Baltzer Journals Feb. 24, 1998

Secure and Mobile Networking *

Vipul Gupta and Gabriel Montenegro

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto CA 94303-4900
E-mail: {vgupta,gab}@eng.sun.com

The IETF Mobile IP protocol is a significant step towards enabling nomadic Internet users. It allows a mobile node to maintain and use the same IP address even as it changes its point of attachment to the Internet. Mobility implies higher security risks than static operation. Portable devices may be stolen or their traffic may, at times, pass through links with questionable security characteristics. Most commercial organizations use some combination of source-filtering routers, sophisticated firewalls, and private address spaces to protect their network from unauthorized users. The basic Mobile IP protocol fails in the presence of these mechanisms even for authorized users. This paper describes enhancements that enable Mobile IP operation in such environments, *i.e.* they allow a mobile user, out on a public portion of the Internet, to maintain a secure virtual presence within his firewall-protected office network. This constitutes what we call a Mobile Virtual Private Network (MVPN).

Keywords: Secure Mobile Networking, Mobile IP, IPSec, Mobile Virtual Private Network, Firewall Traversal, Remote access

1 Introduction

The IETF Mobile IP protocol [21] allows a mobile node to continue sending and receiving IP datagrams using a fixed address, its home address, even when it is no longer connected to its home subnet. A mobile node visiting a foreign network chooses a care-of address on that subnet and registers it with its home agent, a special entity residing on its home subnet. The home agent intercepts datagrams meant for the mobile node and tunnels them to the registered care-of address. Tunneling refers to the process of enclosing the original datagram, as data, inside another datagram with a new IP header [22, 23]. This is similar to the post office affixing a new address label over the original when forwarding mail for a recipient who has moved. The destination field in the outer IP header contains the care-of address, which may belong to a specially designated node, a foreign agent, or may be acquired (perhaps temporarily) by the mobile node, *e.g.*, through Dynamic

*This research was supported in part by the Japanese Ministry of Interior Trade and Industry (MITI) through the NECM (Network Environment for Continuous Mobility) project.

Host Configuration Protocol (DHCP) [10] or the Point-to-Point Protocol (PPP) [25]. In the latter case, a mobile node is said to have a co-located care-of address. This mode of operation obviates the need for a foreign agent. The recipient of a tunneled packet recovers the original datagram before processing it further.

Mobile IP assumes that unicast datagrams are routed solely on the basis of their destination address. Many Internet routers include other considerations in their forwarding decision, *e.g.* to guard against IP-spoofing attacks, source-filtering routers drop datagrams that arrive on an interface inconsistent with their source address [6]. Datagrams sent by a mobile node on a foreign network, using its home address as source, will be blocked by source-filtering routers. One possible solution to this problem is to use a reverse tunnel directed from the mobile node to its home agent [18]. Under this arrangement, datagrams sent from a mobile node carry a *topologically correct* care-of address, rather than the home address, as source. The home agent strips off the outer IP header on reverse tunneled packets to recover the original datagram. From then on, this datagram is forwarded as though the mobile node were on its home subnet. Unfortunately, intervening firewalls can prevent datagrams sent by a mobile node from ever reaching the home agent. Firewalls are typically configured to drop unsolicited datagrams from untrusted external hosts [7]. Unless a mobile node can authenticate itself to the firewall, even reverse tunneled packets can get dropped.

Further complicating matters, organizations often hide the topology of their internal network by using private addresses. Such addresses include, but are not restricted to, those defined in RFC 1918 [24]. These addresses are not advertised to the general Internet and the Internet's routing fabric is unable to route packets to these addresses (resulting in ICMP *destination unreachable* messages). To allow connections from the internal network to the general Internet, application relays (*a.k.a.* application gateways or proxies) are used. In a typical configuration, the internal network is separated from the general Internet by a *perimeter network* on which the firewall and proxies are located [7]. Hosts on the perimeter network use public addresses. When a host on the internal network wishes to connect to the Internet, two separate connections are set up: one between the internal host and the proxy, and another between the proxy and the outside host. To the outside host, the user at the other end appears to be on the proxy host.

For example, Figure 1 shows a sample network with a firewall (FW1) around organization O's network; R1, R2, and R3 are routers and M is a mobile node. Correspondent nodes, such as CN1, CN2, and CN3 may be located in the mobile node's home network, in the foreign network being visited by the mobile node, or elsewhere in the Internet. In this example, organization O uses private addresses within its internal network.

The use of private addresses poses an additional challenge for Mobile IP. A mobile node belonging to a private network cannot use its home address to communicate directly with correspondent nodes while outside its protected network. Replies from correspondent nodes cannot be routed to the mobile node's private address.

This paper presents a unified solution to these and other problems arising from the use of Mobile IP in such a security-conscious environment. For example, a mobile user who carries a portable computer outside his company's firewall-protected network may require automatic encryption of all traffic exchanged with corporate computers. Whenever possible, our mechanisms try to leverage existing technologies with minimal changes.

The rest of this paper comprises seven sections. For illustrative purposes we use a specific security framework described in Section 2. Section 3 analyzes design considerations

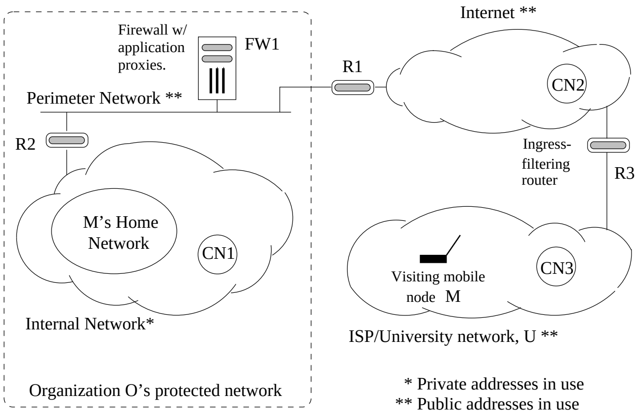
Internet **
Firewall w/ application proxies.
FW1
R1
Perimeter Network **
CN2
R2
Ingress-filtering router
R3
M's Home Network
CN1
CN3
Visiting mobile node M
Internal Network*
ISP/University network, U **
Organization O's protected network
* Private addresses in use
** Public addresses in use

Figure 1: Model Security Architecture

in supporting secure Mobile IP and reviews available security technologies. Section 4 is an overview of SKIP (Simple Key-Management for Internet Protocols) [4] and describes how a Mobile Virtual Private Network (MVPN) can be constructed. Detailed description of such an MVPN's operation is provided in Section 5. Section 6 presents experimental data supporting the practical relevance of our approach. Section 7 addresses related issues, and we conclude with Section 8.

2 Security Framework Model

We use the security framework shown in Figure 1 for our discussions. It depicts an organization, O, connected to the Internet through a screened-subnet firewall architecture [7]. This particular architecture was chosen for its popularity and superior security characteristics, but our ideas apply to other variations as well. The organization's interior network is insulated from the Internet by a perimeter network (or De-Militarized Zone (DMZ)) and two packet filtering routers R1 (exterior or access router) and R2 (interior or choke router).

As an additional security measure, we assume that the internal network topology is hidden using application relays and by advertising to the Internet only the addresses on the perimeter network. It is not uncommon for a large organization to acquire a Class B address for its internal network and a Class C address for the perimeter network. To the rest of the world, these organizations appear as having only the Class C address space. Not only are outside routers (such as R3) unaware of internal addresses, inside routers (such as R2) are, in turn, unaware of outside addresses. Inside routers, however, are aware of addresses on the perimeter network. All routers drop packets with an unknown destination address. Some of them may also drop packets if the IP source is unknown.

Figure 1 also shows another network, U, typical of a university or ISP (Internet Service Provider) environment. These networks impose far fewer constraints on connecting hosts; the only security mechanism in place may be a source-filtering router (*e.g.* R3) to guard against IP address spoofing.

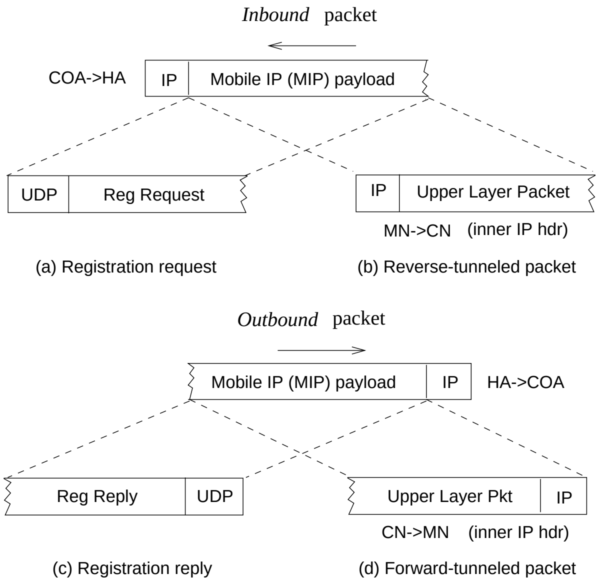

Figure 2: Packets exchanged between a mobile node and its home agent

The mechanisms described in the paper allow a mobile node, M, to move seamlessly[1] not only between different subnets inside organization O, but also between the protected network and an untrusted outside network (*e.g.* U). Irrespective of its physical location, M continues to enjoy the connectivity (except for performance penalties) and privacy it has on its home network. Our solution requires that M and its home agent be able to distinguish addresses in the protected network from those on the outside. We do not address the scenario in which a mobile node connects to the Internet from within a protected network belonging to another organization. Allowing an untrusted "foreign" mobile node to connect to a protected network raises security concerns beyond the scope of this paper. In the interest of simplicity, we ignore the possibility of multiple firewalls between a mobile node and its home network until Section 7.

Note that the internal network shown in Figure 1 may actually be a virtual private network (VPN), *i.e.*, it may comprise multiple, geographically non-adjacent, private networks that function as a single private network. VPNs are built from authenticated and encrypted tunnels between security gateways at the border of each physical network. The mechanisms described in this paper are independent of any such topology within the protected network.

3 Design Considerations in Supporting Mobile IP

We first consider the type of packets that need to pass back and forth through source-filtering routers, firewalls and private address spaces. When reverse tunneling is used with

[1] That is, without disturbing established transport-level connections.

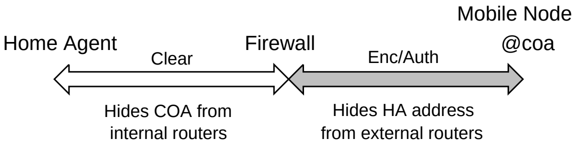

Figure 3: A minimal acceptable secure channel configuration between the mobile node and its protected home domain

Mobile IP, these packets fall into two categories, inbound and outbound (see Figure 2):

1. Datagrams directed from a mobile node to its home agent include registration requests and reverse tunneled traffic. In either case, the (outermost) IP header contains the mobile node's care-of address (COA) as source and its home agent (HA) as destination. We refer to these as *inbound* packets. They are shown going from right to left in Figure 2.

2. Datagrams directed in the other direction include registration replies and forward tunneled traffic. In either case, the (outermost) IP header has the home agent as source and the mobile node's care-of address as destination. We refer to these as *outbound* packets. They are shown going left to right in Figure 2.

Reverse tunneling hides a mobile node's home address (private *and* topologically incorrect) from outside routers like R3. However, there is still the need to hide the care-of address, an external address, from inside routers (*e.g.*, R2); and to hide the home agent address, an internal address, from outside routers. Additionally, a firewall must be able to authenticate the mobile node before accepting packets sent on the mobile node's behalf. Encryption is also necessary if the mobile node's traffic is to be kept private.

3.1 Choosing a Secure Channel Configuration

These considerations suggest the secure channel configuration shown in Figure 3. The communication path between the home agent and the mobile node is split at the intermediate firewall (FW), and additional bi-directional tunneling is employed on the two halves. The tunnel between the home agent and the firewall hides the care-of address from inside routers, and that between the firewall and the mobile node hides the home agent address from outside routers. The second tunnel can also be used to provide encryption and authentication. Such security may be unnecessary between the home agent and the firewall because hosts within the protected network often trust each other.

Two other possible channel configurations are shown in Figures 4 and 5. Both require IPSec capability at the home agent, and in each case, the firewall is unable to examine the IP payload for logging or auditing purposes. The configuration shown in Figure 4 has additional drawbacks: (i) it requires the home agent to be directly addressable from a mobile node on the outside – this is ruled out by the use of private addresses, and (ii) authentication responsibility is shifted to the home agent whose security may be weaker than the firewall's.

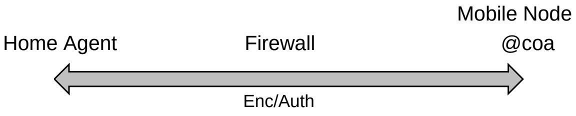

Figure 4: End-to-end encryption and authentication

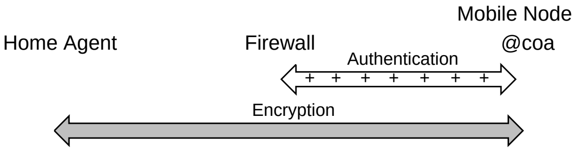

Figure 5: End-to-end encryption, intermediate authentication

3.2 Authentication Technologies

Two technologies that can implement authenticating firewalls are: (i) SOCKS [15], and (ii) IP Security (IPSec) [1].

SOCKS was designed as a general mechanism for firewall traversal. The protocol is conceptually a “shim-layer” between the application layer and the transport layer, and as such does not provide network layer gateway services, like forwarding of ICMP or IP-in-IP [22] packets. IP-in-IP messages from the home agent to the mobile node would have to be encapsulated within UDP or TCP in order to use SOCKS, thereby increasing the header overhead. SOCKS requires that the mobile node – or another node on its behalf – establish a TCP session to exchange traffic with the firewall. This results in a minimum delay of four round-trips (six with GSS-API [16]) before a client is able to pass data through the firewall. This negotiation overhead could be prohibitive if a mobile node changes its point of attachment very often.

The IPSec working group of the IETF has defined an architecture [1], an Authentication Header (AH) [2], and an Encapsulating Security Payload (ESP) Header [3] to provide data authentication, privacy and integrity at the IP level. Network layer security is totally transparent to applications. This technology is rapidly gaining acceptance among firewall vendors. Choosing IPSec for authentication does restrict how a mobile node operates outside its protected domain. The IP Authentication Header protects several fields in an IP header including the identifier. Since a mobile node cannot reliably predict how some of these fields will be filled in by a foreign agent acting as a relay, any authenticator it computes will be invalidated by the foreign agent’s involvement. For this reason, we assume that a mobile node operates with a co-located care-of address outside its protected domain. As it is, a visiting mobile node stands a better chance of acquiring a temporary care-of address (from DHCP or PPP) than of finding an already deployed and – from the point of view of the home agent – trustworthy foreign agent.

In order to make IPSec mechanisms practical, scalable key-management standards are needed. Two popular proposals have been ISAKMP/Oakley [14] and SKIP [4]. The IETF has chosen ISAKMP/Oakley as the mandatory-to-implement key-management protocol for IPSec and support for SKIP is optional. ISAKMP/Oakley, being session oriented, raises performance concerns similar to those outlined for SOCKS. SKIP, on the other hand, offers several features that make it an especially good match for Mobile IP appli-

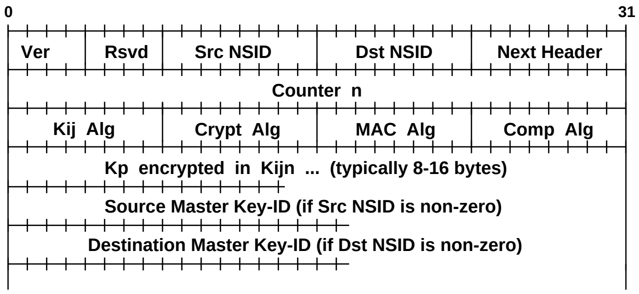
0
31
Ver
Rsvd
Src NSID
Dst NSID
Next Header
Counter n
Kij Alg
Crypt Alg
MAC Alg
Comp Alg
Kp encrypted in Kijn ... (typically 8-16 bytes)
Source Master Key-ID (if Src NSID is non-zero)
Destination Master Key-ID (if Dst NSID is non-zero)

Figure 6: SKIP header

cations. The rest of this paper uses SKIP to illustrate our ideas. These ideas can be adapted for the ISAKMP/Oakley framework and both authors are actively pursuing this goal within the IETF's Mobile IP working group [13].

4 Simple Key-Management for Internet Protocols (SKIP)

SKIP is based on public-key cryptography. It utilizes the fact that two network entities, I and J, can establish a mutually shared secret based simply on knowledge of their own private key and the other's authenticated public key. The well-known Diffie-Hellman algorithm is used to arrive at the shared secret [9]. No prior communication is required between I and J to establish the secret — authenticated public keys can be looked up through any one of several mechanisms, *e.g.* secure DNS [11] or X.509 directory lookup.

This mutually authenticated long-term secret K_{ij} and a counter, n, are used to derive a key, which is denoted K_{ijn}. A one-way hash function is used for this computation, and n is updated whenever the time elapsed or data sent since the last increment exceeds a pre-determined threshold. Typical threshold values are 30 seconds and 512 kbytes, respectively. Individual IP packets are encrypted and/or authenticated using a randomly generated traffic key, K_p. The traffic key is encrypted using K_{ijn} and sent (along with n) in each packet as part of the SKIP header (Figure 6). Since K_{ij} can be cached for efficiency, this scheme allows traffic keys to be modified rapidly, up to once per packet, without incurring the computational overhead of a public key operation. Furthermore, since keys are communicated *in-line*, SKIP avoids the overhead and complexity of a pseudo-session for key negotiation/renewal. The price for this flexibility is the additional SKIP header carried within each datagram.

When a node receives a SKIP-secured packet, it looks up the sender's authenticated public-key. Using that, its own private key, and n, it computes K_{ijn}. From K_{ijn}, the receiver can decrypt K_p and hence the packet. Regular increments to n minimize the exposure of any single key-encrypting-key, making cryptanalysis more difficult. They also prevent reuse of compromised traffic keys. Should a traffic key ever be compromised, it cannot be used later to send forged traffic, since the encryption of K_p under the current K_{ijn} would not be known.

Typically, a receiver uses the source address of an incoming datagram to look up the sender's public-key. However, SKIP allows this lookup to be based on alternate *names*.

Table 1
Sample Access Control List (na = not applicable)

Address	Local	Remote	Security Algorithms	
range	NSID/MKID	NSID/MKID	Kp	Traffic (Enc, Auth)
192.168.4.20	0/na	0/na	DES	RC2,MD5
192.168.4._	0/na	0/na	DES	RC4,MD5
*	0/na	1/10.18.3.17	DES	DES,MD5
default		*deny access*		

This is particularly relevant to a mobile node whose care-of address (IP source of packets sent in reverse tunnels) may change with each move. SKIP's ability to separate a node's true identity from its current address is important for our needs. SKIP supports multiple name spaces which are identified by a name space identifier (NSID). Within a name space, each entity is uniquely identified by a master key identifier (MKID). So, for instance, by setting source NSID to 1 and Key-ID to its home address, a mobile node can effectively tell a receiver "ignore the IP source and use my home address instead to look up my public key". The SKIP header has provisions to carry an NSID-MKID pair for each communication endpoint. When the source (or destination) NSID is zero, the MKID is assumed to be the IP source (or destination). For details on how other NSID values are interpreted, interested readers are referred to [5].

4.1 Access Control Mechanism

SKIP utilizes a flexible access control mechanism. A SKIP firewall is configured with an Access Control List (ACL) that determines which other hosts are allowed to communicate with it and how. There are four kinds of access control entries – host specific, network specific, nomadic, and default. Examples of each entry are shown in Table 1. In essence, each entry describes a SKIP header template for packets allowed to and from the firewall. The first entry in Table 1 indicates that traffic may be exchanged with host 192.168.4.20 provided it is encrypted using RC2 and authenticated using MD5. The key, K_p, used by these algorithms must itself be encrypted using DES. Since the remote NSID is specified to be zero, the IP source of incoming packets is used to lookup the sender's public key. The second entry mandates that RC4 be used to encrypt traffic exchanged with all *other* hosts on the 192.168.4._network. The asterisk in the third entry indicates it is a *nomadic* entry. A nomadic communication partner is not identified by an explicit address. When the firewall receives an authenticated packet with the specified SKIP header characteristics, the wildcard address "*" is dynamically bound to the packet's source. We refer to this as SKIP's dynamic binding feature. A nomadic node identifies itself by including a non-zero NSID (1, in this example) and its unique MKID within that name space. The firewall uses this information to lookup the nomadic node's public key. A zero NSID is inappropriate as the nomadic node may send packets with different IP sources (care-of addresses) depending on its location. Host-specific access control entries are searched before network-specific entries, which are searched before nomadic entries. In the absence of a match, the packet is treated as specified by the default entry. For the configuration shown in Table 1, the default action is to drop the packet.

4.2 Firewall and Mobile Node Configuration

The firewall and the mobile node should be configured in advance with each other's authenticated Diffie-Hellman public values. Strictly speaking, the information could be obtained in real-time, using either of the mechanisms defined by the SKIP protocol: online certificate directory service or certificate discovery protocol. However, preconfiguration eliminates delays associated with real-time discovery. The firewall must also be configured with a nomadic entry for each mobile node authorized to connect from outside the protected domain.

While roaming within the protected network, a mobile node can send datagrams without any SKIP processing. However, upon acquiring an outside care-of address, it must behave as follows:

1. It should prepend a new IP header with the mobile node's care-of address as source and the home agent as destination on all packets sent using its home address. This establishes a reverse tunnel from the mobile node to its home agent.

2. It should enable SKIP processing on all packets destined for the home agent. The security software should be configured to insert appropriate AH, ESP, and SKIP headers before tunneling the resulting packet to the firewall. The source NSID-MKID pair in the SKIP header must match the mobile node's nomadic entry at the firewall. For our discussion, we assume these values are 1 and the mobile node's home address, respectively.

The mobile node must be able to determine when its care-of address does not belong to the protected domain. This could be accomplished by a set of rules defining the address ranges considered internal. User input is another possibility. In actual installations, however, distinguishing between internal and external addresses may present serious difficulties. Because of this, errors in judgment are to be expected. Accordingly, the firewall should be configured to relay packets even if unnecessarily requested by a mobile node with an internal care-of address.

The only configuration required on the home agent is that packets being forwarded to a mobile node at an external care-of address be tunneled to the firewall using an IP-in-IP tunnel. The home agent can determine whether an address is internal or external in several ways. Our testbed uses manual configuration. Another option is to include this information as a new *Traversal Extension* in the Registration Request sent from the mobile node [20].

5 MVPN Packet Formats

This section provides a detailed explanation of how an MVPN works based on our Solaris implementation. We describe packet formats sent over the network and their processing at the mobile node, its home agent, and the intermediate firewall. In Figures 7 and 8, the *vtunl* module [12] offers IP-in-IP encapsulation and decapsulation services. The network driver is shown as *le0*. The SKIP module is responsible for generating and consuming AH, ESP, and SKIP headers.

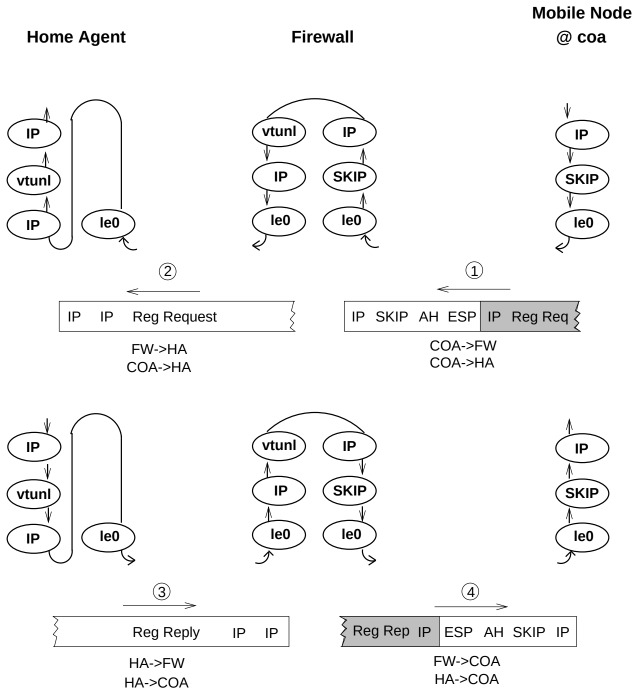

Figure 7: Registration packets exchanged using SKIP

5.1 Mobile IP Registration

Upon arriving at a foreign network and acquiring a care-of address, the mobile node must first – before any data transfer – initiate the Mobile IP registration procedure. This consists of an authenticated exchange by which the mobile node informs its home agent of its current care-of address, and receives an acknowledgment. This first step of the Mobile IP protocol is very convenient, because the SKIP firewall can use it to dynamically bind the mobile node's care-of address to its nomadic access control entry.

With SKIP configured as described in Section 4.2, the mobile node's Registration Request packet is sent to the firewall as Packet 1 in Figure 7. The shaded portion indicates encrypted data. The SKIP header has destination NSID set to 0. The source NSID is 1 and source MKID is the mobile node's home address. The SKIP firewall uses this information to look up a public key for the sender. After the packet is authenticated, the mobile node's nomadic entry is dynamically bound to its current care-of address, COA. In order to hide the care-of address from inside routers (which use the private address space), the firewall forwards the registration request to the home agent through an IP-in-IP tunnel (Packet 2).

The home agent recovers the Registration Request, processes it, and composes a Registration Reply. Since its destination is the care-of address, an outside address, the reply is tunneled to the firewall as shown in Packet 3. The firewall decapsulates the the tunneled packet to retrieve the original Registration Reply. By this time, the care-of address is already bound to the wildcard address in the mobile node's nomadic entry. As

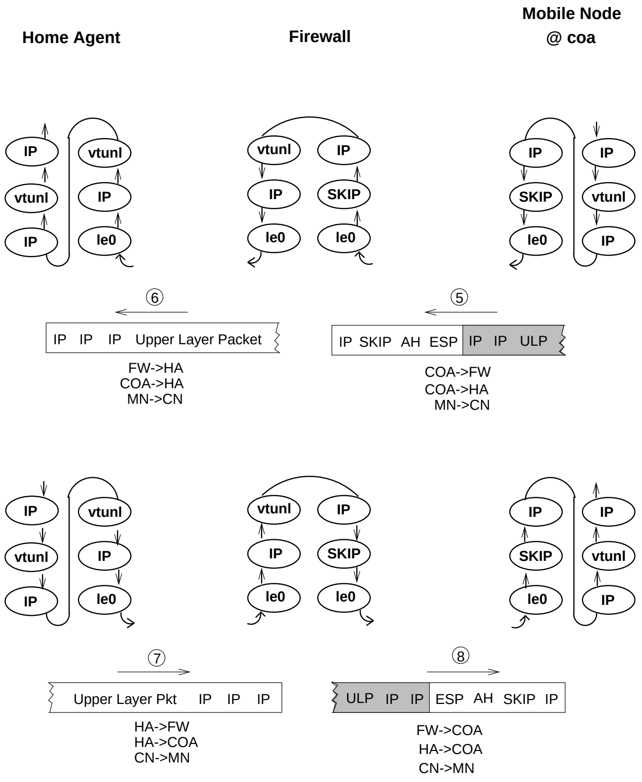

Figure 8: Data packets exchanged using SKIP

a result, the firewall performs AH and ESP operations using the mobile node's public key before forwarding the reply (see Packet 4).

5.2 Data Packets

Once registration is complete, data packets can be sent to and from the mobile node along lines similar to the registration process outlined above. The mobile node reverse-tunnels traffic for the correspondent node to its home agent. In sending each packet, the mobile node's SKIP module, in turn, secures these packets and tunnels each to the firewall as shown in Packet 5 (Figure 8). The firewall consumes the AH, ESP, SKIP, and outermost IP headers to reveal a reverse-tunneled data packet. This packet is forwarded to the home agent as Packet 6. The home agent strips off the two outermost IP headers. From this point on, the datagram follows the same path to the correspondent node that it would have if the mobile node were on its home network. The packet may be processed by a proxy server in the case of external correspondent nodes like CN2 or CN3.

As part of the basic Mobile IP protocol, the home agent intercepts any packet arriving for a mobile node on its home network. The packet is tunneled to the care-of address. Since this destination is outside the protected network, the packet is tunneled again; this time to the firewall (see Packet 7). The firewall removes the outermost IP header and prepares to forward the recovered packet. The mobile node's nomadic access entry causes this packet to be encrypted/authenticated by the firewall on its way out. Packet 8 shows the resulting datagram.

The SKIP module at the mobile node decrypts the packet sent by the firewall. The decrypted packet is decapsulated to recover the original packet sent by the correspondent node.

6 Experimental Results

We have set up a testbed to evaluate the effectiveness of our ideas. It simulates the security environment depicted in Figure 1. Our main interest is to quantify the performance impact of additional processing and increased message sizes due to reverse tunneling and SKIP. We use round-trip latency and end-to-end throughput as our performance criteria.

We chose a fixed correspondent node, CN, within the protected domain and considered the following four configurations. In these experiments, CN is six hops from the mobile node's home network and all communication occurs over 10Mbits/s ethernet links. For the last two cases, SKIP is configured to use DES-CBC for encryption and keyed MD5 for authentication.

Case A The mobile node is on its home network and does not use SKIP or Mobile IP to communicate with the correspondent node, CN. This represents our base case.

Case B The mobile node is outside the protected domain, away from its home network. It uses Mobile IP with reverse tunneling to communicate with CN. However, access control is turned off at the firewall (all packets are allowed) and there is no overhead due to SKIP. Disabling access control isn't really an option in a security-conscious environment but this case helps isolate the overhead due to reverse tunneling.

Case C Same as Case B except access control is enabled and additional overhead is incurred due to SKIP.

Case D The mobile node is on its home network and communicates with another node on the same subnet using SKIP. This establishes an approximate bound on the best performance achievable in the presence of software encryption and authentication.

We measured round trip latency from the mobile node to the correspondent node and back by sending an ICMP echo request (with zero-byte payload) and timing the matching echo reply using *tcpdump* since *ping* does not support sub-millisecond resolution on Solaris. TCP throughput figures were obtained using *ttcp*. Ten ttcp measurements were made and in each measurement, ten thousand 1024-byte TCP packets were sent. Average values are reported in Table 2.

As expected, our base case (Case A) has the best performance and Case C is the worst since it incurs the most overhead. Still, the latency figures are quite tolerable even for interactive applications like telnet.

As for the throughput, we see greater differences. The use of Mobile IP and reverse

Table 2
Preliminary performance results

	Case A	Case B	Case C	Case D
Latency (ms)	3.6	3.8	5.8	2.4
Throughput (KB/s)	795	435	269	288

tunneling (Case B) causes a 45% decline in throughput compared to Case A. The addition of SKIP (Case C) causes an additional drop of 38% over Case B. There are two components responsible for the performance degradation due to SKIP: (i) an increase in packet sizes due to the insertion of ESP, AH, SKIP, and IP headers; and (ii) processing time devoted to encryption, decryption, and authenticator computations. SKIP-related headers increases the size of each packet sent from the mobile node by 116 bytes — ESP header (4 bytes), padding for DES-CBC (8 bytes), AH header (8 bytes), MD5 authenticator (16 bytes), SKIP header (60 bytes), and IP header (20 bytes).[2] For 1 KB ttcp packets, component (i) is responsible for a 10% drop over Case B. The remaining drop can be attributed to component (ii). This is confirmed when Case D is compared against Case A — software-based encryption (using DES-CBC) restricts us to a maximum throughput of under 300 KByte/s.[3]

We can draw some interesting conclusions based on these results. There is a considerable bandwidth gap between the current state of LAN and remote access technologies, *e.g.*, even ISDN, which is a significant step up from 28.8 Kbyte/s modems, only offers 128 Kbytes/s. When a mobile node is connected to its protected domain through an ISDN line or anything slower, software-based encryption is no longer the bottleneck. The additional cost of using Mobile IP, reverse tunneling and SKIP in this case is dominated by the increase in packet size which represents only a 10% to 20% performance loss for moderately sized packets.

These results indicate that our mechanisms are indeed practical and offer a valuable capability without undue overhead.

The use of nested tunnels (due to Mobile IP and SKIP) can delay path MTU (PMTU) discovery [17], causing significant variance in bursty traffic. Several datagrams may need to be sent before the sender's path MTU estimate becomes small enough to account for all intermediate tunnels. Table 3 shows ftp transfer rates for files sent from the correspondent node to the mobile node under Case C. Three file sizes were used and transfer rates were obtained both before and after the sender had deduced the correct PMTU. For the 1 KB file, PMTU discovery isn't an issue because ftp never gets to send a packet size larger than the path MTU. For the larger files, it is clear that the delay associated with PMTU discovery has a substantial impact on overall throughput.

[2] Note that the newer version of ESP can also provide authentication eliminating the usual need for AH.

[3] Note that DES-CBC is not a fast cipher; the use of RC4 nearly doubles the throughput [4].

Table 3
Effect of PMTU discovery on file transfer rate

	File size (in KBytes)		
	1	10	1000
ftp rate in KB/s			
Before PMTU discovery	46.52	1.93	99.8
After PMTU discovery	46.33	92.64	295.3

7 Other Considerations

In the interest of streamlining the prior presentation, we have avoided discussing several related issues. We do so in this section.

7.1 Packet Parsing Firewalls

In Section 5, a SKIP firewall uses its dynamic binding capability to identify a mobile node's access control entry based on its care-of address. Reliance on this capability precludes simultaneous registrations as defined in [21] because each new registration overwrites the previous care-of address associated with the mobile node's nomadic entry.

A firewall that is knowledgeable about the format of Mobile IP datagrams need not maintain dynamic bindings; it can obtain the necessary information from the outgoing packets themselves. The key observation is that both Registration Replies and (forward) tunneled datagrams carry the mobile node's home address and care-of address in well defined fields [21]. Eliminating mobility-related state at the firewalls offers another benefit. In the atypical situation when multiple firewalls are deployed at the periphery, inbound and outbound packets for a given mobile node need not pass through the same firewall.

7.2 Layering Overhead

The combination of various SKIP and Mobile IP related headers artificially increases packet sizes and reduces effective throughput. As indicated in Section 6, extra headers represent an overhead of under 10% for packet sizes close to 1500 (the ethernet MTU). However, for smaller packets (as generated by telnet), the overhead can be considerable and suitable compression schemes should be considered [8]. In some instances, this overhead can be reduced by modifying the behavior of the entities involved. For example, before forwarding Packet 8 in Figure 8, the firewall could harmlessly remove the middle IP header (with the home agent as source and care-of address as destination). Similarly, the middle IP header in Packet 6 could be removed by the firewall. These and other similar changes require close coupling between Mobile IP and IP security software at the firewall [19].

7.3 Securing the Mobile Node

In an MVPN, a mobile node extends the security perimeter surrounding its home domain and must, therefore, share in the responsibility of protecting it from outsiders. This can be accomplished by installing appropriately configured firewall software on the mobile node. In some instances, a mobile node may need to exchange unencrypted packets with another node on a public network, *e.g.* to renew a DHCP lease. Therefore, its firewall software must be able to filter packets based not only on the presence or absence of a valid authenticator but also on protocol and port numbers.

A mobile node's private key should be stored on an external device, such as a smart-card, rather than on its internal disk. Otherwise, anyone who steals the mobile node could automatically gain access to the private network.

7.4 Multiple Firewall Traversal

Consider a mobile node which moves from its firewall-protected home network to a foreign network (*e.g.* another private corporate network) whose periphery is also guarded by a firewall. In such a situation, traffic between a mobile node and its home agent must pass through two firewalls. At a minimum, this requires the mobile node to include multiple authentication headers – one for each intervening firewall. Each authentication header adds one extra level of encapsulation. We recently extended our prototype implementation to handle multiple firewalls as outlined above, and successfully interoperated with another team of researchers for the dual firewall traversal case.

Obviously, given the header overhead, and concomitant delays due to path MTU negotiation, nested tunnels must be used with caution. Nevertheless, we envision that the double firewall traversal case will be quite common.

8 Closing Remarks

Security mechanisms commonly deployed throughout the Internet present serious obstacles to Mobile IP. This paper shows how IP security mechanisms can be integrated with Mobile IP to create a *Mobile Virtual Private Network* (MVPN) and allow nomadic users to roam transparently beyond the confines of their private network.

Acknowledgments

We would like to thank Sunil Madhani for his help in setting up our experimental testbed and collecting the performance results reported in Section 6. We are also grateful to the anonymous referees whose suggestions were very helpful in improving the presentation of this paper.

Ideas in this document have benefited from discussions with the following people: Bill Danielson, Don Hoffman, Tom Markson, Martin Patterson, Tsutomu Shimomura, and Rich Skrenta.

References

[1] S. Kent, R. Atkinson, Security architecture for the Internet Protocol, Internet Draft *draft-ietf-ipsec-arch-sec-02.txt* – work in progress, Nov. 1997 (a previous version appears as *RFC*

1825).

[2] S. Kent, R. Atkinson, IP authentication header, Internet Draft *draft-ietf-ipsec-auth-header-04.txt* – work in progress, Feb. 1998 (a previous version appears as *RFC 1826*).

[3] S. Kent, R. Atkinson, IP encapsulating security payload, Internet Draft *draft-ietf-ipsec-esp-v2-03.txt* – work in progress, Feb. 1998 (a previous version appears as *RFC 1827*).

[4] A. Aziz and M. Patterson, Design and Implementation of SKIP, available on-line at *http://skip.incog.com/inet-95.ps*. A previous version of the paper was presented at INET '95 under the title *Simple Key Management for Internet Protocols (SKIP)*, and appears in the conference proceedings under that title.

[5] A. Aziz, T. Markson, H. Prafullchandra, Assigned numbers for SKIP protocols, available on-line at *http://skip.incog.com/spec/numbers.html*.

[6] CERT Advisory CA-96.21, TCP SYN flooding and IP spoofing attacks, available at *ftp://info.cert.org/pub/cert_advisories/CA-96.21.tcp_syn_flooding*.

[7] D. B. Chapman and E. Zwicky, *Building Internet Firewalls*, O'Reilly & Associates, Inc., 1995.

[8] M. Degermark, B. Nordgren and S. Pink, IP Header Compression, Internet Draft *draft-degermark-ipv6-hc-04.txt* – work in progress, Nov. 1997.

[9] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Trans. Info. Theory*, IT-22, pp. 644–654, 1976.

[10] R. Droms, Dynamic Host Configuration Protocol, *RFC 2131*, Mar. 1997.

[11] D. Eastlake and C. Kaufman, Domain Name System security extensions, *RFC 2065*, Jan. 1997.

[12] V. Gupta, A versatile tunneling interface, distributed as part of the Solaris Mobile IP software package at *http://playground.sun.com/pub/mobile-ip/*, May 1997.

[13] V. Gupta and S. Glass, Firewall traversal for Mobile IP: guidelines for firewalls and Mobile IP entities, Internet Draft *draft-ietf-mobileip-firewall-trav-00.txt* – work in progress, Mar. 1997.

[14] D. Harkins and D. Carrel, The Internet Key Exchange (IKE) Internet Draft *draft-ietf-ipsec-isakmp-oakley-06.txt* – work in progress, Feb. 1998.

[15] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas and L. Jones, SOCKS protocol version 5, *RFC 1928*, Mar. 1996.

[16] P. McMahon, GSS-API authentication method for SOCKS version 5, *RFC 1961*, Jun. 1996.

[17] J. Mogul and S. Deering, Path MTU discovery, *RFC 1191*, Nov. 1990.

[18] G. Montenegro, Reverse tunneling for Mobile IP, Internet Draft *draft-ietf-mobileip-tunnel-reverse-05.txt* – work in progress, Jan. 1998.

[19] G. Montenegro, Tunnel Set-up Protocol (TSP), Internet Draft *draft-montenegro-tsp-00.txt* – work in progress, Aug. 1997.

[20] G. Montenegro and V. Gupta, Firewall support for Mobile IP, Internet Draft *draft-montenegro-firewall-sup-03.txt* – work in progress, Jan. 1998.

[21] C. Perkins, Editor, IP mobility support, *RFC 2002*, Oct. 1996.

[22] C. Perkins, IP encapsulation within IP, *RFC 2003*, Oct. 1996.

[23] C. Perkins, Minimal encapsulation within IP, *RFC 2004*, Oct. 1996.

[24] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, Address allocation for private internets, *RFC 1918*, Feb. 1996.

[25] W. Simpson, The Point-to-Point Protocol (PPP), *RFC 1661*, Jul. 1994.