotiation. The user does not initiate it, but it is very important for security in Mobile IP. Next we describe each use case in detail.

1.1 UC1: Move MN

Actor: User

Description:

The use case begins when the user moves the mobile node and completes when the mobile node gets the COA.

Pre-condition:

User initiates the moving action.

Post-condition:

The system knows whether the mobile node is connected to the home agent or the foreign agent.

Main Path:

1) The user moves the mobile node between the range of the home agent and the foreign agent.
2) The system indicates the location of the mobile node (foreign agent /home agent/neither).
3) If the movement is from the home agent to the foreign agent, the system shows that the mobile node has sent a registration request to the foreign agent.
4) If the movement is from the foreign agent to the home agent, the system displays that the mobile node has sent a de-registration request to the home agent.

Alternative Path:

3a, 4a: If the user moves the mobile node outside the ranges of the home agent and the foreign agent, the system will indicate that the mobile node is out of service.

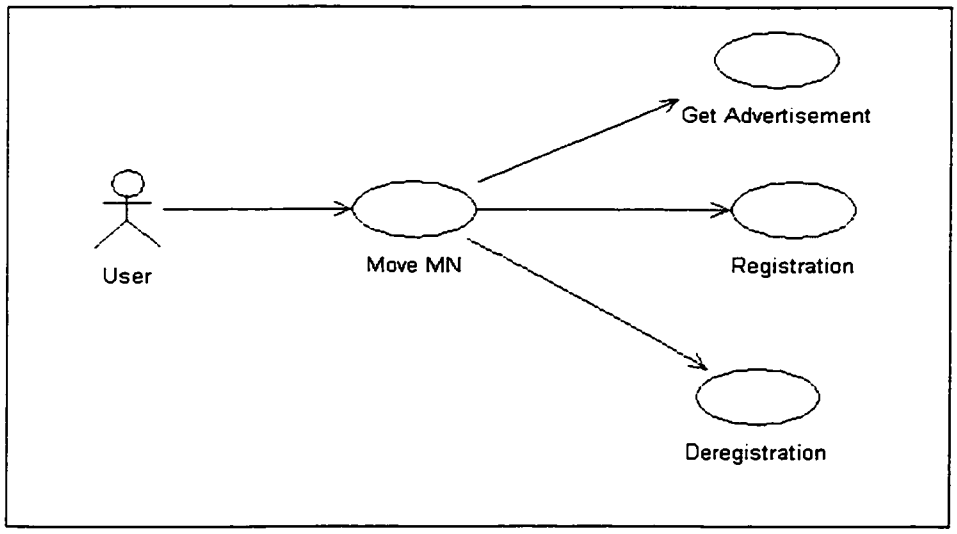

Figure A2. Move MN Use Case

1.2 UC2: Set Agent Secure /Insecure Mode

Actor: User

Description:

The use case begins when the user sets the agent to secure/ insecure mode and completes when the agent is in secure/insecure mode.

Pre-condition:

User initiates the mode setting.

Post-condition:

If the agent is in secure mode, it is able to perform the IPSec processing if the security association is available. If the agent is in insecure mode, there is no security protection available for Mobile IP scenarios.

Main Path

1) The user initiates the mode setting.
2) The user sets the agent to secure mode.
3) The system indicates that the agent is in secure mode.

Alternative Path

2a. User sets the agent mode to insecure mode.

3a. The system indicates that the agent is in insecure mode.

1.3 UC3: Send Message from CN to MN

Actor: User

Description:

The use case begins when the user initiates the sending of a message from the correspondent node to the mobile node's home address without knowing the exact location of the mobile node and completes when the mobile node gets the message.

Precondition:

Correspondent node knows the mobile node's home address.

Post-condition:

The mobile node receives the message.

Main Path:

1) The user initiates the sending of a message from the correspondent node to the mobile node.
2) The system intercepts the message.
3) The system checks the mobile node's location and decides whether the message needs to be tunnelled or not.
4) If the mobile node is within the range of the foreign agent, the system will indicate that the home agent performs Mobile IP processing and then the IPSec tunnel processing if IPSec is available.
5) After tunneling, the system indicates the current security association sequence number.
6) The system indicates that the foreign agent performs Mobile IP processing and the IPSec tunneling processing in reverse sequence.
7) The system indicates the current security association sequence number at that foreign agent.
8) The system indicates that the foreign agent relays the message to the mobile node.
9) The system indicates that the mobile node gets the message.
10) The user looks at the message.

Alternative Path:

4a. If the mobile node is connected to the home agent, the system will indicate that the home agent forwards the message to the mobile node.

4b. If the mobile node is connected to neither the home agent nor the foreign agent, the system will indicate that the home agent sends an error message to the correspondent node.

6a. Any failure of IPSec tunneling and Mobile IP process will be logged and indicated explicitly.

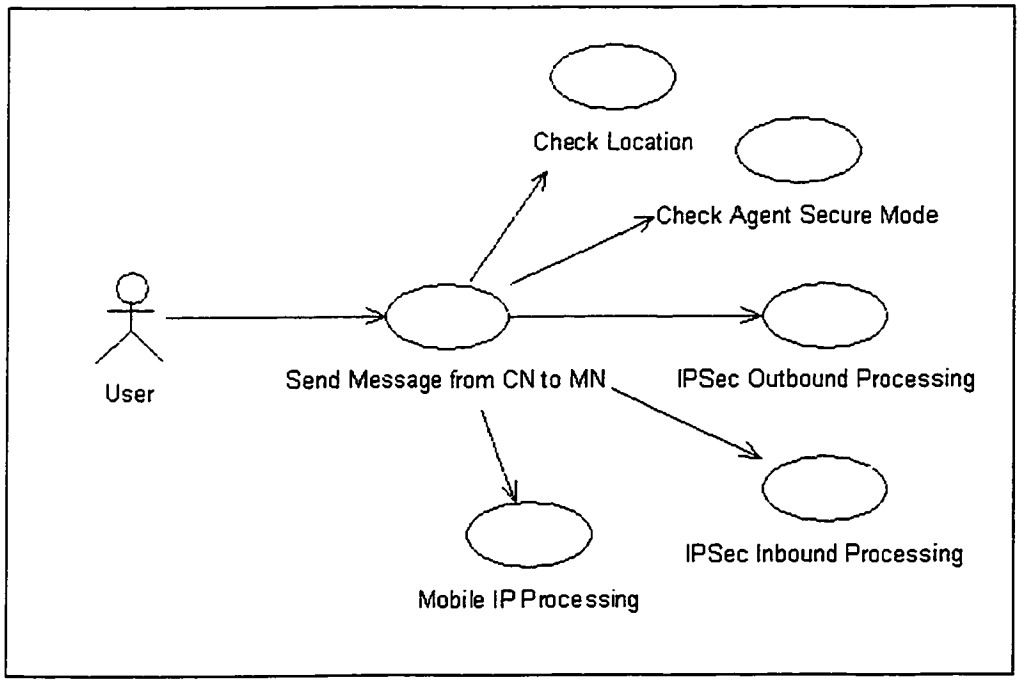

Figure A3. Send Message from CN to MN Use Case

1.4 UC4: Send Message from MN to CN

Actor: User

Description:

The use case begins when the user initiates the sending of a message from the mobile node to the correspondent node and completes when the correspondent node gets the message.

Pre-condition:

Mobile node knows the correspondent node's home address.

Post-condition:

The correspondent node receives the message.

Main Path

1) The user initiates the sending of a message from the mobile node to the correspondent node.
2) The system checks the mobile node status to decide whether the message needs to be tunnelled or not.
3) If the mobile node is at the foreign agent and has finished registration, the system will intercept the message at the foreign agent.
4) The system indicates that the foreign agent performs Mobile IP processing and the IPSec tunneling if IPSec is available.
5) The system indicates the security association sequence number at the foreign agent.
6) The system indicates that the home agent performs the Mobile IP processing and IPSec tunneling in reverse sequence if IPSec is available.

7) The system indicates the current home agent sequence number.

8) The system indicates that the home agent relays the message to the correspondent node.

9) The system indicates that the correspondent node gets the message.

10) The user looks at the message.

Alternative Path:

2a. If the mobile node is connected to the home agent, the system will indicate that the home agent forwards the message to the correspondent node.

2b. If the mobile node is connected to neither the home agent nor the foreign agent, the system will indicate that the message could not be sent.

3a. If the registration process has not finished yet, the system will indicate that the sending of the message is denied.

6a. Any failure of IPSec tunneling and Mobile IP processing will be logged and indicated explicitly.

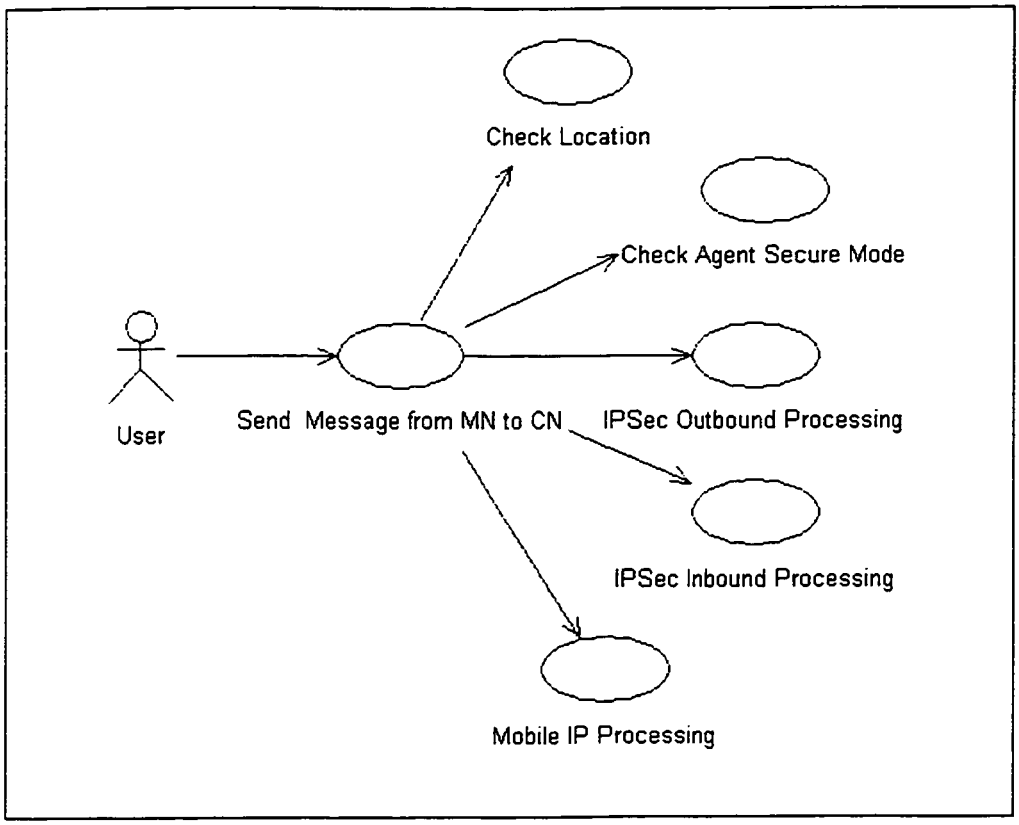

Figure A4. Send Message from MN to CN Use Case

1.5 UC5: Perform Attack

Actor: User

Description:

The use case begins when the user chooses the attack mode of the attack node and completes when the attack node sends the attack packet to the connected agent.

Pre-condition:

For the secure mode, IKE negotiation is finished between the foreign agent and home agent.

Post-condition:

The malicious packet of the attack node is dropped if a security association is available under the secure mode for the concerned agent. Otherwise, the agent is attacked.

Main Path:

1) The user sets the mode of attack (replay/denial of service /session hijacking) and the agent to be attacked (home agent or foreign agent).
2) The malicious data packet is encapsulated according to the mode.
3) The attack node sends the malicious data packet to home/foreign agent.
4) The system indicates that the agents perform the IPSec tunnel processing in secure mode that should result in dropping the malicious data packet and logging the activities.

Alternative Path

4a.If the dedicated agent doesn't have the security association or is under the insecure mode, the system indicates different results depending on the attack mode.

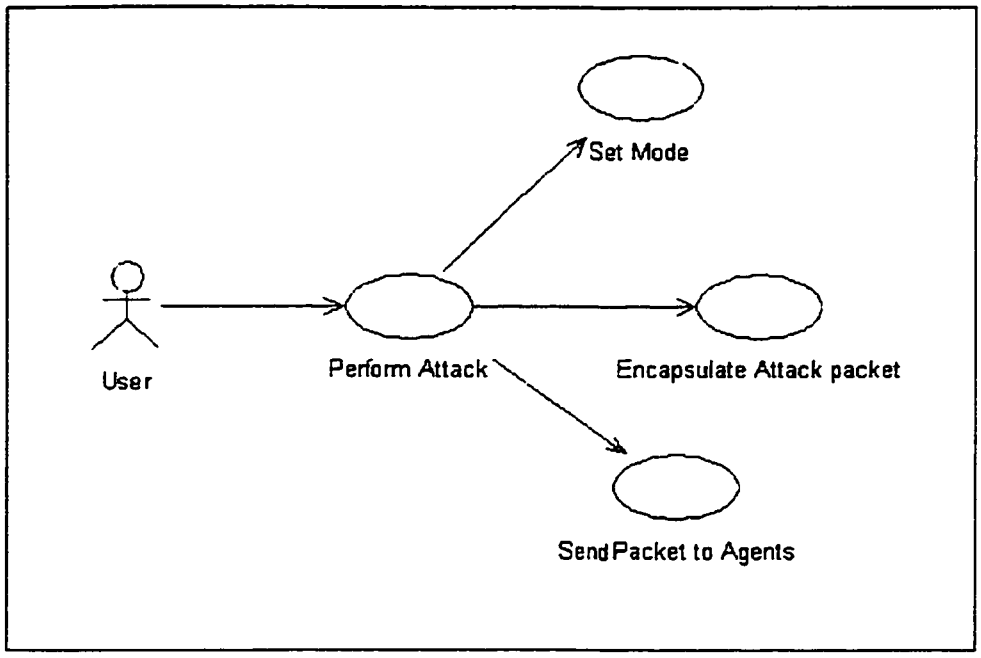

Figure A5. Perform Attack Use Case

1.6 UC6: IKE Negotiation

Actor: Foreign Agent

Description:

The use case begins when the foreign agent finds no security association for the home agent and completes when a new security association is created between the foreign agent and the home agent.

Pre-condition:

Mobile node sends a registration request to the foreign agent, the foreign agent is in secure mode and there is no security association between the foreign agent and the home agent.

Post-condition:

There is a new security association between the home agent and the foreign agent.

Main path:

1) The system negotiates shared secrets between the home agent and the foreign agent.
2) The system negotiates the IKE security association between the home agent and the foreign agent.
3) The system negotiates the DATA security association between the foreign agent and the home agent
4) The system indicates that the IPSec tunnel is successfully established.

Alternative Path:

Any failure of negotiation will be logged and indicated explicitly.

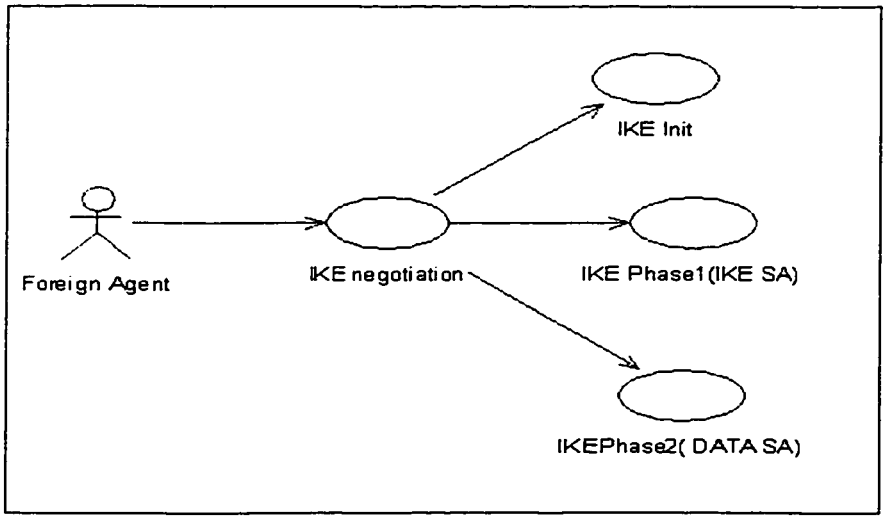

Figure A6. IKE Negotiation Use Case

Appendix B Class Descriptions

The following class descriptions briefly describe the essential elements of every class. The content of a class description includes the relationship with other classes, the purpose, important attributes, and important methods.

1 Utils Package

1.1 Agent

The Agent class is derived from the Node class. This class is used for simulating the typical agent behavior: COA advertisement. Agent has the interface adsArea to record every advertisement packet to MN. It uses the attribute MNServer to send the ads packet and AdsThread to perform the advertisement in a random period.

1.2 AgentInfo

The AgentInfo class is derived from Info class. This class keeps the attributes shared by all thread classes in an agent. In other words, the AgentInfo class is used to communicate the common internal messages between the threads in the agent. Accordingly, the attributes are roughly divided into the communication parts, the SA association part, and the IKE process flags part. Most methods in this class are the access methods (set/get) for the attributes. But the most important methods are related to SA management: adding, updating, deleting and looking up the SPD, IKE_ SA and DATA_ SA for a specific address. It is simpler and more efficient to use the HashTable to maintain the cluster of SPD, IKESA and DATA_SA since the SecMIP simulation is for basic scenarios of integration of MIP and IPSec. In the future, real database tables for complete IPSec implementation can replace the HashTables.

1.3 Authenticator

The Authenticator class is used for getting the checksum of the packet to ensure the packet integrity. In the JCE implementation, there are two authentication algorithms available: Hmac-MD5 and Hmac-SHA1. The default Authenticator constructor uses Hmac-MD5 to initiate the authentication. The attributes are *mac* and *key*. *Mac* is the authentication engine and *key* is the authentication algorithm key.

1.4 ByteClient

The ByteClient class is used for TCP/IP client socket establishment, sending packets to the server and receiving packets from the server. Since data transmission in bytes is important for cryptography and authentication, this class uses DataOutputStream to send the byte packets and DataInputStream to receive them.

1.5 ByteServer

The ByteServer class is used for the TCP/IP server socket establishment, sending and receiving packets. The port numbers used in the SecMIP are following:

AN-FA: 9999, AN-HA: 8888, FA-HA: 6000, FA-MN: 5000, FA IKE-HA IKE: 9000,

HA-CN: 8000, HA-MN: 4000. Since the data transmission in bytes is important for cryptography and authentication, this class uses DataOutputStream to send the byte packets and DataInputStream to receive the byte packets.

1.6 DATA_SA

The DATA_SA class is derived from the SA class (refer to Section 1.15). This class simulates behavior of the data security association. It inherits all the attributes and methods from the parent class. It extends the SA class by introducing the sequence number management, replay flag, protocol name and protocol mode. The lifetime of

DATA_SA is no longer than that of SA. Most methods in this class are the access methods (set/get) of the attributes. The following methods are overwritten.

toString(): overwrite the parent class method and return the DATA_SA string representation.

setAll(String msg): overwrite the parent class method and set all the attributes in DATA_SA according to the string msg. This method is the opposite method of the toString().

1.7 DESEncrypter

The DESEncrypter is used to encrypt and decrypt packets. The packet should be in byte format. According to JCE implementation, the available cryptographic algorithms for DESEncrypter are Blowfish, DES, and DESede. The DESEncrypter default constructor uses DES to initiate the ciphers.

1.8 DH

The DH class is used to generate the shared secret between two different nodes by applying the Diffie_Hellman algorithm. The shared secret is the prerequisite of the later IKE negotiation. The Diffie-Hellman algorithm is processed in the following sequence: The DH class generates the public key for the IKE initiator and sends the public key to the responder; the responder generates its public key by the initiator's public key and sends back the responder's public key to the initiator; the initiator uses the responder's public key to get the shared secret; the responder uses the initiator's public key to get the same shared secret. Both of them can derive the same DES SecretKey using the shared secret.

1.9 ESP_IN

The ESP_IN class is used to perform the ESP protocol processing for the incoming packets, which is the opposite of the ESP_OUT processing. ESP_IN and ESP_OUT are combined to establish the ESP secure tunnel for data transmission.

First, the init method performs the ESP protocol header checking for the incoming packet. The results returned are the following:

0: no policy, 1: no DATA_SA, 2: DATA_SA expired, 3: desEngine and Auth initiation unsuccessful, 4: replay attack, 5: successful.

Then, the authentication and decryption are performed after successful ESP protocol header checking. The decrypted packet will be returned, otherwise, the encrypted packet is returned.

1.10 ESP_OUT

The ESP_OUT class is used for ESP protocol processing for outgoing packets. The class processes the data packet in the opposite way compared to ESP_IN.

The format of the ESP packet is |XX| spi|XX|seq|XX|encrypted payload|XX|checksum |XX|END|XX|.

First, the init method checks the available DATA_SA for the outgoing data packets.

Result value: 0: no policy, 1: no DATA_SA, 2: expiration, 3: desEngine and Auth initiation unsuccessful, 5: successful.

Then, encryption, authentication and ESP encapsulation for outgoing data packet are performed.

1.11 IKE

The IKE class is the most important class for the security association negotiation.

For IKE negotiation in the SecMIP simulation, the FA always acts as the IKE initiator and the HA acts as the IKE responder.

There are three phases in IKE negotiation:

initPhase1: establish the DH shared secret between the initiator and responder.

Phase1: use the DES key derived from the shared secret to encrypt and decrypt the IKE SA, which contains the cryptography algorithm, cryptography key, authentication algorithm and authentication key and other parameters for the DATA_SA packet. If negotiation in this phase is successful, the initiator and the responder update their SPD table and IKE_SA table.

Phase2: use DESEncrypter and Auth in IKE_SA to encrypt, decrypt and authenticate the DATA_SA packet, which contains the cryptography algorithm, cryptography key, authentication algorithm, authentication key and other parameters for normal data packets. If negotiation in this phase is successful, the imitator and the responder update their IKE_SAtable and DATA_SA table.

1.12 Info

The Info class is the base class to provide the log file for a node. The methods are the access methods for the attributes and for initiation of the log file.

1.13 Logger

The Logger class is used to write the string that represents the data packet and the current time to a file.

1.14 Node

The Node class is the base class for all entities in SecMIP since any entity in the network can be considered a node with different functionalities. The node types are Attack Node

(AN), Correspondent Node (CN), Mobile Node (MN), Foreign Agent (FA) and Home Agent (HA). Currently, the Node class just contains the access method for the attribute-hostname, which can be used by any node. The Node class leaves more room for developing common functions shared by all nodes in the future.

1.15 SA

The SA class is used to simulate the IKE security association (IKESA). This class contains the parameters of the security association, which are destination address, SPI, authentication algorithm and key, encryption algorithm and key, start date, and lifetime. Most methods in this class are access methods.

1.16 SPD

The SPD class simulates the behaviors of the security policy. The security policy keeps the address of a peer node, the IKESA index and DATAIKE index.

2 AN Package

2.1 AN

The AN class is derived from the Node class. This class simulates three kinds of attacks: denial of service, replay attack and session hijacking; it also monitors and logs the attack activities. The denial of service simulation is sending 10 useless packets to the selected agent (HA/FA). The attack packet format is |AN|HA|Denial of service|END|.

The replay attack simulation is sending the ESP packet with sequence number 0 to the selected agent. The replay attack packet format is |XX|0|XX|0|XX|0000|XX|0000|END|XX|. The session hijacking simulation is getting the packets between the HA and FA after the MN successful registration, and modifying the data payload without changing the packet header. In the secure mode, the attack format looks like |XX|1|XX|2

|0000|XX|hijacking packet|XX|END|XX|. In the insecure mode, the payload is reversed but the header is unchanged. The attribute currType is current attack type. The values are the following: -1: no attack, 1: denial of service, 2: replay attack, 3: session hijacking. The Init method generates the interface and two threads to listen the traffic from HA and FA respectively. The createThread method creates the threads for session hijacking. The SessHijack method intercepts the agent packets and modifies the packets according the packet mode (secure/ insecure).

3 CN Package

3.1 CN

The CN class is derived from the Node class. This class simulates the CN behaviors. The CN behaviors are sending the packet to the MN and receiving the packet from the MN without knowledge of the location of the mobile node. Packets to and from the MN are forwarded by the HA. The CN is connected to the HA.

4 FA Package

4.1 FA

The FA class is derived from the Agent class. This class simulates the FA behaviors. It inherits all the attributes and methods of Agent except that Init() and disConnect() are overwritten. The main responsibilities of the FA are advertising the COA automatically, acting as the Initiator in IKE negotiation with HA, forwarding packets between the MN and the HA, and protecting them from attacks. As a result, the FA maintains four threads: thread for the HA, thread for the MN, thread for the IKE negotiation, thread for the AN. In addition, there is also a thread for COA advertisement in its parent class.

There are two modes for FA: secure mode and insecure mode. For the secure mode, the MN registration will trigger IKE negotiation and establish the ESP security associations between the HA and the FA. For the insecure mode, normal MIP scenarios work without any security. The secure mode and insecure mode should be synchronized with the HA. There are 3 interfaces to monitor MN, HA and AN packet activities.

4.2 FAANThread

The FAANThread is used for defending the FA from three kinds of attacks: Denial of Service, Replay Attack, and Session Hijacking. The FAANThread gets the packets from the AN, sets the attack types in FAInfo, and displays the results of attacks, according to the FA mode.

The Anti_attack() method defends the attack according to the FA mode and the ESPactive status. If the mode is the secure mode and the ESP is established, the ESP_IN process checks the validity of packets which will result in the AN attack packet being dropped. If the mode is the insecure mode, the FA is attacked. For the denial of service and replay attacks, the attacked packets are displayed. For session hijacking, the FA forwards packets to the AN rather than the HA.

4.3 FAHAThread

The FAHAThread class is used to signal the start, termination and failure for the FA part in the IKE negotiation. It also establishes the ESP_IN and ESP_OUT processes for the data packets in the secure mode of FA. In the secure mode, IKE negotiation will be activated by the packet from the HA: IKE request reply and this thread will be waiting until the IKE negotiation is finished. After that, the active ESP processing for inbound

and outbound packets is established. In the insecure mode, the packets from the MN are forwarded directly to the HA.

4.4 FAIKEThread

The FAIKEThread is used to negotiate the IKE SA and DATA_SA between the HA and the FA. When the FA gets the MN registration packet, the IKE negotiation is activated by the value RunIKE. When IKE negotiation is running, the FAIKEThread gets the highest priority to run. The IKEnegotiation method looks up and creates the security associations between HA and FA. The negotiation sequences refer to the sequence diagrams: Figure 5.7a, Figure 5.7b and Figure 5.7c in Chapter 5.

4.5 FAInfo

The FAInfo class is derived from the AgentInfo class, which inherits all the AgentInfo attributes and methods. The extensions to the AgentInfo class are the IKE attributes and interface attributes.

4.6 FAMNThread

The FAMNThread is used to communicate with MN. The MIP encapsulation and ESP encapsulation (secure mode) will apply to raw data packets from the MN. MIP and ESP encapsulation use the tunnel mode (IP in IP). There are three kinds of packets from the MN. The first one is the MN location information packet to indicate that the MN is in the FA's service range. The second one is the registration packet. The registration is encapsulated with the MIP header in this thread. For the FA secure mode, the registration packet will also activate IKE negotiation and be waiting until the IKE negotiation is finished. After the IKE negotiation, the ESP_IN and ESP_OUT processes will be initiated and all the traffic will have to be subject to the ESP processing. If the FA is in

the insecure mode, the registration packet is forwarded to the HA directly after MIP header encapsulation. The third kind of packet is a normal data packet to the CN. A data packet is encapsulated in a MIP header and subject to an ESP processing in secure mode.

5 HA Package

5.1 HA

The HA class is derived from the Agent class. This class simulates the Home Agent behaviors. It inherits all the attributes and methods of agents except that the disConnect method and init method are overwritten.

The HA responsibilities are automatical home address advertising, acting as the Responder in IKE negotiation with the FA, forwarding messages between the CN and the FA, protecting data against attacks and deregistering the MN. So, the HA maintains five threads: thread listening to the FA, thread listening to the MN, thread listening to the CN, thread for the IKE negotiation, and thread listening to the AN. In the parent Agent class, there is also a thread for address advertisement.

In addition, there are two modes for HA-secure mode and insecure mode.

In the secure mode, the FA IKE negotiation request will trigger the IKE negotiation responder process and establish the ESP security associations in HA. The secure mode and insecure mode should synchronize with FA. The HA has 3 interfaces to monitor the FA, the MN, and the CN packet activities.

5.2 HAANThread

The HAANThread is similar to the FAANThread.

5.3 HACNThread

The HACNThread is used to communicate with CN. If the CN sends a message to the MN, the HA will forward the CN message to the FA or the MN after checking the MN location and registration information.

5.4 HAFAThread

The HAFAThread is used to communicate with the HA. In the secure mode, there are two kinds of packets. The first kind is the IKE control packets to signal the IKE negotiation start, termination and failure. Once IKE negotiation is activated, the HAFAThread will be waiting until it is finished. The other kinds of packets are the normal ESP encapsulated data packet. The ESP_IN and ESP_OUT processes are initiated after the IKE negotiation is finished. All traffic between FA and HA is subject to ESP processing if ESP is active.

5.5 HAIKEThread

The HAIKEThread class is used for IKE negotiation between the FA and the HA, which is very similar with the FAIKEThread except the IKE responder phases are executed.

5.6 HAInfo

The HAInfo class is derived from the AgentInfo, which inherits from its parent class all attributes and methods. This class is used for sharing data between threads in the HA. Once initiated in the HA class, every thread in HA class can get the reference to update HAInfo.

5.7 HAMNThread

The HAMNThread class is used to communicate between the HA and the MN. Since the MN originally stays in the HA, we assume the HA can identify the MN without extra secure operation. The HAMNThread has three responsibilities: get the MN location information, forward packets to the CN when the MN is in the HA, deregister the MN when the MN moves from the FA to the HA after successful registration.

6 MN Package

6.1 MN

The MN class is derived from the Node class. This class simulates the MN behaviors.

The MN behaviors are the following: Moving into the FA, moving into the HA, moving outside of the HA and the FA, getting the agent advertisements, sending packets to the CN, and getting packets from the CN. There are MNHAThread and MNFAThread listening to the HA and the FA respectively.

6.2 MNFAThread

The MNFAThread is used to communicate with the FA. The thread is used to listen to the FA advertisements, registration reply packet, FA control packets, and the data packet from the FA. This thread also sends the registration packet to the FA.

6.3 MNHAThread

The MNHAThread is used to communicate with the HA. The thread is used to listen to the HA advertisement, and HA control packets, send the deregistration packet and receive the deregistration reply packet.

6.4 MNInfo

The MNInfo class is derived from the Info class. The extensions to the Info class are attributes and access methods for the MN location information, registration information and interfaces.

Glossary

AH: Authentication Header

AN: Attack Node

CN: Correspondent Node

COA: Care-of-address

DATASA: Security Association for DATA

ESP: Encapsulated Security Payload

ESP_IN: ESP Inbound Process

ESP_OUT: ESP Outbound Process

FA: Foreign Agent

HA: Home Agent

Initiator: The node that initiates the IKE negotiation session.

Responder: The node that responds to the Initiator's IKE request.

IKE: Internet Key Exchange

IKESA: Security Association for IKE

MN: Mobile Node

SA: Security Association

SPD: Security Policy Database

SPI: Security Policy Index