Secure Mobile Networking
Final Report[1]

James R. Binkley and John McHugh
Portland State University

June 1999

[1]Supported by th U.S. Department of Defense Advanced Research Projects Agency and the U.S. Air Force Rome Laboratory under contract F30602-95-1-0046.

Contents

Chapter 1

Introduction

This document is the final report for the Portland State University Computer Science Secure Mobile Network project. In this paper, we will summarize our accomplishments, discuss high and low points of the project, and suggest further work that might be done to further mobile/wireless security research. We will also present a final overview discussion of Mobile Security Policy.

We would like to point out that a recent IETF draft authored by Jim Binkley while at Oregon Graduate Institute and John Richardson of Intel [2] presents a number of thoughts on the subject of how cross-domain (inter-site) Mobile-IP might be made secure. We regard that document as part of this final report and urge any interested readers to obtain it.

Interested readers should note that source code releases and related papers may be found at the PSU Secure Mobile Network site: http://www.cs.pdx.edu/research/SMN. Although our DARPA funding period is over, we intend to pursue more modest secure wireless research, and will use this site to post any future results in software or technical reports. We would like to thank DARPA for giving us the opportunity to get started in this research arena.

Chapter 2

Accomplishments

In this section, we will briefly present our accomplishments (non-accomplishments will be relegated to the low points section below). We will minimize details and provide only bare bones summary text.

Please note that Appendix A below serves to tie these accomplishments to our quarterly reports. We hope that the appendix may serve as a bibliographic guide to content in previous reports.

Our accomplishments are discussed in the following subsections.

2.1 Creation of a secure enclave model for wireless mobility that includes both inter and intra-domain Mobile-IP.

We constructed an integrated Mobile-IP/IPSEC system in which Mobile Nodes abroad can use 2-way tunnels to securely tunnel all their packets to and from their Home Agent (assumed to be at home in the secure enclave, of course). This solution only addresses one possible facet of a many-sided security problem. We also invented two forms of ad hoc routing (including multi-hop) and tied them to end to end (but network-layer) IPSEC-based routing. Thus hosts that a priori belong to the same security enclave may choose to securely talk to their security peers. Further our Home Agent and Foreign Agents use one-way tunnels authenticated with AH. This allows all agents to reject any arriving (tunnel) packets that do not have an IPSEC binding using agent IP addresses.

Our 2-way tunnels deal with the problem of what to do about "our" mobile nodes, but neglect dealing with non-local mobile nodes. Issues here are complicated and we cannot claim to have the last word. However we have addressed many security issues in this area in our Internet draft [2] and refer interested readers to that document. In brief, we suggest that foreign nodes simply be logically treated as being "outside" the enclave and their packets need only be tunneled across the enclave to a firewall access point.

2.2 Integration of Mobile-IP and IPSEC in terms of routing and security.

A key focus of our work was to tie Mobile-IP and IPSEC directly together. At this point in time, Cisco (for example) has made both Mobile-IP and IPSEC available in their routing devices in IOS version 12.0 [5]. However there is no evidence that the two have been integrated. We believe that the combination of IPSEC and Mobile-IP is far superior to virtual link-layer tunneling schemes a la L2TP [6] or Microsoft PPTP [7] simply because IPSEC has wide-spread architectural utility, generality, and is liable to be more supported than more proprietary schemes. For example, IPSEC can cover both the link layer (by being at the network layer) or the transport layer OR run router to end system, router to router, etc. There is also no point in separate security mechanisms for the latest virtual link tunneling scheme when IPSEC can be used in all places. Mobile-IP needs IPSEC by definition as all packets between a remote Mobile Node (or Mobile Node on a wireless link) to/from home should be made secure.

In our system, we tied IPSEC to routes. For example, when a Mobile Node installs a default route, it is aiming that route at an agent. A route binding that included an indirection mechanism (tunnel IPSEC to the Home Agent) was part of the picture. When we did non Mobile-IP work (say using our multi-hop ad hoc routing protocol) to tie two Mobile Nodes together, we also naturally tied IPSEC to the host routes installed in each Mobile Node. All other IPSEC implementations we have seen so far tie IPSEC to some sort of additional packet-filter like access list mechanism. Our mechanism seems more powerful and in some ways simpler, but to be fair we cannot make a compelling case for its superiority over access list mechanisms (other than one less lookup in the IP layer, but even that is not terribly important given the relatively blinding speed of processors these days). Our IPSEC route binding mechanism can however easily be used to setup manual Virtual Private Networks between two routes simply by installing symmetric keys in a key file, and using the route(8) administrative command to install a route (to a network, subnet, or host). Our mechanism also applies to ARP/link-layer bindings. Our architecture seems to have general utility.

2.3 Simplified Link-Layer Only Ad Hoc Routing

In this protocol, we do not use ARP on a link. We instead use a protocol (like the ISO ES-IS) in which all nodes, agents, and Mobile Nodes send authenticated beacons. This "not" ARP mechanism is intended to serve a number of purposes. First, it tries to mitigate possible ARP spoofing by insisting that the (IP address, MAC address) binding be authenticated. Note that we have implemented this mechanism with both symmetric and asymmetric key systems (in the former case, we have a network-wide key; in the latter, a per host signature). Secondly, the mechanism serves to tie networks together by key possession. It is not important if two laptops do or do not share a subnet. All systems beacon. Therefore if you share a key, you can talk. The low-level IP subnet semantic that requires a router for communication between two hosts from different subnets is obviated. The mechanism also serves a gateway function so that systems who do not possess the secret cannot penetrate into a secure enclave through a "firewall-like" mobility agent. Lastly the mechanism serves a very important purpose in that we assume that if we can hear your beacon, we can talk to you. Beaconing (unlike ARP) is done at a relatively high rate of speed. If a system disappears, we will use other routing mechanisms to try and find it (and not believe an ARP cache entry that is going to hang around for twenty minutes).

The only toothless worn-out criticism that can be made of this system is that if everyone beacons the link itself may be less scalable/useable in terms of throughput.

Given that current wireless links cannot support many simultaneous hosts anyway, it is hard to understand how this criticism can be valid. ES-IS originally was criticized along these terms, but the critics apparently did not notice that beacon rates were extremely slow (once per minute for fixed ethernet systems). (See our criticism of wireless loading in the Suggested Future Work section below for more discussion on this topic). Slow beacon rates for Mobile Nodes along with a combined unicast ACK "reply" to agent beacons make the mechanism more scalable. Agents can beacon and Mobile Nodes can simply append their MAC address under Mobile-IP authentication when they use Mobile-IP to register. This takes care of Mobile Nodes that only desire to talk to the wired infrastructure and do not want ad hoc service. Non-motile Mobile Nodes do not need to beacon very often and can probably slow down their beacon rates (as long as agents do not remove them from the routing state in the agent). Highly Mobile Mobile Nodes need to beacon at higher rates in order to talk to each other.

We suspect the real problem here is wireless loading. If the link itself should be able to tolerate 100 FTP transfers at the same time – one should not worry overly much about 100 nodes sending out 100 byte beacons, even if the beacon rate for all nodes is 1 per second. Of course, this would be too much for WAN wireless systems with small overall bandwidth, but in such cases an IEEE 802.11 [8] style link-layer registration protocol only between agent and Mobile Node can make sense; i.e., one could well neglect the ad hoc function when there are too many nodes in a cell, or one simply doesn't care to talk to anything other the agent (highly likely in many usage scenarios).

2.4 Multi-hop Multicast Ad Hoc Routing (MADRP)

Our multicast ad hoc routing protocol was based on ideas freely borrowed from early suggestions by Dave Johnson [3] of CMU and Scott Corson of the University of Maryland. Our system differs from others in that

1. We did not neglect security between the systems.,
2. We considered redundant communication paths as possibly important., and
3. We integrated our system with Mobile IP so that it allowed systems more than one hop away from an agent on a wireless link to still communicate with the Internet.

The basic idea is that a source Mobile Node would use a multicast discovery packet to do an expanding ring search for another destination ad hoc host across any number of participating mobile hosts acting as routers. A multicast discovery packet would be sent out and "flooded" or forwarded until it reached either the desired Mobile Node or any Agent. Note that the flooding occurs across systems with only one interface, which is unlike how conventional routing protocols work, as in general, they will take a control packet in one interface and flood it out other interfaces; i.e., Mobile Nodes are assumed to have only one wireless interface. Intermediate systems and the final system would setup host routes pointing back towards the sender. A discovery ACK would be returned to the sender by the receiver which could either be multicast or unicast. The return trip packet would also cause routes to be set up (the send/receive discovery packets always cause routes in the opposite direction to be set up). Thus two-way communication would be enabled. Ordinarily the returned ACK might be unicast back across the "best" path. However if redundancy was desired, one could choose to multicast that packet back, thus informing the source of possible multiple paths to the target.

We also improved redundancy by considering agents as possible potential default routes; i.e., agents would always claim that they knew the way (even if they didn't). Thus a Mobile Node might get at least two path answers back:

1. from the true Mobile Node target if available., and
2. from a default agent.

The latter would be used only if the former was not available.

We made Mobile IP work in the multi-hop case by considering the Home Agent one more remote ad hoc node, and we would search for it. This would produce a host route (through a nearby agent) that would allow Mobile IP to work. We submit this is a rather elegant recursive architectural idea. End to end Mobile Node security was implemented by setting IPSEC route bindings up to the source and destination host routes respectively. Thus end path (from source to destination or destination to source) used IPSEC. Note that this mechanism is end to end, not end to router. It does not suffer from the possibility of a proposed plaintext attack. Intermediate systems are presumed to usually NOT impose additional IPSEC bindings. MADRP routing security itself is similar to Mobile-IP authentication. The protocol itself was implemented in the summer of 1997 and designed in 1996.

We submit that the routing metric system we developed was interesting but we never fully tested it. The basic idea was that each participating node would have the same function for metric computation based on a network-wide settable weight function consisting of

$m = f(\mathrm{hopcount}, \mathrm{power} - \mathrm{remaining}, \mathrm{signal} - \mathrm{strength})$.

Thus there were three possible variables in the metric, and weights could be set by the administrator to determine which of the three or combinations therein would be used. For example, one could decide that power at Routes with better power and signal-strength would be chosen. Hop count would be ignored. Or one might setup a link based singly only on power, hop count, or signal-strength. We felt that a combination of signal strength, and power would make the most sense.

We should point out that our protocol name "MADRP" stands for Multicast Ad hoc Demand Routing Protocol. Other possible expansions for the acronym never crossed our mind.

2.5 The Home Agent Redundancy Protocol (HARP)

HARP is documented elsewhere (for example, see [4]). Briefly we developed a protocol that is used between two or more Home Agents to share Mobile registration state (including IPSEC bindings). The agents essentially tunnel in parallel to Mobile Nodes. This mechanism is opaque to Mobile-IP as a whole. Mobile Nodes and Foreign Agents are not affected. HARP is an Interior Gateway Protocol. We assume that a protocol like OSPF is in use, and that a partitioned subnet scheme will be used. As a result packets sent to HARP agents may be evenly distributed by OSPF due to equal multi-path routing. We suggest that Home Agents not be on the same subnet, so that local power or routing failures cannot take out both Home Agents simultaneously.

The downside of HARP is that one must manually administer at least two sets of tables including Mobile-IP registration and IPSEC keys. One might attempt to cleverly "simplify" here by introducing yet another single point of failure. Beware.

We have experimentally developed a version of routed under FreeBSD that can live with our Mobile-IP daemon ("mipd"). We did this so that HARP Mobile-IP subnet availability can be dynamically advertised. However, we never made a production release of this code.

We have demonstrated the extreme usefulness of HARP on several occasions. Last Christmas, we shut down the main Home Agent in order to improve the ethernet wiring in its building area. We had a priori switched a static route in local Ciscos (which is advertised into our OSPF mesh) to the backup agent only. Mobile-IP use was unaffected. No one even noticed that the traditional Home Agent had disappeared and no one noticed when it came back two weeks later.

2.6 The Establishment of Two Wireless Campus Infrastructures

We established a wireless network that has a few users (mostly IT staff, faculty, and a few graduate students) within the School of Engineering at PSU [21]. We intend to maintain this network and extend it where possible. A more limited three-node network was established at OGI last year and is still in use by OGI IT staff and a few researchers.

2.7 Availability of Wavelan drivers for PCCARD and DESKTOP platforms

We have made our Wavelan drivers available to the FreeBSD community. The ISA driver is currently in FreeBSD. The fate of the PCCARD driver has been more complex, but is at least available in the so-called Hosokawa FreeBSD PAO (PCCARD package), although we still make it available on our SMN site as well. It has become obvious over the years that a small but hardy bunch of radio researchers who use point to point directional antennae have adopted Wavelan for long-haul WAN links. We are pleased that our efforts have aided them.

2.8 Use of RSA-based digital signatures to secure the NARP (ad hoc #1) protocol

Given that ad hoc #1 [1] was originally implemented with a symmetric key system (two symmetric keys, one for our side, and one for limited use by visitors), it could be claimed that it was challenged in terms of key scalability. As an experiment, we re-implemented beacons using signatures as authentication and tied our signature schemes to a DNSSEC server (this was lookup by IP address to get a public key). Thus agents could either use a flat file to obtain address/key bindings or could dynamically look up key bindings via DNS. The chicken and egg problem of how a Mobile Node "verifies" a Foreign Agent remains of course. But it is simply a matter of policy. One either brings needed certificates with you, you can choose to "not care", based on the observable notion that communication is proceeding with systems elsewhere, or you ask the Home Agent how it feels about a particular Foreign Agent. Agents, of course, are configured to care, and will not allow Mobile Nodes access unless certificate verification can occur. DNS at least offers relief from yet another single point of failure due to its master/slave setup. Our experiences with DNSSEC (even in a rather limited way) were positive.

Note that ad hoc relationships with pairs of Mobile Nodes are exactly like possible relationships between Foreign Agents and Mobile Nodes; i.e., you may decide a priori that you must somehow have the other sides certificate installed "manually" before communication can proceed. Face to face floppy transfers remain a distinct possibility (of course). Please show two forms of ID.

2.9 Use of RSA-based digital signatures to secure the Mobile-IP protocol

Signatures were also used to replace Mobile-IP's own authentication system. The symmetric authentication system used in Mobile-IP applies to all three possible relationships, 1. MN/HA., 2., MN/FA., and 3. FA/HA. The former is scalable but the latter two are subject to many to many explosions. Thus one of the obvious virtues of a asymmetric key system based on DNS is that one secure enclave with possibly many HAs, FAs (belonging to our side), and MNs (belonging to our side) might centralize its keys with DNS. We did not have the administrative time or manpower to make this so. It seems like a very good idea.

One of the interesting aspects of our implementation was that we decided to solve the MN/FA chicken and egg problem by allowing the MN to ask the Home Agent how it felt about a particular FA. The HA had an access control list for "approved" Foreign Agents and could act as the MN's proxy in terms of any possible DNS lookups and key verification activities where the MN/FA relationship was concerned. In general, it is extremely useful for security (and possibly other systems) to take advantage of the notion that the MN has a proxy partner at home, able and willing to act on its behalf. We believe this is a very important and key concept for trusted mobility.

Chapter 3

High Points

Our accomplishments as listed above are "high points". However we thought we would briefly mention a few positive tangential aspects that we feel are of interest.

3.1 Combined Mobile-IP and IPSEC system truly works

Given the complexity of many of the sub-systems, we are pleased that our system is in use at the PSU campus by small numbers and that the combined IPSEC/Mobile-IP system itself is also in use. Given our UNIX basis, it is not unknown for people to use slogin and scp on top of IPSEC, with the additional ad hoc #1 defence mechanism, thus giving us an in-depth set of security mechanisms, roughly at application (ssh), network (IPSEC), and link-layers (NARP).

3.2 The Bjorn-cam

As we had ported our mobile node daemon to linux/Redhat, we decided that a natural application would be to meld a wireless laptop with a color quickcam, thus producing a mobile camera (mobile within the wireless infrastructure of course) that takes periodic snapshots and puts them on the web. This is known as the "bjorn-cam", as Bjorn Chambless, a graduate student who has been with us for quite some time, developed the system. The bjorn-cam has evolved into a minor sociological phenomenon. Typically one person will take it and put it in his or her office for a week or two, and then send email or call friends and relatives and invite them to observe daily office activities on the web.

3.3 Adoption of wireless systems by IT administrators at OGI and PSU campuses

School of Engineering staff and IT administators at OGI have seemingly become very fond of our mobile system (especially network engineers). From the network engineering point of view, wireless is a natural out of band access system to the network, and represents a secondary path for troubleshooting. An IT staff person may typically be called to service a "broken" computer, that may simply be broken due to network setup problems. It is extremely useful to be able to bring along a working mobile computer that can then be deployed to troubleshoot the fixed wired computer's problem.

Chapter 4

Low Points

All efforts of this scope have their share of frustration and other low points. We are no exception. This section discusses some of the things that did not go as well as we would have liked.

4.1 Death By Integration

Our system involved kernel drivers, kernel security code, modifications to routing table code, complex application routing daemons, symmetric and asymmetric key-based systems, including an experimental version of DNS. The integration work included at least one kernel port (a new release) in the middle of the project. We ended up with roughly twenty systems that needed software replacements or upgrades. Certainly, in general, security code is complex, and requires complex skills especially when combined with fundamental operating system level system's programming. The complexity of our system required ever increasing release testing and integration testing. The worst aspect of the problem was probably the number of moving targets including frequent "upgrades" by FreeBSD itself. We cannot blame them for that, of course, but it seemed like everytime we tried to catch up, before we could make a release, a new FreeBSD release would appear. During the last year, when graduate student code was finally released, we did not have the resources to test the more complex parts and fully integrate them. We did however make a last gasp attempt to capture our kernel modifications in a set of patches. This was released in the fall of 1998 (this school year). Our hope is that the patch release may aid in future ports. Our belief is that complex security systems require long term commitment and serious test and integration phases.

4.2 U.S. Export Laws Deemed Not Helpful

We chose to use NRL's IPSEC work inside FreeBSD. FreeBSD has chosen to not implement our work. One of the reasons is that FreeBSD wants an IPSEC solution that is exportable and as a result is considering the Japanese Kame work which is an IPv6/IPSEC solution for FreeBSD.

Nevertheless, we wish to publicly thank MIT for helping to make our system available.

4.3 Wireless a Moving Target

802.11 as an IEEE project started in the early 1990's (we believe), which was before our project started. Ironically, the 802.11 standard only became available when our project was winding down. In the last few months, IEEE 802.11 wireless hardware has begun to be available. Unfortunately all of our Wavelan hardware was pre-IEEE. Lucent is in the process of making the previous Wavelan hardware unobtainable. The old Wavelan (naturally) does not interoperate with the 802.11 hardware as the fundamental link-layer mechanisms (and encapsulation) are different. When Wavelan (original) was available, Lucent advertised third-party drivers on its web pages. When they released the IEEE code, they expunged all mention of third-party drivers and only provided a limited number of PC OS-centric drivers for the IEEE systems, thus making things difficult for the UNIX-based research community. Only recently has a linux driver (which still has limited features) been made available [9]. We have already purchased a limited amount of IEEE hardware and intend to begin to move our test systems and wireless infrastructure in that direction. (We may simply transition to linux as well from FreeBSD).

The good news is that prices per NIC card have fallen approximately to 1/2 or 1/4 of the prices when the project started. (From $1000 per unit when we started to $250 for some (non-Lucent) IEEE cards).

Chapter 5

Mobile Security Policy Overview

In this section, we will briefly review our overall thoughts on mobile security policy. We will first look at the situation from the point of view of a given secure enclave trying to implement mobility, and then we will review the situation from the point of view of the traditional Mobile-IP architectural elements (Mobile Node, Home Agent, Foreign Agent).

5.1 Secure Enclave approach

By "secure enclave" we mean one set of hosts under a single security administration that is protected by some sort of a priori security mechanisms. For example, the secure enclave may be hooked up to the Internet but is protected by one or more firewall systems which might either be of the packet filter or bastion host variety. We do not rule out "defense in depth" for interior hosts. We do however assume that some hosts are exposed to the Internet and others are not. We are interested in hosts that are exposed to the Internet. We are also interested in hosts with wireless link layers, which for the sake of argument, we will assume are more susceptible to attacks like promiscuous mode sniffing.

In combining Mobile-IP and IPSEC in Mobile Nodes, we first of all made a number of simplifying assumptions. We assumed that a Mobile Node when at home could maintain a two-way IPSEC tunnel connection between it and its Home Agent. We assumed that a Mobile Node when abroad at either a DHCP link or Foreign Agent link could use two-way IPSEC to tunnel home to the Home Agent. The mobile node would dynamically discover these situations and setup tunnels with appropriate security mechanisms. The Home Agent was thus always a bastion host or security gateway to the secure enclave, via any link on it (wireless or wired). Foreign Agents in our first-cut model only serve as link-layer wireless gateways to a secure enclave and may or may not serve external visitors (but must serve internal wireless users, else they are not of interest). IPSEC associations between Mobile Nodes and Foreign Agents did not exist. A Mobile Node abroad might thus be viewed as an extension of the home secure enclave. The two-way tunnel to and from home would serve as an umbilical code to extend the umbrella of the secure enclave to the Mobile Node.

In more detail, let us look at the relationship between a Mobile Node and Home Agent with the Mobile Node at home. If the Mobile Node is a wireless node, our IPSEC system gives it a functional equivalent for link-layer security. (If it was wired, the user (or the local security authorities) might disable this function). What is important is that all packets barring ARP or Mobile-IP itself would be subject to network-layer IPSEC security, which might be more or less good depending on the specific IPSEC implementation and security transforms in use. We replaced ARP with a more secure ad hoc mechanism that simply made traditional ARP spoofing more difficult and made two-way exchanges into the secure enclave (via any agent) difficult as well. Mobile-IP had its own authentication mechanism (and we developed a digital signature replacement scheme for its authentication). There is nothing here to prevent the use of additional security at non-network layers including IPSEC at the transport layer, or transport-equivalent security mechanisms like the Finnish ssh (which we use all the time).

This mechanism is very general. The Mobile Node at home is simply equivalent to any current system using its default router. What is curious is that one still finds weak mechanisms such as the 802.11 WEP

(Wired Equivalent Privacy) in which an algorithm like RC4 may be used for confidentiality between Mobile Node and "bridge" (access point), but is unlikely to be sufficiently strong both because the key length will be restricted due to export reasons, and there is no provision for a key management protocol with the power of protocols in IPSEC. Ironically our ad hoc #1 system seems to be in competition with 802.11 and we believe our overall integrated system is far superior in many ways.

5.2 Us versus Them

When one thinks about a secure enclave, one must consider the enclave from two possible points of view. First one must consider mobile systems that belong to "our side". One must then consider mobile systems that may belong to less trusted visitors. Obviousally different security policies might exist for wireless (or wired) mobile systems that belong to the home team as opposed to possible potential visitors (or "evil" crackers trying to gain access via a wireless link available via a passing motor vehicle). For example, one might choose to allow local wireless nodes access to the secure enclave and might disallow visitor nodes. Or one might allow visitors access to the wireless network, but disallow access to key internal secure enclave areas. Certainly any number of policies might exist. It seems that flexibility in this area could be useful. On the other hand, rich security policy implementations could easily be confusing and hard to get correct.

We did some policy work here, and probably could have done more (see below under foreign agents for some discussion of one possible interesting security extension). Our work can be summarized in two ways:

1. We made it possible to limit access via "mobile" link-layer interfaces into the secure enclave. Visitors might be totally disabled or allowed access on a case by case basis. This was done via the ad hoc #1 authentication mechanism. A mobile node was required to know a shared secret (or present a signed beacon) to a agent. If the agent recognized the beacon, it would install a route to give the mobile node access. If the route was not installed, the mobile node might be able to initiate one-way attacks on the enclave, but it would lack the means for two-way communication. Visitors could be given a temporary key to allow them temporary access into the enclave. We implemented ad hoc #1 with symmetric keys to start with, but experimented with a DNS-based database system for asymmetric keys which would give more key scalability. The critical idea here is that agents act as routers and do not bridge packets naively into the interior infrastructure.

2. We suggested that network design might be employed to deal with the problem of remote visitors (or local wireless systems) by simply rerouting the internal pipes. For example, any packets coming in on an external unsecured link might simply be tunneled to come back in via an outside external firewall interface. Thus one can easily enable visitors (or local untrusted wireless access) by designing them "outside" the firewall. Packets from trusted external hosts might be allowed direct interior access as long as IPSEC is used (and this risk is deemed worth taking). Mechanisms used here might include known tunnel technologies like CISCO's GRE or IPIP, or even IEEE virtual lans at the link-layer. The tunnel endpoint (from agent to firewall) must tie to the same sort of input packet filter checks imposed on ordinary Internet packets coming back into the enclave through firewall systems.

5.3 Foreign Agent Considerations

Security policy for foreign agents in our original system was simplified and very natural. We class foreign agents as either trusted or non-trusted and implemented mechanisms to allow foreign agents to both exclude Mobile Nodes at the link-layer (ad hoc #1) and securely accept tunnel packets from the Home Agent (basically with IPAHIP). Confidentiality was left to the Mobile Node; i.e., the Mobile Node is responsibile for making sure that its own packets are secure to/from the Home Agent as it might be using an untrusted Foreign Agent. The reason for using IPSEC authentication in tunnels is to exclude tunnel spoofing possibilities; i.e., the possibility that an attacker might use barebones IPIP to send packets into an infrastructure at an agent, have them decapsulated, and thus appear to be local with a local IP source address. This is not possible if "our" Foreign Agents only accept IPAHIP packets from Home Agents that they trust and throw away IPIP packets.

We will discuss a more complex security policy system (and implementation) below that we did not implement, but could serve to allow even more complex policies vis-a-vis Mobile Nodes and Foreign Agents.

5.4 Home Agent Considerations

Home Agents serve as a bastion-host for mobile systems. Two-way tunnels terminate (or originate) at the Home Agents. It is assumed that barebones (HA to FA) IPIP tunnels are not used with Mobile-IP. Instead one ties IPSEC into the tunnel mechanism. We have already discussed how Foreign Agents could insist that all tunneled packets must a priori have some sort of IPSEC association between the Foreign and Home Agent. The Home Agent also serves as the tunnel destination for Mobile Node packets coming back to the enclave. It can enforce a similar semantic; i.e., insist that all inbound IPIP packets must have a Mobile Node/Home Agent (or "our-side" Foreign Agent/Home Agent) security relationship. Barebones unsecured IPIP packets would be tossed. Thus the Home Agent can defend the security enclave. One downside here is that Home Agents in this system are not end systems; they are intermediate systems (routers). Thus IPSEC packets may be subject to proposed plaintext attacks, as a "man in the middle" attacker might send packets to the Mobile Node to the Home Agent, and then observe the encrypted packets arriving at the Mobile Node. Defenses against this problem can include session key mechanisms that limit the exposure of keys and/or firewall mechanisms that do not allow Mobile Nodes abroad to talk to systems that are not in the secure enclave.

As a matter of policy, it would be reasonable to assume that Home Agents cannot suffer from a single point of failure scenario. We implemented the Home Agent Redundancy Protocol (HARP) so that Home Agents could act in parallel. We did this in such a way that IPSEC associations were shared and that in general, Mobile Nodes had no knowledge of HARP.

5.5 Mobile Node Considerations

In summary, we suggest that Mobile Nodes at home might use two-way IPSEC to talk to the Home Agent when an unsecure link is in use. (Note that this implies that a Home Agent should have an exterior and interior secure enclave interface). When abroad, they should use two-way IPSEC tunnels to both defend against malign influences on less secure links, and/or possible interception across the Internet. We regard concerns about "triangle routing" as irrelevant to security concerns. In general, security between parties who have no trust relationship is an oxymoron. The real security policy considerations for systems outside the secure enclave are two:

1. Should that system be allowed to talk to home? If the answer, is yes, mechanisms such as two-way IPSEC tunnels could be employed.

2. Should systems that are away be allowed to talk to untrusted systems outside of the secure enclave? If the answer is no, this would obviate such notions as routing redirection targeted to fix "triangle routing".

We also want to point out that our integrated solution includes the possibility of so-called "ad hoc" Mobile Nodes engaged in secure communication. With both our ad hoc systems, Mobile Nodes with a priori trust relationship could setup end to end IPSEC tunnels and thus securely communicate using legacy protocols like telnet and ftp. These tunnels were at the network layer, but unlike the Mobile Node to Home Agent relationship, they were end to end. Thus it is not possible for any attacker to forward packets through a Mobile Node and create a proposed plaintext attack.

Another key idea (pun intended we fear) was the notion in the first ad hoc protocol that an ad hoc network could be based on shared trust. In this case shared trust was blatant as ad hoc #1 uses two shared symmetric keys network-wide. One key was intended for our side and one key was intended to be temporarily created and shared with strangers deemed temporarily trustworthy (and then revoked). However our digital signature experiment assumed that there was one asymmetric key-pair per Mobile Node. We used a digital signature scheme for all beacons (Mobile Node and Agent), and also used digital signature with Mobile-IP authentication itself. This offered a very novel policy mechanism that as far as we know has not been considered before. In our signature scheme, we solved the chicken and egg problem of how Mobile-IP nodes can trust Foreign Agents, by simply asserting that the Foreign Agent's trust statement had to be "mailed" home (sent in the Mobile IP registration) from the Mobile Node to the Home Agent. Thus the Home Agent could use trusted infrastructure to both decide if the FA was trustable and also implement a possible access list control mechanism on non-acceptable Foreign stations. We suggest that the tie between the Mobile Node and Home Agent (as a basic trust duo) is important and can be used to solve many mobility problems. Yes, the Mobile Node is mobile, but it has a fixed surrogate at home that it can interrogate about possibly sticky situations.

Chapter 6

Suggested Further Work and Other Parting Shots

In this section we are going to present a few ideas that we would like to have pursued but lacked the means or time.

6.1 Mobile Nodes Abroad

It is important to note that Mobile-IP may be viewed as either an Interior Gateway Protocol or Exterior Gateway Protocol (or neither), simply based on security policies. Put another way, it is a reasonable security policy to claim that one is not going to allow foreign Mobile visitors using Mobile-IP (or simpler DHCP loaned addresses) to appear "inside" ones secure enclave. In [2] we noted that anti-spoofing measures currently used in the Internet make cross-domain Mobile-IP problematic. We implemeneted DHCP-based IPSEC tunnels to show that Mobile Nodes abroad could securely tunnel home and not have any problems with Mobile-IP source address spoofing (the Mobile Node source address is inside the external IPIP encapsulation and will not be seen until the packet arrives home). However we did not implement any mechanisms that would allow an agent to automatically setup tunnels to tunnel non-trusted Mobile Node packets outside the realm of the secure enclave. This is one mechanism for making smarter security agents that could make cross-domain Mobility more feasible.

With such work, the implementors should pay attention to both verification of the security mechanism and should consider risk assessment for what might happen if such mechanisms are breached. One of the problems with "dynamically poking holes in firewalls" is that one may get unintended holes. One of the problems with the notion of "active networks" is that they may be more active than you anticipated.

6.2 Smarter Foreign Agents

As another possible mechanism for dealing with both trusted and untrusted Mobile Nodes, Foreign Agents could be made smarter. We suggest two possible optimizations: 1., our ad hoc #1 protocol, served as a very primitive (but brutally effective) mechanism for disallowing cross enclave traffic by Mobile Nodes lacking beacon keys simply because the Foreign Agent would not install a local route for unwanted Mobile Nodes. The result (again) is that the Mobile Node could send packets but could not receive them through the Foreign Agent. One could improve this mechanism by tying a packet filter access control list to the registration process (802.11 can do this, but at the MAC level). A Mobile Node could present a certificate to a Foreign Agent, and upon verification, the Foreign Agent would then install an ACL that would permit two-way traffic for the Mobile Node. Note that this mechanism should not stand alone (as it is still spoofable) from IPSEC measures. For example, a Foreign Agent might also then insist that all forwarded packets must first be send directly to the FA itself using a MN/FA IPSEC association. Packets lacking that relationship would be thrown out or unceremoniously tunneled outside the secure enclave. Flexible policy for Foreign Agents may be useful, but appears complex. Of course, the risks inherent in such systems should be considered as well.

6.3 The hard work - integration

To make a long story short, security software is complex and integration although unloved is extremely important. Any key management system seems to have inherent and possibly untoward complexity that informally appears non-linear in nature. This should be recognized and time should be given to security and redundany oriented research projects that takes these problems into account. Given that we tried to combine Mobile-IP, IPSEC (including ISAKMP), digital signatures, redundancy in many forms, ad hoc routing, various versions of various operating systems, DNSSEC, etc., it is no wonder that we have had a nightmare integration problem. Our entire software system could stand thorough review and slow and patient replacement of key sub-systems with better quality code. We respectfully suggest that programs oriented towards security and redundancy need to be carefully nurtured and managed in terms of time and budget.

6.4 Keys as a basis for networking

Lastly, our integrated system seems to contain a rather curious notion. In ad hoc #1, we suggested that IP subnets were not the basis of networks. Instead shared trust (keys ...) should be the basis of networking. Our ad hoc #1 and ad hoc #2 systems could be viewed as small groups of Mobile Nodes that formed a network based on individual atomic key knowledge of members. Routing in truth may present a chicken and egg problem as you cannot speak ordinary data before you setup routing as control. But routing protocol security is usually solved by an a priori assumption that a set of routers share a shared secret. We suggest it is not unreasonable to assume that all packets might have a trust relationship and that networking might be done from the ground up (even ARP ...) on that basis.

6.5 Wireless Loading

In the course of the project we discovered that in general Wavelan did not "load" very well. What we mean is that the number of simultaneous transfers between N laptops and 1 agent is strictly limited and certainly does not scale to the level of ethernet. For example, at the height of our project in terms of group numbers, we had a meeting with seven people present all using a Wavelan equipped laptop. All members attempted to simultaneously do an ftp download of a large file. About only four succeeded in simultaneous download. In fact, several of the ftp transfers totally failed, while others would simply wait. This was an informal experiment but we believe it is truly illustrative of the problem. This has never really been a problem for us due to limited numbers of users, lots of cells (10), and a wide distribution of users on campus (in two different buildings). However we expect that anyone who widely adopts a wireless lan link and then faces > 5 users in the same cell will experience difficulties when simultaneous bulk transfers are attempted. We expect this problem is due to a number of factors including the limited spreading inherent in current spread spectrum techniques (currently limited by FCC regulations), the smaller overall bandwidth (1-2 megabits), and the fact that such a technology is only capable of "listen while send"; i.e., there is no out of band channel mechanism for true collision detection.

Solutions might lie with techniques borrowed from DARPA "Software Radio" techniques; i.e., a very wide spectrum spreading in terms of frequency. More narrowly based devices might use techniques for sampling limited ranges and then switch to another range that does not show so much current use. An agent could easily include the number of current users in its beacon. More research in this area is necessary.

Chapter 7

Acknowledgements

We would like to thank the following PSU students and staff for their participation in the project: David Reeder, Bill Trost, Bjorn Chambless, Radheka Godse, Jennifer Ye, Zhong Chen, Xu Hao, Eric Bergren, and David Burns. We would like to also thank Don Westlight of OGI for his support and interest. We wish also to thank Robert T. Morris Junior of Harvard for supplying us with the original set of Wavelan drivers, Jon Inouye (then of OGI) of Intel for his freebsd/pccard infrastructure support, and Ted Tso of MIT for making our security bits available.

Bibliography

[1] Jim Binkley. Authenticated ad hoc routing at the link layer for mobile systems. Technical Report 96-3, Portland State University, Computer Science, 1996.

[2] Jim Binkley and John Richardson. Security considerations for mobility and firewalls. Internet-Draft, November 1998.

[3] Josh Broch, David Johnson, and David Maltz. "the dynamic source routing protocol for mobile ad hoc networks". MANET IETF draft, draft-ietf-manet-dsr-01.txt, December 1998.

[4] Bjorn Chambless and Jim Binkley. Harp - "home agent redundancy protocol". Internet Draft, October 1997.

[5] http://www.cisco.com. Internet web site. Search on IPSEC and Mobile-IP.

[6] http://www.cisco.com. Internet web site. Search on L2TP.

[7] Counterpane systems announces crack of microsoft's point-to-point tunneling protocol. http://www.counterpane.com/pptp.html. Bruce Schneier's Cryptanalysis paper should be read first. More information can be found at http://www.microsoft.com.

[8] Wireless lan medium access control (mac) and physical layer (phy) specifications. IEEE Std. 802.11-1997, November 1997.

[9] linux ieee wavelan driver. http://www.fast.fh-dortmund.e/users/andy/wvlan.

[10] John McHugh and Jim Binkley. Secure mobile networking, fourth quarterly report. http://www.cs.pdx.edu/research/SMN/3q96.ps, Oct. 1996.

[11] John McHugh and Jim Binkley. Secure mobile networking, second quarterly report. http://www.cs.pdx.edu/research/SMN/1q96.ps, April. 1996.

[12] John McHugh and Jim Binkley. Secure mobile networking, third quarterly report. http://www.cs.pdx.edu/research/SMN/2q96.ps, July. 1996.

[13] John McHugh and Jim Binkley. Secure mobile networking, eighth quarterly report. http://www.cs.pdx.edu/research/SMN/3q97.ps, Oct. 1997.

[14] John McHugh and Jim Binkley. Secure mobile networking, fifth quarterly report. http://www.cs.pdx.edu/research/SMN/4q96.ps, Jan. 1997.

[15] John McHugh and Jim Binkley. Secure mobile networking, seventh quarterly report. http://www.cs.pdx.edu/research/SMN/2q97.ps, July. 1997.

[16] John McHugh and Jim Binkley. Secure mobile networking, sixth quarterly report. http://www.cs.pdx.edu/research/SMN/1q97.ps, April. 1997.

[17] John McHugh and Jim Binkley. Secure mobile networking, 11th quarterly report. http://www.cs.pdx.edu/research/SMN/2q98.ps, July 1998.

[18] John McHugh and Jim Binkley. Secure mobile networking, 12th quarterly report. http://www.cs.pdx.edu/research/SMN/3q98.ps, December 1998.

[19] John McHugh and Jim Binkley. Secure mobile networking, ninth quarterly report. http://www.cs.pdx.edu/research/SMN/4q97.ps, Feb. 1998.

[20] John McHugh and Jim Binkley. Secure mobile networking, tenth quarterly report. http://www.cs.pdx.edu/research/SMN/1q98.ps, May 1998.

[21] Seas wireless network. http://guinness.cs.pdx.edu.

Appendix A

Ties to the quarterly reports

In this appendix, we tie our accomplishments to the quarterly reports which give mode details. For completeness, the the individual reports are included (in slightly modified form) as appendices to this report. The references are to the appropriate pages in this volume. Also given are references to the URLs where the individual reports may be found and the section titles that may be used to locate the reference.

1. Creation of a secure enclave model for wireless mobility that includes both inter and intra-domain Mobile-IP.
   - Mobile-IP Security Analysis, (section C.4, p. 42. See also [11, p. 6])
   - Appendix - IPSEC, Mobile-IP, and Policy, (section G.7 p. 103. See also [16, p. 19])
   - Security Considerations for Mobile and Firewalls, (section M.4 p. 177. See also [18, p. 16])
2. Integration of Mobile-IP and IPSEC in terms of routing and security.
   - Mobile-IP, (section C.3 on p. 40. See also [11, p2].)
   - Mobile-IP Status and Planning, (section D.2 on p. 47. See also [12, p. 2])
   - IPSEC status, (section D.3 on p. 50. See also [12, p. 6])
   - Towards A Secure Mobile-IP, (section D.4 on p. 51. See also [12, p. 7])
   - Mobile-IP, (section E.2 on p. 58. See also [10, p. 2])
   - Mobile IP and IPSEC, (section E.7 on p. 66. See also [10, p. 15])
   - IPSEC && Mobile-IP, (section F.6 on p. 76. See also [14, p. 5])
   - IPSEC/Mobile-IP Security Considerations, (section F.13 on p. 84. See also [14, p. 15])
   - IPSEC and Mobile-IP, (section G.2 on p. 93. See also [16, p. 4])
   - ISAKMP, (section H.6 on p. 113. See also [15, p. 7])
   - Use of DHCP in Mobile IP - Zhong Chen (section K.2 on p. 149. See also [20, p. 2])
3. simplified link-layer only ad hoc routing
   - Authenticated Ad Hoc Routing, (section E.4 on page 59. See also [10, P. 5].)
4. multi-hop ad hoc routing (MADRP).
   - Multi-Hop Ad Hoc Routing, (section G.3.2 on p. 98. See also [16, p. 11])
   - Ad Hoc Routing, (section I.3 on p. 120. See also [13, p. 3])
5. the Home Agent Redundancy Protocol (HARP).

- Home Agent Redundancy, (section F.14.1 on p. 85. See also [14, p. 17])
- Home Agent Redundancy, (section G.5 on p. 101. See also [16, p. 16])
- Home Agent Redundancy Work, (section I.4 on p. 121. See also [13, p. 5])
- HARP Internet Draft, (section J.5 on p. 132. See also [19, p. 5])
- HARP, (section L.2 on p. 163. See also [17, p. 1])

6. the establishment of two wireless campus infrastructures.

   - See http://guinness.cs.pdx.edu.

7. availability of Wavelan drivers for both PCCARD and DESKTOP platforms.

   - See http://www.cs.pdx.edu/research/SMN

8. use of RSA-based digital signatures to secure the Mobile-IP protocol itself.

   - Public-Key Cryptographic Work, (section H.5 on p. 112. See also [15, p. 6])
   - Signed MN Registration, (section I.7 on p. 125. See also [13, p. 10])

9. use of RSA-based digital signatures to secure the NARP (ad hoc #1) protocol.

   - Signed MN Registration, (section I.7 on p. 125. See also [13, p. 10])

Appendix B

Quarterly report – Fall 1995

B.1 Introduction

The Secure Mobile networking project at PSU got off to good start, but almost immediately encountered administrative difficulties with which we have yet to come to grips. After a rush on the part of ARPA to get us under contract by mid September, we were informed that the ARPA reorganization had resulted in our ARPA point of contact being moved from Teresa Lunt to Mike St. Johns. Major St. Johns then informed by email that it would be necessary to reduce the scope of our effort.

These issues have not been completely resolved and in the face of conflicting information, we decided to proceed with limited system development efforts in the absence of directions to the contrary. The remainder of this report discusses our activities in setting up an initial system and out participation in the The International Cryptography Experiment, a requirement added by ARPA during the contracting process.

B.2 The Initial Mobile Network

B.2.1 Current Project Status and Operating Assumptions

As of winter quarter, one of the PIs (Jim Binkley) is working nearly full-time on the project and is only teaching one course this quarter. We are assuming that we will be able to do at least a two year project. Under this assumption, the budget could be cut back by one third; i.e., we are dropping the final year from the original proposal. The first year will be as previously planned; i.e., we will implement mobile-ip and network layer security and integrate them with attention to tunnels. We will use NCR Wavelan as a link-layer medium and will construct at least a radio-based networking infrastructure in the CS/EE building. We have already mapped the building and have determined that 4-5 base stations are enough for coverage. We will also attempt to understand and analyze those protocols in a mobile LAN-based radio environment. We would drop (or restructure as a third year option) the previous second-year effort aimed at distributed access protocols ("mobile firewalls"). We still think research in this area is interesting and it is certainly going to be necessary for widespread secure mobility, but based on the current state of mobile computing in general and secure mobile computing in particular, we think that the work on routing and agents, originally proposed for the third year is more critical. We will move the proposed third year research to the second year. This will focus on routing redundancy; i.e., multiple foreign agents, multiple home agents, and "ad hoc" networking; i.e., direct secure communication between mobile hosts.

B.2.2 Winter Quarter Results and Plans

It was hoped that we could put together a suitable development environment in fall quarter but that did not happen. One question we faced immediately was what operating system to use for a development environment. Two possibilities seemed reasonable. We could use linux, since wavelan drivers were available for it. Or we could use FreeBSD, because we had previous experience based on the BSD stack. Unfortunately drivers existed for operating systems that were close (BSDI) but not quite the same. We chose to not use BSDI because their stack source would be proprietary.

Due to the availability of NCR wavelan drivers for linux, we first decided to install linux and check it out. At this point, we have decided to not use linux, if we can use FreeBSD. Certain aspects of the current linux TCP/IP stack are immature. E.g., the routing/forwarding code in the kernel consists entirely of a linear search that does not even bother to try and find the longest matching prefix in subnet terms. The TCP protocol itself has some questionable areas. We foolishly managed to bring down our own local network for a while by blindly trusting the linux installation of "rip" (the switches are wrong by default as they assume that the box in question is meant to be the "internet gateway"!). Jim Binkley has considerable previous engineering experience with BSD stacks, including BSDI and vxWorks (a real time operating system) plus we are in contact with primary FreeBSD developers who have contributed cdroms to students at PSU. The BSD stack is probably a good choice for other reasons too since there is a widespread basis here for sharing with other efforts elsewhere. As an example, we are hoping to at least examine and possibly reuse work done by Perry Metzer (ipsec) as he implemented prototype ipsec protocols in NetBSD (a close ally of FreeBSD).

Work is underway to port the BSDI wavelan drivers (supplied by Harvard) to FreeBSD. We need drivers for both ISA bus cards (for wired infrastructure agents; i.e., HAs and FAs) and for the pcmcia card that goes in laptops. We have successfully ported the ISA driver and a port for the laptop pcmcia card is underway.

We have ordered and purchased a first round of equipment. The continuing drop in the overall prices of Intel/PC equipment is frankly delightful. We have purchased 3 IBM thinkpad laptops (2 cheaper 701c laptops and one pentium 755 thinkpad). We have unix running on one thinkpad so far and are doing driver development on the machine. We purchased 2 pentium desktop stations for mobile agents. Both machines cost less than $2k apiece. We have purchased 6 wavelan cards initially, 3 pcmcia cards and 3 ISA cards. We decided to not purchase the Wavelan access points at this point, as we felt that getting the ISA cards going would be cheaper in the long term. This gives us the basis for an initial lab.

We have used the link-layer point to point diagnostics supplied by NCR to determine how best to deploy radio in our CS/EE building. Please see the attached report.

We should mention some possible assistance to our work. Jon Inouye is a Ph.D student at the Oregon Graduate Center who has chosen to work on network layer adaption to change. He is specifically interested in researching how a network stack can adapt to changes in adapters; e.g., pcmcia adapters and is interested in collaborating with us on FreeBSD/mobile-ip. Also one master's project student at PSU, Craig Hondo, who is an experienced industry engineer, has chosen to work on IP network security o.s. architecture for his master's project.

The plan for the remainder of winter quarter is to get the link-layer drivers established and then work on the design and implementation of mobile-ip.

B.2.3 Spring Quarter Plan

In the spring quarter, we will finish mobile-ip, design and implement network-layer security, and integrate them. Testing and analysis will probably continue into the summer. We also hope to have an implemented protocol deployed in the building to allow mobile roaming.

B.2.4 Research Notes

Ad Hoc routing protocols We have done a literature survey on the subject of ad hoc routing protocols. We will report on this in a little more detail in the next report and include references there. All we want to say for now is that there is a small body of highly theoretical work on the subject at the moment and that in general, you can divide it into two camps (barring flooding). Camp 1 wants to somehow reuse existing routing protocols (distance-vector and/or link-state) even though the need to minimize broadcast and involvement of every mobile node is not easy to achieve. Camp 2 consists of Dave Johnson at CMU who proposes that something might be learned from previous radio efforts and it is possible that a routing protocol based on demand source routing might minimize involvement of other hosts. A demand source routing system coupled with network authentication for security might be a very interesting research alternative indeed.

Link Layer Security Please see the section at the end of the attached report (Appendix B.4) "Link-layer Connectivity."

B.3 International Cryptographic Experiment participation

During the contract negotiations, ARPA asked us to attend the meetings of the International Cryptography Experiment. This experiment, is best described by WWW entry at http://www.stc.nato.int/public/ice/ice.html from which we quote:

> ICE is an international programme to demonstrate the use of interfaces whereby computer based applications can invoke external cryptographic functions supplied either in hardware or software form. The overall aim is to provide and demonstrate Cryptographic Application Programming Interfaces and Cryptographic Algorithm Portability Interfaces (CAPI) which are available for international use and which are independent of particular applications, algorithms and vendors in order to achieve the following advantages:
>
> - application development without the need to become involved in cryptography
> - common secure applications working with a wide variety of different cryptographic modules
> - common cryptographic modules supporting many different applications
> - the ability to easily integrate cryptographic functions designed for civil, commercial and military needs
> - cryptographic modules can satisfy the regulations within each country without impact on applications

John McHugh attended the second ICE workshop at The Hague in September. The presentations from ranking NSA and NATO officials indicated that an approach like the one advocated by the ICE sponsors is likely to achieve official sanction because of the need by both national and multinational military forces for interoperability. The September workshop addressed the problem largely at the policy level, but a call for volunteers to develop pilot systems was made at the end of the workshop. Since our systems development effort is intending to use embedded cryptography, we have offered to be a participant in these developments.

The next ICE workshop will be held in the DC area in March.

B.4 Report on Wavelan/Link-layer Connectivity in the CS/EE PCAT Building

Jim Binkley

B.4.1 Introduction

I took an IBM laptop configured with a Wavelan PCMCIA radio lan card and measured the connectivity between the laptop and a PC workstation located in my office, 128D. The goals were to:

1. establish that Wavelan would work with the PC hardware in question. There were two types of wavelan cards (PCMCIA for laptop) and ISA for a workstation box. Everything worked fine and this question will not be further discussed.

2. determine how many wavelan "base stations" (workstations equipped with ISA cards) would be needed on the wired infrastructure in order to establish building-wide connectivity.

PCAT was built as a showplace building for Blue Cross many years ago. Our offices are all on one floor in a quadrant with an empty courtyard containing a disfunctional fountain in the center of the building. My office is well situated at one angle of the quadrant, in the NE corner.

B.4.2 Equipment

Software: The laptops were both running PCDOS 6.3. I used the wavelan PCMCIA enabler device which configures the PCMCIA socket and simply ran the ptpdiag.exe utility on both boxes. The ISA card needed no setup or drivers. The diagnostic software implements a link-layer diagnostic between both pieces of hardware used that measures the signal quality (signal to noise ratio) in percent terms 0..100% and also gives a thumbnail rating in terms of "good", "acceptable", and "poor". "Poor" effectively means you are losing packets. With "Acceptable" you are not losing packets. 4 packets are sent and echoed per second.

Hardware:

Laptop: IBM 701c laptop. This is a cheaper 486-based system that is quite compact and has two PCMCIA slots.

Workstation: IBM PC clone pentium 90mhz with a number of PCI and ISA slots.

Wavelan cards: Wavelan is a 2mbit LAN technology; i.e., broadcast in the bus-based or ethernet sense is supported. Both cards use different Intel-based lan controllers, and function in some sense as close-to or equivalent Ethernet systems; i.e., they have IEEE MAC addresses. In general the card architecture consists of Intel lan controller and a NCR "radio-modem" components. Both are addressable from the ISA/PCMCIA buses in typical Intel fashion (i.e., ports/IRQs, etc.) Connectivity is said to be 800 feet in a radius from a base station; sans walls, metal cabinets, etc. It is limited by power restraints due both to physical reality and the FCC.

- 1 ISA card in the workstation equipped with an omni-directional antennae.
- 1 PCMCIA card in the laptop with a roughly equivalent antennae (the lan controller is in the PCMCIA card. The "antennae" actually includes the radio modem hardware.)

B.4.3 Raw Data

The raw data given in Table B.1 shows signal strengths based on a single entry point located near the NE corner of the PCAT building. The actual locations can be seen in Figure B.1.

B.4.4 Interpretation

Certainly we can get good to acceptable coverage with little or no packet loss in the CS/EE building. We could cover the building in an adequate fashion including lecture and conference rooms by putting one "base station" in each corner. 4 or 5 base stations would do it. We might be able to cover the building by putting one center base station outside in the fountain area, but we would probably certainly lose PCAT 28. Possibly two base stations might work here. (However the building seems to have extra walls in its corners and that might be one reason for putting base stations in the corners).

Later on, we should repeat this experiment with two laptops both equipped with PCMCIA cards. We can station one laptop in the center of the building and walk around the edges (both in the inner hall and out on the street).

As an additional equipment test, I also configured both systems with a PC TCP/IP stack and using the supplied ODI (Novell) drivers ran ping between the stations. This worked with no problems.

B.4.5 Observations

One interesting point is that there is an appreciable difference between Len Shapiro's office and the 113A area in the NW corner. Len's office is almost line of sight down the hall back to the base station in my office in the NE corner. 113A has to go through a number of walls, the cs office, etc to get there. John McHugh points out that the Wavelan radio frequency at 900 mhz is on the order of having a wavelength around the size of a metal cabinet. Walls and metal cabinets do indeed seem to cause signal loss and limit the range.

I have tested the wavelan setup at home with the same equipment. I have a 2 story wood house with about 3000 sq. feet. There was signal loss but the results were always rated "good" from one end to the other even with the base station at the far end downstairs.

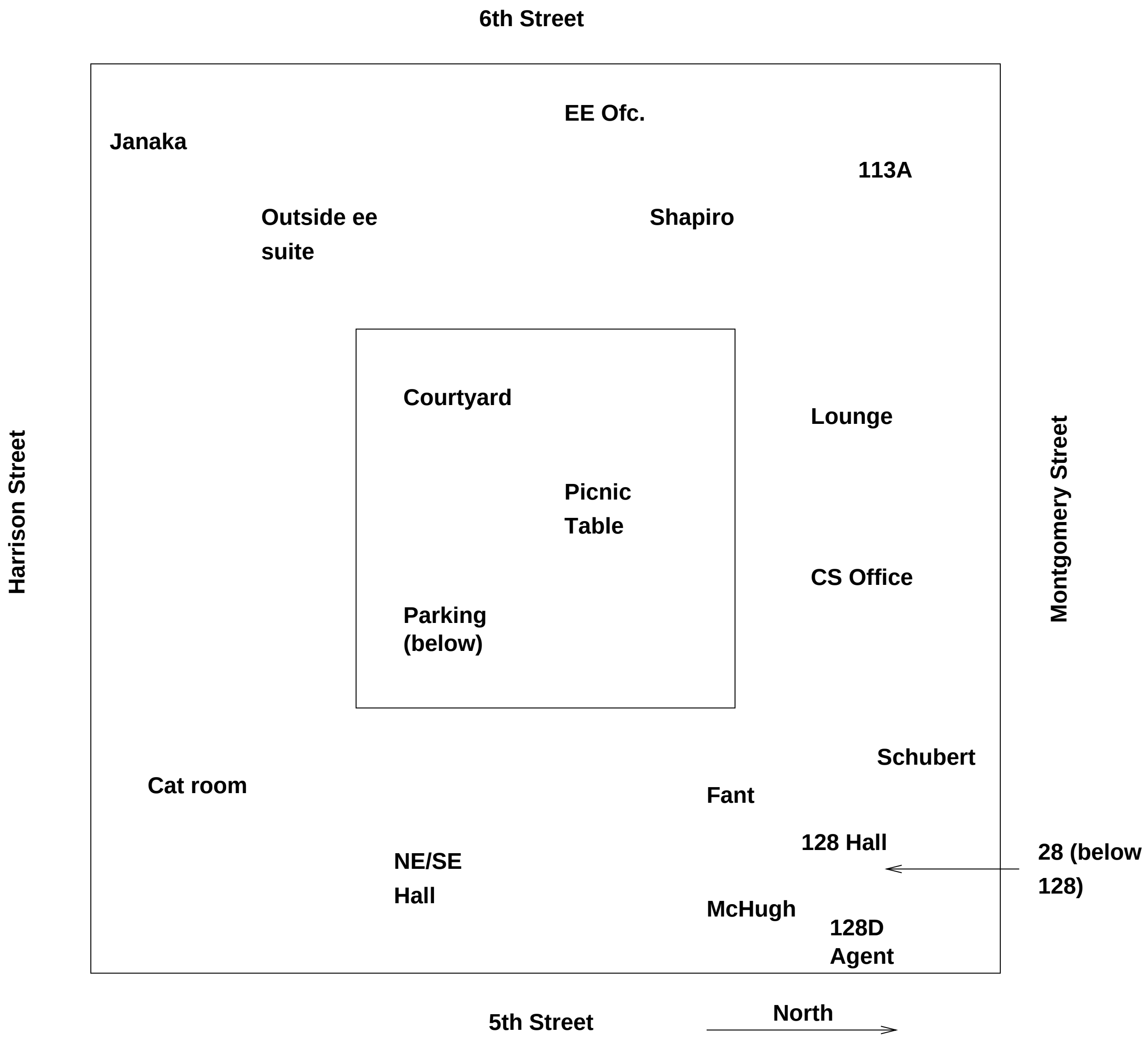

Figure B.1: Floor plan of PCAT

	Location	**Signal %**	**Signal Grade**
1.	Jim's office (128D) comment: Still some dropoff here probably due to metal surfaces in office. My office is in the NE corner of the building.	90	good
2.	outside in 128 hall	70	good
3.	Schubert's office	52	good
4.	PCAT 28 comment: Downstairs lecture room. No loss on acceptable. signal quality varied acc. to how far from my office you were. Not as good in the front. of the room near the blackboard.	40-60	acceptable-good
5.	downstairs parking lot	< 20	poor
6.	John McHugh's office	70	good
7.	Karla Fant's office	50	good
8.	outside McHugh's office	65	good
9.	c.s. office (mid North quadrant)	39	acceptable
10.	student lounge	32	acceptable
11.	113A area, across from Joe Maybee's office	≤ 20	acceptable/poor
12.	Len Shapiro's office (end of hall)	40	acceptable
13.	EE office	14	poor
14.	outside EE office suite (where Janaka's office is) diagonally opposite from my office in quadrant (SW corner)	23	acceptable
14.	outside Janaka's office in far SW corner	10	no connection
15.	Picnic table in fountain square at center of building	37	acceptable
16.	cat room	varies 15-34	poor to acceptable
17.	halfway in hall on NE/SE side	34	acceptable

Table B.1: Wavelan Signal Strength Measurements

B.4.6 Link-layer Routing and Security Observations

There are two possible physical networking configurations that might be employed for constructing a Wavelan-based LAN (not a WAN). The constant here is that you would use PCMCIA cards in laptops for mobility. What varies is how you might choose to setup connectivity to your wired infrastructure. You could choose to 1: employ ISA cards in PCs that would function as either bridges or routers, or 2: you could buy the Access Points from ATT that are effectively a packaged version of #1. The Access Point (AP) is effectively a 386-based computer with an ethernet card and a wavelan ISA card in it. In point of fact, you can buy the ISA card ($600) and construct your own AP or you can buy it. The AP itself is around $2k list. The interesting difference here though is that the AP comes with so-called "roaming" software.

The vendors of such products as Wavelan are limited to what they can do to enable mobility and can only act at the link or hw layers (the domain of device drivers). They are limited to "same network" in terms of the network layer; e.g., IP routing will fail if you moved from one base station to another and the base station was on a different IP network segment (subnet change). Mobile LAN vendors by definition have to have a bridging solution. It is also desireable that the number of packets on the radio link be minimized. One partial solution is called "roaming" which effectively means that a link-layer registration protocol is used between PCMCIA-based clients and base stations (APs). This is a software solution. APs send "beacons" which are periodic special link-layer packets that have a beacon header on top of the normal link-layer header. The mobile node (client) uses a link-layer sign-on protocol in response to any received beacon that has the higher signal level. As a result, packet duplication is eliminated. APs that receive packets destined for a given mobile node can discard them if the node is not registered with them. This may be a useful optimization, although it does not address network layer mobility. There may also be room for criticism on a couple of points in terms of secure networking.

Link-layer beaconing may have a flaw in that an outside attacker might pull up in a van and run an Access Point that could have a better signal strength and thus could suck up all packets. This is in some sense a general problem with any routing protocol with a central system that says "send the packets to me". For example, this is also true for the Mobile-IP Home Agent since tunnels to and from it might be manipulated by attackers or the HA itself might be spoofed. It may be better to simply beacon at the network layer and thus take advantage of IPSEC mechanisms like authentication. In general, routing control (or all packets) would be much more secure with network-layer authentication mechanisms. For example, ARP could stand to be authenticated as well, although possibly the mechanism here should be kicked upstairs and become an ICMP registration protocol for ad hoc systems.

From a redundancy point of view, it may be that signing-on with one system and only using it for access to the wired infrastructure is not a good thing either. If the link-layer is under stress (packet loss and packet corruption), which is certainly a likely attribute of radio by definition of, multiple paths may be useful.

Appendix C

Quarterly report – Winter 1996

C.1 Project Status Overview

We have gotten our project up and going this quarter and feel that we are making progress. We are about on track with what we predicted last quarter. We have built a small lab, and put together a development environment. A few laptops equipped with Wavelan radio pcmcia devices are being given out to project team members. We are in the process of building and deploying the mobile routers that are needed to implement Mobile-IP in our CS/EE building. We have made an informal security analysis and design of Mobile-IP. The security analysis is presented in more detail below. The implementation of Mobile-IP is underway. We include a short discussion of Mobile-IP issues below. We think we have some interesting results in terms of routing design in that area. We are at least engaged with IPSEC issues. Our analysis team is looking at both the cryptographic issues involved with the authentication algorithms (how MD5 is to be used) and is planning on reviewing the current set of session key management papers. We are also investigating the architectural issues where IPSEC is concerned. We hope to be able to "reuse" an existing UNIX based IPSEC implementation and integrate it with our version of Mobile-IP. We expect to make that determination in spring quarter. We believe that we should have a functioning Mobile-IP by the end of next quarter and that IPSEC should be started and integrated with Mobile-IP (and hopefully finished by the end of summer, although it may drag on into fall). For next quarter, we are starting up our mobile security reading group again and plan to read a number of papers on both mobility and related network security issues. We hope to encourage student involvement from both our department, Electrical Engineering, and also from the Oregon Graduate Institute, with whom we have ties.

We have made contact with our counterparts at BBN during the March PIs meeting and hope to exchange visits with them during the early summer.

C.2 Hardware/Development Environment

We have purchased a number of laptop computers, Wavelan ISA and PCMCIA cards, and desktop pentium systems to use as mobile routers. We had already ascertained that around 4 mobile routers would cover our CS/EE building. We have gotten 2 mobile routers deployed at this point, and are working on getting the other 2 out. We have ported Wavelan drivers from BSDI for both ISA and pcmcia cards to FreeBSD UNIX, which we have settled on as a development system. The laptops in question are IBM thinkpads, 701c, and 755cx. Buying laptops and running UNIX on them is problematic. There is a danger that hardware might not be compatible, especially the pcmcia controller and display graphics chipset. The Wavelan pcmcia driver assumes an Intel pcmcia controller. Where graphics cards are concerned, probably the hardest thing with putting UNIX on a laptop is making sure that X works, since after all, given the popularity of the World Wide Web, we need to be able to run netscape. (By the way, we have successfully run the Mbone nv app over Wavelan). IBM has complicated things by discontinuing the 755cx (we managed to purchase one) and apparently the 701c is headed down that path too. However the FreeBSD, 2.1 release seems to work well on the laptops so far. We have managed to save money by avoiding the purchase of ATT's Wavelan access points ($2k list). Instead we ported the ISA Wavelan driver and are simply integrating it into the mobile routers (Pentium-90s) as the Mobile Agents in question would have to exist on subnets in our building and elsewhere anyway. The ISA cards list at $600.

Our entire building will be covered with wireless access by the end of April. This will give us a Mobile-IP test bed as most if not all of the mobile routers just happen to be on different IP subnets. Our current building is essentially a square with a central courtyard and halls with offices around the edge. It covers one (small) city block. Offices exist with doors, walls, and a fair number of metal cabinets; i.e., there is not much line of sight access. We find that by putting a router in each of the corners, we should be able to ensure reachability for the entire building. Penetration in one corner through walls is not a problem. Wavelan actually is mostly reachable down any given hall from one corner to another. However, there are a few spots where end to end reachability doesn't exist (too many walls).

C.3 Mobile-IP

Having followed Mobile-IP since almost the inception (1992-1993), we would like to point out that it is a classic case of "design by committee". Mobile-IP has many, many feature points (too many), and we feel we would not be able to get anywhere by trying to implement them all. As a result, we are going to produce a Mobile-IP that is more limited in terms of features, but includes all security features, plus features that we deem essential in terms of security (two-way tunnels). For example, Mobile-IP allows authentication between all FA/HA/MN pairs and we intend to implement and use that capability. There are two forms of replay protection permitted and we intend to implement both forms and experiment with them.

Here are a few examples of the overdone Mobile-IP semantic set and what we plan to do about it:

1. Mobile-IP somewhat unclearly assumes that you will use ICMP router advertisements and solicitations (RFC 1256). We have already implemented a simple form of Mobile-IP router advertisments and do not plan on using solications at this point. Agents (FAs/HAs) will solicit and we regard the ICMP router advertisements as network-layer beacons, analogous to the HELLO messages used in the ISO/OSI ES-IS protocol. Our infrastructure will rely on Foreign Agents anyway as the odds of our having enough administrative control to get other people to deploy DHCP (or bootp) are slim to none. Further, beacons from agents can provide link-layer signal strength status. This information can be used to improve how and when Mobile Nodes switch between Agents.

2. Mobile-IP permits 3 different kinds of encapsulation, the basic required IPIP form, Cisco's GRE, and so-called minimal encapsulation. We plan on implementing only the basic unicast IPIP form to start with. The Mobile-IP draft states that the basic form of IPIP should have a so-called "soft state" mechanism that basically allows a HA to keep track of tunnel state and do something more intelligent with certain ICMP messages returned regarding encapsulated packets. Further, the tunnel mechanism should use the PATH MTU mechanism. We are in sympathy with these ideas but we do not intend to implement soft state at this time. We only want bare unicast tunnels between Mobile Nodes and Agents as appropriate. The required tunnel mechanism is complex, given PATH MTU, soft state, etc. Possibly it is too complex. There does not seem to be any implementation experience here. The draft doesn't note that unicast tunnels are also the exact same transport protocol as used with the current Mbone IPIP tunnels. These tunnel mechanisms are going to have to coexist in an operating system that supports mrouted. We are implementing a simple form of IPIP tunnel that will be accessable as a route to a virtual device in the routing table. Our bare bones IPIP should at least interoperate with other implementations.

3. Mobile-IP systems may choose to use one of two forms of protection against replay attacks. A timestamp mechanism based on NTP timestamps may be used or a nonce mechanism may be used. For timestamps, the bottom line here is that NTP must be used and presumably authenticated possibly from source to mobile node.

4. Mobile-IP optionally permits forwarding of broadcast and multicasts (via a bizarre unicast encapsulation scheme) to a remote Mobile Node. We do not intend to implement these mechanisms.

The above are just examples. Consider that Mobile-IP as it stands MAY contain semantics involving NTP, ICMP router solicitation, PATH MTU, DHCP, multicast routing, and it comes in two basic modes (FA-based and MN sans FA). Mobile-IP routing daemons may have to live alongside of mrouted, gated, routed, xntpd, and rdisc (routing discovery). Although the protocol itself is simple, the resulting architectural and test considerations are not.

One problem with IETF output is that in general, a working group might produce a protocol, but there is no guarantee that resulting application/operating system implementations and architecture will make sense. We feel that both IPSEC and Mobile-IP seem to suffer from some problems in that regard.

In the case of IPSEC, it is not clear at what layer security associations should be made in the operating system. If we use FreeBSD as our reference, we might tie IPSEC security associations (key binds to X) to sockets (application/transport), protocol control blocks (transport), or to routes (network layer). The choice ultimately will profoundly affect the ease of use of a system. It is also important for system security of course and in fact there may be no right choice. For example, both sockets and routes may need security associations.

With Mobile-IP, one problem is that the current IP subnet model has to be violated. For example, Foreign Agents must be able to talk to any Mobile Node from any subnet and not pay attention to the MN's subnet home ip address. This is contrary to the way current kernels work as they determine the subnets they are on from looking at the IP addresses associated with their interfaces. The BSD stack has another problem which is that in general it is not possible for an application to determine the input interface for an incoming packet. It is difficult (but not impossible) for applications to direct packets out a given interface. In general, the latter mechanism exists in a crude form and is called a "route to interface". Typically a routing daemon (that only talks to the local links anyway) may send a UDP packet to the directed broadcast address (e.g., via subnet.255) and the subnet address can be used to route the packet out the desired interface. Routing daemons that receive packets determine the interface by matching the incoming broadcast subnet with their own set of interfaces.

Without going into great detail, we have broken the design of Mobile-IP down into state machines based on the FA, HA, and/or MN role. There is a state machine model for each. Inputs are timer timeouts or ICMP or UDP packets. Outputs are actions like inserting or deleting routes, dropping a node from a visitor list, and/or sending control packets. The outcome of this decomposition is that we feel that we have successfully separated the functionality between application/routing daemons that will implement policy and a few bare-bones kernel mechanisms. It is obviousally desireable that as much functionality as possible should be put in the application layer and as little mechanism as possible in the kernel. Our design is complete and our implementation is under way. We are finishing the kernel implementation and beginning to focus on the routing daemons (applications).

One interesting aspect of our design is the routing table mechanisms we have added to the FreeBSD kernel. These mechanisms are necessary to make Mobile-IP work and also to make it secure. To implement Mobile-IP, we have added 3 types of routes to the kernel. This mechanism is important because Agents (FAs/HAs) need to maintain "visitor lists"; i.e., what MNs are registered at any given time. We want these visitor lists to be maintained by the application layer daemons. However the daemons have to be able to insert or remove routes in correspondence with the MNs in the visitor lists. As a result, new routing semantics are probably the most important kernel mechanism needed to enable Mobile-IP. Where FreeBSD is concerned, the routing table is the central focus at the network layer and it is a very efficient and useful mechanism. Even ARP is integrated with the routing table.

Our new routes take the following forms:

1. for unicast tunnels, we have implemented a virtual tunnel driver. A tunnel route can be added to the kernel as follows:

   ```
   # route add -tunnel <to Mobile Node> <via Care Of Address>
   ```

   The routing mechanism will direct an outgoing packet to the MN through the tunnel driver. The COA is used as the destination IP address in the outer IPIP packet. This mechanism is two-way and can be used by MNs to tunnel packets back to the HA or to the home network. We may want to implement a socket option mechanism that will cause a MN to switch ALL routes (except for the default or local link routes) to the tunnel back to the HA.

2. the MN itself needs to be able to setup a default route to the Foreign Agent. The problem here is that the FA's subnet address may not be the same as the MN's home subnet. We have introduced a default route mechanism that allows a default route to be inserted according to a predefined mobile interface (an ip-layer socket option). The route is of the form:

   ```
   # route add -foreign <to default> <Foreign Agent IP>
   ```

   The foreign switch coupled with the default destination tells the kernel that it had better use the mobile interface and avoid looking up the gateway in terms of its own set of subnet addresses.

3. the FA needs to be able to insert host routes to MNs that are present. It will do this (and hence enable routing for the MNs) once a successful HA UDP request is returned to it subsequent to an initial MN request. The result is that routing will not occur until the MN is cleanly registered with the FA. The route here takes sly advantage of the kernel's mechanism for determining an interface binding as we pass in the interface IP address that we desire to use as the Gateway portion of the route. This will allow the FA to be multi-homed, which would be useful for a FA system that wants to allow mobility on say ethernet or other wired links. The route is of the form:

   ```
   # route add -foreign <to MN> <local interface ip address>
   ```

   The kernel mechanism here basically only adds an ARP stub route (sans MAC address) and as a result we actually have optimized the number of routing table entries. More important, there is a very interesting and potentially useful result associated with this form of "foreign" route (foreign implies one or more IP addresses in the route that are not local to the system). This form of foreign route enables an initial and simple form of Ad Hoc routing between MNs. MNs can insert this form of route and talk directly sans agents even when their subnets do not match. It is important to note that they will not be able to talk to each other unless an explicit policy decision is made by a human or routing daemon to permit them to talk; i.e., local host routes must be inserted. Once the routes are inserted, ARP will work. Without them, ARP requests are ignored. We plan on constructing a simple UDP based form of host beaconing that when coupled with authentication will at least allow a primitive and simple form of Ad Hoc routing.

The upshot of the above routes is that they allow clean separation of policy and mechanism between the mobile routing daemons and the kernel.

C.4 Mobile-IP Security Analysis

John McHugh and Jim Binkley made an analysis of what we feel are potential security problems with Mobile-IP. (We are not entirely ignoring the issue of MAC authentication algorithms, but are deferring the analysis of those issues to both our cryptographer, Sarah Mocas, and to cryptographers working with the IPSEC working group). We present a discussion below of potential problems and solutions, where solutions exist.

Some potential issues/and solutions:

- Mobile-IP naively relies on ip address authentication; i.e., when a packet is coming into a Home Agent from somewhere remote; it will appear that Mobile Node ip addresses that should be originating inside the domain, are coming from outside the domain. Put bluntly, Mobile IP cannot be distinguished from a remote spoofing attack. Of course, this has long been recognized as a security issue [Bellovin 89]. The solution is that all packets between home networks and a MN should be at a minimum be authenticated using IPSEC AH (with static keys to start with). Better security may be obtained by combining authentication with encryption and tunnels for privacy (more below). An issue has been raised in the IETF as to whether or not it makes sense for Mobile-IP packets to authenticate their Mobile Node to HA registration packets. The point was that IPSEC should be used. We agree in part, but we do not know that it hurts to also authenticate the UDP registration packet. In effect, it is a request to the HA to install a route and extra authentication there may add a small amount of utility and security redundancy as long as different keys are used. Certainly it makes logging easier.

- There is lack of encryption and or authentication for normal (non-control) packets. We presume that IPSEC AH combined with encryption (ESP) is a first cut here. One point to make is that encryption without authentication should not be used. This of course does not preclude the use of application layer security mechanisms.

- The Mobile-IP implementation on the HA/FA should be careful about denial of service attacks based on spurious registration attempts. For example, Mobile Agents (HA/FA) are supposed to maintain a visitor list (in effect a routing table) for registered MNs. In order to register with the HA, a MN must first have its UDP registration packet authenticated. If a bogus registration packet for the MN appeared with an invalid authentication, and the HA dropped the MN as a result, routing service to the MN could be disrupted. The implementation should ignore invalid authentication requests (except for logging) and allow currently valid registrations to timeout.

- There is the issue of MN location secrecy. How do we minimize information about our current location to the outside world? We combine the discussion here with the next item.

- There is a question about what circumstances make use of a FA a good idea, since the FA by definition has easy access to packets sent from a MN. In general, passive or active attacks could come from an arbitrary point between an MN and its HA, but they are most likely either on the MN's current link or via the FA. The FA is a good candidate for passive snooping attacks. For maximum security, the HA-FA and FA-MN relationships should be authenticated as Mobile IP permits. Of course, this means that the FA in question is effectively under the control of a given authority. There needs to be terminology to distinguish between Foreign Agents that are controlled by a given organization seeking security and FAs that are truly foreign to that organization. We will call the latter "untrusted FAs". The more interesting question is what if anything can be done with untrusted FAs? We believe that the principle here should be: "do everything possible to minimize knowledge about network traffic between the MN and HA". All packets should be authenticated and encrypted, and an MN to HA tunnel should be used (we will consider the HA function to be colocated with a firewall) for any "home" traffic and/or all traffic. The tunnel will minimize the exposure to a FA, since we presume that the only IP source and destination addresses exposed (not encrypted) will be those of the FA and HA. Packets will have the following encapsulation, outer IP (FA/HA), Authentication, Encryption (authenticated), inner IP, etc. A FA could attempt to disrupt and or slow down traffic, but it certainly could not decode it or fake authenticated packets (i.e., denial of service is a possibility of course). There is an question here about how much if any security might be added by an MN obtaining its own COA (Care Of Address) and not using FA services. This would simply hide the MN's own home IP address from the foreign network and is a small but potentially useful service.

- Link layer access to packets in the clear is of course a consideration. The arguments from the previous discussion hold. We assume at least network layer security and do not rule out application layer security as desired or appropriate.

- UDP registration packets and IPSEC presents an interesting problem where untrusted Foreign Agents are concerned. The problem is that the UDP registration packet itself cannot be encrypted or the FA will not be able to use it. The obvious solution is that the registration packet should not be encrypted. However it is not yet obvious as to whether any elegant architectural solution exists to make that happen. The operating sytsem will be responsible for IPSEC crypto mechanisms and an application routing daemon will be responsible for sending and receiving the UDP registration packets. Hence we have a disconnect. This is in fact a generic problem with IPSEC as we mention below.

- There may be potential pitfalls with the use of simultaneous registration with ¿ 1 Foreign Agents. Simultaneous registration in Mobile-IP is currently specified as meaning that packets outbound from HA to MN should be duplicated for as many FAs as the MN is registered with. This may certainly be useful from a redundancy point of view. However it offers an untrusted FA a distinct opportunity to keep the MN's registration a little longer (or forever) if the MN moves on. We will keep this problem in mind during our second year research on redundancy.

- One of the replay protection mechanisms is based on NTP timestamps. Presumably an active attack might be started by spoofing NTP packets, adjusting time appropriately, and then launching a replay. NTP has an authentication mechanism, but if you are hoping for total source to end system authentication, that may be unlikely. The NTP deployment situation here is complex and we do not know what capabilities may really be available. Certainly a Mobile Node can get NTP packets from home. We may be able to use IPSEC to guarantee that those packets are authenticated in lieu of cradle to grave NTP authentication mechanisms. We will look into this. NTP is deployed on our routers on campus, but it is not authenticated.

C.5 IPSEC

We are tracking developments in IPSEC and attended the recent L.A. meetings. We feel that we at least are beginning to grapple with the issues. At this point, one might feel that IPSEC is coming somewhat unglued. The fundamental authentication and encryption "transforms" (algorithms + mechanisms to secure various network headers) are undergoing change. IPSEC has a tension between cryptopgraphers who always feel that more analysis is needed and engineers who want to do something and throw it over the wall. Lately the cryptographers are winning. The AH MD5 transform is probably going to be replaced by a better transform in terms of cryptography (HMAC from Hugo Krawcyzk et. al. of IBM). Our cryptographer Sarah Mocas has investigated Hugo's work and feels that his reasoning is sound. Basically his HMAC algorithm will offer much more authentication security for a small amount of computational overhead. The ESP (encryption) packet was originally assumed to be "standalone"; although many thought that it would of course be combined with an AH authentication packet. This was a point of confusion. As a result, a new ESP transform is under way and the original will be replaced with one that subsumes authentication inside the transform; i.e., it will not be possible to produce encrypted packets sans authentication. Our analysis folks also intend to look at the proposed session key protocols (Oakley, SKIP, ISAKMP, Photuris). Although this is preliminary, we rather like the NSA ISAKMP protocol.

It has been pointed out by the IETF community that IPSEC has some built-in difficulties where encryption is concerned. Both RSVP and DHCP may need information that is present in higher-level headers in order to function. (Anyone who believes in strict layering separation may be in for a rude shock). This problem applies to Mobile-IP's UDP registration header too.

In terms of implementation, we are currently investigating the kernel architecture of the Naval Research Labs initial IPSEC kernel. This kernel is of BSD 4.4 vintage and includes IP next generation code. If possible, we would like to borrow as much (or all) of an existing implementation and combine it with our version of Mobile-IP. The NRL implementation by this point is somewhat flawed since the security mechanisms for AH and ESP are changing. On the positive side, it has a key socket mechanism that is certainly something we would want to use. It curiously binds security associations with sockets as opposed to protocol control blocks or routes. We are not sure that is the right choice (although it may be a valid choice) and we need to think about this issue more.

It was announced recently on the IPSEC mailing list that Cisco would be releasing a version of ISAKMP including fundamental IPSEC headers as a UNIX reference implementation sometime in April. FreeBSD in particular was mentioned. From what we understand of ISAKMP at this point, we think that the architectural aspects are important and useful. ISAKMP offers a model where security associations may be exchanged and policy is negotiated and decided on by application-layer daemons. Of course it also includes session-key management. We hope that this code is released soon, as it may be a better choice for us.

It should be pointed out that until we have IPSEC, we are naked at the link layer when it comes to sending telnet/ftp passwords in the clear. As a result, we are installing the Finnish "secure shell" that is a drop-in replacement for rlogin/rsh, etc. on our laptops and mobile routers. Secure shell uses static RSA keys for authentication and IDEA for encryption. We can use slogin as a replacement for telnet, and you can run ftp over a secure shell tcp link to hide passwords (and data). We have managed to get ssh deployed on local CS/EE systems by systems staff. It will be interesting to informally compare the two mechanisms (ssh versus IPSEC) once we have IPSEC implemented.

C.6 Redundancy and Ad Hoc Routing

Although this is work for next year, we have done a little research on redundancy mechanisms and have some initial ideas. We have already mentioned an initial form of Ad Hoc routing that will at least allow direct but authenticated contact between "consenting" Mobile Nodes. Our review of papers in this area and our own ideas lead us to believe that this protocol can be enhanced to allow more indirect communication between Mobile Nodes by developing a limited flooding source routing protocol based on multicast, and sequence numbers. We also think that we might investigate pairing the use of multiple Foreign Agents (allowed by Mobile-IP) with packets multicast to the "default" FA. This would give two-way redundancy between MNs and FAs. Multicast looms important here and we need to test and make sure that our Wavelan drivers support it.

C.7 Outreach

It is ironical that we teach Computer Science in places where computers and networks do not (and sometimes cannot) exist. We hope to eventually deploy a number of mobile routers elsewhere, as for example, we have labs and offices in another building across the street. We also want to cooperate with central PSU computer management who manage access to classrooms elsewhere that we use. We have demoed Wavelan to them and they are quite interested in using it to address certain wiring problems in PSU buildings that are not capable of being wired. (By coincidence there are classrooms that we use in one such building). Naturally they brought up link-layer security as a concern. Of course, that is the focus of our project. As a result, it appears that we may be able to do outreach within our own campus in the next year, at least to extend coverage to a few remote classrooms where we teach. Our initial goal here is to enable the use of mobile systems via some of our faculty. Ideally, we may be able to interest the university in some scheme to allow students (at least engineering students) access to mobile systems as well.

There is Ph.D research on-going at Oregon Graduate Institute that is focused on mobile concerns. One issue being investigated there is how to do dynamic switching of network pcmcia cards. One of our PIs is on the Ph.D committee and OGI is going to buy and deploy a small Wavelan network.

C.8 References

These are informal and are all available on the Internet.

Randall Atkinson. IP Authentication Header. RFC 1826, August 1995.

Randall Atkinson. IP Encapsulating Security Payload. RFC 1827, August 1995.

S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. ACM Computer Communications Review, 19(2), March 1989.

S. M. Bellovin. Problem Areas for the IP Security Protocols. draft. March 1996.

Mihir Bellare, Ran Canneti, Hugo Krawczyk. Keying Hash Functions for Message Authentication. preliminary version. January 1996.

Charles Perkins, editor. Mobile-IP. Internet Draft 15 – work in progress.

Secure shell URL - http://www.cs.hut.fi/ssh

Appendix D

Quarterly report – Spring 1996

D.1 Project Status Overview

We have made a considerable amount of progress this quarter. We are in the final stages of completing a Mobile-IP implementation for FreeBSD and hope to make an alpha release to interested parties this summer. The Mobile-IP system currently contains a number of security features, including timestamp replay prevention code based on NTP and security logging, plus all of the basic security association relationships possible in Mobile-IP. We have deployed Mobile-IP in two buildings and a number of students and faculty are making use of it. Our implementation of Mobile-IP will serve as a basis for our security research.

We will begin to develop and integrate IPSEC mechanisms with Mobile-IP this summer. Our goal is a first cut secure mobile infrastructure that supports the notion of virtual secure enclaves via routing and in some sense glues Mobile-IP and IPSEC together. Our ideas are based on the analysis of Mobile-IP security that John McHugh and Jim Binkley presented in the previous report. Policy mechanisms will depend on knowledge of location courtesy of an enhanced Mobile-IP, and secure tunnel routes using the combined AH+ESP transform in the kernel. We may be able to complete this work this summer, but it may continue into next fall as there is a lot to do. We will make an informal presentation of our current design notions later on in the report.

We have gone through a round of planning for redundancy work next year and will make a short report on that subject later in this document.

As of this summer, we are bringing on board one full-time staff programmer and 3 graduate Research Assistants, two of whom are just starting the Master's program at PSU. We hope they will be with our program for a while.

We have established a Mobile-IP infrastructure in two buildings on campus. There are three agents (1 HA, 2 FA) in our PCAT engineering building and one Foreign Agent in the Mill Street CS Lab building. Three graduate students, 4 professors (3 CS, 1 EE) and two staff members have mobile laptops. Over summer, and during the next year, we hope to build up our lab with a machine or two and extend our mobile wireless infrastructure to more buildings on campus. Our goal with the latter effort is to bring wireless laptops to the numerous buildings where CS and EE professors teach. PSU is located in downtown Portland and includes a number of "white elephant" buildings that currently lack network connections in the classrooms. Bringing a wireless laptop to the classroom allows faculty to demonstrate programming concepts with real code and real running programs, which ironically up to now has been difficult if not impossible to do.

D.2 Mobile-IP Status and Planning

We have a few things left to do to Mobile-IP before we make a release. As we mentioned in the previous report, in our opinion Mobile-IP is a prime example of design by committee; i.e., there are too many features, requirements, and optional parts. Our implementation has attempted to minimize kernel involvement and maximize application daemon code. We have only implemented the use of Foreign Agents and neglected a number of other features (some required); e.g., we are not trying to implement the MN as a "do-it-yourself"

FA, we have neglected to implement soft state in tunnels, or any other kind of tunnel mechanism except for the required IPIP, or the nonce replay option. We are currently NOT planning on doing much more other than cleaning up some hastily written code and working long term on improving the hand off mechanism. Even so, we have over 10000 lines of C source for the daemons. Our goal at this point is to use Mobile-IP and to begin to modify it to make all network traffic secure.

We have five executables for Mobile-IP:

1. a simple key generation program, md5gen, that spits out a 128 bit key. We hope to improve its "randomness" in the future. We understand that newer versions of the freebsd kernel (newer than the current release) will have a device in it that can be used for random number generation and intend at some point to look into that. The current application uses a number of sources of random input; e.g., "netstat -in" which provides a count of packets seen so far.

2. a mobile node daemon, mnd. It uses a configuration file /etc/mnd.conf that contains various addresses including that of the HA, and key information. Note that for purposes of our experiment we assume that laptops are single user systems, and that in fact neither our routers or laptops are ever to be multi-user. We are focused on network security, not o.s. security. Having said that, we are taking elementary security precautions; .e.g., the daemons and the utilities that query them are only runnable by root. The configuration files are only readable by root. The mnd is primarily in charge of determining the state of the mobile node, and it sets the default route to the HA or FA as appropriate. It keeps a security log file in /var/log/mnd.log. Certain events of interest security-wise are logged there. Two kinds of security relationships may exist for the mn daemon and associated manual keys are stored in the /etc/mnd.conf configuration file. The security relationships are MN to HA (which we use by default) and FA to MN (which we have tested but are not using). (The third relationship that exists in Mobile-IP is between the FA and HA and does not apply to Mobile Nodes. We do not intend to use it either and have not as yet even tested it.)

3. a utility named mnstat that queries the mobile node daemon for information. It uses a UNIX datagram socket to communicate with the mnd so that messages from outside the machine have no way to get to the daemon. mnstat allows a user to do the following:

    (a) determine the agents available locally. Eventually this information will include wavelan signal strength.

    (b) manually command a hand off to a particular agent (and override the current choice if desired. This is extremely useful for testing).

    (c) determine the current state of the mn daemon; i.e., ATHOME (at the HA), AWAY (at a FA), and NOWHERE (no agents found).

    (d) print information of a statistical nature; e.g., how many beacons have been received.

    Basically Mobile-IP requires an authenticated and sequenced UDP request and reply. As a result, we have been able to teach mnd how to determine that connectivity with the HA has been lost. When mnstat shows the state, it can indicate that the connection to the HA seems to have vanished.

4. a Mobile-IP daemon, mipd, that runs on both FAs and HAs and is capable of being mobile out more than one interface. It uses a configuration file /etc/mipd.conf that may contain key relationships for keyed md5 as appropriate. It is assumed that the mobile router's console is secure and that this daemon and its config file are accessible by root only. The daemon maintains a visitor list and sets routes appropriately. If it supports the HA function, it sets tunnel routes via a virtual tunnel device to the MN. For a FA function, it sets a link-layer route to the MN. The latter allows direct connection between two different IP subnets; .e.g., we typically set the wavelan card on FAs to use a private net 10 IP address. Our MNs and HA are all on the subnet 204.203.71.192 (netmask 255.255.255.192). Thus the connection between the FA and MN breaks the IP subnet rule.

5. a utility called mipstat is available on agent routers to query the current status of the agent daemon (mipd).

We have attempted to minimize the kernel modifications. They consist of the following:

1. We added three new kinds of routes which primarily affected a few core routing modules and added a virtual tunnel driver. The kernel now supports two kind of "foreign" route, one for a default used by the MN, and one used by FAs to link directly to an MN. The foreign route may or may not be on the same subnet. If the subnet is not the same, information must be supplied to the kernel so that it can determine what interface to link the route to. The virtual tunnel driver (mvif0) basically puts a tunnel on an packet where the route gateway is the tunnel destination. For Mobile-IP, we run a tunnel from the HA to the FA, so the FA's Care Of Address (COA) is the tunnel endpoint. The driver basically adds an IP header to the packet with the IP src = HA, IP dest = FA, and then sends the packet to ip_output again so that it is routed normally. This driver uses code borrowed from mrouted multicast code in the kernel. A major difference with the multicast setup is that our driver ONLY handles unicast packets (multicast is left to mrouted) and tunnels are added dynamically by the mobile agent daemon (mipd). Of course, with the MBONE mrouted code, IPIP tunnels are added by hand. We believe that we have not damaged the BSD kernel's multicast functionality (in fact we fixed a bug in it).

2. UDP input is modified so that any packets coming into port 434 (the Mobile-IP port) have the interface IP address appended to them (after the UDP checksum). This is an OUTRAGEOUS hack, but it allows the mobile agent daemon to know what interface a mobile node is using and thus allows us to decouple the daemon from the kernel. It may be possible to make this a general facility say with a setsockopt so that any routing deamon could use this. This would fix a general problem with routers and the BSD stack; i.e., routers could use UDP packets (or raw packets) and thus know which interface they received a packet on.

3. There is a little bit of Mobile-IP glue code that is needed for routing and the tunnel input routine had to be carefully melded with the existing multicast input code.

4. We added a setsockopt to be used with the existing ROUTETOIF option, so that routing daemons could instruct the kernel to turn packets sent to the interface address into LIMITED BROADCAST (255.255.255.255) IP packets as required by Mobile-IP.

In terms of security features, we have implemented all 3 of the key association mechanisms as outlined above. We have also implemented timestamp (NTP) based replay attack prevention. We have tried to isolate normal error logging and security logging. State switch messages and connection messages, plus anything deemed to be a sign of peculiar behavior is logged in the security log. Fatal errors and the like go into the system log.

The most curious and hardest part of Mobile-IP has been the implementation of the timestamp replay feature. We assume that the HA has a secure source of time (e.g., authenticated NTP). In our case this is not actually true. We are running xntpd on our HA and getting the time from local routers that are hooked up to the Internet NTP time infrastructure. The MN uses what is essentially a seconds-only NTP timestamp as a security nonce (and sequence number) to timestamp authenticated packets sent to the HA. If the HA determines that the MN's time is out of whack by greater than a slop factor (currently set to 2 seconds), it NAKS the time and returns the current time (still authenticated) to the MN. The MN updates its time and tries again. The result here is that the HA is synchronizing the MN's time to seconds. If the HA receives a packet from the MN that is behind the last known time from an MN that is currently registered, a security warning is logged and the packet is ignored. It might be a time spoof, but most likely it is a delayed registration request. Note that we do NOT drop the MN from the visitor list as this would allow a denial of service attack. In general, it would seem that our IBM laptops are having their time updated only a few times a day. This mechanism could perhaps be improved by use of stable storage at the HA and by network header authentication between trusted FAs and the HA. It could also be improved by making the HA's source of time secure to the source. We don't intend to do those things. However at some point, we would like to try and use GPS to see if we can do something clever with two sources of time. (In order to do that, we will need a kernel infrastructure that can support > 1 PCMCIA card at a time).

We have tried hooking MNs up via slip connections to a remote FA connected back to our network over a modem and have noted that the time code still operates fairly well as does Mobile-IP. There are a couple of curious network topology points one can make about FAs. First of all, as a very minor security modification, we set the wavelan IP addresses for FA systems to use subnet 10, the private subnet. As a result, any host trying to access our networks through the FAs will not be successful as our ciscos refuse to route net 10 packets. Only the router itself can be attacked. Of course, the use of subnet 10 here also reinforces the notion in Mobile-IP of FAs and MNs being on different subnets. More curious is the discovery that if we take a FA home say and use it over a modem connection via SLIP or PPP, we have essentially managed to take PSU IP addresses home with us. This is either an extremely bizarre or important point (we don't know which). The upshot is that Mobile-IP could be used to allow networks at home, without requiring that internal routers actually know or support additional routes and that IP address space might be better utilized. Assume that a network provider has a class C network address and that PPP routers at home support a wavelan card with a private IP address. The network provider keeps a HA close to the terminal emulator (if not co-located). Mobile Nodes at home can be on the Internet and use the Home Address associated with the HA, and never actually be "at home" (presumably home would be on the premises of the IP provider). If the HA is close to the terminal emulator, tunnel overhead is trivial compared to circuit overhead imposed by a analog phone line or an ISDN line.

D.3 IPSEC Status

We have struggled mightily to catch up with IPSEC work and at this point have at least read the major drafts. We are sending our staff assistant to Montreal to attend the next IETF meeting. Our co-investigators are beginning to investigate how to use a formal logic system to prove (or disprove) assertions of trust about IPSEC network protocols.

We intend to use the combined AH+ESP transform (coupled with tunnels) as a major tool for Mobile-IP security. We have a concern with the combined transform and have expressed it at least to the author of the transform. We don't understand why the authentication part of his combined transform does NOT cover the IP header addresses too. His notion of supplying a nonce in the header though is interesting and we may see if we can do something with that eventually. One could point out that a notion of nonce at the IP layer requires a per key association "counter", and is hence extremely stateful. BUT if said counter is actually the time, it seems that statelessness might be possible.

For the last quarter, we have had a graduate student working with Jim Binkley, investigating the Naval Research Labs IPSEC code. Our hope was that we may be able to use it and thus save some time in the implementation of IPSEC facilities. We intend to integrate some parts of that stack into our mobile FreeBSD kernel this summer. However the parts that relate to policy-management will have to be changed as we require policy mechanism to be associated with routing, not sockets. We should point out that the NRL effort included an IPv6 (IPNG) stack, which we do not intend to use.

As a short summary, the NRL stack offers the following mechanisms:

1. a key socket which is totally analogous in nature to the current BSD routing socket (and in fact was based on it). The key socket currently allows keys to be bound to IP src and dst addresses (and spis) in a kernel key association table. Suitable keyed-md5 and des keys may be put into the kernel. We will use this mechanism.

2. AH and ESP transforms exist that basically slice a packet up just before it is shipped based on an extremely simple policy mechanism and insert AH or ESP headers before an external IP header (which may be a tunnel). These transforms are based on the original IPSEC RFCS (1826, etc.), which are now out of date. We do not intend to use ESP by itself and will change the code so that we can support basically AH, or combined AH+ESP. For a first cut, we will use the current keyed-md5 and des-cbc crypto algorithms, but we also plan to start working on a general kernel internal security driver infrastructure that will allow both hardware (via pcmcia) and software cypto drivers to be installed and used more easily. The current des code is based on the Australian free library implementation. Over the summer, we intend to have one of our RAs, working with Sarah Mocas, implement the new HMAC-MD5 and HMAC-SHA transforms.

3. The policy mechanism in the current NRL IPSEC stack is based on two hooks. It is possible to set a global kernel base mechanism that governs all policy. This has to be compiled into the kernel at configuration time. This is an oversimplification, but for the sake of argument, you can assume that the policy is either, ALWAYS use IPSEC, or never use IPSEC. On top of this, one may also choose to bind policy to a particular socket, and in fact the socket descriptors include a simple policy description variable. So one ends up with the possibility of per socket keys and some sort of notion along the lines of "user-oriented keying". However on the downside, application code needs to be modified to take advantage of this facility and there will be no way to interoperate with existing socket/applications code that knows nothing of IPSEC. As a result of this design choice, on packet output, IP looks at the socket policy variable just before a packet is handed off to a driver. On input, the design is much more baroque. The transforms (AH or ESP) are carried out as normal layers (e.g., IP — AH — TCP — DATA and note that ESP is not intended to be nested on top of AH), and a linear search is performed to map a spi to a key table entry. A policy check must be made at the transport layer only after the TCP or UDP ports are mapped to protocol control blocks (which have socket pointers). One should point out that this is the inverse of the notion of checking policy first and then doing security transforms which was put forth in John Ioanniddis's SWIPE. We think this policy model is too simplistic, not interoperable, and more important not useful to us. We want policy to be based on routes at the network layer, which will more easily facilitate the notion of a secure virtual enclave (e.g., between a remote MN and a HA). On the other hand, there is nothing inherently wrong with binding security associations with sockets. User-oriented keying and "host-oriented" keyed are not mutually exclusive and probably both mechanisms should exist. We just want to bind policy to routes and intend to explore in that direction.

D.4 Towards A Secure Mobile-IP

The following section will outline how we intend to implement IPSEC and integrate it with our Mobile-IP implementation. We will use our Mobile-IP source base, and various IPSEC mechanisms, including the useable parts of the NRL implementation. Our theory here was presented in the previous report. The guiding principle is simply that we wish to do everything possible to "hide" (where hide means location privacy plus authentication plus encryption) data transfer between a Mobile Node and a Home Agent. As a result, for mechanism we desire to use a combined authentication and encryption transform from IPSEC (AH+ESP) and in general will nest packets in tunnels in order to hide participants and network topology from "interested" intermediate snoopers. This theory leads to some interesting routing ramifications. We do not blindly apply it to all cases. The architectural goals have three basic network topologies as a focus:

1. Network topology #1: An MN located at an untrusted FA (which may be trusted for that matter) establishes a secure network relationship that is two-way between itself and the HA. In essence, this is a virtual private network and the "circuit" between the MN and the HA is made secure.

2. Network topology #2: An MN establishes a secure link that ends at either a FA or HA, where we assume that we only want to secure the link. The network "inside" the HA or FA (seen as routers) is deemed secure.

3. An ad hoc relationship is established between two MNs that may or may not be on the same IP subnet. Eventually we may want to look at a multicast means for key distribution. The point should be made that ad hoc situations, especially when they are cut off from network infrastructure, have some interesting peculiarities that are different from generally connected networks. Key generation and distribution need not be very scalable (at least at the moment). One could manufacture a key, put it on a floppy, and pass the floppy back and forth. On the other hand, designs that assume either DNS or a general world-wise key + certificate architecture need to take into account that mobile nodes may be disconnected and disconnected for long periods of time.

D.5 Mobile Network Authentication

In order to make Mobile-IP secure, we are going to work in two areas, one of which we will call "network authentication" and the other "network security". Network authentication is going to apply to link-layer addressing and is actually a form of secure ad hoc routing. However in this first cut at ad hoc routing, we will not allow Mobile Nodes to be routers. Instead we will merely authenticate (IP, MAC) address pairs with shared secrets using keyed-md5 (or whatever the latest AH transform turns out to be). Essentially this will allow us to do two things: 1. we will not be using ARP anymore except that we will build a user interface for our systems so that if ARP is needed, it can be used, and 2. there will be no "same-subnet" else can't talk limitation.

The bottom-line here is that every host will beacon, both MNs and agents. We will augment the ICMP beacons currently used with Mobile-IP in such a way that theoretically they will still be usable with other Mobile-IP implementations, but in our case they will include extra Tag-Length-Value extensions that authenticate the information for those who have shared keys. The packets logically will contain: (the sender's IP address, mac address, a timestamp nonce, and an authentication hash). Authenticated beacons will cause link-layer routes (or ARP table entries) to be installed. Without them communication will not be possible since normal ARP will be ignored. The MN and agent daemons will time these entries out so that if beacons are not heard, connectivity will go away. One upshot of network authentication of course is that the ARP attack mentioned by Bellovin in his classic 1989 paper on TCP/IP security problems will no longer be possible. Another result is that we will be able to divide our systems from "their" systems. For example, we will know which FAs belong to us and which do not.

In order to set policy, we will create a user interface called xmnstat that will include network authentication and network security dialog boxes. The network authentication dialog will provide a set of policy choices based on knowledge of the current Mobile-node topological state; i.e., what to do when at home, when at a trusted foreign agent, when at an untrusted foreign agent (beacons have no keys or keys don't match), and what to do about link layer contact from other Mobile Nodes. Each one of these topologies will have 4 choices plus a yes/no checkbox for ARP compatibility. The choices will include:

1. never - don't install any routes period.
2. always - always install routes and ignore keys
3. key - only install routes if the keys match
4. ask - ask if ok via a popup if a mn or agent exists but doesn't have a matching key

Our default configuration will probably be "key"; i.e., we will only talk to agents and MNs that share a secret key with us. The ARP check box will basically allow or disallow ARP functionality as needed according to the topology. We will turn it off here but we might use it for reverse compatibility as needed. In terms of kernel functionality, there is currently a NOARP switch in the interface configuration flags set. We will modify the kernel so that the Mobile Node daemon can turn ARP on/off for a particular interface. On agents we will turn ARP off on mobile interfaces and of course, leave it on for true Ethernet interfaces.

D.6 Mobile Network Security

We plan on putting the NRL key socket mechanism into our kernel, and we also will implement the AH and AH+ESP transforms. However our policy work will focus on matching policy with routing.

As we mentioned above there are three topologies that we find interesting. For discussion here, we are only going to theorize about the first topology, MN via untrusted FA to the HA.

As usual, with routing there are two problems, MN to HA, and HA to MN. In the former case, we could supply in the configuration a canned list of static network or host routes. When we arrive at a untrusted FA, we would know the FA was not one of ours because of the authentication (or lack of it) in its beacons. We would install tunnel routes with an AH+ESP flag so that all packets from the MN to those sources would have the following structure:

```
| ip:  ip dst = HA, ip src = MN | AH + ESP | ip inner ...
```

As a result, we hide who the MN is sending packets to and of course authenticate and encrypt the packets. As a form of stronger policy, it would be nice if we could teach the kernel's routing mechanism how to clone packets sent to specific destinations (except for the default!, as the Mobile-IP packets sent to the FA and forwarded to the HA cannot have IPSEC applied to them). For each specific destination, we would automatically make a specific tunnel. This would represent a stronger policy choice (we might call this a "tunnel cloning" mechanism).

Sending packets back from the HA to the MN in the same form is more curious. Basically the HA already has a tunnel that almost (but not quite) reaches the MN. The HA needs to send packets to the MN in the following format:

```
ip:  ip src = HA, ip dst = FA | ip:  ip src = HA, ip dst = MN | AH + ESP |
ip inner, ip src = X, ip dst = MN | ...
```

Thus in order to hide who is sending the MN packets and thus not reveal internal topology to the world, we need to have what we will call a "double tunnel" header. The outermost IP packet would be stripped at the FA. The next header (that is still outside the crypto header), would get the packet from the FA to the MN. The MN would strip that header, deal with the crypto header, and then deliver the packet to itself. This is somewhat baroque but it meets out goal of hiding as much as possible between HA and MN.

A long-term research question here that we hope to begin to supply one answer for is: "what form should network security policy mechanisms take in a TCP/IP stack?". IPSEC and Mobile-IP only propose certain network headers, but they do not seem to take into account how those protocol requirements might be implemented (and implemented reasonably) in an operating system. The question of separation of policy and mechanism remains central. In the latest draft of Mobile-IP, there is a requirement that MNs at FAs are not supposed to use ARP, and are supposed to somehow cache MAC addresses (from the FA beacons) in order to allow the MN to talk to the FA. This is probably bad network design and it is worse kernel design. There is no attempt to deal with the central question of network authentication, which is probably crucial to Mobile-IP in the long run. With IPSEC, there has been a long term tension between "user-oriented" and "host-oriented" keying. It is not clear how policy should be architected in the kernel to supply both of those needs. We suggest that there is nothing wrong with either of those mechanisms and that security bindings might be made to both sockets and to routes. We suspect that socket-level keys should be end to end at the application layer (or socket layer) and should not rely on network-layer IPSEC mechanisms (but they could of course use Oakley or SKIP). A real-time system might get by with one key (there might be only one protocol that accesses it). On the other hand, a multi-user system by definition needs one or more keys per user. In fact, it may be more important to worry about how to override "host" keys in some cases. For example, the current Mobile-IP manual authentication mechanism is essentially socket layer (or session layer). AH authentication between a MN and a HA, when the MN is at a untrusted FA, cannot be used as the untrusted FA will not be in the key relationship. If we blindly applied the same key (and AH) to all packets going out of a MN, we wouldn't be able to use Mobile-IP. In summary, architectural flexibility must be a long-term goal.

D.7 Theory of Mobile-IP Redundancy

In this section we are going to briefly discuss our ideas about making Mobile-IP more secure in terms of redundancy; i.e., there are three areas of work: 1. ad hoc routing; i.e., how MNs can route amongst themselves and also find paths to agents through other MNs; 2. redundant FAs., and 3. redundant HAs. Along with the Secure Mobile-IP work, this will be our focus for next year.

D.7.1 Ad Hoc Redundancy

Our goals here should both include both network authentication and network security. Our first cut ad hoc protocol is focused on link-layer routing and will not allow MNs to route amongst themselves. However eventually we want to develop a second-cut ad hoc routing protocol based on Dave Johnson's (CMU) ideas of source-routing; however in our case, authenticated source routing. A key idea here is that MNs should also be able to find a path to Mobile-IP agents that are further away than one hop. We also want to see if we can use signal strength as a metric and see what we can do to learn about the cons of beacons and how they might be alleviated. If we have enough time, we would also like to investigate some sort of ad hoc multicast key distribution protocol.

D.7.2 FA Redundancy

We find that to some extent we are already working on this as we happen to have 2 FAs within hearing of most MNs in our main PCAT building. Our current hand off algorithm is based on starvation (no beacons) and we need to teach it to be based on signal strength (if available). We can do that with wavelan and currently are working on getting the signal strength information out of the wavelan driver. We also think we can improve the hand off algorithm by imposing a test in the registration back off mechanism, so that when there is indeed greater than one agent possible, if no ACKS are heard, the system can switch and try the other agent. Since the registration is MN to HA, this means that we may be able to both deal (to some extent) with weak connectivity to an agent, and a router failure between the FA and HA.

Our long-term approach might be as follows: A mobile node would need to simultaneously register with any FAs it can hear. Mobile-IP explicitly allows that. Assume that the MNs in the presence of FAs can setup a default route to a multicast address. This means that every packet sent to default (a FA) will go to every FA without multiple resends. This is a one-way process from MN to FA. The FAs would continue to do what they do now (install a link-layer route to the MNs). Multiple registrations at the HA means that routing table entries must support a LIST of destinations. A packet sent to such a LIST route would have to be resent for every gateway in the LIST.

D.7.3 HA Redundancy

There are probably any number of ways one might approach the problem of removing the HA as the single point of failure in the Mobile-IP design. One HA is truly a terrible design flaw in terms of presenting an opportunity for total catastrophic system failure due to failure of one system. We suggest that for survivability reasons, it would be best to design a system where the HAs are not necessarily on the same subnet or link. They may not be far apart (although they could be), but it would be nice to have at least two HAs, where the survival of the mobile system is not dependent on one central router that leads to both HAs.

Assume that the MN has a list of HAs that it might use. It will use one HA at a time, but if it does not get ACKS from that HA, it can switch over to the other HAs in the list. Any HA will serve as “home”. HAs basically have two interfaces, one wired and one “mobile”. The mobile interface is a interface to a partitioned network by definition. Ironically, that means that the notion of “home” becomes blurry and the system begins to resemble FAs, and in some ways will ultimately resemble John Ioannidis’s original Mobile-IP design at Columbia. HAs are paired by definition and if one of them installs a tunnel route to an MN, the other must be told about the MN and must install the same tunnel. If a link fails to a HA, normal dynamic IP routing will send packets to the other HA. This of course, may happen anyway since HA2 may be closer to a host sending packets to the MN than HA1.

D.8 Outreach

One of the PIs has met with a couple of small companies (e.g., Solution Logic) in the Portland area that are interested in network mobility and also gave a talk on the project at a recent Portland technical conference called ETACOM. Both of the PIs will be meeting with Intel security folks this summer in Hillsboro.

We have continued cooperation with OGI and one OGI Ph.D candidate on his mobility research. He is helping us this summer upgrade our kernel environment so that we can take advantage of his PCMCIA kernel infrastructure support. He is making use of basic infrastructure work on Wavelan drivers that we have done. We have submitted a joint paper with him on dynamic network interface card switching to Mobicom.

Lastly we should mention something that might be called “Inreach”. The CS professors involved in this research effort are supporting an effort made by our Electrical Engineering department to start a laboratory devoted to mobile research. Of course, their research will have a focus on various hardware research topics including mobile cellular radio. However they have recognized our interest in network stacks, mobile security, and cryptography and have made the effort to include courses we teach in cryptography, security, and networking in their program. Long term, we hope that this collaboration will pay off.

Appendix E

Quarterly report – Summer 1996

E.1 Project Status Overview

Summer was very busy and much progress was made. It is reasonable to say that the project is up to speed at this point.

At the beginning of summer we were able to add 1 fulltime person and 3 Research Assistants to the project. In spring, David Reeder graduated from PSU with a B.S. in Computer Science and at the beginning of summer he joined our project as a fulltime employee. The three R.A.s are Xu Hao, Radheka Godse, and Bjorn Chamblis. Radheka and Bjorn are new graduate students as of summer.

Accomplishments for the summer include:

- In early summer we made an alpha release of our Mobile-IP implementation for FreeBSD 2.1 which included routing daemons for mobile nodes and agents, utilities for making status enquires of the routing daemons, and drivers for ISA and pcmcia cards.

- New personnel were trained in our Mobile-IP implementation and brought up to speed in terms of project plans.

- An X-based user interface was created for the Mobile Node daemon. The aim of the tool is to allow the user to control and observe various aspects of mobile status including especially security policy and setup, mobile configuration management, and various runtime statistics including agent signal strength. There is a section below with more details on the tool.

- We have completed a port of the NRL IPSEC stack into our BSD based kernel (FreeBSD) and intend to use this as a foundation for our fall work on integration of IPSEC and Mobile-IP routing. We used the NRL June release.

- We have for the most part finished the coding of our authenticated ad hoc routing protocol (referred to in previous reports as "network authentication"), which is intended as a replacement for ARP. Initial testing has shown proof of concept. We intend to finish testing this quarter, integrate control of this sub-system with the X interface, and deploy it on our Mobile-IP network at the end of the test period. This protocol has both network security and important redundancy aspects. We will provide more details below.

- We enhanced our Mobile-IP implementation to allow quick handoffs between radio-based agents. This work was primarily aimed at providing redundancy along the lines of switching between agents when one agent should fail. It also applies to path testing between MN and HA. If the route via one FA to the HA should fail and another FA is available, we will try it. This work will provide us with a basis for what we call Foreign Agent redundancy later down the road.

E.2 Mobile-IP

An alpha release of a Mobile-IP based on draft-15 was made around the end of June. Our implementation at that time had around 12000 lines of C code (counting *.c, and *.h but not kernel mods or the X user interface which was produced after the alpha release). The implementation included our ISA and PCMCIA Wavelan drivers that have been ported from the original MACH versions developed by Anders Klemet. The release included man pages for the routing daemons (MN and agent (HA/FA), man pages for the configuration files, source, makefiles, kernel patches for FreeBSD 2.1 and for a few related binaries (route/netstat), and an implementation guide. To our horror, we discovered a nasty kernel bug in August, and patched the implementation to fix it.

We have also stumbled upon a few Wavelan enhancements, in one case provided by a user, and have kept the drivers available and up to date. Although we have not kept statistics on how many times either the Mobile-IP implementation or the Wavelan drivers have been fetched, they seem to be reasonably popular. For example, we have been contacted by a student of JJ Garcia-Luna-Aceves at UC Santa Cruz doing mobility research who wanted the drivers.

The ftp site is:
`ftp://zymurgy.cs.pdx.edu/pub/mobility`.

The alpha release is also findable from our Secure Mobile Network web page:
`http://www.cs.pdx.edu/research/SMN/index.html`.

We should point out that the effort to complete a Mobile-IP implementation is huge. We are skeptical that there might be a complete Mobile-IP implementation anywhere. For example, our release is aimed only at Foreign Agent support and includes no support for a COA-based Mobile Node. This is because we intend to work on Foreign Agent redundancy issues (and in fact have already done work this summer along those lines). Also the COA-based system is probably impractical (at least for us) in a useful sense, as we lack the needed administration to setup DHCP (or some other mechanism) to dynamically allocate per-subnet IP addresses on real local networks. We have reviewed draft-17 over the summer and made a list of missing features. However we are attempting to minimize any more Mobile-IP work. Our current implementation suffices for our research needs, although we may want to add a IETF required missing feature or two along security lines as we go along. Our goal at this point (and in fact at mid-summer) was to only *enhance* the system along our desired lines of research in security and redundancy. This work is beyond the limits of the IETF Mobile-IP specification.

E.2.1 Considerations Regarding Timestamp Replay In Mobile-IP

We implemented the timestamp replay protection mechanism in Mobile-IP. Timestamp replay protection is done between the Home Agent and Mobile Node and applies to the UDP registration and reply messages. The HA provides 32 bits worth of NTP time (seconds) and the Home Agent may send a NAK and a better time if it believes the MN is out of sync. We did the obvious thing here which was to make the MN implementation willing to accept a time update from the HA (there is also a crude attempt at calculation of round trip latency made in our system). Our implementation allows the MN to choose between ignoring the HA and somehow updating its own time or accepting a HA update. However we currently have no better choice than to accept the HA's time. We have noticed that our PCs will typically get out of synchronization a few times per hour which may be acceptable. However latency due to congestion or number of hops combined with the need for synchronized time is worrisome. We performed an outrageous latency test by taking a MN home, and by running it over a local Foreign Agent connected to the PSU network over a 28.8k SLIP connection. We subjected the slip link to extreme degradation by using the popular UNIX ncftp client to perform an FTP download which used up all bandwidth on the slip connection. There were 100s of queued packets typically (and certainly some packets were dropped) as NCFTP typically does 30K i/o chunks (presumably in an effort to take advantage of ever-increasing TCP windowing sizes). Mobile-IP registration packets had difficulty getting through and we found that we had to tune our retry mechanism to make it slower in order to make it more robust. More surprising, log messages in the Mobile Node's security log showed that it was receiving HA NAKS with the new HA timestamp the same as the current MN's time. The HA logs showed that the MN was truly out of time syncronization with it on the order of 5 seconds or more. The queueing of packets at the slip client/server was real and caused an immense slow down of MN control packets. The bottom line here is that latency is problematic. We currently tolerate "time slop" of about five seconds. That may not be enough. From a security point of view, the time slop needs to be as small as possible. From a network latency point of view, it needs to be as long as possible.

One thought is that Mobile-IP packets should be given higher priority by IP than other packets (certainly bulk FTP packets).

There is interest in Mobile-IP circles in building FA systems where there is a hierarchy of FAs, say an interior "mother FA" that communicates status between MNs and HAs, and leaf FAs that actually deal with handoff between visiting MNs and local cells. The goal of such an architecture is to cut down on MN to HA exchanges over the Internet and thus reduce Internet traffic. The timestamp mechanism where HAs need to sync the MN mitigates against such an architecture.

We submit that Mobile-IP's approximation of NTP time via a few bits in a Mobile-IP registration packet is probably insufficiently complex. However what other possibilities exist for synchronization of time?

1. one might wish for an atomically-powered PCMCIA clock card, but certainly we know of no such card available now. We have been told that such cards might actually be available in the future.
2. we might use GPS, but GPS will not work indoors.
3. we might try and use NTP itself, and once we have IPSEC protection between the HA and the MN, we will try it and see how much bandwidth is actually used.

E.3 Lab and Wavelan Considerations

At this point we are only attempting to modestly extend our Mobile Network. We intend to add a few Mobile Routers (agents) for other buildings on campus over the course of the year. We also hope to place one Foreign Agent (FA) at Oregon Graduate Institute, which should be a rather interesting exercise both in network administration (they have a firewall. The FA will be installed as a "bastion host") and may shed light on how timestamp-relay based Mobile IP works across a number of Internet hops (currently 7 and subject to change).

We made a minor modification to our Wavelan drivers over summer that allows us to dynamically change the Wavelan Network ID (NWID). The NWID is used to allow Wavelan networks to logically distinquish between NWID Network I and NWID Network J. I and J will not receive packets from each other (they actually still use the available bandwidth, but the radio modems ignore packets from NWIDs not their own). We can boot laptops or mobile routers and set them to a different NWID. This allows us to setup a logical Wavelan link that occupies the same physical space as another network and is incredibly convenient for "lab testing". As a result, we can setup FAs on another NWID, and in essence have a small "virtual" lab that exists in the same place as a real radio network.

E.4 Authenticated Ad Hoc Routing

We are currently primarily engaged in our work to make Mobile-IP more secure and are in the process of finishing one of the two major portions of that work which we have called in the past "Network Authentication" and will refer to in the future as "Authenticated Ad Hoc Routing". (The second portion is a combination of Mobile-IP and IPSEC at the routing layer and will be discussed below). It should be emphasized that this work is an extension of Mobile-IP and is fundamentally a link-layer routing protocol that is analogous to the ISO ES-IS protocol and somewhat similar in function to ARP. The protocol is important from both routing and network security points of view.

Let us briefly outline how the system works. We have experimentally enhanced the ICMP router discovery protocol (RFC 1256) along lines suggested by the Mobile-IP implementation. In the current Mobile-IP system, only agents beacon. In our system, all entities including both agents and Mobile Nodes beacon. Beacons are either broadcast to limited broadcast (which is what we are currently doing) or multicast. We have added three Tag Length Value (TLV) suffixes that imitate and physically follow the Mobile-IP extension that is added to the RFC1256 router discovery packet. (From here on in, we call this a "beacon"). Logically the packet may consist of the following sections:

```
(ip header,
 ICMP router discovery header,
 Mobile-IP extension portion (required by Mobile-IP),
 new MAC address TLV,
 optional id string TLV,
 network authentication extension,
 optional ad hoc authentication extension)
```

The IP address of the sender is found in the ICMP router discovery header. The MAC address portion contains the MAC address of the sender. It is up high for reasons of "convenience". Our daemons are applications after all, and it is not easy in a BSD kernel stack to get MAC info upstairs to the application layer. Convenience here is close to necessity.

An optional information string may be provided. Our agents currently state where they are, along the lines of "Guinness: PCAT room 128D". MNs might state their user or might simply not use this facilty. The location string may optionally be placed in the agent or MN configuration file. If found, it will be placed in the beacon. This information is displayed in our X user interface for the MN daemon (section on this below).

One or two authentication extensions must follow. Our authentication mechanism here is analogous to the Mobile-IP authentication mechanisms; i.e, we authenticate via a shared secret based on a MAC algorithm (currently MD5, but there is a spi in the authentication extension, and the algorithm in use could of course be changed). We will discuss the authentication header more below, but for now there are several important points that need to be made:

1. note that we are essentially associating a secret key with a link. Key distribution (static for us of course) is viewed as belonging to the multicast key distribution problem, as obviousally one system sends out a broadcast/multicast beacon and N systems receive it.

2. The authenticated information logically includes at least the following information: (ip address, MAC address, and some control information about the system (it's a FA, HA, or MN)). The authentication is done over the ICMP header (where the IP address is), and all higher headers up to and including the spi in the authentication extension.

3. there are at least two possible keys that may be concatenated.

Any agent (FA or HA) or MN that belongs to the same logical link define themselves in terms of a fundamental network authentication extension. A second ad hoc key may be generated on the spot and given away for a brief period for ad hoc (face to face) meetings between MNs. For example, the key might be generated via a shared passphrase fed to MD5 when visitors come to your site or when you have a face to face rendezvous with some other Mobile Node. This key exists primarily so that no user ever gives away the site network authentication keys. It should be pointed out that an implementation might choose to have a list of keys on a per agent (HA or FA) basis for the network key and index these via agent IP address. In our implementation, we currently only have the network key (and ad hoc secondary key) supported, but this is an implementation detail and for now we our implementation will suffice for proof of concept. It should be noted that we accept beacons if either key is successfully authenticated. This allows MNs that are at home to speak to the ad hoc systems (the visitors) directly and retain their own current agent connection. It also allows a site manager to teach a FA about the ad hoc key for the duration of a meeting, thus visitors may be allowed controlled access to the Internet. The more important aspect there is that a site manager would NOT have to give a visitor a local network key that was in use by others (especially Mobile Users who are local and can thus not be knocked off a net link due to a key change). We reject beacons that fail authentication or lack it. In summary, beacons provide an authenticated MAC and IP address pairing.

Some basic design choices were made here. For the most part they were made due to reasons of Mobile-IP protocol coherence. One might ask the following pointed questions about our choices:

1. why not enhance ARP? One might argue that ARP was not designed to be enhanced (even though this has actually been done in the past). We felt that since Mobile-IP had already added extensions to

ICMP router discovery and had a reasonable TLV way for doing that, it was natural to enhance the Mobile-IP beacon. The MAC enhancement and location enhancements actually have general utility for Mobile-IP by themselves.

2. why not use IPSEC AH? The strongest reason is that we wanted to have a mechanism where we could have two authentication keys available, even if one of them was rarely used and it is not clear how AH keys might be provided in pairs. There would need to be a definition of AH made that would allow more than one key per packet. It is also much more convenient to imitate Mobile-IP where we can do the work at the application layer. One of the more compelling aspects of our Mobile-IP implementation so far is that we have attempted to minimize kernel modifications as much as possible. (We will outline the kernel modifications needed for this work below). Of course the most honest reason is sheer convenience because we did not until just recently have AH available. There is probably some way AH could be used. It should be pointed out that by putting an IP, MAC pairing above IP (as an ICMP payload), one has made the use of IPSEC possible here and there is no IPSEC definition or applicability for ARP.

E.5 Ad Hoc Routing Considerations

From the routing point of view, the major benefits are as follows:

1. link-layer reachability between nodes (agents and MNs or MNs and MNs) is estabilished simply because hearing a beacon defines reachability. A higher-level ad hoc routing protocol along lines discussed elsewhere by Dave Johnson (ad hoc routing driven by a source-routing technique), or Charlie Perkins (ad hoc routing according to a vector distance technique) might use parts of this protocol or be based on top of it. It is not unusual for routing protocols to seek to establish the existance of systems via "echos" or logical "hellos".

2. network redundancy is improved simply because the rather absurd restriction on same IP subnet-only access is removed. Two Mobile Nodes from different subnets can talk to each directly as long as they can hear each other. This is a simple form of ad hoc routing (link-level only) and does not allow cross-hop routing. No intermediate router is needed.

Before we go on, we review a couple of points we have made before in regard to IP subnets and radio-based topologies. We also will explain the routing theory behind our beaconing sub-system.

First of all, no one should ever make the mistake of assuming that radio based links (when they are broadcast based and not point to point) are topologically similar to ethernet or other wire-based technologies. With ethernet, the same subnet prefix implies that system A on a wire CAN actually hear system B (barring too many collisions). The notion of an IP subnet is 1 to 1 with the notion of a wire. Subnetting is not a meaningful concept on a radio link. It is an optimization on the problem of how a sender determines who is local and who is not. After all the classic IP routing algorithm is:

```
if same subnet
        use ARP
else
        send packet to default router
```

With radio the fact that 3 systems share subnet IP addresses even if they are close does not mean that they can talk to each other. Imagine a radio cell with three systems, A, B, C. B is an agent in the center of the cell. A and B are Mobile Nodes and are diametrically opposed at the edge of the agent's cell. Thus A and B might not be able to talk to each other directly. The consequences here are profound. If A tries to talk to B, according to most IP kernel stack implementations, it will attempt to use ARP. This won't work. It must instead send the packets to the agent. On the other hand, the agent B, must not try and use a routing redirect to tell C to talk to some other router D. B has no way of knowing whether or not C and D can talk directly. We simply use beacons here to install link layer routes and our resulting routing algorithm is roughly:

```
if a link-layer route exists (because of beacons)
        use it
else
        use the default
```

Of course one may also make the observation that Mobile-IP renders subnetting absurd too. After all a Mobile Node may wander into a link at an agent that contains other Mobile Nodes or may simply find itself in an ad hoc situation with other MNs, none of which share the same IP subnet. Mobile-IP in the current draft attempts to fixup some of the link-layer toplogical glitches via fiddling with ARP. For example, at a Home Agent, the MN should issue a gratuitious ARP when it returns from aboard. The HA should also proxy ARP for the MN while it is away on business. However these mechanisms are limited and do not address all of the topological problems, some of which may be built into implementations. For example, if you have a BSD kernel and two MNs that share the same subnet are at different Foreign Agents, they will not be able to talk to each other. This is because as long as the assumption that same subnet == direct link is present (due to a routing table entry (BSD) or worse built into the code (brand X)), then one MN will attempt to ARP for its peer as opposed to using Mobile-IP (which would work). In our current routing system, the MNs (shared subnet, but at different FAs) would simply use the default route (to the agents), and thus can take advantage of Mobile-IP in order to find each other.

In order to implement our current version of link-level only ad hoc routing, we had to change some fundamental kernel assumptions. Kernel modifications themselves were small but profound. First we modified the operating system so that ARP could be turned off on a per interface basis. This is done with an ifconfig -arp flag. Ironically the flag existed in the BSD kernel, but did nothing. We taught the ARP code in the kernel (which is actually fundamental to driver use since the ARP code is always used for mapping IP addresses to MAC addresses) to ignore incoming ARP packets and not use ARP on output. The ARP cache mechanism is still maintained for mapping IP to MAC addresses. However in the case of the ad hoc system, in general, (but see below for an important tie-in between Mobile-IP and this routing system at agents), an authenticated beacon will cause the Mobile-IP routing daemon (MN or agent daemon) to insert a IP/Mac mapping. Unlike with ARP, the daemon tracks the routing entries and times them out. This gives us link-layer routes (IP to MAC as gateway). It also gives us daemon control over link-layer routes as opposed to having them magically appear just because an ARP was done due to a packet sent to a local subnet IP address.

A more subtle and invidious hazard had to be negotiated too in terms of how link-layer routes were inserted. The ARP mechanism in 4.4BSD is tied to a particular per interface route table entry that has the flags UC. UC here means "user" and "clone". We call this the "clone route". The clone route is of type LLINFO; i.e., it is a link layer route and its gateway may be a MAC address. Its destination is the subnet ip address. The clone route is installed when an interface is configured with ifconfig and an IP address and subnet mask is bound to the interface. The clone route itself has an empty gateway, which represents the link layer part. When ip output tries to send a packet to an ip destination that matches the clone route in a routing table lookup, a link-layer route is created (cloned) for that ip destination. ARP will go off and fill the MAC address in. Thus the clone route causes link layer routes to same subnet destinations to magically appear in the routing table. On the input side, packets coming in from a subnet/link peer will cause ARP to allocate via cloning a link-layer route for the peer. Thus a routing table entry is used to implement the subnet/ARP idea.

This semantic is not exactly hard-wired into the BSD implementation. We choose to rework our Mobile-IP routing mechanism for adding default routes and link-layer routes so that they did not depend on having the clone route present. Thus if we choose to use our authentication/ad hoc mechanism, our daemons on a per interface basis remove the clone route (and turn ARP off on the interface). With no clone route present in the radix tree, and with a link-layer route (dst=IP/gateway=MAC) installed due to an authenticated beacon, packets will be forwarded locally when a system is actually local. If it is not, packets will be sent to the default and Mobile-IP will come into play for Mobile Nodes (or not) that are elsewhere.

We expect that agents will be configured at boot with ARP on their ethernet side and with ad hoc on their wireless side. MNs are capable of being configured dynamically via our X user interface and can change between "modes" (routing state must be flushed). Our user interface will show the user whether or not beacons are being received that are authenticated, incorrectly authenticated, or lack authentication (presumably you are in ARP land). The user may then decide to dynamically change the link mechanism.

However the system itself does not currently change the mechanism, which is a conservative approach. We simply want to use the user interface to better inform the user about toplogy and security aspects.

The traditional ugly question of scalability should be raised. Some have denigrated ES-IS in the past and promoted ARP, but if you can't make the subnet "net==link" assumption it is hard to see how you can ask on demand as opposed to simply be told somehow. Beacons are one way to be told. We also must keep the lack of transitivity with radio links in mind.

So let us accept for the moment that beaconing is OK, but is less scalable. How can we make it more scalable? We observe that Mobile Nodes do not need to beacon as much as agents. So far we have only done minimal testing (more follows this quarter) but we are considering setting the MN beacon rate to be 5 to 10 times slower than that of agents. We feel that MNs for the most part need to correspond with the agent, and MN to MN interaction may exist but is rare. As a result, our implementation sends a beacon out in front of the MN UDP registration messages (current rate at stable state is about 1/3 of the agent lifetime, which is 1/3 of 2 minutes, hence one reregistration every 40 seconds). A beacon rate of once per 10 seconds is probably fine for MNs with current usage. If a group of us sits down in a conference room, we will have established connectivity by the time we get around to actually doing anything. We suggest that the beaconing rate should be tied to motion and to the rate of MN beacons received from other MNs. If an MN discovers MANY MN beacons, it may slow its own rate down. On the other hand, if it moves it might speed its rate up. Of course, the question to ask here is how does an MN tell its rate of speed? There is room for future research here.

Ad Hoc Routing Security Considerations

There are probably at least four threats that are addressed here:

1. ARP spoofing in the active sense; i.e., a spoofing system steals the IP address of another system and uses it. With traditional ARP spoofing, the spoofer only changes its IP address and retains its MAC address.

2. denial of service due to loss of a link-layer route courtesy of an ARP spoof.

3. denial of service due to unauthenticated ICMP beacons sent by a fake agent. This is more dangerous in a radio topology if handoff mechanisms (like our current handoff mechanism) are based on signal strength.

4. the possibility that a user might due to frustration decide to give away the network link key. This is why we want to emphasize and make available a key that can be easily manufactured, destroyed, and given away in short term face to face ad hoc situations. In such a scenario, MNs from different domains may be present and all may be disconnected from the Internet.

#1 and #4 are probably the most important cases that need to be considered. #2 is an outgrowth of #1. #3 is a very real possibility and should not be considered unimportant. We will not say anything more about it as our beacons are authenticated by definition.

Without explaining exactly how to do this, (although we fear the mechanism is all too obvious), we have "successfully" performed an ARP spoof at a Foreign Agent. One MN (the attacker) overwrote the ARP cache for another MN (the victim) at a Foreign Agent. The result was that the attacking MN (which assumed the IP address of the innocent MN) was able to steal both the link layer routing entry AND the tunnel between the Home Agent and the Foreign Agent. As a result, without running Mobile IP, the attacking system was able to use the spoofee's Mobile IP setup. The only Mobile-IP attribute that the attacking system needed was the ability to bind the default route to a agent not on its subnet. Of course, at a Home Agent the attacker would not need that ability, but at HOME, the scenario is exactly equivalent to traditional ARP spoofing. On a radio link, it is further possible that an ARP override might not be detectable by the victim, due to the non-transitive topology we discussed earlier in this report. Possibly, something even worse might be accomplished, as there might be a way for an attacker to feed the Mobile-IP UDP reply packets returning from the HA back to the victim and thus keep the Mobile-IP connection fresh.

In summary we don't believe that ARP is a very good foundation for Mobile-IP, especially on radio links where a radio cell might easily extend outside buildings, even to passing vehicles. Mobile-IP has decent mechanisms for authentication of its UDP control messages including replay prevention. However we hijacked an authenticated MN/HA UDP Mobile-IP connection. Further if there had been authentication between the Mobile Node and the Foreign Agent, it would not have mattered. Mobile-IP authentication mechanisms only apply currently to its UDP control packets (registration/reply). We might judge the hijacking of such a Mobile-IP connection to be a more serious breach than a traditional ARP spoof, since we highjacked an authenticated tunnel.

One might think that ARP over ethernet is fine since ethernet links are (maybe) more subject to managerial controls than radio links. Whether this is an illusion or not is something we are not going to discuss here. However it may very well be a dangerous illusion and there is nothing to prevent our authenticated beacons from being used on ethernet too. IPSEC can be used to authenticate and encrypt ordinary data traffic, or applications like ssh/slogin might be used for authenticated and encrypted TCP sessions. However neither Mobile-IP or IPSEC address ARP as a protocol. Use of IPSEC might reduce such a highjacking to simple denial of service (we hope). However our mechanism would prevent such a denial of service attack (loss of link-layer routing) or at least make it more difficult.

The use of authenticated MAC and IP addresses changes the playing field for spoofing. An attacking system must now spoof both the MAC address and IP address. Certainly some lan controllers have programmable MAC addresses and it is possible to adopt some other system's MAC address. Note however that since the attacker intends to use the MAC address of the victim, the victim might see returning packets (or all packets) due to the shared unicast MAC address. For example, think again about our radio setup with the three systems, A, B (FA), and C. If A spoofs C, C might not be able to see A's packets. However packets returning from B to A would be seen by C. C therefore has an opportunity that it would not have had with a simple ARP/IP spoof, where the attacker's MAC address is different. If A starts a TCP connection, C would issue a TCP reset against it. Therefore what is needed here is that C should be "alert" and notice anything peculiar going on at the link. It should be able to see spoofing. What is needed is a user interface that informs the user about network stack security events of interest, TCP resets, UDP ICMP no such port errors and the like.

On the other hand, an attacker might simply be passive, but if it is passive it isn't gaining anything that it can't do by analysis of local traffic without going to the trouble of capturing packets and reprogramming its ethernet controller. IPSEC addresses such a threat.

IPSEC provides network and transport authentication, encryption, issues and we feel that this ad hoc mechanism is complimentary with IPSEC. An authenticated ad hoc link-layer routing protocol coupled with IPSEC mechanisms between MN and HA may be said to form the "moat" and "castle wall" of a more secure roaming node.

One interesting question remains and that is what if anything should be done about possible replay attacks? What if an attacker soaks up your link-layer authenticated beacon at one FA and goes to another and plays it back? In this case, the victim would not be present to notice what is going on. Possibly replay attacks are a higher level problem and we should not worry about them here. After all, all you gain is a route in either a Mobile Node or agent. However, we have ruled replay attacks out at agents due to careful implementation. Our agents will not install the link-layer route (at either HA or FA), until the Mobile-IP UDP registration packet returns successfully. This means that an attacker must also minimally successfully breach the MN/HA Mobile-IP security relationship. Hence to some extent, we are taking advantage of Mobile-IP replay protection and authentication. The remaining question may be reduced to what, if anything should be done about MN to MN relations? Of course in that case, spoofing must be done locally. It is possible that we might use a challenge-response mechanism of some sort but we would like to get advice on this problem space from other quarters. Opinions and thoughts of any readers are appreciated and requested. For now, we observe that close attention to what is happening on the link layer (is anybody using my MAC address?) will suffice.

E.6 X-based User Interface for Mobile Routing Daemon

We feel that it is fundamentally necessary to build a GUI as a visible controller for mobile interactions, especially security interactions. Our visual display utility is called "xmnstat". We will use this tool to control and view various state and security parameters of the Mobile-IP node. For example, there are screens that show current agent status, a running histogram of available agent signal strength, and prototypes of

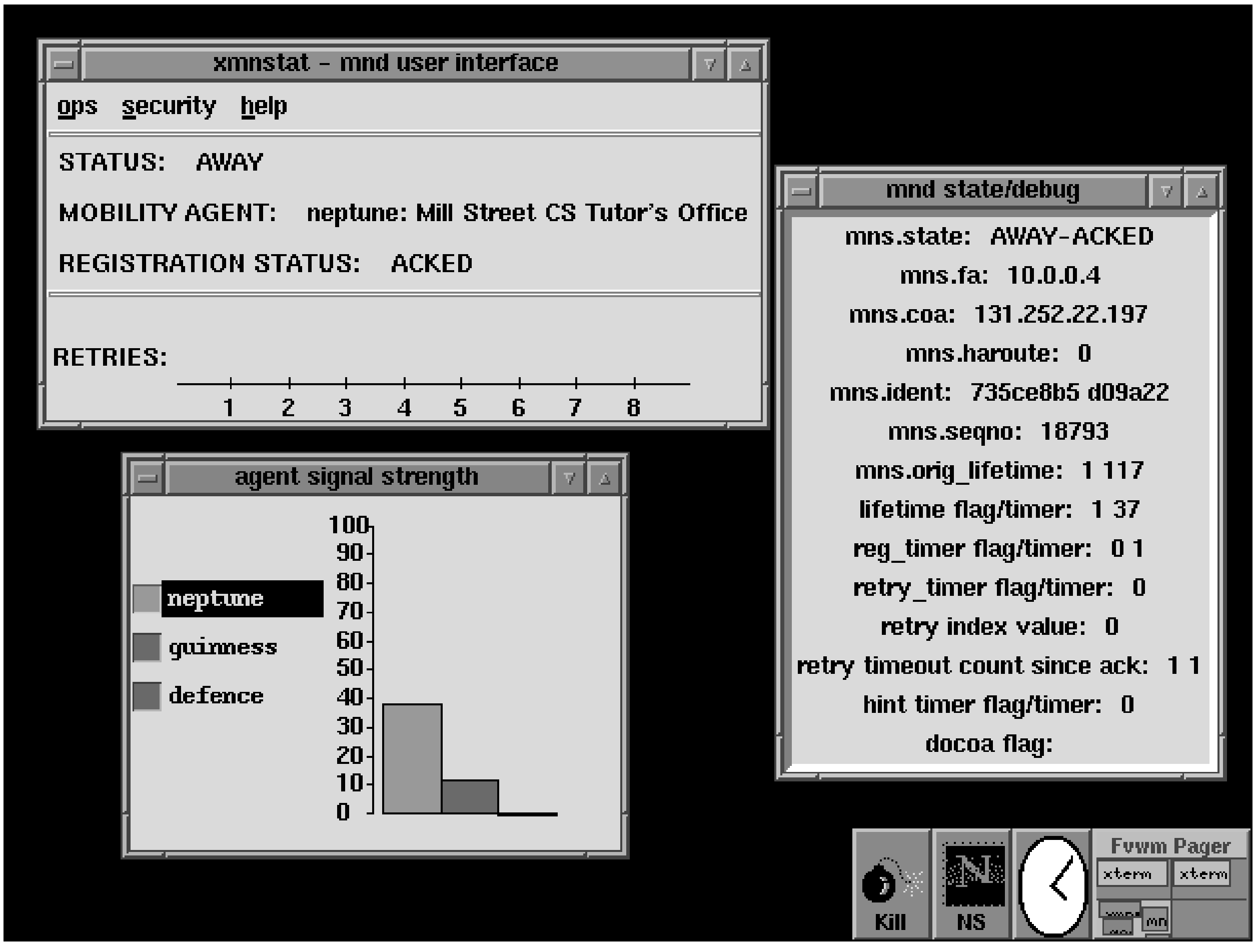

Figure E.1: Mobile-IP Status Windows

network authentication (link/ad hoc) security and (not yet, but soon) IPSEC policy. We present a couple of the screens here and will briefly discuss them. The implementation is being done in TCL by one of our graduate student RAs. In terms of security mechanisms, the ui will offer certain policy mechanisms to the user, and then if the user changes the policy, the ui uses a UNIX socket (no off machine control - we view our laptop systems as single user systems) to instruct the Mobile Node daemon, which in turn will manipulate the operating system as needed.

(The screens are available in color on our web site as the following urls:

1. http://www.cs.pdx.edu/research/SMN/index.html/summer96/status.gif
2. http://www.cs.pdx.edu/research/SMN/index.html/summer96/linkauth.gif)

Figure E.1, the "status" windows show various windows associated with normal Mobile-IP operation and is NOT a prototype. One may see the current state (ATHOME or AWAY) and information about current agent connections in either small or large amounts of detail. The latter of course is primarily useful for debugging. The most "demoable" part here is the running 5 second histogram display of agent strength. We scale Wavelan signal strength on an arbitrary 0..100 percent scale (which is not how the Wavelan driver scales it), and regard signal strength $< 10\%$ as bad. The agent strength sub-window shows that we currently can hear 3 agents, and are hooked up to neptune (FA) as it has the strongest signal strength. Connectivity to guinness and defence are weak, which makes a lot of sense given that they are in another building.

Figure E.2, the "link/ad hoc authentication" security screen is currently a rough prototype and will change. We will allow the user to do the following:

1. choose between ARP and ad hoc (authenticated beacon) modes. All state will be flushed in the MN daemon as appropriate when a mode switch is done.
2. try and supply various hints about what kind of link you have wandered into; .e.g, if we are in authenticated link mode, we increment counters (but do not store or setup routing) that will allow the user to see that there are agents out there that are not speaking authenticated beacons. Thus the user may decide what the policy will be for this link. To begin with, information about agents "in the other mode" will take more of a "dumb lights on the dashboard" approach; i.e., we will show the existance of entities in the other mode, but not provide a nice list of them. We may (as time permits) attempt to enhance the user interface in this regard and show a list of agents that we can't talk to.
3. If the user is using authenticated beacons, the user may make certain simple policy choices about those beacons as we have two possible keys in place. The only real functionality here is focused on the secondary optional ADHOC network key. The user may choose to accept/reject beacons with such a key. By default, such beacons are rejected. The user can put a ad hoc key in place, start sending beacons with the ADHOC MAC hash, and at the conclusion of a meeting, disable the ADHOC interface. We can also generate such a key for face to face meetings and distribute it (you can assume the distribution mechanism is out of band BUT in point of fact we have a simple user interface that feeds a passphrase to a MAC algorithm and generates such a key. This is a mere feature). This mechanism may be of use in disconnected operational situations.

The "ask" buttons in the figure should be ignored. They are going to disappear.

E.7 Mobile IP and IPSEC

We are going to be brief here as this will constitute the bulk of our work next quarter (fall) and our fall report will cover our work in more detail. We have ported the NRL (June) netbsd IPSEC code into freebsd and are testing it as given. As of this date, we know that the key socket works, and that we have not damaged normal non-IPSEC operation. The most difficult part of the port was that the IPSEC code is not cleanly differentiated from the IPV6 code, and we have no interest in the latter and have not compiled it into the kernel. (For example, the mods for IPV6 would also modify IPv4 code. We want to minimize IPv4 changes since it would introduce unnecessary risk and force us to completely retest the IPv4 stack). We are testing the AH-transport and ESP-tranport socket-level mechanisms.

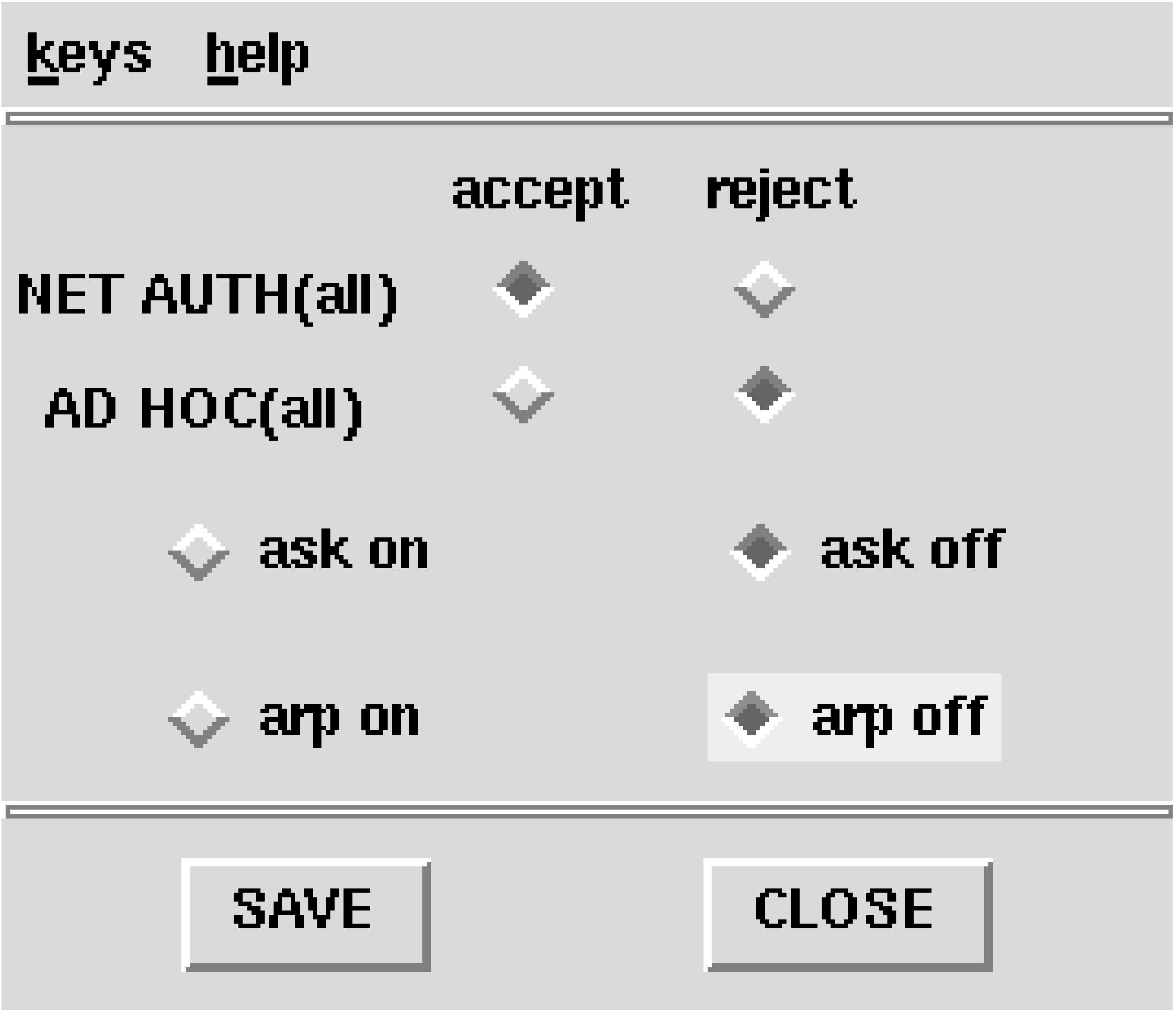

Figure E.2: Mobile-IP Link Authorization

We have decided what we want to do to support routing and how that will be combined with Mobile-IP. The work to teach the kernel to bind routes to IPSEC is now under way. We include some discussion of high-level assumptions here.

As we outlined in previous reports, we seek to support a number of Mobile-IP toplogies and these toplogies will change dynamically as a system moves from HA to FA. We will need a few extensions to the Mobile-IP UDP registration messages. Simply put, the MN will tell the HA (or a FA with which is shares a out-of-band a priori key relationship) what security it desires, e.g., none, AH, or ESP (the default). (We are assuming that NRL will eventually supply the IPSEC community with an up to date ESP that includes authentication. For now, we will use what we have been given, which is ESP according to RFC 1827). Our user interface will inform the user as to which policy choices have been made and allow a certain amount of dynamic switching. Basic topologies of interest include:

1. MN to default router (HA or FA with which a priori key relationship exists).
2. MN to HA when away (to HA and back across untrusted or any FA).
3. MN to MN (for ad hoc situations).

All route level bindings will assume an outer IPSEC security tunnel. Spi parameters, and ipsec src and dst will be supplied to the route mechanism and used to determine the a priori loaded key material in the NRL key table. It appears that we can make fairly straightforward use of the NRL key socket and kernel lookup mechanisms.

A very important general question here is: "how does IPSEC interface with the os"? A more focused question along those lines is: "how do we bend IP policy on output to combine route-based IPSEC with NRL socket-based IPSEC?".

Very roughly we intend to change ip_output as follows:

```
if (!forwarding and there is a socket ipsec binding)
        do ESP transport
        do AH transport
else
        if there is an IPSEC binding to a route
                do ESP (with the ipsec tunnel)
                do AH (with the ipsec tunnel).
```

The above is an over simplification. The first part of the if clause roughly represents what NRL does now, and the second else part represents what we will add.

The upshot of this is that socket-level IPSEC bindings which are end to end will override any route level bindings which are probably not end to end. Socket-level bindings will not be associated with tunnels. All route bindings will have a tunnel slapped on the end according to the route in question (and decided by a route daemon). For example, for a MN at home, it could choose to bind AH or ESP flags to its default route, and the packet coming out of it would have a structure as follows:

```
ipsec tunnel ip header | ESP mn/ha | ip original | transport
ip src = MN            |           |             |
ip dst = HA            |           |             |
```

We suspect we should also implement a setsockopt that would allow a socket to proclaim that it desired NO ipsec route binding be applied to it. This would allow applications like slogin which uses TCP/IDEA to avoid having a DES outer binding applied. The question of exactly how should IPSEC bind to a kernel is very interesting and the above represents an incremental improvement but is by no means a final answer. We do not state that it would be wrong to have both AH/ESP due to routes piled on top of AH/ESP due to socket, but we do not want to implement it or test it. Possibly IPSEC mechanisms in the kernel should be tied to firewall mechanisms as well and/or to ports in a general sense. There is room for further research here.

E.8 Redundancy

In this section, we are going to briefly discuss some preliminary redundancy work we did this summer (including one curious and bizarre radio link situation we ran into), and also present a few principles of how we think HA and FA redundancy should work. Our thinking along those lines has crystalized over the summer.

Of course, the authenticated ad hoc beaconing mechanism proposed and implemented above is at heart a routing redundancy mechanism. We choose to not use subnetting and ARP. MNs can talk to each other without routers present. Certain (bizarre) problems with Mobile-IP and ARP that are primarily due to subnet assumptions built into routing tables are cleanly solved as well. At some point, we would like to build a higher-level routing protocol that would use signal strength as a metric, and would allow > 1 hop host to host routing based on an IPSEC tunnel (say ESP). We need to consider how such a mechanism would both interface with our link-layer routing and interface with the kernel routing table.

We enhanced our Mobile-IP handoff algorithm so that in the case of the Wavelan interface only, instead of basing handoff on agent starvation, handoff is now based on best signal strength with a bit of hysteresis. The PCMCIA Wavelan driver keeps a signal strength cache indexed by MAC address. Each cache entry consists of (MAC, ip, various Wavelan specific parameters including raw signal strength). The cache is only updated when a broadcast or multicast IP packet is received, thus only beacons update the cache. The mobile node daemon can read this information and it sorts its possible list of agents by signal strength. The algorithm is more complex than we will present here, but in summary it can be said to consist of a running sum of N samples where N is set to the last 10..15 beacons. Thus our handoff is not fast, but it is fast enough for current uses. We can walk around our square building with agents in the corners and watch handoff occur cleanly as we walk by. More important, if a FA (or the HA's Wavelan interface) goes down, an agent switch does not take long. Thus we can cleanly overlap agent cells and deal with the loss of an agent's Wavelan interface. This list will also be useful later on for FA redundancy work since we will choose to use the top N (likely 2) agents for simultaneous registration.

We also implemented a redundancy modification to the UDP registration cycle. When an MN attempts to send a registration packet to the HA, it enters a registration cycle that consists of N retries, with a backoff mechanism built in. When we are in the presence of multiple agents (FAs), we short-circuit the registration retries (say halfway). Thus if registration via one FA does not seem to be successful, we try another FA as long as its signal strength is above the low-water signal strength mark (10%). The FA currently deemed "useless" will be retried later on (a few minutes) since its signal strength is higher. This allows us to attempt to use a possibly different end to end (MN to HA) route and will effectively deal with a nearby and short-term routing partition.

When we began to test the improved handoff algorithm, we found a very bizarre short-term situation that would manifest itself in our building with the 3 overlapping agent cells (2 FAs, 1 HA). For a period of time, typically 5-15 seconds, agent signal strengths would mysteriously rise and fall so that even for static MNs, an agent within a yard could lose its signal strength, and a normally weak agent at the other end of the building would appear stronger. There are only two possibilities at this point: 1. either we have a bug in the Wavelan driver in terms of how the information is cached (but we doubt it)., or 2. this is a real phenomenon (not reported elsewhere as far as we know) and might be due to multipath or some as yet unknown radio interference. We adjusted our hysteresis mechanism to compensate for this.

We believe that our current work has nicely aimed us towards HA and FA redundancy mechanisms. We already have some capabilities in the latter direction since we can switch between available overlapping FAs now, if one should go away. Our ad hoc routing mechanism will also help us deal with certain partitioning problems inherent in subnetting. This will actually have applicability to HA redundancy since if we have two HAs on different IP networks (as we desire), the HA subnet itself may be partitioned (physically split to two places) and that is toplogical case we feel we can now deal with.

We intend to modify the fundamental BSD radix routing mechanism so that a gateway can logically be list of gateways; i.e., one destination can have multiple gateways. In point of fact, we will probably limit this to two. This is a profound change. The ip output function would take a packet and cycle through the list of gateways sending a packet to each gateway in turn. Of course, this means we are trading redundancy for bandwidth. This mechanism will be used both at HAs and MNs since a HA might have two COAs for two different FAs and an MN might have two different FAs associated with its default route. An interesting variation on the latter idea would be to bind a multicast address to the default route and thus multicast default route packets to agents.

We believe that HA redundancy should have the following attributes:

1. at least two HAs should be supported and they should ideally be on different subnets so that the failure of one router does not eliminate both HAs. This can be done with conventional unicast IP routing. However it means that any packet from a peer host to an MN might go to either HA at any time according to unicast IP routing. It should also be possible if desired to put both HAs on the same link.

2. We believe the HAs should have a protocol (say based on TCP so that the messages are reliable) so that they can keep each other informed about MN location. They need to tell each other about MNs at HOME or AWAY. If an MN is at HOME, the peer HA must be told so that it can setup a tunnel with the other HA as the COA. Note that there are at least 2 HOMEs. If an MN is AWAY, both HAs must setup tunnel routes with the FA/s in question as COAs since packets may be header towards either HA at any time.

3. The MN will have a list of HAs, and will not try to use them both at the same time, but will merely switch from one to another when it deems that HA #1 is no longer available; i.e., UDP replies do not show up during one complete registration cycle. We choose to have the HAs communicate MN state between themselves as opposed to having the MN keep both HAs up to date as this would minimize potential cross-internal traffic. In general HAs will be within an enterprise.

We hope to begin this work when we have completed the IPSEC/Mobile-IP work.

E.9 Visit to BBN

John McHugh and Jim Binkley on Sept 27 1996 made a one day visit to BBN in Cambridge in order to compare project directions with John Zao of BBN. We met with John Zao, Joshua Gehm, and Steve Kent. Jim Binkley and John Zao made overviews about their respective Mobile projects. BBN is focused on key management issues, i.e., how to tie certificates and session keys into the Mobile-IP protocol, e.g., between a Mobile Node and its HA. They are also interested in the notion of hierarchical Foreign Agents and how certificate exchange between Mobile Node and Home Agent might be minimized. PSU is interested in fundamental integration of IPSEC and Mobile-IP, and redundancy and survivability issues including ad hoc routing, toleration of multiple agents and the elimination of ARP spoofing. The one point in common we have is the desire to use NRL's IPSEC code. In that regard, we (PSU) are currently ahead of BBN as they have apparently focused on their certificate work first. We would be happy to share any work with them along these lines. We, in turn, will adopt BBN results in the areas of certificate management and Fortezza integration.

E.10 Plans for Fall Quarter

In brief, our plans are as follows:

1. the primary focus is the design, implementation, and testing of binding our Mobile-IP to NRL's IPSEC. We need to make o.s. changes to FreeBSD so that routes can be bound to their key table, change our Mobile-IP daemons to install those routes, etc.

2. write a test plan and test the authenticated ad hoc mechanism, and also get the X user interface policy windows fleshed out and real so that it actually controls the MN's behavior.

3. commence formal modeling of the implemented security protocols.

4. develop a cumulative acronym list for the quarterly reports as requested by RADC.

5. begin integration of the Fortezza cards with the system.

6. occupy new laboratory space.

In general, we hope to get our fundamental security work done this quarter, and begin HA/FA (and further ad hoc) redundancy efforts next quarter. At some point, possibly next quarter we should make various measurements of IPSEC transforms over Wavelan links.

Within the next few weeks, we hope to occupy laboratory and temporary office space in a building recently acquired by PSU. The building itself will become a major facility in our experimental plan as it's layout will accommodate a variety of mobile-ip configurations.

Appendix F

Quarterly report – Fall 1996

F.1 Project Status Overview

Overall we feel we have made reasonable progress this quarter.

The two most important items accomplished were deployment of the ad hoc sub-system, and first cut work on IPSEC/ESP bound to routes for Mobile-IP.

We have finished debugging, and deployed our ad hoc link-layer replacement for ARP on our Mobile wireless infrastructure at PSU. All users and almost all agents (barring one left behind for mode testing between ARP and ad hoc) are running the authenticated link-layer ad hoc system.

The IPSEC work we intended to do turned out to be too large of a bite for one quarter, and as a result we divided the project up into two parts. In the first part, which is almost complete, we have constructed a kernel route binding for IPSEC/ESP (and partially for IPSEC/AH). We wanted to make a smaller pass over a first project, so that we could gain some assurance regarding "proof of concept". The route binding has involved much kernel routing work. At this point we have statically tested the Mobile-IP topologies that we described in the last report. We are currently in the middle of modifying our Mobile-IP routing daemons (applications) to take advantage of the new kernel mechanisms. We should have a simple form of "Mobile-IP meets IPSEC" under test in a few weeks. In the second part, which we will complete this quarter, we intend to finish our kernel IPSEC/route mechanism and carry that work to our Mobile-IP daemons.

F.2 Accomplishments for Fall 1996

1. Jim Binkley has written a paper on the ad hoc protocol entitled: "Authenticated Ad Hoc Routing at the Link Layer for Mobile Systems". The paper is currently a PSU Computer Science technical report, TR 96-3. It has been submitted to a special ACM journal on mobility for possible publication.

2. We have debugged and deployed the ad hoc routing protocol. We hope to release this code soon (probably in January).

3. We have created a kernel route binding to IPSEC, including a route API in the `route(8)` and `arp(8)` commands. We have extensively tested the NRL IPSEC ipv4/socket port into FreeBSD and feel that it is sound at this point. We are in the process of teaching our MIP (Mobile-IP) daemons how to use IPSEC/routing. We will discuss this work in greater detail below.

4. David Reeder has attended Mobicom/IETF meetings and has kept the group here up to date on events at those meetings.

5. Over the course of the quarter, we have made several internal improvements to our Mobile-IP source base. These improvements were either due to bug fixes, IPSEC work, internal security improvements, or as an attempt to make the system more portable.

6. In spite of the fact that growing our Mobile-IP/wireless infrastructure is NOT a current focus, we still have deployed three more agents. We have deployed two Foreign Agents, one in the PCAT central engineering building and one at the Oregon Graduate Institute. The latter has led to some interesting "cross-routing-domain" experiments, which we will discuss briefly later in the report. We intend to put a new Foreign Agent in another engineering building next quarter (U.S. West). We also finally deployed a second "test" Home Agent, which has proved invaluable in the final stages of ad hoc testing, and is currently being used for IPSEC testing.

7. We have made some progress with redundancy theory. This work is primarily being driven by graduate students. We are in the process of completing a specification for multi-hop ad hoc routing. We have started specification work on a redundant Home Agent system (two HAs per Mobile Node subnet). We expect this work to bear fruit either by the end of winter or spring quarters.

8. Jon Inouye, a Ph.D candidate at Oregon Graduate Institute has ported our Mobile-IP into his link-layer redundancy system. Jon's Ph.D work is focused on teaching a Mobile Node how to retain network connectivity when link-layer devices are switched. For example, one might start with WaveLAN (which is now available at OGI), and switch to a PCMCIA Ethernet device. Jon has made available to us his kernel PCMCIA infrastructure work, which gives us hope of eventually being able to use a second simultaneous PCMCIA card, along with the WaveLAN card, which is a permanent fixture in our laptops.

F.3 Winter 97 Plan

In this section, we are going to briefly discuss our winter term goals and planning. Our planning is primarily personnel based; that is, person X works on project Y. We will present current people and the projects they are associated with.

F.3.1 Winter goals

In summary, we will make a couple of code releases, and continue our current focus on IPSEC/routing this quarter. We also are turning more to our redundancy work as two graduate students are making good progress in that area.

1. A Mobile-IP release. We intend to release our ad hoc version of Mobile-IP soon. We need to finish some release engineering and write some final documentation. (There is no IPSEC in this release). This release will replace our Mobile-IP release of last summer.

   Major features made available in the release include:

   (a) improved FreeBSD kernel features for Mobile-IP. These are all enhancements to the BSD routing mechanisms.
      i. use of Bill Fenner's RECVIF function, which allows routing daemons to determine the interface a packet came in on.
      ii. a revamped tunnel driver (discussed more below, however there are no IPSEC features in this version).
      iii. we fixed and revamped the mechanism our routing daemons use to tell the kernel how to bind a route to an interface. All of the above changes were useful and were almost all bug fixes as well as being part of an attempt "to get it right".

   (b) of course, the authenticated ad hoc routing mechanism is included.

   (c) We implemented per MIP entity MIP authentication. For example, all of our Mobile Nodes can now use a per MN Home Agent key. This was an important security improvement.

   (d) The release will include the fast and redundant handoff work we did last summer.

   (e) xmnstat, our visual X/tcl-based interface for the Mobile Node daemon will be included.

2. ipsec kernel. We want to release our current IPSEC/FreeBSD 2.1 kernel to the FreeBSD security community. A number of people including engineers at TIS and firewall companies have expressed interest. This release will ONLY include the kernel and various applications, including the `PF_KEY key(8)` utility, test applications developed by us, and our first cut ESP/route binding. It will also of course include the original NRL IPSEC/transport mechanisms including the crypto algorithms of last summer. For example, AH includes md5, hmac-md5, sha, hmac-sha, and "dummy" (the NULL algorithm). It will not include Mobile-IP daemons, since that work is not complete. The kernel will however contain our Mobile-IP routing modifications. This is important as our IPIP tunnel driver, originally developed for Mobile-IP has become a central part of our IPSEC work. We would have liked to have gotten this our earlier, but the ITAR-specific code has been held up in transit elsewhere. It would be extremely nice to have other FreeBSD people assisting us in this work.

3. We intend to finish our first cut IPSEC/Mobile-IP route binding that uses only ESP/DES in various routing topologies. We have tested transport IPSEC for all the AH and ESP transforms. We have implemented a first cut routing API. We are in the process of taking advantage of our current kernel mechanism dynamically in our Mobile-IP daemons. We intend to finish the routing API (we know what we want at this point) and carry the new mechanism (primarily AH) into our Mobile-IP daemons. We should be able to finish this work this quarter. We also hope to do a bit of tunnel testing, since we think that AH may be useful in tunnels between Home Agents/Foreign Agents/and Mobile Nodes.

4. We will finish our multi-hop ad hoc specification and redundant Home Agent specification. We may be able to get the HA mechanism implemented and under test, although we will need to do some work to setup a reasonable testbed for redundant HA testing. It is likely that multi-hop ad hoc coding and test will stretch into spring quarter as that work is both theoretically complex and difficult to test.

F.3.2 Optional goals

1. as a group we hope to write a technical report on our Mobile-IP/routing security work. This paper will focus on the route/IPSEC binding.

2. we intend to do a little more investigation of CERT/firewall/Mobile-IP ingress filtering problems. We can do this because of the Foreign Agent we have deployed at OGI. OGI is outside of the NorthWestNet border router. As a result, PSU-based IP subnet addresses are filtered out when packets from a PSU laptop at OGI try to cross into the PSU domain. (The irony level here is high. When at OGI, we can talk to any Internet system except our own).

F.4 Personnel

Jim Binkley acts as project manager. He supervises David Reeder, and the graduate students. Jim is usually involved in design, code release, review, specification review, and kernel work.

John McHugh has recently resigned as Computer Science Department chair which will allow him to focus more on DARPA research. John will eventually drive Fortezza work in this project (See the Fortezza note below).

David Reeder is our full-time employee. David worked last quarter on IPSEC kernel routing issues and is currently completing the Mobile-IP application/daemon work in that area. In winter quarter, David will finish the IPSEC/routing API, rework the routing daemons in IPSEC phase #2. When this work is complete, David will begin work on ISAKMP and how it might be tied into Mobile-IP.

Xu Hao is a M.S. graduate student who will graduate in winter quarter. Last summer he worked on the first cut ad hoc protocol, and during fall (and winter), he has been working on the specification of our multi-hop ad hoc protocol.

Bjorn Chambless is a M.S. graduate student who has just started his M.S. work as of last summer. He has completed xmnstat for us and has recently started work on Home Agent redundancy.

Jennie Ye is a new M.S. student and has just joined the project this quarter. She will probably spend most of this quarter coming up to speed. We intend to deploy Jennie to work on redundancy as soon as we can. We need someone to carry on Xu Hao's work and also to begin work on Foreign Agent redundancy.

Mathew Mead is currently a Teaching Assistant, but we intend to hire him in spring quarter to replace Xu Hao. Both Mathew and Jennie are currently taking a reading class from Jim Binkley that focuses on TCP/IP, network security, and Mobile-IP. We intend to have Mathew work under John McHugh's direction in spring on Fortezza.

F.5 Fortezza-note

Due to lack of time, people, and a kernel infrastructure stumbling block, we have not yet been able to do anything with Fortezza cards. Of course, we have just gotten an IPSEC infrastructure in place and tested in the last two quarters. Presumably Fortezza would somehow fit into that infrastructure. A major kernel infrastructure problem has been that our current FreeBSD 2.1 kernel does not support more than one PCMCIA driver at a time. By definition, to use Fortezza, we feel we must use our WaveLAN PCMCIA cards simultaneously. We originally intended to wait for FreeBSD 2.2 in the hopes that it might contain a general PCMCIA kernel infrastructure. However, due to Jon Inouye's recent work, we now have a source of 2.1 kernel code that includes our Mobile-IP routing infrastructure and includes a reasonable PCMCIA infrastructure. We hope to be able to "digest" that code into our current source base over winter quarter. If we cannot, it still gives Mathew Mead and John McHugh a starting base in spring. It would seem that our OGI association has born fruit.

F.6 IPSEC && Mobile-IP

In this section, we are going to discuss our major focus of the quarter, in four sub-sections:

1. IPSEC-Mobile-IP architecture.
2. The MVIF tunnel driver.
3. OGI/Firewall experimentation.
4. Conclusions and Observations.

This work is still under progress and will continue into this quarter and possibly beyond (if we include ISAKMP. We expect to talk to the DARPA project manager about ISAKMP when she visits us in January).

We decided to break the IPSEC work up into two phases in order to test our most basic routing notions as soon as possible. The first phase consists of a complete Mobile-IP implementation that would dynamically use IPSEC-based ESP/DES routes only. The Mobile-IP focus in the first phase is primarily on routing topologies ONLY between a Mobile Node and a Home Agent. The second phase would finish the routing work and include AH, AH/ESP in combination, possibly a statically deployed FA/MN relationship (we are not sure yet if that is really useful).

We can divide the first-phase project into the following phases:

1. initial port of NRL/IPSEC to FreeBSD 2.1 (done last summer by Jim Binkley).
2. development of test code for testing (and understanding) the NRL transport-based IPSEC code. (done this quarter by a graduate student and Jim Binkley).
3. testing of NRL/transport with all algorithms AH and ESP (done this quarter).
4. definition and implementation of a route/based API for the BSD kernel for binding a NRL security association (SA) in the kernel to a BSD route table entry. (started this quarter by David Reeder/Jim Binkley. We will finish this early in the winter quarter).
5. static testing of route API. (David Reeder. done).

6. modification of MIP tunnel driver to support IPSEC/routing. Jim Binkley. (done)

7. static testing and debug of various (3) route topologies of interest to Mobile-IP by Jim Binkley (done).

8. modifications of Mobile-IP daemon applications to dynamically use IPSEC/routing (primarily HA/MN away or at home). David is currently doing this work.

9. modification of xmnstat to support simply IPSEC policy choices. To be done winter quarter by Bjorn Chambliss.

10. internal deployment and test.

In our second phase, we will finish the kernel/IPSEC routing work, and teach our Mobile-IP implementation how to do AH and AH/ESP simultaneously. We also need to decide what if anything to do about the FA/MN relationship. We are of mixed minds about that problem. Something useful can probably be done with static keys and FAs that belong to a given organization's routing domain. In particular, an authenticated tunnel between Home Agents and Foreign Agents may be interesting. It may be useful for a given administrative domain to setup all of its tunnels with authentication (in order to protect itself from the outside world). On the other hand, interaction between a Mobile Node and another domain's Foreign Agent in terms of setting up secure routes requires dynamic negotiation of authentication via certificates. An interior domain might of course with to use certificates for authentication too. We wish to point out that authentication should be considered not only for the current MIP relationships but also for tunnel endpoints.

F.7 IPSEC-Mobile-IP architecture

In this section, we will try to briefly explain the gist of our Mobile-IP/IPSEC architecture. The goals for the architecture may be summarized as follows:

1. route-based IPSEC. Our Mobile-IP implementation will bind IPSEC/ESP/DES (and eventually AH with various transforms) to routes; i.e., the security is done with tunnels at the network layer between entities such as Mobile Nodes and Home Agents.

2. We are for now only using static key distribution. For example, we assume that both Mobile-IP and IPSEC secret keys are installed at an MN and a HA before the MN "leaves home".

3. Our Mobile Node daemon application dynamically requests, and sets up secure (or insecure) routes in a number of routing topologies according to Mobile-IP topological information (secure enclaves). All routes are always two-way in terms of security. The MN to HA route and HA to MN route either both use IPSEC (or do not).

4. A MN user may dynamically or statically specify IPSEC security in terms of a simple policy interface. As a consequence, routes are installed dynamically and bound to pre-installed key material. This is done with the xmnstat controller (and may be done by rebooting mnd and changing its boot-time config file as well). For example, in our first phase, a user will be able to specify two modes of security, "none" or "esp/des" to a HA, either at home (link connection to HA as default router) or "away" (two-way security tunnels between the MN and HA over the FA without the FA knowing about it). In summary, the MN (and hence the user) will control whether ESP tunnels or no tunnels are setup to the HA.

F.8 BSD(4.4) IPSEC/routing API

It is probably easiest to simply present a routing API in terms of the switches/flags we currently use with the `route(8)` command. We have 4 possible IPSEC tuples that may ALL be used together in a few rare cases or more likely used as ones or twos in normal routing cases:

1. ESP on a normal route
   `# route add <destination> <gateway> -esp -spi N -itsrc <ip src> -itdst <ip dst>`

2. AH on a normal route
   `# route add <destination> <gateway> -ah -spi N -itsrc <ip src> -itdst <ip dst>`

3. ESP at the front of a tunnel route packet (special case)
   `# route add -tunnel <destination> <COA> -esptunnel -spi N -itsrc <ip src> -itdst <ip dst>`

4. AH at the front of a tunnel route packet (special case)
   `# route add -tunnel <destination> <COA> -ahtunnel -spi N -itsrc <ip src> -itdst <ip dst>`

The first two route bindings may be used with most normal routes. They are of generic interest in terms of routes and Virtual Private Networks. The latter two are only for Mobile-IP and apply to the virtual tunnel device that is typically used at the HA to tunnel to the FA.

We assume that the SA is loaded via `key(8)` and it specifies a Security Association in terms of (ah — esp, spi, IP src address and IP dst address). The security associations have been loaded via the `PF_KEY` socket by `key(8)`.

Note that each route type above specifies a 4-tuple; for example, (esp, spi, ip src, ip dst). When a route is "added", the routing code makes sure that it can find the SA and binds itself to it in the kernel. The 4-tuple is used as an SA signature. In general, the order of operations for routes and keys is:

1. add a key via `PF_KEY` and `key(8)`

2. add a route and bind to the key

3. delete the route

4. delete the key

On output, the IP layer simply uses the SA route pointer to do the IPSEC transformation.

It is possible that a given route might have more than one of the bindings associated with it. E.g., a given route might have both ESP and AH bound to it. In that case, AH would be performed first in ip output before ESP.

The `-itsrc` and `-itdst` switches are of special interest. The idea is relatively simple. These addresses must match the ip src and ip dst addresses specified in the /etc/keys security association. They represent an IP tunnel from this host to the destination. We assume that all routes are IP tunnels and that this relationship always holds. The semantics here are very interesting as routes become essentially double jointed with 4 IP hosts involved. The itdst and gateway may or may not be co-located. If not co-located, the tunnel endpoint obviously goes over the routing destination, which will simply forward the packets to the itdst router and is not involved in the security association. We might draw the following picture:

```
host(itsrc) ----> gateway ----> itdst ---> destination
     ========================> (security tunnel)
```

Routing is by definition one-way. Packets are forwarded for the destination to the next hop gateway. The security tunnel however is between the itsrc and itdst. On output, IP will (for example) take the IP datagram that reached it and append an IP — ESP header pair. So if we have say a UDP packet, we would have the following output for route binding #1:

IP outer — ESP — IP inner — UDP — data payload

We call this an "ipsec tunnel". It may take the following forms:

1. IP — AH — IP,

2. IP — ESP — IP, or

3. IP — AH — ESP — IP.

The `-ipsrc` and `-ipdst` values are used in the IP outer header as ip src and ip dst. ESP's next protocol is 4; i.e., IPIP. On arrival at the ipsec tunnel destination, the packet is fed first to ESP and if decryption works, it is then handed to our mvif tunnel driver for IPIP processing. ESP essentially removes itself from the header and hands the packet to the IPIP mvif driver. The mvif driver removes the outer IP header and forwards the inner datagram to `ip_output`; i.e., it forwards (routes) the packet.

The first two kinds of bindings MAY be bound to tunnel routes as well. For example, we might use binding #1 with ESP on a tunnel route (at a HA to a FA),

```
# route add -tunnel <destination> <gateway> -esp -spi X -itsrc <S> -itdst <D>
```

This route allows the HA to "ipsec tunnel" packets to the MN; i.e., send packets using DES to the MN where the security association is only between the HA and MN. This is probably the most important routing mechanism trick for Mobile-IP.

On the other hand, there might be a security association at the HA in terms of mechanism between it and the FA that the MN currently is visiting. And we might also have the HA/MN security association in place as well. In this case, the latter two route bindings (#3, and #4) could be applied as variations that ONLY apply to MIP tunnel routes (routes bound to mvif). As a result, one might produce the following example packet:

```
ip outer       |  AH     | ip inner      | ESP     | real IP datagram
ip src = HA    |  HA/FA  | ip src = HA   | HA/MN   |
ip dst = FA    |         | ip dst = MN   |         |
```

Assume we have bound both `-ahtunnel` (#4) and `-esp` (#1) to a tunnel route. The ip destination for the tunnel route is the MN. The ip gateway is the FA's "Care-Of Address". On output, first ip output would append an "ipsec tunnel" between the HA and MN. This would prepend the "ip inner" and ESP portions. Then the tunnel driver itself would note that the `-ahtunnel` association existed and instead of performing its normal operation (appending just an IPIP outer IP header), it would append the IP header AND perform an AH operation over the packet. In this case, all four bindings could be performed on one packet. If this router was also an end system, 6 IPSEC headers could be inserted if we add in two transport headers.

F.9 BSD route types and the IPSEC route binding

It should be noted that routes in a routing table perform different functions and their semantics are complex. Our Mobile Nodes typically set their default routes to an agent, HA or FA. They must also set a "link-layer" or ARP route to point to the agent as well. (In 4.4 BSD, the arp table has been merged with the traditional routing table, and "arp" routes simply have MAC addresses instead of IP addresses for the gateway portion). Our Home Agents set link-layer routes to MNs and tunnel routes for MNs that are "AWAY" at Foreign Agents. A Foreign Agent will set a link-layer route for a MN that is resident on a given FA link. In addition to default routes, and link-layer routes, there are other kinds of routes. We can distinguish at least the following kinds of routes in the BSD routing table:

1. default routes. The default route has a cloning feature and destination addresses that match it will cause the insertion of a specific host route that will time out if not used for 20 minutes.
2. link-layer routes (flag=LLINFO, gateway is MAC address)
3. per interface clone routes (typically used with ARP and closely tied to IP subnet semantics).
4. host routes (destination is a specific host) and the gateway is a next hop router.
5. net routes (destination is a specific network).
6. tunnel routes (new from us via the mvif virtual tunnel device - the tunnel driver simply appends an IP header and hands the packet to `ip_output` to be routed again).
7. multicast clone route - the multicast clone route is associated with a given interface. An attempt to multicast sans interface information, will cause a specific multicast route to appear in the routing table.

It should be pointed out that we have also modified the `arp(8)` command so that ESP and AH bindings may be associated with LLINFO (link-layer) routes. `arp(8)` and `route(8)` both use the BSD `route(4)` interface internally and are not dissimilar. We will not document the arp switches here at the present time. However this gives us needed mechanism so that a HA can "tunnel" with an ESP/route binding to a MN, when the MN is at home.

Our route bindings may potentially be applied to all of the above routing types. The semantics of the above set of "route types" are complex and we will not claim yet that all of the interactions between them and our bindings are clear or understood.

The "ipsec tunnel" when applied to the default route causes IPSEC security to be inserted, but does not apply to the default route gateway itself UNLESS of course the itdst is that of the gateway. In that case, the gateway's link-layer route must also have the same SA applied to it. Thus it is possible to tunnel over the next hop router or to tunnel directly to it.

One of the routing flags possible is called a CLONE flag (actually two, but never mind). Typically the cloning flag causes a more specific host route to be dynamically inserted in the routing table. E.g., a small 'little c' clone flag is always associated with the default route. If a specific ip destination, matches the default route, a host route is inserted in the routing table with a timeout. (These may be seen with `netstat -ran`. The clone routes will have a W flag associated with them.) We puzzled over what to do about clone routes in that particular case and decided the only correct response was for a clone to inherit the SA. If a user decides to use ESP to tunnel over a FA (default) to a HA. The user would want specific clones to carry over that attribute. There are other cases where cloning is used however, and in those cases we are not sure about proper behavior.

There are still however a number of semantic TBDs (To Be Determineds) that are lurking here. E.g.,

1. multicast routes. We have not had time to even play with these yet or think about it much.
2. per interface clone routes as used with ARP. In our link-layer ad hoc mode, we remove them. But a IPSEC binding might be used for some sort of defensive measure here.
3. cloning caused by IP redirects. A security association between one router cannot be blindly inherited by another router. We should make sure that SAs are not inherited by dynamic redirects. (More fundamentally, we need to have a per interface way to turn dynamic redirects off).

The semantics associated with the kernel routing table are NOT simple. We are still feeling our way here.

F.10 Static Routing Topologies

We have tested three different MIP-related routing topologies "by hand" at this point using primarily the ESP/binding. The topologies are:

1. MN to HA at home. The HA is the MN's default router. Both the default route and link-layer MN/HA routes at the MN point to the HA (gateway==itdst). The HA sets up a link-layer ESP tunnel to the MN.
2. MN to HA over FA. Since we assume tunnels are two-way, we of course setup a tunnel from the MN to the HA and from the HA to the MN. We will look at these tunnel types more below as they are extremely curious. Note however that the FA is not part of the equation. We tunnel "over" it. The MN sets the FA as its gateway and sets up a link-layer route to the FA as well. The itdst for the default route is set to the HA. The HA binds a `-esp` binding to the tunnel route with the `-itdst` set to the MN. Thus the routes are complimentary.
3. MN to MN. Given that we have a link-layer routing authentication protocol (ad hoc), it is reasonable to try and automatically setup secure routes between two MNs on the same link. Both MNs must insert link-layer routes to each other that for example, might be bound to ESP.

The most curious trick here is the MN to HA over FA topology. The MN to HA route is not so strange. We can simply set the gateway to the FA, and the itdst to the HA. Packets emitted from the MN to the default route will have an outer IP header where the ip dst is set to the HA, and ESP would be the next header.

As we have pointed out before, however the HA to MN packet is very interesting. If we assume no security association between the HA and FA, we have:

```
ip outer: | ip inner:     | ESP    |  ip datagram
ip src = HA              | ip src = HA   | HA/MN  |
ip dst = FA (COA)        | ip dst = MN   |        |
```

The packet has three ip headers in it. We might call it a "double tunnel". The innermost header is the original ip datagram, where the src may be any system. The destination of course, is the MN. Since we associated `-esp` with the tunnel route itself, ip output will append the "ip inner" header and the ESP header first to the original datagram. Here the `-itdst` == MN and `-itsrc` is the HA. That gives us the second IP header. The third one is appended naturally by the tunnel driver itself as a natural consequence of its operation. Please note again that it presents another opportunity for insertion of IPSEC as we could bind IPSEC between the HA and FA.

When the packet arrives at the FA, the outer IP header will be stripped and the inner part will be delivered to the MN (since the inner IP header has the MN ip address for its dst). When the packet arrives at the MN, it will process the inner IP and ESP headers. If we assume that the ESP portion is successfully decrypted, the mvif driver will deliver the packet to `ip_output` for forwarding, but since the real ip datagram is sent to the MN, the packet will be delivered (finally) to the MN, itself. Thus we are able to tunnel "over" the FA.

F.11 Tunnel Driver

Our MIP tunnel driver, called the "mvif" device, is integral to our IPSEC/MIP implementation. To paraphrase Churchill, it is "an enigma wrapped up in a mobius strip" and its operations may not be easy to understand.

Sans IPSEC, we may describe it as a virtual pseudo-device. A route is bound to it and the packet on input to the mvif driver, simply has an IP header prepended to it. The ip destination in the outer ip header is set to the gateway, which is typically the FA's ip address, or tunnel endpoint. That packet is then delivered to `ip_output` again for forwarding. (Note that we have an inner ip src address and an outer ip src address, since we have IPIP appended to the packet). On input at the tunnel endpoint, the mvif input function strips the outer IP header, and delivers the remaining IP datagram to `ip_output` for forwarding again. We typically use this for MN specific host routes between the HA and a FA where the MN is resident. The rule is that whatever mvif does, it always delivers the result to the ip output function so that the packet can be forwarded (routed) again. At a HA, on mvif output, a packet will typically be routed via the default route and head out to the FA via an Ethernet interface. We have described normal operation, but the reader should note that mvif tunnel routes may be used anywhere, including between a FA and another FA, or between a MN and agents at home.

The driver has 3 IP addresses of interest associated with it:

1. the normal IP address bound to an interface via ifconfig. We will call this the "ifconfig" address.
2. the inner ip header ip src address.
3. the outer ip header ip src address.

All 3 of these addresses are setable. In normal operation at a HA, we use a private IP address for the "ifconfig" address, as typically that address is not useful. We have modified the TCP/IP stack so that if it tries to choose the inner ip src and finds that the device is the mvif device, the driver itself will be asked what its IP src address should be. As a result, it can lie and will provide the setable #2, inner ip src. We typically set this to be the same as that of the Ethernet ip address on a HA. Thus the inner ip src is a "real" ip address. In normal HA operation, we typically do not set the outer ip address (which the driver itself can do), as the rest of the stack will set that according to the default route's real interface. The two inner IP addresses can be set via `ioctl(2)` calls to the driver.

Why so twisted? Because Mobile-IP itself is possibly a little strange since we are told that a Mobile Node might act as its own Foreign Agent if it can acquire a COA somehow (for example, via DHCP). We have some architectural flexibility because we can use the mvif device as a place to push the "original" MN ip address (home ip), since the COA will have to be bound to the real interface in use. This would allow the MN to at least recognize packets sent to its home ip address that arrive when it is abroad, and might enable the Mobile-IP MN as COA mode (although we have done nothing else to enable that aspect of Mobile-IP).

However the primary reason for this particular aspect of the mvif device is that it makes the driver very flexible in terms of border routers and firewalls that seek to toss packets arrived from abroad (exterior domain) where the ip src of the packet should be inside the firewall. Of course, this mechanism at a border router is an anti-spoofing function. We will talk about this aspect of the tunnel driver more below in a section on the MIP/Firewall problem.

In IPSEC terms, the mvif driver is playing the following functions:

1. it uses the NRL `ah_transport` and `esp_transport` functions and implements all output tunnel functions; i.e., it knows how to append a header and do AH/ESP. It is called at various times to do all four route bindings we have mentioned previously.
2. on input it is called after AH or ESP have gone off, as the next proto value for either of those headers is set to 4; i.e., IPIP.

Given that mvif is above AH/ESP on input, we have implemented two possible simple input filters. We have a filter for all input and one for IPSEC only. An administrator may choose to only allow mvif to be used for output, and any input will be discarded. This might be of use at a HA, where we do not expect any packets to be tunneled back. Or he/she may choose to allow only IPSEC processed packets; i.e., a packet must have successfully passed through AH or ESP before it reaches mvif and can be forwarded. Both switches may be set with `sysctl(8)`. This gives us some simple defenses against tunnel spoofing.

F.12 Firewalls and Tunnel Considerations

The Mobile-IP working group has recently discussed the problem of how Mobile-IP can work in terms of current CERT advisories that basically block external domain Mobile-IP connections. The CERT advisories in question are:

1. CA-95:01.IP.spoofing.
2. CA-96.21.`tcp_syn_flooding`.

The first basically calls for border routers to filter out IP packets trying to get into the local domain, based on an IP src that should belong to an interior domain. This is being called "ingress filtering".

The second expands on the first and calls for an additional filter on packets trying to leave a domain, where the packets have an ip src that is outside the domain. Logically it could be called "outgress filtering" (if that term makes any sense, but since when does routing terminology make sense?).

Filter #1 was probably deployed in 1995 on a number of border routers (routers that perform an Exterior Gateway Protocol such as BGP). It was apparently a response to external attacks on protocols that only use IP addresses as "authentication". For example, NFS, X, and the BSD rsh based utilities such as rsh, rlogin, and rcp. (We submit no one should ever use the term "IP address authentication" again, because it is an oxymoron as the authentication mechanism is so poor, it is a disgrace to the term "authentication").

The consequences of filter #1 for Mobile-IP are severe and ironical. If a Mobile-Node wanders out of its interior domain and tries to use a Foreign Agent found elsewhere, the following observations hold:

1. it will be able to register with the MIP UDP registration packets, because the FA acts as a proxy; i.e., the FA forwards the registration to the HA, hence the FA ip src address is used (and is legal). Hence the border router will not filter out the packet from the FA. Mobile-IP as a UDP-based routing protocol works.

2. the Mobile Node will be able to talk to any end systems that are not within its interior domain.

3. the Mobile Node will NOT be able to talk to any end systems that are within its interior domain. The most serious lack here is probably that it will lose its DNS service.

There is one simple way (barring the use of tunnels, which we will discuss later) out of this impasse. A remote FA could be considered as a "bastion host", and could provide proxy services for any visiting MNs via socks, proxy web servers or the like. There are trickier possibilities. One must observe that DNS should be used locally, and one probably should not be doing DNS over the Internet under these circumstances. There is much to be said for work here in general terms that would offer up services to a Mobile Node, possibly via DHCP. Of course that leads to another observation, a Mobile Node that could use DHCP under these circumstances, would not have the filtering problem, since its "interface" ip src address could be legal. At times, people in the Mobile-IP working group have wondered what the MN-COA mode of MIP is for. Possibly its greatest use should be in external domain situations, since the acquired COA would make packets legal in CERT terms.

We ran into filter #1 when we installed our FA at OGI. OGI is external to our NWNET border-router.

Filter #2 would additionally not allow a Mobile Node to send packets out of an interior domain because its own fixed MN ip src address should not be present locally.

We have performed a series of experiments at the OGI FA to see what would and would not work. We have not observed Filter #2 (yet), but we have observed that Filter #1 is deployed at the border router "entrance" to PSU. We have played a bit with tunnels and have performed two "interesting" (and scary) experiments that confirm that IPIP tunnels can be used to "solve" (in truth, cover up) the problem.

1. we setup two host tunnels from the OGI FA back to the only two PSU hosts likely to be of use for us; our own DNS server, and our compute server. As a result, any number of local PSU MNs at OGI could access PSU services.

2. we "borrowed" the Foreign Agents COA and setup tunnels directly from the MN in question to PSU. The two tunnels were host tunnels again to the PSU DNS and compute servers. The COA was used as the outer ip src address in the mvif driver.

We have always felt that in terms of topologies, there were at least three major Mobile-IP topologies that were of interest:

1. MIP within an interior domain. We are doing this now of course. We regard this as probably the most likely short-term use of MIP.

2. MIP across exterior domains. This is an interesting possibility, but it is hard to know at this point, how common it might be. Certainly there are interesting security problems here that need to be solved (the firewall and HA/FA tunnel questions are two such problems). Possibly a solution might lie in the realm of expecting MNs to borrow COAs when abroad and use them "intelligently". We also firmly believe that trust negotiation between MNs/FAs and HAs/and firewalls are possibly and indeed likely components of a solution space in this topology.

3. MIP at home as a way to deploy more Internet-capable computers at home. A MIP box at home might fall into either the interior or exterior routing domain areas.

We intend to upgrade the FA at OGI to an ipsec-based kernel soon and try setting up AH based tunnels between it and a PSU agent. We would like to point out that agents are fundamentally a security exposure if they are capable of acting as tunnel endpoints, because they can be used to violate the CERT advised border router filters. We submit that tunnels should probably be tied down by AH and or AH/ESP. We also intend to see if we can setup AH-based tunnels directly from an MN abroad using the FA's COA back to "home".

F.13 IPSEC/Mobile-IP Security Considerations

1. route table type semantics are complicated, for example, cloning and routing redirect semantics. There is danger here at the Mobile Node, in that one might choose to tunnel to a local-link HA and use ESP on all packets, but a dynamic redirect might spirit packets away. At a minimum, MNs must be able to turn dynamic routing redirects OFF.

2. at a MN, if one chooses to use DES via the default route to the HA, it is always possible that a more specific non-DES host route might somehow be inserted (especially on a multi-user system). We claim that our laptops are single-user, but this possibility should be kept in mind. Furthermore, host routes might already exist and thus not be overridden by the default route. The first two points here can be summarized by simply pointing out that our route binding is a tool, but we must still study long-term how to use it.

3. by definition, routing means network layer security. This is a useful tool in some situations, but it is not end to end security which is something only app or transport layer security can accomplish. IPSEC/routing does not rule out IPSEC/transport OR ssh for that matter.

4. traffic analysis leading to cryptanalysis by an evil MN at a remote FA is always a possibility. Bellovin has stated that it is desirable for a node to always meet an encrypted packet with an encrypted packet, or plaintext with plaintext. Ideally a node should not respond to encryption with plaintext or vice versa. For example, imagine that EVILBART is pinging MNALICE. With our routing based encryption, EVILBART could notice that MNALICE is at a given FA, and try and ping MNALICE. The packets would go to MNALICE, who might choose to enable a chosen plaintext attach by encrypting EVILBART's packets. EVILBART could use various routing techniques to make sure that the packets were delivered through MNALICE's HA and hence would go through the ESP tunnel. A MN should be able to filter traffic by end systems. By definition, ALICE might only talk to end systems, X, Y, Z, and EVILBART would not be amongst their number. Normal ip firewall mechanisms at MNALICE could be useful. We might also need more dynamic mechanisms so that ALICE's software systems might be informed that packets from an "as of yet unknown" end system had arrived and could make a policy decision.

5. The ability for a FA to act as the tunnel endpoint means that any FAs available in a local routing domain can be used by external spoofers to defeat current CERT "ingress filtering". At a minimum, we suggest that tunnels be able to turn off input (not possible at a FA by definition, but possible at a HA), and/or use IPSEC AH (at least) and insist that any incoming tunnel packets must be like this:

   `IP | AH | IP (etc).`

   In other words, HA to FA tunnels that cross enterprise domain boundaries should use IPSEC as mechanism. This suggests that tunnels themselves might need to be negotiated by a higher level trust mechanism across domains. Firewall filters can filter out IPIP packets and presumably can be taught to allow IP — AH or IP — ESP. Possibly they may allow packets to only certain IP src addresses (known FAs designed as bastion hosts). Without IPSEC (and higher dynamic key negotiation mechanisms), it is hard to see how Mobile-IP sans IPSEC can be anything other than a "cert'ifed" spoofing threat.

F.14 Redundancy

We are going to briefly touch on the current theory behind our redundancy work. We have "theory" in place for the following areas:

1. multi-hop ad hoc routing

2. home agent redundancy

3. foreign agent redundancy.

In addition, in working with Jon Inouye, we have realized that redundancy could also focus on allowing a Mobile Node to bind more than one interface at a time to a route (especially the default route). We do not intend to currently pursue that area, but the notion is interesting and could be useful.

We are currently in the final stages of working out a theory for multi-hop ad hoc routing. We intend to explore "redundancy" aspects of such a routing protocol and will base it on authenticated source routing via multicast. We will go into more details in the next report. For now however we only wish to raise a few points:

1. we are architecting a source-rate based protocol that will be based on top of our current ad hoc protocol. The current protocol is authenticated and establishes link-layer reachability. It is subnet-free. The next generation will also be authenticated and subnet-free. We hope to be able to explore multi-path aspects; i.e., give this protocol a redundancy orientation.

2. we have encountered a difficult kernel routing implementation problem. Basically we need an "input" (besides an IMCP host unreachable message, which may not arrive) from the routing code in the network layer in order to drive the "on demand" source route query in the MN routing daemon. The problem is that if a default route is present, it always matches any destination lookup where there is not a more precise answer (a specific host route). If the default route is not present, the kernel will generate a "`route_miss`" message and send it upstream to readers of the `route(4)` socket (routing daemons). It is not satisfactory to simply remove the default route, since for MNs that can hear an agent, the default route will be set to the agent. We have experimented with adding a function to the routing code of the kernel that will cause the kernel to generate a "`route_clone`" message whenever cloning occurs. This will allow us to learn that the kernel has generated a "host route" when the default route is used, and we can use this as an input to drive the multicast source query mechanism at a Mobile Node.

F.14.1 Home Agent Redundancy

The HA redundancy specification work is started. For a change, we do not need any kernel work. We think that HA redundancy might take two flavors: 1. "serial" HA redundancy or 2. "parallel" HA redundancy. "Serial" means that a MN would have a list of two HAs and would only use one HA at a time, but would be able to switch between them, if the current HA failed to respond to a Mobile IP UDP request. Parallel would mean the use of two HAs at once. We prefer the former, since it would lighten the Internet load and seems more practical.

The gist of the idea here is that the HAs will use normal unicast IP routing and act as unicast routers to the MN subnet, which will be partitioned by definition (i.e., the wavelan subnets on the other side of the HAs will not be co-located). In other words, we intend to use normal unicast IP routing for reachability to the Mobile IP subnet. The important point here is that at any time, an individual packet from a host sent to a MN might go to either one of the HAs. As a result, the HAs must keep each other up to date about the location of MNs.

We intend to configure the two cooperating HAs so that know about each other and keep a TCP connection open between the HA pairs. They will use that connection to exchange information about MN status as they must keep a parallel routing table setup for all cooperating MNs. A TCP-based application routing protocol will be defined that allows the HAs to check that their peer HA is up, and exchange MN location information. The HAs will install tunnels in parallel for MNs at remote FAs. If an MN is actually "home" at one HA, the peer HA will install a tunnel to the other HA for packets that are naturally routed to it.

One idea that we should consider here is whether or not MNs could be dynamically informed of a new HA. This would allow for a smooth handoff in case one HA was taken down and replaced by another that did not share the same IP address for the HA. This is a feature that could be built on top of the "serial HA redundancy" system. On the other hand, this may be a feature that is not useful in the short term.

F.14.2 Foreign Agent Redundancy

We have already implemented one possible aspect of FA redundancy; that is, we have a "faster" handoff algorithm that allows us to switch between FAs and gives us some ability to deal with FAs that are partitioned from our current HA. We hope to soon (probably in spring) begin an experimental implementation that will allow a MN to talk to two FAs at the same time.

The fundamental feature point needed here is that we must teach the routing code in the kernel to allow us to load a second gateway. Logically we want to be able to bind a list of gateways to a given destination. IP output would send each packet to every gateway in the list. In terms of the routing table, we might have:

```
Destination IP Gateway 1/if1 Gateway 2/if2
1.2.3.4 2.2.2.2/ed0 3.3.3.3/wlp0
```

When IP sends a packet to destination 1.2.3.4, we would send two packets. One would go the next hop router at 2.2.2.2 via the Ethernet ed0 interface. The other would go the next hop router at 3.3.3.3 via the wavelan interface. (We would probably use the same interface in both cases, but the capability to use two interfaces is interesting). We are simply trying to do what we can to improve the odds that a given packet might make it. On the other hand, the cost to the network is obviously twice as much. Still this may be useful in extreme cases, and as is often the case, a little mechanism in the kernel may give a lot of flexibility at the application layer. We intend to implement the mechanism in the next quarter and think about how to use it in the long term.

FA redundancy would use this mechanism as follows:

1. at the HA, the HA could setup a tunnel route to the MN, and bind two FA gateway/COAs to it.
2. at the MN, a MN using WaveLAN currently has a list of agents (FAs) sorted by signal strength. Instead of using the top single agent, we would use the top two.

We expect this dual gateway mechanism would make intermittent connectivity "better". Possibly it should only be used in the case of intermittent connectivity to the top two FAs. How to measure any cost and advantage here remains an open question.

F.15 Rough Measurements of IPSEC and Finnish ssh over WaveLAN

We have made some crude measurements of our current IPSEC/ESP route binding and the Finnish ssh (which has other encryption algorithms besides DES) on a few pentium machines, over WaveLAN, and over "localhost". "Localhost" means we run the client and server on the same machine, thus we get a rough idea of a machine's compute power. The goal here was simply to get a feeling about IPSEC versus ssh in terms of algorithms and to get a feeling about the cost of IPSEC encryption over the WaveLAN radio modem. The ostensible goal was not to measure crypto speed on a pentium machine, although there are certainly some inferences in that regard that can be made. These are rough benchmarks and it is important to point out that the conclusions are rough as well.

We will first characterize the test environment in terms of hardware and software tools, then present the numbers, and finish with a few conclusions.

- Hardware/hosts:
  1. IBM thinkpad laptop, 75mhz pentium
  2. Home agent running on 166mhz pentium machine. This is not a pentium pro.
- Hardware/WaveLAN:

  On paper, WaveLAN is advertised as having a 2mbit speed. we typically see speeds between 1mbit and 2mbit depending on collisions and on timeout errors that are built-in (unfortunately) to the Intel 82586 lan controller used in the ISA WaveLAN system. The testing here was done when the local WaveLAN link was otherwise unused. We were able to verify this by using a new BSD tool called "trafshow" that is a simple "real-time" network analysis tool that shows local link traffic. As a specific example, we typically see speeds of 100k bytes per second with ftp transfers over WaveLAN.

- Software/IPSEC route binding:

  Using the `arp(8)` command, we installed routes between the two hosts over the WaveLAN links that were bound to a DES (or dummy) crypto algorithm. We used ftp to transfer data and compared to this ssh transfers using DES (or other algorithms). It should be noted that the IPSEC/NRL DES algorithm is coded in assembler and is more efficient than the C-based algorithm used in our current ssh code.

- Software/tcpclient + tcpserver:

  As part of our IPSEC verification effort, we created a tcpclient/tcpserver application pair that allow us to send various sized tcp packets over a TCP connection. We used this to test the NRL/IPSEC transport mechanism. Tcpclient sends packets to tcpserver that discards them.

- Software/ssh:

  Ssh is the Finnish tool that allows RSA-based authentication, followed by use of encryption across a TCP-based connection. Scp can be used to replace FTP or rcp. Slogin can be used to replace telnet or rlogin. By default, ssh uses idea with 128 bit keys for encryption but can be told to use DES or 3DES as well. Ssh is entirely an application layer application. It does not use IPSEC and might well be considered as a competitor for IPSEC. (Frankly, our laptop users are making a great deal of use of ssh at this time).

- Software/ncftp:

  Ncftp is a replacement ftp client for the traditional BSD ftp client. Observation has shown that it is more efficient in a number of transfer situations than the traditional ftp. It also has a better user interface. The reason why it is more efficient is that it typically does 30k byte reads or writes and hence has a tendency to fill any TCP window that is available. As a result, the TCP window is often *not* the bottleneck where ncftp is concerned.

Note that all of our applications here were TCP-based. Our numbers are presented in kbytes per second. We only ran our tests a few times in each case and present a rough average. For reasons of comparison, note that maximum Ethernet bandwidth is 1280 kbytes per second.

We have not yet benchmarked IPSEC/ESP/DES over Ethernet and hope to be able to do that this quarter (we need to purchase another PCMCIA Ethernet card or two).

F.16 Actual Measurements

F.16.1 Using WaveLAN between the two hosts:

- ncftp with no security in place: 95 kbytes
- scp with only plaintext encryption (-c none): 100 kbytes
- scp with crypto:
  - -c des: 103 kbytes
  - -c idea: 110k kbytes
  - -c 3des: 100 kbytes
- using IPSEC/ESP/DES/route binding over WaveLAN (with ncftp): 100 kbytes
- using IPSEC/ESP/dummy (ncftp): 100 kbytes

F.16.2 Localhost testing on bridgeport:

- tcpclient/tcpserver using IPSEC/transport:
  - IPSEC/ESP/DES: 800 kbytes
  - IPSEC/ESP/dummy: 3,200 kbytes
  - no IPSEC (just plaintext with no ESP): 10,000 kbytes

F.16.3 scp testing over Ethernet

This was done between a second 90 mhz pentium desktop machine and the 166mhz desktop pentium. The Ethernet cards in both machines are 100mbit/10mbit SMC Ethernet cards used on 10mbit Ethernet. These cards are 32 bit PCI-bus cards and are probably among the fastest Ethernet cards available for PCs.

- scp with plaintext (-c none): 478 kbytes
- scp -c des (56): 313 kbytes
- scp -c idea (128 bit key): 262 kbytes
- scp -c 3des (3 * 56): 159 kbytes

F.16.4 ncftp testing over Ethernet - plaintext only

- 1076 kbytes (80+

F.16.5 Benchmark Conclusions

Our primary conclusion is that the computationally intensive cryptographic algorithms DES, 3DES, and IDEA are still fast enough on a pentium machine so that WaveLAN (at roughly 1/10 of Ethernet) speed becomes the bottleneck. All of the presumed scp encryption algorithms that have bigger key sizes than DES were slower than DES over Ethernet, but ran at the same speed over WaveLAN. There was no measureable difference between IPSEC/DES and scp/DES. It is possible that faster LAN radio systems may come onto the market in the next few years, but most likely compute speed will go up as well, and probably will be able to keep up with the encryption cost in the near future. We have tried a few encryption benchmarks on very slow 486/33mhz laptops. On those systems the speed of DES encryption begins to interfere with transmission over WaveLAN. A few of us have used slogin/3des over WaveLAN and have not noticed any apparent difference. This result explains why. In summary, WaveLAN simply isn't fast enough at the moment to get in the way.

On the other hand, a Home Agent acting as the tunnel endpoint for N remote Mobile Nodes using DES/tunnels back to the HA, will need all the compute power it can get. We will attempt in the next quarter or two to measure simultaneous DES stress on the Home Agent (if we can figure out a meaningful test case).

The IPSEC/DES algorithm that we are using was done in assembler and appears to be much more efficient than the C-based ssh DES algorithm. On a fast pentium machine, over localhost, it runs at a rate that is still less than Ethernet bandwidth, but is not bad (800 versus 1260 kbytes). Certainly over 100mbit fast Ethernet, any of the encryption algorithms would become a bottleneck. On the other hand, our measurements show that ncftp is twice as fast over an unused Ethernet link than scp in plaintext mode. Scp could probably stand some application code tuning as well as use computer specific encryption code.

The tcpclient/tcpserver test that used IPSEC/DES versus IPSEC/dummy in transport mode on the same pentium server is interesting. It points out that there are two costs involved with IPSEC and both are very real. Of course, there is a compute cost involved in DES encryption and decryption. The dummy transform simply carries out mbuf (message buffer) manipulations in the kernel and inserts the ESP header in front of the plaintext TCP and data portions. One can see that there is a heavy cost involved in just the empty transform. As NRL has pointed out, there may be room there for future tuning.

F.17 Outreach

John McHugh and Jim Binkley paid a brief visit to BBN in late September to compare notes with Steve Kent and John Zao. At this point we have an idea as to what each group is up to in terms of security and mobile networking. Up to now, we have been working on low-level mechanism, and they have been working on high-level signature work.

John McHugh gave a presentation on the project as part of a panel at the National Computer security Conference in Baltimore.

David Reeder has attended recent IETF and Mobicom meetings and made various useful contacts in the mobility research community. We are hoping that we might be able to obtain a copy of FTP Inc's Mobile-IP release for interoperability testing. David will go to Boston for their announced interop testing before the Memphis IETF. Jim might and David will attend the next IETF meeting.

Jim Binkley plans on giving a talk in the next quarter at OGI and later in February at Intel to their mobility group on our current work. John will continue to attend various PI meetings and give briefings on our current status.

In winter quarter, at least 3 professors (Binkley/Daasch/Schubert) used mobile laptops to give software demos to their classes in the PCAT building. Professor Daasch of Electrical Engineering was able to demo Mentor CAD tools available on sparc systems elsewhere for his EE class and pronounced the radio system extremely useful. There is certainly much opportunity for deployment of mobile radio lans in campus teaching environments.

Appendix G

Quarterly report – Winter 1997

G.1 Project Status Overview

We have made progress on both security and redundancy issues this quarter despite the fact that system complexity and the number of moving targets (Mobile–IP, FreeBSD, IPSEC, ISAKMP, etc.) are ever–increasing.

We made two important releases to the Internet. We released a FreeBSD–IPSEC kernel in cooperation with MIT. We also made a second release of Mobile–IP at PSU that included our ad hoc and radio–based handoff code from summer and fall. We also have managed to acquire BBN's code and we sent a newly hired engineer to Boston in order to improve contacts and our knowledge of their system.

At the beginning of the quarter, we were extremely fortunate to hire Bill Trost as our second programmer (along with David Reeder). Both are focused on various security related issues. Bill has a fair degree of UNIX internals and networking experience. He is working on PCMCIA including Fortezza, kernel porting, and higher–level security issues including session–key mechanisms. David Reeder is continuing to work on tying Mobile–IP to IPSEC in terms of lower–level kernel and routing daemon mechanism. Our graduate students are working on redundancy issues.

G.1.1 Specific Accomplishments for Winter 1997

1. FreeBSD–IPSEC kernel release. This was a port of last summer's NRL code to FreeBSD modified to include our routing interface to IPSEC (ESP only). The code is split into two parts, one at PSU, the other at MIT. The first contains the PSU work, without the cryptographic code, the second contains only the cryptographic code.

   PSU – `http://zymurgy.cs.pdx.edu/freebsd--ipsec/freebsd.ipsec.tar.gz`

   MIT – `http://web.mit.edu/network/isakmp`

2. FreeBSD–PSU Mobile–IP release number 2. We released a second version (not IPSEC at this point) of our Mobile–IP for FreeBSD 2.1.0 that includes our authenticated ad hoc work, fast(er) radio–based handoff, and X–based mobile node daemon. It also includes improved and simplified kernel routing mechanisms.

3. IPSEC–integration with Mobile–IP. When the DARPA Project Manager visited us in January, we were able to demo static routing tunnels that used ESP/DES but the mobile node routing daemon were not yet able to use that kernel mechanism. At this point, our version of Mobile–IP can use 2–way ESP/DES tunnels between the Mobile Node and Home Agent, whether at home or away over a Foreign Agent. We can also, according to static configuration on a per IP address basis, install end to end 2–way DES routes between MNs that use the ad hoc routing protocol. We will discuss current status and emerging issues in more detail below. This work will form the basis of a future integrated Mobile–IP/IPSEC release.

4. PCMCIA kernel work bears fruit including Fortezza "contact." Bill Trost has been able to get Jon Inouye's (OGI[1]) supplied PCMCIA kernel infrastructure working and we have made "contact" with the Fortezza hardware. There will be a short section on this subject later in the report. There has been another interesting result here due to a better PCMCIA kernel infrastructure. We can now deploy a "mobile router" that has either a ethernet or serial card AND a wavelan card; i.e., we can make agents out of laptops. We have made use of this facility several times. We deployed a Foreign Agent at the Baltimore DARPA PI's meeting that was a laptop that made a serial connection and had a WaveLAN card. Jim Binkley gave a recent talk at OGI where we deployed a FA that used a lan card and a wavelan card.

5. MIP interoperation. We have successfully interoperated at FTP Inc.'s sponsored Mobile–IP interop. David Reeder attended the recent FTP Inc. interop and our implementation has successfully interoperated with other Mobile–IP implementations.

6. Bill Trost visited BBN in Boston. We have acquired their software. Bill's job will include integration of this code either in spring or more likely in summer.

7. Redundancy work. We have finished a specification of our multi–hop ad hoc routing protocol and also a first draft specification for the Home Agent Redundancy Protocol. We are also in the final stages of constructing a small lab for the Home Agent protocol. There will be a short discussion of both of these efforts below.

G.1.2 Spring and Summer 1997 projects:

In this section, we are going to briefly discuss long term goals and planning. It is not clear that we can make a release in Spring as it has become readily apparent that we must include FreeBSD 2.2 in our next release. We also would like to include ISAKMP in the routing mechanism and as part of the release.

Our next release minimally will include the following attributes:

1. an integrated Mobile–IP/IPSEC based on routing where policy is specified very simply in configuration files.

2. Mobile–IP/IPSEC should include the ability to setup IPIP "tunnels" that use either AH/ESP as supplied in the current NRL kernel infrastructure and agents should be able to only accept packets over tunnels that have IPSEC attributes.

3. ISAKMP will be supported as part of the routing infrastructure. We will use previous Cisco releases for this (and in point of fact, have already gotten the daemon working on FreeBSD, but not the route to PF_KEY interaction).

4. the release will be targeted towards FreeBSD 2.2, as defined by the CDROM shipped by FreeBSD (that has not yet happened, although the release has been formally announced).

We do not choose to predict when we will make this release other than to say it will probably happen this summer. We must point out that we MUST use what we have and cannot afford to deal with all of the many moving targets at once. The list of possible targets includes ISAKMP, IPSEC proper (AH/ESP/combined) including PF_KEY, Mobile–IP, and FreeBSD itself. We must first limit the moving targets to FreeBSD. Of course, we still must integrate other existing sub–systems architecturally which includes more original routing/IPSEC work and ISAKMP, itself. For now, we plan on porting our entire software system to FreeBSD 2.2 as soon as possible. In point of fact, that work has already started and we have just gotten our mobile node daemon working under 2.2. When summer comes, we can hopefully address other moving targets that might include a new NRL release and/or PF_KEY changes.

[1] Oregon Graduate Institute

G.1.3 Personnel and sub–projects:

Jim Binkley acts as project manager. He supervises David Reeder, Bill Trost, and the graduate students. Jim is usually involved in design, code release, review, specification review, and kernel work. Jim will also take on some of the 2.2 porting work as he is the most experienced kernel person.

John M^cHugh is working on the Fortezza part of the project and is in charge of high–level key issues. John will help manage Bill Trost.

David Reeder is a full–time programmer. David has worked this quarter on integration of Mobile–IP daemons and IPSEC/ESP in routing. He is currently working on finishing the kernel routing mechanism needed for binding IPSEC to Mobile–IP tunnels and other IPSEC related routing work in both the operating systems and Mobile–IP routing daemons.

Bill Trost is our new full–time programmer. Bill brings with him a fair degree of system programming experience. Bill is in charge of "higher–level" security issues such as ISAKMP, and the BBN session–key protocol. He visited BBN a few weeks ago and has been leading our effort to understand their work. For now, he has four sub–projects: 1. Fortezza AND kernel PCMCIA work, 2., taking part in the 2.2 port with Jim Binkley, 3. integration of ISAKMP into our IPSEC routing infrastructure, and 4. integration of the BBN session–key protocol into our version of Mobile–IP.

Xu Hao is a M.S. graduate student who graduated at the end of winter quarter. He completed a first cut multi–hop ad hoc routing during winter quarter.

Jennie Ye is a M.S. student and joined the project last quarter. She has taken over Xu Hao's work and has begun to implement the ad hoc protocol. Jennie spent last quarter coming up to speed on the project.

Bjorn Chambless is a M.S. graduate student who has been with us since last summer. Bjorn made a first draft of the Home Agent Redundancy Protocol (HARP) during winter quarter and will start implementation in spring. He has also setup a small lab as HARP minimally needs 3 routers for testing. (Two routers will act as peered Home Agents, and the third will act as an upstream control router).

G.2 IPSEC and Mobile–IP

In this section we will discuss the keys that are currently available in our system, current IPSEC/MIP project status and emerging issues, and provide a short discussion of firewall issues and work we have done in that area.

We have written a rough draft "appendix" on Mobile–IP and Security Policy issues. The appendix is at the end of this document.

G.2.1 Keys In Our Current Software System

There are currently three classes of keys in our Mobile–IP system. The classes include: 1. Mobile–IP authentication keys, 2. IPSEC keys (both AH and ESP), and 3. ad hoc authentication keys. The first and last might be viewed as "routing" keys.

The Mobile–IP keys are indexed by IP–address. There are three kinds:

1. MN–HA,
2. MN–FA,
3. FA–HA.

We support all three but use of the latter two would make the mobile system closed; i.e., external FAs (and MNs) in terms of key administration could not participate. We typically only use the MN–HA keys. Each MN has its own unique HA key (although it is actually indexed by the HA IP address, thus the mechanism could support multiple HAs). The Home Agent configuration file of course is indexed by MN IP address. A sample key at the MN looks like this:

`mnha_key 204.203.1.2 md5 256 0x12345678901234567890123456789012`

The key type is named by a tag, and is followed by the HA IP address, transform, spi, and key. MN–FA and FA–HA key syntax is very similar. We borrowed the syntax from NRL's /etc/keys format.

The IPSEC keys apply to IP source and destination addresses and not only may be unique in terms of each Mobile Node but can be unique for outgoing and incoming packets. We do not distinguish between socket keys and route keys. The key simply is mapped into a kernel Security Association and is used by either mechanism in the stack. A typical ESP key key pairing as found in a Mobile Node might look like this:

esp 5000 204.203.67.203 204.203.67.210 des-cbc 0123456789abcdef 11223344

esp 5001 204.203.67.210 204.203.67.203 des-cbc 0123456789abcdef 11223344

The NRL format names the type, spi, IP source, IP destination, transform, key and iv (in the case of ESP).

A Home Agent with manually distributed keys would have the same per MN key pairing. Note that we also use the same mechanism with the MN to MN relationship.

The ad hoc keys are currently network wide but could be agent specific or even end system specific. There are two keys since one is intended to be used at home (the network authorization key) and one is intended to be manufactured on the spot and given away (the ad hoc authorization key). The formats are very similar to the Mobile IP key format but currently lack an IP address as an index. Note that any beacon can have BOTH keys or only one.

G.2.2 IPSEC/MIP status and issues

We have spent a good deal of the last quarter debugging our current Mobile–IP and IPSEC routing system. At this point we are able to dynamically install (or remove) 2–way ESP/DES tunnels between the MN and HA, whether the MN is away or at home. We use a very simple policy extension to Mobile–IP (that could *charitably* be viewed as a form of in–line ISAKMP) whereby the MN tells the HA whether or not it wants ESP. Of course the keys are a priori loaded and known to both systems. FAs are not involved and are simply used as routers, hence this 2–way tunneling could also work with the MN that has acquired a COA and has no FA. Further note that the proverbial chicken and egg problem of routing before IPSEC is solved by a simple hint in the MIP packet itself. Presumably we can modify this mechanism to handle either static key management or tell the other side to setup routes that can kick ISAKMP into play.

Getting the system working has not been easy. We have experienced some rather large bugs including the bug (recently solved by us) whereby systems under stress corrupted TCP data in transit that had been encrypted with DES. We finally determined that the problem was an NRL bug and removed the offending code which offered a security feature that we do not need. This gave us a cheap workaround. NRL has been informed and is in the process of fixing the bug. Ironically the Finnish ssh utility, which might be viewed as IPSEC competition, blew up when it detected file corruption - hence it showed us that the bug existed).

During the next quarter we intend to finish our IPSEC/route binding and deploy the IPSEC system internally. At this point we can only bind one SA (e.g., ESP) to any route. We may however want more than one binding, particularly for IPIP tunnel routes, so that we can establish authentication between a HA and FA, and at the same time have an IPSEC association between the MN and HA. We also of course want to be able to do AH as well as ESP. Much of the work to do the latter two mechanisms has been done, but route(4) API work, final hookup, and testing remains.

In addition, we can also address ad hoc topologies where keys have been manually distributed. This is doable because of our beacon–based authenticated ad hoc protocol. A mobile node daemon when it receives a authenticated beacon can lookup the beacon sender by IP address in a policy list and determine whether or not there is a policy installed for that MN (e.g., use ESP or do not use ESP). Given a pre–loaded SA again, when a beacon arrives for MN X, each system simply installs a link layer route with ESP for the other system. We assume any such routes will be symmetric (2–way). Note that unlike the MN to HA case, this use of route/IPSEC is End System to End System and that as always we assume our systems are single user systems.

We should point out that we have somewhat refined various problems that are built–in to the notion of binding IPSEC to routing, especially between IS to ES. We implemented two new switches in the TCP/IP stack that act as IPSEC–oriented packet filters. One switch applies to only IP forwarding (routing and not ES functionality). We call this the IPSEC–forwarding switch. If a system forwards packets, it may or may not apply IPSEC attributes to them. As a result, our MNs only apply IPSEC attributes to packets that originate on them. A spoofing system can not make a proposed plaintext attack on a MN where the IPSEC–forwarding switch is in use. By definition, this switch cannot be used at a HA that forwards packets into a combined IPIP and IPSEC tunnel. The other new switch applies to the input side of the IPIP virtual tunnel driver in the operating system. It can insist that packets that arrive to be detunneled, must either have already had IPSEC successfully performed on them lower down in the stack OR will have IPSEC performed on them later on in this computer. As a result, we have a simple filter that can guarantee that only IPIP packets with an IPSEC attribute can be forwarded off of a particular machine.

During the course of the quarter two thorny issues have emerged. The first is how do we combine ARP and IPSEC/routing? The second is how do we describe policy in terms of configuration? We will not discuss the second issue at this time and only wish to point out that when one combines certain topological possibilities that include using IPSEC, acquiring COAs (there are a number of possible ways), always routing to home or to individual nets or hosts, and other issues, the policy combinatorics become non–trivial.

The ARP issue emerged because we simply lacked kernel mechanism to control ARP access to computers. For example, assume a Home Agent with a wireless link and further assume that you want to control access to the interior so that only Mobile Nodes using IPSEC can qualify. One might (wrongly) think that the MIP registration packet could be used as a policy lookup token so that a agent daemon could determine if ESP was wanted or not. Unfortunately this does not apply to systems that do not play the Mobile–IP game. Any system that uses ARP can get access and send packets through a router as long as its subnet IP address makes sense.

There is a further worrisome detail and that is that the BSD link layer clone route would automatically clone a link layer "host" route (gateway == MAC address) without notification of any daemon upstairs. As a result, our agent daemon that implements the HA function really does not manage ARP routing state for Mobile Nodes that are at Home since the routes are automatically created by the operating system (although it does delete them when a MN moves away). We must be able to determine at the application layer if a system has sent us an ARP message and act on it. We also must make sure that IPSEC attributes apply to all packets sent through a link layer route (created by ARP) if that is the desired policy.

After much puzzlement, we decided on a plan and have implemented the kernel mechanism part of it. Next quarter we will modify our routing daemons to take advantage of the kernel mechanism and thus finally implement our MN–HA (at home) and MN–MN secure topologies using ARP. The basic idea is that we will install the clone route (a necessity for ARP) so that it has two new attributes:

1. the XRESOLVE attribute. This causes an UPCALL to be made when ARP cloning occurs; i.e., a daemon listening on a route(4) message socket will be notified that a system with IP destination X has shown up. This mechanism applies both to packets send and ARP packets received. As a result, we can implement a IP address based policy lookup mechanism.

2. a new BLACKHOLE attribute. The clone route will be loaded on a per interface basis with this flag as well. As a result, cloned routes will not let packets escape until the routing daemon shows up and changes the route (e.g., binds ESP to it and/or removes the BLACKHOLE attribute or logs a breakin attempt). This mechanism oddly existed in terms of route(4) flags but did not actually do anything.

Both of the above attributes should give us enough control to implement IPSEC based routing using ARP.

G.2.3 IPSEC and MIP - Firewalls and Tunnel Considerations

We have made more trips to the Oregon Graduate Institute because the Foreign Agent we installed there (which is now a HA for OGI at the same time) is as previously reported, outside our NWNET border router, hence we cannot talk to home without taking "measures."

It would appear that we have done something clever (or foolish?). We were able to demonstrate that two MNs could simultaneously "borrow" the FA's COA and setup IPIP tunnels back to home systems where the individual tunnels either had or did not have ESP bound to them. We wrote a script that would determine the COA from existing MIP utilities and then automatically install the tunnels. There are at least two points worthy of mention:

1. With ESP, all packets sent from the HA were sent using IPSEC, but we could either bind the default route to send all packets home using IPSEC or leave it alone, and bind IPSEC to the individual tunnel routes that sent packets home. This sort of policy choice is yet another mechanism that we would like to somehow build into our configuration.
2. There is no problem in using the FA's COA since the FA itself (as is probably the case with routers where firewall–oriented checking is not on) do not examine incoming packets to see if their own IP source is used. Since the FA's COA is really only applied to get the packets from the FA to the HA where the outer wrapper is stripped and thrown out, there is no reason more than one MN can't avail itself of that address.

We do not understand the current Mobile–IP effort to allow MNs to "reverse tunnel" home courtesy of the FA. We do not think that MNs should ask FAs to break into their homes and we do much like the idea that routers that are not performing firewall functions, should examine the IP source addresses of incoming packets. Of course, this whole notion is lunatic in terms of "ingress filters" since it serves to institutionalize bypassing them. As a result, a future CERT advisory may very well suggest that unicast IPIP should be blocked. Might one timorously suggest that IPSEC is the answer here?

It is true that stealing the FA's COA may be tacky. Possibly the right answer is that the FA might make COAs available to MNs. A COA could be a useful disguise since an IP outer header with such an address can cross a firewall and if followed by ESP could even hide the MN's own IP address. Of course, in that case, the MN must dynamically setup a Security Association between itself and the HA that binds the SA to the MN's COA.

G.2.4 Fortezza under FreeBSD

We have ported the Fortezza interface library to FreeBSD 2.1.0 using the "PAO" PC–Card support for built–in laptop PC–Card slots distributed by Tatsumi Hosakawa. In so doing, we discovered an apparent bug in the interface library's code to check for operation completion, which we had to correct before the library would function properly. At this point, it appears that we can talk to the Fortezza cards and other PCMCIA cards at the same time under our hybrid FreeBSD 2.1.0 system.

We have timed the SHA implementation present on the Fortezza card, thereby demonstrating our ability to access the card's cryptographic functions. The Fortezza's implementation of SHA is significantly slower than that provided with NRL's IPSEC distribution, which is software in the operating system. The NRL version is approximately 200 times faster on a 100 MHz Pentium with large (800K) inputs, and faster still with small inputs. The Fortezza documentation points out that the card's SHA implementation is slow, and they recommend using a software version of SHA they provide with the documentation that is approximately half as fast as the NRL supplied SHA. (The NRL version was supplied by NIST.)

G.3 Redundancy

In this section, we will present a short description of our redundancy work over the last quarter. This work is being done by graduate students and has focused on the specification of protocols for:

1. a multi–hop authenticated ad hoc routing protocol
2. and a protocol/system for Mobile–IP Home Agent redundancy.

We have also developed and implemented a very simple "Mobile Router" protocol that allows radio–based mobile agents to extend the wired infrastructure. This was an unplanned effort that occurred due to project synergy.

At this point we have developed three different ad hoc protocols, although the multi–hop version can be viewed as a logical extension of the first cut ARP replacement system. These protocols are:

1. a single–hop authenticated ad hoc replacement for ARP. Key ideas here include using a shared key as opposed to a shared subnet, and making address spoofing more difficult by authenticating the MAC address, IP address pair. All systems beacon. This work has been commented on in previous reports.

2. a simple mobile router system where Foreign Agents can also act as mobile nodes and hence become "mobile routers." Such systems only use one wireless interface and have no wired interface. Thus they act as a mobile node and a FA out the same interface. We have invented a very simple protocol that we will discuss below that allows wireless–based mobile routers to not choose each other (and avoid a route loop) if they can hear an agent that is closer to the wired infrastructure. This work is complete and will be included in our next release.

3. the current multi–hop source–routing based ad hoc protocol that we have been specifying this quarter.

G.3.1 Redundancy - Mobile Routers (Mobile FAs)

We discovered that mnd (the Mobile Node daemon) and mipd (our HA/FA agent daemon) can operate on the same computer without noticeably interfering with each other's operation. Such a system consists of one laptop with one PCMCIA wavelan card (radio only). The mobile node side communicates with a wired Foreign Agent and hence can roam. The Foreign Agent side allows any Mobile Nodes that are too far from the wired FA to operate. This allows us to deploy a wireless–based Foreign Agent as long as the Mobile Node (mnd) side can hear a wired infrastructure agent. We have actually used this system in a real life situation to extend the reach of our wireless infrastructure.

However, running multiple mobile nodes as FAs causes problems if two such mobile routers can hear each other. If the two Mobile Nodes are adjacent to each other and distant from any wired FAs, our current Mobile Node radio signal–strength heuristics will cause the two mobile nodes to attempt to use each other as Foreign Agents, causing a "routing loop" that prevents the Mobile Nodes from establishing long–term Internet connectivity.

To address this problem, we added a new optional tag–length–value extension to the ICMP Router Advertisement messages broadcast by Foreign Agents. This TLV contains an integer value that indicates the Foreign Agent's distance from the wire in hops. Zero or the absence of the TLV indicates the FA is wired. Thus, our modifications should interoperate with existing Mobile IP implementations that may or may not use our extension. (In theory, MIP implementations that do not use our extension should ignore it because the tag is in the range of Mobile–IP tags that may be unknown but should be ignored.) Mobile routers must have a specific configuration switch in their configuration file to cause them to even include the extension.

The protocol extension is extremely simple. Mobile routers (as FAs) send out the "hops" extension and calculate it by sending out a value that is one more than the smallest value they can hear. If they cannot hear any Foreign Agent, the value gallops to infinity (255) via a timer based mechanism. Thus a FA that is by itself "goes bad" relatively quickly (in seconds). This is a very simple vector distance protocol. The Mobile Node side (be it normal non–router MN or mobile router) uses the information as a first cut on the signal–strength based heuristic. We only keep the set of agents sorted according to signal strength where that set has the same "hops–to–wire" value. This system is not perfect but it does allow Mobile routers to be sloppily deployed so that agent cells can overlap. A con is that a normal MN that hears both a "mobile router" and a wired infrastructure FA may very well choose the wired FA over the wireless FA, even though the wireless system might be a better choice. On the pro side, we can deploy a daisy–chain of Mobile Routers so that we do not have to worry about whether or not they will point at each other as Mobile Nodes. The picture below shows what is possible:

```
------------ wired FA  (hops = 0)------------------
               |
               |
               |
             wireless FA #1  (hops = 1)
               |
               |
               |
             wireless FA #2  (hops = 2)
               |
               |
```

```
                        |
                wireless MN hooked up to to FA #2
```

Although we have not measured the latency of such a system (from the MN point of view), we should do so.

G.3.2 Redundancy – Multi–Hop Ad Hoc Routing

In this section we will provide a short discussion of our proposed multi–hop ad hoc routing protocol. We will first provide an overview, a few details, and some possible experimental goals.

G.4 Protocol Overview

The protocol establishes routing from one MN to another MN or to an agent when the MN is not available locally. It will be an on–demand source–route based protocol. Roughly when MN X wishes to find end system Z, it may establish a short–term path from X - Y - Z, where all systems are routers. The routing system is end host oriented because we want it to be subnet free. X will send a multicast–based request out and intermediate systems will read that request, subject it to a (anti-) flooding filter, and if the request survives, forward the request until it wears itself out (due to too many hops) or until it arrives at the desired destination. Source requests are made only on demand; i.e., when a system actually wants to talk to another system. The request mechanism may be viewed as a expanding multicast ring search. Agents always answer the request as a possible destination. Thus an agent is essentially advertising itself as a possible default route. Each request (or multicast reply) builds up a variable length hop tuple that consists of (IP address, metrics) as the source request goes from start to finish. Forwarding nodes must setup routes pointing backward as the request goes through them. The reply packet must also cause routes to be setup to the destination as it goes back to the sender.

The protocol should be viewed as consisting of requests and replies which if multicast are very similar to the request.

Replies may be multicast or unicast. We will make this a configuration option. It might make sense for agents to only unicast their existence (although multicast of agent availability may also be useful - usage situations may differ.) Multicast replies (like requests) cause the forwarding node to append its own IP hop tuple to the payload of the packet. Unicast replies are simply returned hop by hop in a unicast fashion from the destination (or agent). The source path is frozen in a unicast reply and represents sender to destination but rebuilt from scratch if multicast and may find more than one path back. Multicast replies hence consume more resources but at the same time are more adaptive. In fact, multicast replies have one very interesting property and that is that a multicast reply essentially informs the mesh of the "answer" (destination). Thus nodes might learn about a shared DNS server or agent (or command node). It may be useful to think more how this quality might be exploited.

The protocol is source route based as intermediate systems that choose to forward a packet will append a IP "hop tuple" that consists of (IP address, metric) to the packet. Thus a source route is built up. Each system that gets such a packet must set a route back to the sender (or multicast destination for replies) and may also set a route back to any intermediate system in the path. Routes already contained in the routing table should be refreshed. Routes timeout according to sender supplied values and/or forwarding system parameters, whichever are smaller. This might help to accommodate motion as for example, a truly mobile node would have to keep state based on a smaller timeout set by any sender or forwarding system (if it is moving).

How flooding is handled is very important. The true essence of such a protocol is that it is flooding–oriented and one would like to predict that it would improve the odds of packets getting through in a mobile mesh (as opposed to traditional routing protocols that try to achieve convergence first and simply use flooding as a tool to achieve that end). Flooding attributes include:

1. Requests that arrive at agents are not forwarded onto the wired infrastructure. We expect only mobile mesh systems sharing the same kind of link to use this system. However a agent may forward the packet to the mobile mesh itself (on the same kind of link).

2. Any multicast request has a id value set by the sender. Forwarding systems will cache the packet and not forward it if they have seen the id previously and the id itself has not timed out. On the other hand, forwarding systems MIGHT use the built–in route metrics to determine the best path and ignore the id. The latter notion is more costly again in terms of mesh bandwidth but might improve redundancy.

3. Forwarding systems will subject the packet to a loop test. If they see their own IP address in the packet, they will drop it.

4. The sending system will set a high–level (application, not IP) TTL. If the TTL is exceeded, the packet will be dropped. This is not the IP TTL which will be set by any forwarding or originating sender to 1 to prevent any possible multicast routing of the packet.

5. Packets that fail authentication are dropped.

The protocol will be authenticated, and will use shared secret keys in a manner similar to the first ad hoc routing protocol. It will carry over the major attributes from that protocol in that it will be subnet–free and authenticate the MAC, IP address pair of the sender. In our first cut, we will actually build the protocol on top of the current ad hoc protocol; i.e., the current ad hoc protocol can be viewed as establishing the link layer and the next generation establishes multi–hop connectivity. However the protocols can be combined since the multi–hop protocol can carry a MAC address over the first hop. Thus we can get to a complete demand driver system for link–layer establishment. In addition, we will carry over our current ad hoc authentication for now, but later on we will also try and make use of AH authentication tied in to the multicast addresses (it is not clear that our current NRL code supports that, but it should not be hard to make it work).

The metric part of the packet tuple will consist of a composite metric: (hop count, signal strength (or snr), and power). To begin with, we will only apply the metric at the endpoints of the source route path; i.e., forwarding agents will simply toss multiply received requests (same sender ID). By endpoint, we mean the destination (or sender on the multicast return route). The composite metric formula is:

$m = (PV * PS) + (SV * SS) + (HV * HS)$

where P stands for POWER, S for SNR, and H for HOP COUNT. The PV, SV, and HV values are variables that will arbitrarily range from 0..100 where 0 is worst and 100 is best. The PS, SS, and HS are weights (scalars) and are chosen at mesh configuration time. PS, SS, HS must add up to some integral value (say 1) and m itself will be scaled to 0..100. Thus one can choose to emphasize hop count, signal strength, or power according to various hardware attributes of the mesh itself. Forwarding systems will store the signal strength as recorded from the last link and their own current power value in the tuple as they forward it. Hop count can be determined by counting the number of tuples. It should be obvious that this composite metric is oriented towards mobile systems.

G.4.1 Implementation Details

The protocol will use a multicast IP address in the range 224.0.0.1 to 224.0.0.255; i.e., it is not routable in MBONE terms. It will use UDP. Mobile routing daemons will send and receive multicast packets (and optionally unicast packets sent to the IP address indicated in the hop path tuple). It should be pointed out that routing table entries are setup by this protocol and normal traffic will hence be carried by normal forwarding mechanisms until timers remove routes.

One important question that we have finally settled in the last quarter is how this integrates with our current routing daemons and kernel routing mechanism. We need a mechanism that can drive the source routing algorithm in the sender. We want said mechanism to not be dependent on the state of the routing table but to be driven by routing misses; i.e., the lack of specific host routes in the routing table. We also want the mechanism to be kernel initiated but inform an application daemon so that policy and mechanism can be cleanly separated. We do not want the mechanism to be affected by whether or not a default route is present as this would require a potentially hazardous and painful rewrite of the current code in our mobile routing daemons (which is complex enough). Currently a default route is present in Mobile Nodes if they can hear an acceptable agent directly (agent beacons). It is not present if they cannot hear an agent. We need to tolerate both sets of circumstances. There are also redundancy arguments for keeping a default route if available (since it gives you a default path where a node might be found).

In order to satisfy these requirements we have determined that the pre–existing BSD routing attribute called the XRESOLVE switch can be used. As a result, we can use the route(4) socket together with the pre–existing raw ICMP socket in our daemons to look for certain messages either from the network or the kernel which can drive the source route lookup. We had to make a small modification to the IP_output function so that XRESOLVE would also apply to raw socket packets (for example, ping). We do not understand why raw sockets would not cause XRESOLVE to work, as it was originally intended as a routing upcall so that a routing daemon could somehow externally resolve a non–ARP link address (X.25).

There are three possible events that can drive the source routing:

1. if the default route is present, we tag it with the XRESOLVE flag. When IP packets are sent to a new host IP destination, that has not previously been present in the routing table, a route(4) upcall is made to the routing daemons informing them that a cloned default host route to X was created. Normally such a route might be used for TCP metrics or for PATH_MTU state. The upcall will be used to drive the source route request. If source routing fails, packets will still be routed to the default route (which might be correct although it may or may not have an optimal path). Thus there is an important redundancy aspect here since our goal is to try and get packets through end to end even if the path is not optimal.

2. if no default route is present, the kernel will generate a routing MISS message on the route socket. We can of course use this to initiate a search.

3. In addition, we may receive an ICMP_UNREACHABLE message. We can use this to try the source routing again, although we might build in a back off timer on such retries. Note that the unreachable message is by definition not generated by multicast since one does not return ICMP errors for multicast messages. However it may at any time be generated by a path failure and should be regarded as a potentially useful input.

G.4.2 Experimental Goals

1. Obviously the protocol should work and not immobilize the local mesh with a broadcast or multicast storm. We need to observe local operation and simply make sure that the protocol does indeed operate as intended. It should be noted that we do not have enough computers to make wide scale testing in terms of even 10's of units. We will have to do the best we can with a handful of machines.

2. We need to make some attempt to quantify at least the following:

   (a) how much traffic overhead is induced?

   (b) can we determine any measure of "packet success" in terms of survival as opposed to loss?

   (c) how far can we actually get in terms of nodes away from the wired infrastructure and still have decent performance?

3. multicast versus unicast replies. We need to ask ourselves whether or not the multicast mechanism's cost is worth the overhead as opposed to the less–costly but less redundant unicast reply. As part of this, we must ask if the multicast return of a given destination to nodes that didn't even ask the question may be useful. Certainly it would seem a priori that a trait whereby many nodes learn and share a DNS or local web server could be very useful. Nodes could cache information that they might use later.

4. effectiveness of metrics. We have access to signal strength information currently in our WaveLAN drivers (although we still need to deploy the signal strength code in the desktop platform (agents)). We do not have means to actually gauge power strength but we can configure values in by hand and test it. We intend to make a small trial of a number of signal strength and power metrics.

5. motion and loss of intermediate systems. We certainly intend to setup paths and then eliminate forwarding systems “in between” the sender and destination. Hopefully this will give us some feel about how the algorithm will react in a truly mobile environment. In the final analysis, this is certainly an area that could stand simulation. We do not presently have the means for such a study.

G.5 Home Agent Redundancy

The Home Agent Redundancy system and protocol is still not completely specified as we are conducting some simple prototyping experiments. However we can tentatively outline how the protocol itself will work and outline a few requirements. When we have this system working, we will write an Internet draft on the protocol.

The protocol itself is called HARP, which stands for “Home Agent Redundancy Protocol.” HARP will be layered on top of ordinary unicast routing and as a protocol will be very simple. In general two Home Agents will be designated as HARP peers (yes, they will be called “harpies”) and will send HARP packets over both UDP and TCP. For a change, we can certainly use IPSEC/socket mechanisms (authentication in particular) as these are simply ordinary unicast sockets.

System requirements of interest include:

1. IPSEC (which is easy to do) for all protocol messages.

2. the harpy peers will present a consistent virtual mobile IP subnet to the world; i.e., they advertise one subnet to the interior routing domain. This will be discussed below in more detail.

3. any RFC 2002 compliant Mobile Node will not have to know about HARP. Thus agents take part, but mobile nodes do not need software changes. One consequence of this is that a Mobile Nodes’ Home Agent can be changed, but the MN itself will not need to know that the change occurred. The MN need be configured with only one HA IP address.

4. both harpy systems must maintain routing tunnels (say IPIP from HA to MN) in parallel. When MNs are at home, one harpy will tunnel to the other HA where the MN is actually resident.

HARP consists of three kinds of messages that will be exchanged over a UDP socket. We expect that peered HARP routers will not be on the same network since one would not want them both to disappear due to the loss of a shared network card in a single router, but they are not likely to be far apart either. Exterior routing protocols cannot support the idea of a subnet being in two directions. It is likely that the internal path between them will be better than the external “to the MN” (or from the CH) path. As a result (unlike say with BGP), we feel that we can get by with mostly stateless servers and let the MNs send registration messages to one harpy, which in turn will forward the message to the other system. The MNs must retry with a certain periodicity since a HA might reboot at any time. As a result, the MNs will essentially keep the HA state “fresh” and the protocol (or server implementations) itself need not overly concern itself too much with redundancy. Redundancy boils down to having two HAs in two different places.

On the other hand, as an optimization, we will allow one HA that has rebooted to use TCP to read out the entire MN routing table from the other HA. Thus the protocol in some respects will be somewhat like DNS (use TCP to exchange tables, and UDP to exchange messages). We agonized over whether to use TCP in all cases (more BGP like), but decided that we simply could not trust the semantics of TCP when routing connections fail or backup due to buffer latency. We do not want any given MIP agent to hang on an internal write because TCP cannot find room in the peer receiving window. The surviving agent should still be able to process all messages with a rough but constant fairness. If an interior network path fails, we expect dynamic routing to send all packets to the remaining peer. We don’t want it to fail because its peer failed. We could mitigate this TCP worry by always opening a connection, writing a message, and closing the connection, but this could lead to too much overhead. We suspect that taking advantage of the MIP stateless property here will lead to making HARP simple and more fault tolerant as well. It may also make it a bit more extensible, as possibly a unicast UDP protocol could be easily replaced with a reliable multicast UDP protocol.

UDP–based:

1. hello messages. When one harpy has not heard from the other for awhile they will exchange simple UDP unicast "hello" packets. If the other harpy is deemed not available due to the failure of N out of Y tries, a message will be logged, but it will not stop forwarding MN requests. Eventually more aggressive TBD "out of band" mechanisms might be implemented in order to notify local network managers that a serious internal routing failure has occurred.

2. MIP registration forwarding: Any MN registration message received by one peer will be encapsulated in a very small wrapper and forwarded to the other harpy, where it will be processed "normally" (routes may be setup to either a COA or peer HA), but no MIP ACK will be returned by the HARP receiver peer. The system that initially receives MIP messages is responsible for ACKs and NAKs and will not forward NAK'ed registrations.

3. upon boot, a HARP peer will know its peer due to static configuration. It will initiate a TCP connection to its peer and attempt to exchange in bulk the other's MN routing table. Thus one peer can learn the entire routing table of the other. If two systems reboot at the same time, there will be nothing to learn. HAs minimally should have UPNs. One might think about stable storage at some point.

G.5.1 The basic HARP topology

The two most important topological aspects of HARP are that we intend to interoperate with Mobile–IP MNs "as is" and in addition we will take advantage of normal IP unicast interior gateway protocol routing (e.g., OSPF or RIP). The trick is that we expect dynamic routing to find a path to the "nearest" HA since both peered HAs advertise a path to the "same" subnet. If one peer is lost, we expect dynamic routing to send all packets to the other peer. (Note that the mobile subnet may be partitioned and hence "virtual"). Thus a Mobile–IP registration request and a normal packet sent from a Correspondent Host will likely follow the same path to one or the other HA. Either packet may at any time go to either HA. Each paired HA must tell the other where the MN has gone, and they both must track the MN in terms of routing state. A MN will really have "two" homes with two HAs (and there is nothing to limit the protocol to two, but two will suffice for our purposes.) For the HAs to hide their differences from the MN, both must share the same "virtual subnet" IP address. They cannot communicate with each other as a result with that address, but we expect such systems to be multi–homed by definition. One "real" IP address must be used at each HA, paired with one virtual IP subnet address that is shared and attached (possibly as an alias) to an interface. This is a small price to pay for MN interoperability.

Our lab topology will be something like the following: note that both HAs are multi–homed systems and each has both a WaveLAN and ethernet card.

```
Cisco router ---------- test router (FreeBSD)
                            |
                            |
                        | hub  |  interior ethernet subnet Z
                        |      |
                HA #1       HA #2
                !           !
                wavelan subnet  X
```

The Cisco router routes mobile subnet X to the test router. The test route statically routes that subnet to either HA #1 or HA #2 in terms of their respective IP addresses for interior ethernet subnet Z. This gives us a way to simulate dynamic unicast routing as we can simply change the route to the mobile subnet by hand. The Wavelan interfaces will be configured with both the same (and different) IP addresses. From a practical standpoint, the HA WaveLAN cells should not overlap since the use of the same IP address (Home is the same) will cause confusion to the MNs when they are actually resident in both cells. We can either only really have one home (by not sending HA beacons from one system) or configure the WaveLAN Network IDs differently so that HA beacons from both systems cannot simultaneously be overheard.

G.6 Outreach

Bill Trost has visited BBN for a few days to discuss issues of porting their code into our Mobile–IP/IPSEC system.

David Reeder has attended recent IETF and the recent FTP Inc. Mobile–IP InterOP.

Jim Binkley and John McHugh have given recent talks of various lengths at the Oregon Graduate Institute, at the Baltimore DARPA PI meeting, and the GLOMO PI meeting on the UCLA campus.

G.7 Appendix - IPSEC, Mobile–IP and Policy

John McHugh has recently mailed out a short treatise entitled "Keys, Chickens, and Eggs" in which in the final section he discussed capital P 'Policy' vs little p 'policy; i.e., the desired attributes of a security system versus how the attributes were actually implemented. The point needs to be reiterated that it would be useful to actually think aloud and possibly talk to industry, firewall vendors, and to IPSEC people, about what kind of Policy might make sense in terms of Mobile–IP. Once we know, we could try and implement it. Up until now, we have addressed a very limited range of topologies, but as we begin to explore "dynamic key" issues, it would be nice to try and determine what actually might be useful and/or feasible. We shall attempt a short discussion here in order to begin to explore some of the issues. This discussion will not be complete, but it is a start. Hopefully it might fuel some high–level discussion about what needs to be done and how such Policies might be accomplished.

As a word of caution, the issues for security and Mobile–IP are complex and they are changing . Mobile–IP itself is a moving target as recently the subject of a "reverse tunnel" protocol setup by MNs where the local FA tunnels back to the HA has come up in the IETF Mobile–IP working group. As is often the case, it seems little attention is paid to security issues, even though an entirely new topological dimension has been revealed.

For the sake of argument, let us establish first some topological notions. We will try to bind our Policies according to the topologies. In conclusion, we will try to summarize any lessons learned.

Mobile–IP topologies of interest might include:

1. MNs/HAs.
   - .at home
     - on a wireless system (WaveLAN)
     - on a wired system (ethernet)
   - .away
     - via a FA
     - via a COA (but no FA)
2. MNs/FAs It is possible that a FA might have a pool of COAs. This might be useful although Mobile–IP currently does not address it as an possibility.
3. MNs/COA Here an MN acquires a COA as a local–link IP address. The IP address may be acquired via DHCP or some other mechanism including PPP and manual administration.
4. FAs/HAs
5. ad hoc MNs
6. route optimization (MNs talking directly to CHs).

There are two kinds of protocol considerations that are fundamental. One can fundamentally distinguish between "control" packets (routing including link–layer mechanisms like ARP), and "data" packets that can be transmitted after routing is established. Routing must be setup first. In certain topologies (barring ad hoc), this is what Mobile–IP tries to do. This is of course, the chicken and egg problem that BBN has referred to elsewhere. Without routing, data cannot be sent.

In the discussion below, we will mostly neglect any Policy that says "you may not do this," and always neglect Policies of the form "we don't care if you do this." We will also neglect the likely useful area of IPSEC/Firewall policy combinations.

G.7.1 MNs/HAs

At home

Policies:

1. A wireless link may be felt to be less secure than a wired link. As a result, one might require wireless security along the usual lines with the usual possibilities: encryption, authentication, session–keys.
2. Ideally such security should encompass both routing and normal data.
3. For reasons of scalability, one may very well require dynamic key management (as opposed to static). If all employees at General Motors have laptops, that will be a lot of keys. (There is not much difference between /etc/keys and /etc/hosts from the scalability POV.)
4. One may view a Mobile system as a threat simply because it can get up and walk away, possibly with valuable information. Hence it may be subject to any number of special restrictions even when at home. (can only use subnet X, and must use Mobile–IP authentication to even get routability).
5. One may have policies for the HA as a router that limits ingress to interior non–mobile networks.

Observations: Each relationship (like any other network conversation) is symmetrical. Here we have MN to HA and HA to MN. As John pointed out, in some cases, one of the sub–relationships may be less important. In this case, the HA is probably more important simply because it represents an entrance to the secure enclave. (This is of course, even more important when we are talking about routing to and from the HA when an MN is abroad).

One might view the MN at home as being within the secure enclave, unless one is worried about uncontrolled wireless exposure or uncontrolled taps into local links. The MN at home is not very different from conventional systems at home. ARP is a security exposure anywhere, although it may be deemed adequate for links like ethernet.

Certainly small–scale static key distribution for both control/data is possible. But if dynamic key distribution will occur anywhere, surely it can occur here.

Away

Policies:

1. Mobile–IP may be banned. Of course, such a Policy may be viewed as not interesting but it is of fundamental importance. Until recently firewalls existed to keep spoofers out and one form of spoofing is borrowing IP addresses that should be "inside." Therefore Mobile–IP can be viewed as institutionally (IETF) approved spoofing. Certainly many organizations will never permit it either for their employees to get back in or for "visitors" from abroad. We will not refer to such a negative policy again, but it is a fundamental challenge for IPSEC to see what inroads if any can (or should) be made about such world views.
2. One may require the MN to authenticate its control packets to the HA (and vice versa).
3. One may require the MN to secure all of its packets including data and either:
    (a) send packets to home systems securely but not care about packets sent to CH systems not at home
    (b) send all packets home first so that the MN appears to originate to CHs from home. The enclave itself may choose to not allow CHs to forward packets to MNs to suppress proposed plaintext attacks through the HA.

(c) only talk to systems at home.

(d) never talk to systems at home.

Note that again "secure" can be extended to include authentication, encryption, and session keys.

4. One may require the MN to NOT form any sort of security relationship other than with Home. For example, MNs may use FAs but may not form SAs with them, and must not have the FA forward packets into an enclave (reverse tunnel). Why, because we choose to not trust non–local FAs.

5. One may require the MN to hide any IP addresses that are native to "home" since exposure of such address topology may lead to external attacks.

Observations:

We are talking about the MN and the HA when the MN is not present on the HA's subnet. The MN/HA relationship is potentially useful in many ways, simply because the MN can pre–establish a security relationship with the HA before it wanders abroad. We can also view a MN that is abroad as creating a possible extension of a local secure enclave. Another variation on this theme is that the MN and HA by definition belong to one adminstrative/security domain.

As we have pointed out previously, ARP spoofing could be viewed alone as a denial of service attack, but if coupled with a previous acquisition of any home IP address or telnet/FTP password sent in the clear, it may become a very potent means of attack on the home enclave. As a side issue, note that not giving out IP addresses for home systems (which our current 2–was DES tunnel scheme implements) is an important attribute as well.

IPSEC as a mechanism here is very important. We also need to be able to dynamically setup "webs of trust" that include HA–FA–MN pairings.

Specification of routing policy based on IP addresses that leads to flexible routing could be useful. One might want to talk to CHs directly and avoid home OR send everything home, etc.

G.7.2 MNs/FAs

policies:

1. One may rightfully worry about wireless links again, in that they may be deemed less secure. If a network security officer cannot control wireless links at home, he/she can certainly not control foreign links. We may hence require link security when abroad and might also ban ARP. Foreign links can be viewed as much more dangerous.

2. A reasonable rationale for the existence of FAs is to charge MNs as customers. As a result, a likely policy here is that any MN will have to use a very good cryptographic means to prove its identity. Dynamic negotiation of trust (for identity as well as usage) would be a requirement.

3. Local security may require that FAs form a closed trust system (no outside MNs may use them).

4. Local security may require that FAs be outside of any secure enclave (and hence there may be no need for security relationships).

5. Certainly the home security organization might desire that MNs reveal as little as possible about themselves to a FA. This might include a requirement that the MN is NOT to form a SA with a FA.

6. FAs detunnel IPIP packets. One may require a SA between the tunnel endpoints.

Observations: It is not clear in how many ways an MN and FA might combine (or not) in security terms. It is clear that it runs the gamut from complete and dynamic security both for routing control and data (wireless link) to nothing at all. In our current Mobile–IP/IPSEC system we can run all packets to home with IPSEC and simply use a FA as a router. We could currently setup our FAs to form a "closed" trust system since we have implemented MN/FA static keys. By definition, this would exclude MNs from other domains. However we currently have no means to dynamically negotiate anything between MNs and FAs.

Note that the relationship could be assymetrical. The MN might not care so much about whether or not the FA can prove itself, since the MN is very much at the FA's mercy. If the FA will ship packets out and in, that may do from the MN's point of view. The FA however might care greatly since it may represent a portal to a "secure enclave" or might be interested in economic transactions.

On the other hand, the truly paranoid MN may be concerned about the FA since the FA is certainly a "man in the middle" and might do funny things to the MN's packets. The MN may be able to take advantage of its permanent trust relationship with the HA and use the HA to determine the FA's trustworthiness. Is "Joe Fa" really a FA? Possibly such a determination could be used to setup 2–way trust between both the MN–FA and FA–HA.

G.7.3 MNs/COAs

policies:

1. an MN might be required to ONLY use a COA ip address on outer IP packets sent home so that it can tunnel packets through a border router. Thus even the MN's IP address could be hidden from view.
2. The MN may be required to setup a SA based on the COA with the HA. This would prevent external non–secured IP addresses from entering a secure enclave. Such a Policy could extend to both control and data.

Observations: An MN may acquire a COA via some TBD mechanism (DHCP). It might even do so at a FA. Some of the previous Policies apply here (so we left them out). There is however a new wrinkle, and that is that the MN has acquired an IP address with local significance. This is a two–edged sword. It may be useful in that an MN with a local COA can use it to bypass an ingress–filter. On the other hand, one could view such a COA mechanism as a security liability.

G.7.4 HAs/FAs

policies:

1. One may require either a static or dynamic trust negotiation for at least routing packets (MIP). Dynamic keying would be needed for FAs not under local control.
2. IPIP packets may not be permitted. One might require IP IPSEC IP or IP IP IPSEC where there is some way to bind at least one of the local headers to the box "at home" that performs the IPSEC operation.

observations: We feel the most important point to make here is that IPIP can be viewed as a liability especially across security domains as it is basically a mechanism for poking an IP datagram anywhere (and shedding the IP outer header or skin that got the datagram to the tunnel endpoint). One could use conventional firewalls to protect internal "detunneling" devices that lack protection of their own. On the other hand, IPSEC is an obvious mechanism for making IPIP safer. Further one needs dynamic key distribution to enable IP IPSEC IP as a mechanism for boxes not in the same security domains.

G.7.5 ad hoc MNs

policies:

1. MNs should authenticate routing packets.
2. MNs should secure control information.
3. MNs should only talk to their own kind and maybe only when they see their own kind.

observations: The first two policies above are nothing new and other policies previously mentioned can certainly apply (e.g., concern about wireless links abroad).

It is important to note that MNs in an ad hoc situation may be worse off than an MN at a FA, simply because they can be cut off from the wired infrastructure. Both DNS and any existing Certificate Authority may not be available. (The lack of DNS certainly stops some not very bright applications from even running).

G.7.6 Route Optimization

policies:

1. We may simply disallow it as we do not want our MNs to talk to CHs except through home. Of course we may not want them to talk to non–home CHs either.

2. control packets should be authenticated.

3. previous concerns about tunnels apply as well.

observations: One can point out that routing optimization is an interesting name for a noble attempt to make MNs behave like ordinary hosts; i.e., avoid the routing triangle problem. However from a security point of view, one might not only like CH packets to go through the HA (you could log who the MNs are talking to), but worse you may want all of your traffic from the MNs to come back home first as well. Hence one would have a double triangle (MN - HA - CH and CH - HA - MN) and be happy about it.

Routing optimization is a form of source routing (as is any use of IPIP tunnels) and one might think that minimally authentication for tunneled packets is a requirement.

G.7.7 Summary

1. MNs in ad hoc situations and at FAs are at a disadvantage because they either lack connectivity or must do it through a man in the middle. How do you get a certificate from the DNS if there is no DNS? You may have to bring something with you to either minimize exchanges due to lack of bandwidth or fast motion, OR you may simply not have access to the wired infrastructure in an ad hoc situation OR you may have to establish a "face–to–face" web of trust.

2. MNs can take advantage of the a priori security relationship with the HA. We agree with BBN that a MN need not ship a certificate to the HA to prove itself since it can do something more lightweight and/or prepare a bit before it leaves. The MN can also ask the HA about a FA since the MIP protocol itself is extensible. Some sort of "in–line keying" could be done there in the MIP protocol. A FA might ask a HA about an MN (as well as about the HA itself) as we can presume that the FA/HA bandwidth might be greater than the MN/FA bandwidth.

3. Tunnels need to be secured by IPSEC and be setup via some dynamic exchange of key material since security domains may be crossed. Note that Home Agents may detunnel packets as well as Foreign Agents.

4. If Policy dictates that FAs need keys, FAs need more than manual key mechanisms, although such a mechanism may have limited utility in a small network infrastructure. FAs need to dynamically negotiate with HAs in order to setup tunnels and to make sure that all packets coming through a tunnel are secure. This is an absolute MUST. FAs MAY need to negotiate with MNs where a security policy dictates that MNs must authenticate themselves and MUST negotiate with MNs wherever FAs might charge for their services.

5. One must utter the "scalability" word in terms of manual keying. This may be deemed obvious, but one will have more FAs than HAs, and if you use manual keying with FAs, you must add each new FA key to all MNs and teach that FA about all MNs. This is less scalable than the MN–HA relationship. Manual key distribution may also be hard to automate. It can be done where FAs are concerned since we assume they are reachable. But given that laptops may be off or away, the laptop side is very hard to automate in practical terms.

6. In an environment where an MN acquires and uses a COA, the neighborhood local router might choose to impose security policies on the MN. The kinds of policies between such a router and a FA may not differ very much.

7. This last set of router policy considerations may be extended to filtering border routers and might require some TBD protocol that can discover the number of policy–oriented routers between here and there (and back). Such a system begins to sound similar to other problems facing the Internet (e.g., RSVP setup management, or nested tunnels and "soft state"). We need to consider the dangers of creating a system that requires "hard state" end to end, so that if dynamic routing must occur, a "connection" will be lost. Such a system would not be very much like the current Internet and might be very hard if not impossible to debug.

Appendix H

Quarterly report – Spring 1997

H.1 Project Status Overview

Our Spring work was focused on finishing a couple of key infrastructure projects and there are now two pieces we can point to that are complete. Bill Trost and Jim Binkley ported our entire code base from FreeBSD 2.1.0 to 2.2.1. We have for the most port upgraded roughly 26 agent and mobile node systems with a very few remaining exceptions. This was a tremendous task and we are over the hump at this point. In addition, David Reeder finished the kernel work so that our IPSEC/route binding that includes Mobile–IP tunnels is now complete and in 2.2. We can now build on that work to include more IPSEC functionality in our Mobile–IP implementation.

We have made progress in other areas but the work is not complete yet. Our two graduate students continued their work on redundancy issues. Papers being worked on include one by Sarah Mocas and Jim Binkley, and one by Sarah and Tom Schubert.

We will probably make at least one summer release of our IPSEC/Mobile–IP and possibly two (say in early and late summer, depending on when ISAKMP has been successfully integrated into our Mobile–IP system).

Since the previous winter report was so long, and this quarter has for the most part been spent on implementation as opposed to design, this report will be briefer than average. It will however include a short summary of ISAKMP work by Bill Trost, and a commentary by Jim Binkley on Mobile–IP and firewall issues.

H.2 Accomplishments for Spring 1997

1. Bill Trost and Jim Binkley ported our kernel sub–systems including Mobile–IP and IPSEC, WaveLAN drivers, affected routing apps (`route(8)`, `arp(8)`, etc.), and Mobile–IP code from FreeBSD 2.1.0 to FreeBSD 2.2.1. The port is complete. We will touch on it briefly below. This version of FreeBSD will form the basis for subsequent releases by us of our IPSEC/Mobile–IP system.

2. our WaveLAN driver ports are becoming more widely used. Michael Smith, a core member of the FreeBSD team, ported our ISA WaveLAN driver into FreeBSD 2.2 and we were able to reuse that work in our port. Tatsumi Hosokawa has included our PCMCIA driver into his FreeBSD PCMCIA package. We also reused that code in our 2.2 port.

3. David Reeder has finished the kernel IPSEC/route binding that we have been working on for quite some time. This means that AH/ESP in any combination can be used with kernel routes including IPIP tunnel routes of special interest to Mobile–IP. The latter feature is apt to be unique to our implementation.

4. Our FreeBSD 2.2.1 system (at least for the kernel including WaveLAN drivers) has been internally released to BBN. We have given accounts on our internal server to Greg Troxel and Matt Condell of

BBN. We know that they are in the process of internal deployment of a WaveLAN based radio network.

5. Tom Schubert and Sarah Mocas have continued efforts on formal analysis of Mobile–IP protocols. These activities have included:
   (a) developing SMV (model checker) specification of Mobile IP.
   (b) continued development of journal paper.
   (c) preparation/presentation of 5 minute talk for Oakland.
   (d) preparation of DIMACS abstract (submitted mid June).
6. John McHugh prepared a position paper on initial key establishment to guide discussion at the Memphis IETF meeting.
7. A paper entitled *Dynamic Network Reconfiguration Support for Mobile Computers* by Jon Walpole, Jon Inouye, and James Binkley was accepted for MibiCom'97. The abstract of the paper is as follows:

> Hot swapping technology combined with pervasive heterogeneous networks empowers mobile laptop users to select the most appropriate device for their current environment. Unfortunately, the majority of system software remains customized for a particular network configuration, and assumes many characteristics associated with the network environment remain invariant over the runtime of the software. Mobility produces changes in the environment and nullifies many of these assumptions. This leads to severe loss in system functionality when resources are lost and failure to benefit when resources are gained.
>
> This paper presents a model for device availability where assumptions about a network device are made explicit. When these assumptions are invalidated, cross–layer notifications trigger reconfiguration operations. Each layer adjusts in an intelligent manner, adapting to a new set of assumptions based on a user–configurable policy. An implementation of this model demonstrates its effect on a variety of applications run while migrating between different network environments.

H.3 Current Personnel and Projects

Jim Binkley will act as a half–time manager for Bill Trost, David Reeder, and lead the research of our two graduate students Bjorn Chambless, and Jennie Ye.

John McHugh will as usual edit reports, represent us at any necessary off–site meetings, and consult on related security issues.

Sarah Mocas is going to lead the research for a new graduate student (replacing Xu Hao) named Zhong Chen, who will be joining us this summer. Sarah will begin investigating the use of a simple (and new) public–key based authentication sub–system to be used with both Mobile–IP and ad hoc routing protocols. Sarah is working with Jim Binkley on a paper that will summarize our Mobile–IP security views. We hope to submit that paper to an upcoming IEEE Transactions on Software Engineering, special issue on Mobility and Network Aware Computing.

Sarah Mocas and Tom Schubert have submitted an abstract entitled "Formal Analysis of IP Layer Security" to the DIMACS Workshop on Design and Formal Verification of Security Protocols, to be held September 3–5, 1997 at the DIMACS Center, Rutgers University. Tom and Sarah have modeled different security mechanisms and protocols that are currently being developed for use at the IP network layer. This work has been done as an interplay between formal logics and mechanized systems. In the context of Mobile IP, they are currently using formal logics to examine mobile registration. Specifically, mobile registration requires the use of an untrusted route to establish a secure tunnel. We model the security association implicit in mobile registration.

Bill Trost (programmer) has been working with Jim on 2.2.1 porting and deployment, and is now focusing on the integration of ISAKMP into our route–based IPSEC security system. Both Bill and David Reeder attended the Memphis IETF meetings. Bill's principle area of focus in addition to kernel internals is integration of our Mobile–IP IPSEC architecture with public–key cryptography–based protocol systems.

David Reeder (programmer) has just completed the kernel work needed for our IPSEC/route bindings. David will now focus on the actual use of those IPSEC bindings in our Mobile–IP system. We intend to finish the implementation (started long ago) of a manual configuration system that would cause Home Agents and Foreign Agents to automatically install authenticated IPIP tunnels as opposed to non–authenticated IPIP tunnels. David will also fix various Interoperability problems found at the Spring FTP Interop and these fixes will be available in our next Internet release.

Bjorn Chambless, a Master's Student, is working on the implementation of HARP or Home Address Redudancy Protocol, which we hope to finish this summer. He will be full–time employed this summer by the project. This quarter we discovered that although we had designed a reasonable and simple protocol, the integration of the protocol into our Mobile–IP agent daemon infrastructure was anything but simple. As a result, we are in the process of rearchitecting the state machine side of the Home Agent in order to make the integration of HARP easier. When we finish the HARP work, we will write a draft RFC on HARP and submit it to the IETF. As Bjorn will not graduate for awhile, we will probably put him to work on Foreign Agent redundancy eventually.

Jennie Ye, has implemented the Mobile Node side of the Multicast Ad hoc Routing Protocol this quarter, and is beginning to test the protocol itself. She will be full–time employed this summer by the project. Our goal for summer with Jennie is to try and finish at least the Mobile Node side of the protocol. Eventual milestones in this project include:

1. MN to MN multi–hop routing discovery
2. MN to agent
3. integration of the protocol with Mobile–IP
4. integration of this protocol with the previous ad hoc beacon protocol to form one demand based (as opposed to broadcast based) protocol.
5. a written technical report on the protocol.

We expect this small sub–project to continue through next year.

Zhong Chen is a 1st year M.S. student at PSU and has considerable experience with TCP/IP, which he has taught in the past.

H.4 Future Release Deliverables

We have essentially just made an internal release of our combined FreeBSD 2.2.1 and IPSEC/Mobile–IP system that includes 2–way DES tunnels (HA/MN). During the early part of the summer we will issue DES keys to our laptop users in order to stress this system and in order to see what actually happens when a number of users simultaneously use DES with a single Home Agent.

We are looking at probably two releases to the Internet over the summer and early fall depending on the exact system deliverables we wish to include. We suspect that an early release of a combined Mobile–IP/IPSEC would be a good idea simply to get it out. On the other hand, we want to eventually include ISAKMP and the HARP protocol, but it is hard to predict when those efforts will be done.

We are currently working towards a early summer release that will minimally have the following attributes:

1. Our IPSEC and Mobile–IP systems including WaveLAN drivers ported to FreeBSD 2.2.1
2. The now completed kernel route binding for IPSEC. This includes `arp(8)` and `route(8)` commands and various IPSEC combinations for routes including for example:
   (a) `route dst gateway -ah -spi ...`
   (b) `route dst gateway -esp -spi`
   (c) `route -ahtunnel dst gateway -ah -spi`
   (d) `route -esptunnel dst gateway -ah -spi`

In other words, ordinary routes can have either ah/esp bound to them, and tunnel routes can have both those bindings and two more (-ahtunnel and -esptunnel) that apply to the appended COA outer IP header itself. Thus those bindings can be applied say to packets sent from a Home Agent to a Foreign Agent (or Mobile Node using a DHCP acquired COA).

3. A manually configured system that allows secure tunnels to be setup between Home Agent and Foreign Agents. Policy is expressed in the agent configuration file, and the routes between HA and FA are automatically installed with IPSEC bindings when needed.

4. Bug fixes in our Mobile–IP daemon system based on the FTP Interop experience.

Note that for any release we will have to cooperate with MIT again and rerelease the cryptographic part of the IPSEC code (/sys/netsec) as it has changed due to bug fixes and porting.

Later on when we finish ISAKMP work (i.e., change the kernel and daemons so that we can dynamically bind security associations to routes used in Mobile–IP) we will make a release that includes it. We would also hope to include the HARP protocol which should be done by the end of the summer and an integrated IPSEC/ARP mechanism.

One stumbling block that we hope to fix over the summer is that our Mobile–IP implementation changes routes by executing the `route(8)` and `arp(8)` binaries. This is not good in that it has always made error logging more difficult, and has certainly not made hand offs efficient. We intend to fix this problem over the summer and replace the Mobile–IP routing module with one that uses the `route(4)` routing socket directly. This will allow us to finally integrate our IPSEC/routing as used in Mobile–IP with traditional ARP as opposed to our current authenticated ad hoc beaconing protocol. In order to do IPSEC with ARP, we need to be able to detect when the kernel installs a low level link layer route. We will be able to do this by binding a rarely used BSD routing flag (XRESOLVE), which causes `route(4)` messages to be sent to daemons that have a route socket open. The per interface “clone” route (which is bound to a subnet) causes incoming ARP requests to create new link layer per IP destination routes. ARP itself fills in the MAC address as the route gateway. Since the clone route will have an XRESOLVE flag bound to it, each new instance of a ARP route will cause a message to be sent to the routing daemons. We will also bind a BLACKHOLE flag to initially created routes which will prevent packets from exiting the system until a security policy lookup and binding takes place. Thus our daemons can learn about ARP requests and bind IPSEC attributes safely to different link routes in a policy–based fashion. We hope to get IPSEC/ARP functionality into our late summer release.

H.5 Public–Key Cryptographic Work

Given that there were a number of possible avenues to go down in this area including ISAKMP, possible integration of BBN’s MOIPS, and Fortezza, we decided to prioritize on getting ISAKMP integrated into our Mobile–IP architecture and system first. Once we have done that, we will reconsider what to do about Fortezza and MOIPS (although possibly BBN might adapt our Mobile–IP and do that work for us). Bill Trost has written a short report on his ISAKMP work which will follow in the next section.

John McHugh, Jim Binkley, and Sarah Mocas have decided to use one new graduate student and attempt a small–scale authentication protocol that would simply use digital signatures. Sarah Mocas will lead this work and Jim and John will advise as needed. Given that BBN’s protocol uses a session–key based approach, we thought it would be interesting to attempt something different and simple that would use signatures for authentication (e.g., RSA or DSA). Our goal would be to design a Tag–Length–Value based scheme that could easily replace our MD5–based authentication mechanisms used in both Mobile–IP registration and ad hoc routing protocols. The same scheme would be used in both cases. All these protocols do after all, is establish routing. ISAKMP would then be used once routing was established for all higher–level protocol security. In essence, we would have a two–tier security system (like our current ad hoc + static IPSEC system), but public–key cryptography would be available in both cases.

Such a protocol should minimize the number of messages needed and the size of the packets. Hence we would simply sign the packets, but not include certificates. (This is in accord with BBN’s thinking in MOIPS). We have briefly investigated the speed of signed MD5 hashes and think that a signature based mechanism is not unreasonable given the infrequency of registration packets and the current (and ever increasing) speeds of CPUs. In the first cut, we might simply use a flat file lookup system for public keys. Eventually such a system can be replaced with a certificate lookup system.

As an upper bound, http://www.rsa.com/rsa/developers/bench.htm, the RSA web page claims the following speeds for their proprietary BSAFE implementation for signing and verification on a Pentium–90 system (which is slow by currently available Pentium standards. We have several laptops that are faster).

512 bit keys	**seconds**	**ops/sec**
RSA/P90/sign	.024	41
RSA/P90/verify	.0027	370
DSA/P90/sign	.029	34
DSA/P90/verify	.052	15

These numbers suggest that signed packets when sent with the infrequency of current Mobile–IP registrations are computationally feasible. In our implementation, Mobile–IP nodes reregister on the order of once per minute. The more important number is probably the verification speed. This is because a given agent (especially a Home Agent) may be called upon to verify many Mobile Nodes. The scalability of verification at a Home Agent should be a concern. Of course, we have not yet factored in any costs that might be caused by a higher level certificate lookup system (perhaps available to the HA via DNS). It may be useful for us down the road to consider whether or not the HARP protocol can be modified to allow distributed authentication amongst a cluster of Home Agents. As a secondary consideration, verification speeds are probably also important for our new generation multi–hop ad hoc routing protocol. In our protocol, intermediate Mobile Nodes would have to both sign and verify multicast packets if the intermediate system were to modify the packet (e.g., appends its own path tuple).

We will have a design to present for this sub–protocol in the summer report.

H.6 ISAKMP (Bill Trost)

We have been working with the implementation of the Internet Security Association and Key Management Protocol (ISAKMP) and the Oakley protocol that Cisco systems is distributing. This implementation, called ikmpd, supports Oakley's Main Mode and Aggressive Mode exchanges for setting up the key management protocol, and Quick Mode exchanges for creating subsidiary security associations for use with data traffic. Under some circumstances, ikmpd will also use Informational Exchanges to let another party know when there is a problem with the creation of a security association. It may use DSS signatures, RSA encryption, or preshared keys to create the initial security association. The implementation does not provide for the expiration of security associations, but this is more a result of limitations of the IPSEC implementation being used than of Cisco's implementation of ISAKMP.

Cisco's most recent distribution of their ISAKMP implementation has a variety of problems. As distributed, the program is not even usable, as it has IP addresses hard–coded into the program in various places, and the program does not attempt to perform a DSS exchange, even if that is the only keying information available. When creating a security association in Quick Mode, the distributed program would use the IPSEC algorithm identifiers when instantiating the security association in the kernel, instead of using the operating system's algorithm identifiers.

Once these simple problems have been fixed, the program is still very brittle. Missing RSA keys cause the program to transmit malformed ISAKMP packets, and failure to create a security association sometimes induces fatal memory errors in ikmpd.

In addition, our research environment has uncovered a problem that is relatively unlikely in a workstation environment. Many of our computers have two network interfaces, one wireless to provide support for the mobile laptops, and an ethernet interface to provide Internet connectivity for the computer and the laptops it serves. If a laptop attempts to establish a security association with the wired interface of a computer it was using for its Internet connectivity, responses from the wired computer would be transmitted using its wireless address. A Mobile Node, unable to determine that the wireless address corresponds to the same computer that it is trying to establish a security association with, discards the response it received and retransmits requests to the other computer's wired address, until it eventually abandons the attempt to establish the security association because it has seen no responses.

We have submitted fixes to these problems to Cisco, and the implementors there welcomed our suggestions and have indicated that our fixes will be included in the next release. In the case of the problem with multi–homed hosts, the solution we provided is to have ikmpd use the hostname assigned to the (agent) system when it starts up for all ISAKMP transmissions. This solution is adequate for our environment but will not work for computers which have multiple addresses assigned to the same name.

Cisco's implementation of ISAKMP is surprisingly slow. Profiling analysis indicated that the program spends a large amount of time generating random numbers for use in the protocol. FreeBSD provides a random number generator in the operating system based on keyboard activity and other interrupts received by the system. We modified ikmpd to use this source of random numbers should it be available. This change dramatically improved the program's performance. As Cisco does not use FreeBSD, they will be unable to take advantage of this improvement, although they may choose to distribute it.

The purpose of our work with ISAKMP and Cisco's implementation thereof is to integrate its dynamic, public key session creation mechanism with our changes to allow IPSEC security associations to be attached to routes. When this work is complete, a security association can be attached to a route using a special "wildcard" security association. This special security association will tell the operating system that it must request the key management service to negotiate and install a security association, which will then be used for all traffic on the route. Most of the changes to support the attachment of dynamically–negotiated security associations to routes must be made in the operating system itself; only one small change to ikmpd will be necessary. We hope to design and finish a ISAKMP route binding work this summer. Our spring work has helped us gain familiarization with the Cisco implementation.

H.7 Considerations on the Mobile–IP Firewall Problem

The Mobile–IP working group has spent some time attempting to deal with the perceived problem that current Internet border router filtering schemes may prevent Mobile–IP from working in a inter–routing domain context. There are security problems with the solution the working group has proposed. We believe we have a solution, and hope to pursue research and implementation in the next year that will demonstrate that the problem can be solved. To some extent, we feel we have already demonstrated a working solution to the problem.

The Mobile–IP "ingress filter" problem is this: CERT advisories suggested that routing filters should be installed that would prevent packets with IP source addresses that should be internal to a site, from crossing from the outside world into the site. The advisories were prompted by attacks on protocols like X, BSD lpr, NFS, BSD rsh and the like which have weak authentication systems based only on IP addresses. More recently an advisory was issued that suggested that packets that leave a routing domain possessing IP source addresses that are only externally valid should also be blocked. This was done due to the well–known TCP syn attack problem. A filter of this type may be called an "egress filter".

Both ingress and egress filtering will cause problems for Mobile–IP. By definition, Mobile–IP systems retain their "internal" IP source addresses. (Note that this does not apply to Mobile–IP itself as a registration protocol since Foreign Agents forward Mobile–IP packets as proxy systems; i.e., the UDP registration packet will have the FA's "outside" IP source address and hence that packet will not be filtered out.) Ingress filtering means that a Mobile–IP system in an external domain will not be able to talk to systems at home. The "egress filtering" problem is actually more serious in its consequences to a Mobile–IP "visitor". An "egress filter" would prevent the Mobile–IP system from talking to any external systems, at home or anywhere on the Internet.

The recently proposed Mobile–IP solution is that agents must support "reverse–tunneling" upon the request of a Mobile Node. Normal Mobile–IP tunnels might be called "forward–tunnels" and would apply to packets forwarded from a Home Agent to a Foreign Agent. "Reverse tunneling" means packets that are forwarded from the Mobile Node (or possibly Foreign Agent) back to the Mobile Node's home routing domain. IPIP (proto 4) would be used in either case.

The idea that a Foreign Agent might reverse tunnel packets back to a routing domain is not a good idea for security reasons. In our current PSU IPSEC/Mobile–IP implementation, we are capable of using IPSEC to reverse tunnel home directly from the MN to (and back) from the HA. We make sure that all (non Mobile

IP registration) packets that go through a FA are encrypted. Hence the FA can only route and cannot "peek" at the MN's traffic. From a high level point of view, we believe that a Mobile Node should minimize its reliance on the FA and maximize self–reliance. A Mobile Node ideally should minimize security exposure and associations with external "unknown" hosts. It can (and must) always rely on its Home Agent, since all its traffic traverses the Home Agent. Hence we simply tunnel over the Foreign Agent using IPSEC.

Further, we have pointed out to the working group that such FA–based tunneling is not a solution since it simply violates the spirit of the CERT based filters. IPIP tunnels are a way to sneak in over the top of the CERT routing filters. IPIP itself as a protocol can easily be blocked (IP protocol 4) and may eventually be blocked on a wide basis if Mobile–IP becomes widespread. Indeed, in terms of current firewall infrastructure (cite Chapman) IPIP must be blocked in order to prevent a new mechanism for attacks via Mobile–IP agents on NFS and other protocols that rely on IP address–based authentication. It is hard to see a solution to this problem that does not involve strong cryptographic based solutions like IPSEC.

We have tested the ingress filtering problem by placing a Foreign Agent at Oregon Graduate Institute, which is external to us in routing terms, and where there has been at least one NorthWestNet border router with CERT ingress filters installed. The problem is real. Without extraordinary measures, a visiting PSU Mobile Node cannot talk to home.

We found that we could however set up Mobile Node based "reverse" tunnels and tunnel through our Home Agent at PSU to selected hosts. We combined IPSEC–based routing tunnels (using ESP/DES) with an IP src address stolen from the local OGI Foreign Agent. The packets sent to the Home Agent had the structure <IP><IP><ESP>, where the outer IP header used the FA's IP source address. These packets could cross the border router because the FA's IP source address was internal to OGI. The security association on the inner IP header was between the MN's "home" address and the Home Agent. Hence we will claim we were able to securely send packets home.

The notion of allowing a MN to borrow the FA's local address (as a COA) is a hack, and although it does seem to work, is not the best network engineering. The Mobile Node COA should be unique for a number of reasons; e.g., the IP identity field on the outer header should be unique per system and there might be clash if the same COA was used causing fragmentation to fail. However objecting to this mechanism does not disprove our solution, since FA's could easily be combined with DHCP or some other mechanism to give out unique local–link IP addresses (COAs) to visiting Mobile Nodes. During our testing we demonstrated that more than one Mobile Node could simultaneously use such a FA address and setup IPSEC/ESP host routes back to specific hosts and networks at PSU. Our current implementation is not elegant, but it allows us to suggest a more reasonable solution.

Our suggested solution has three fundamental principles:

1. When abroad, MNs should be able to acquire unique COAs (care–of addresses; i.e., local link IP addresses). It does not matter if an MN is at a Foreign Agent or not. We will call this the "COA acquisition" principle.

2. IPIP is a routing mechanism that allows tunneling across or through current CERT ingress filters. It must be made secure by the use of IPSEC. We will call this: "IPIP needs IPSEC".

3. IPIP itself must be blocked at firewalls for the foreseeable future. We plan to initiate such blockage on our own PSU engineering (and OGI) border routers within the next year.

H.8 COA acquisition

Mobile–IP has tended to view MNs at foreign links as possibly having two completely different operational modes. MNs may forward through Foreign Agents or they may act as their own FA if they can acquire a local COA (for example, via DHCP). These modes are not really mutually exclusive and there is no reason they cannot be combined.

From a high–level routing point of view, the most important idea in Mobile–IP is that MNs, when away, prepend a COA IP address to their conventional "home" IP address. The MN must inform the Home Agent, which will tunnel packets for the inner address to the outer address. Where the COA comes from is immaterial. Hence we believe that Mobile Nodes can fundamentally solve the "ingress filtering" problem by simply acquiring a COA and making sure that packets sent abroad (and possibly routed through home on the way) leave with a reasonable COA.

The real question here is what routing implementation model is needed in terms of the Mobile Node's own operating system stack. As pointed out by cite Cheshire 4x4 routing paper, Mobile Node routing flexibility is needed and may even require user intervention in some cases. Our "theft" of the FA's COA works. Making DHCP available for "visitors" is a better solution. Teaching Mobile Nodes how to juggle both local COAs and their real "permanent" IP address (along with combining it with IPSEC) in terms of a given operating system routing implementation is the true challenge.

H.9 IPIP security issues

Mobile–IP might skirt the "ingress filtering" problem in a trivial way anyway since a site that has a firewall may well have a perimeter defense system and agents that have external access could easily be bastion hosts (be in the "no man's land" between the external firewall and the internal harder firewall). This would solve the egress problem for visitors. Conventional site security administration would probably forbid same site Mobile Nodes from talking to systems within a perimeter defense anyway (unless strong cryptography is used, say with SSL, ssh, IPSEC, one time passwords, or the Microsoft PPTP tunneling scheme). The ingress problem is solved by insisting that all foreign hosts (including Mobile Nodes) talk only to bastion hosts and/or use protocols that involve strong cryptography. It is not clear that new firewall technology just for Mobile–IP is needed. Current firewall techniques (bastion hosts and carefully designed filters) can be used to make Mobile–IP safe for all. Frankly, it has never been clear that there would be that much inter–routing domain Mobile–IP anyway. Same domain use is much more likely.

IPIP itself is dangerous from a security point of view and should be blocked unless combined with IPSEC. We have implemented a simple filter in our agents that can be turned on when packets are detunneled. The filter asserts that IPIP packets must either already have successfully passed a IPSEC based security association check with the agent or have a security association yet to come in the packet header with the same agent. In effect, this means the sender must have a IPSEC security association with the agent. Packets might have either an <IP><IPSEC><IP> structure or a <IP><IP><IPSEC> structure. The former case could be of possible use between Home Agents and Foreign Agents for packets sent to remote Mobile Nodes. We are currently in the process of creating a manual–key based IPSEC security relationship system for Mobile IP that only applies to HA to FA packets. The latter case (<IP><IP><IPSEC>) applies to packets sent from Mobile Nodes (bypassing the FA) to the Home Agent, when the MN is away from home and is something our current implementation can do. In both cases, IPIP turns into IPIP + IPSEC and can hence be rendered safe for Internet use. Either system could eventually use public key cryptography and certificates for dynamic agent and Mobile Node security relationships.

H.10 Formal Methods for Protocol Analysis

To reason about Mobile-IP, a simple SMV specification is being developed that does not utilize IPSEC mechanisms. The specification describes agent discovery, registration and HA-FA tunneling. I hope (not) to find attacks by adding an adversary to the specification and modifying the specification to include the IPSEC mechanisms. I will then attempt to extend the specification to include ad hoc routing and redundancy mechanisms (use of multiple FAs, HARP).

After developing a simple SMV model of mobile registration, it was determined that the simplicity of SMV's specification language was not adequately expressive to define general routing structures (in particular, SMV does not provide arrays). A brief examination of the Murphi checker suggests that it might be a more appropriate tool. However, the focus of this summer's activity will be on further developing a model in the HOL90 theorem prover.

The abstract submitted to the DIMACS workshop appears below.

Formal Analysis of IP Layer Security
Sarah Mocas and Tom Schubert

{*sarah,schubert*}@*cs.pdx.edu*
Department of Computer Science
Portland State University

The use of formal methods during the development of a security protocol permits the designer to identify subtle assumptions that can lead to security flaws. Automated support for using formal methods can greatly assist this process, however, for many applications, correct formal modeling of a protocol and identification of its underlying assumptions is initially of greater importance than automated proofs. With this in mind, we have modeled several different security mechanisms and protocols that are currently being developed for use at the IP network layer. This work has been done using an interplay between modeling protocols via formal logics and embedding protocols in a mechanized system.

The use of informal modeling has motivated the development of a mechanized system. We develop a framework to formally reason about cryptographic protocols using belief logics with the HOL theorem proving system. The prover allows identification of the sequence of inferences used to conclude why a property about a protocol does or does not hold. The automated procedure informs the user what rules are relevant and when rules are applied to a goal.

Using formalizations, we have examined key escrow protocols[4] and proposed IP security mechanisms. Current work pursues the analysis of secure mobile networks.

We have modeled several of the proposed IPSEC security extensions to determine the soundness of these propositions. Specifically, we examine the services that should be provided by the addition to an IP packet of the authentication header, AH, and the privacy header, ESP (encapsulation security payload). Through formal methods, we show where protocols using "secure packets" succeed in providing the service for which they are intended and where they fall short.

In the context of mobile IP, we are investigating the use of formal logics to examine IP layer security mechanisms. There are several security mechanisms needed for a reasonable implementation of a secure mobile network. Initially, mobile registration requires the use of an untrusted route to establish a secure tunnel. We model the security association implicit in mobile registration.

Belief logics are designed to consider what conclusions individual parties (principals) in a communication dialog can deduce based on messages received and a set of initial assumptions or beliefs. Belief analysis attempts to show that only desired properties are guaranteed by the communication (data confidentiality, message authorship, non-repudiation, no replayed transactions, etc). Note that proofs about idealized protocols are not a guarantee that the concrete protocols are correct. There are many implementation assumptions that if invalid, would cause a secure, idealized protocol to actually be insecure. For example, these logics all assume that the crypto algorithm is secure. Several logics for analyzing cryptographic protocols and authentication schemes have been proposed (see for example [1, 2, 3, 5, 6]).

To better support the use of mechanized belief logics, we have developed a general infrastructure for creating automated and interactive procedures to prove goals about inductively defined relations. The infrastructure provides tools to build interactive functions and goal-directed support functions, procedures to eliminate existentially quantified variables from terms, and tactics to generalize existentially quantified rules. Also provided is a tactic generating function that returns a tactic specialized for a list of inference rules and a list of combinators. The tactic searches for and then applies, an appropriate rule to apply based on the current goal. In practice, this tactic appears to provide a significant improvement over tactics that try every rule. The tactic also outputs what rules where chosen to apply in a given situation. The displayed list of decisions can assist the user understand why a proof is correct and often, why it fails.

Bibliography

[1] M. Abadi and M. Tuttle. A semantics for a logic of authentication. In *Proceedings of the Tenth ACM Symposium on Principles of Distributed Computing*, pages 201–216. ACM Press, 1991.

[2] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1), Feb. 1990.

[3] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy*. IEEE Computer Society Press, 1990.

[4] Tom Schubert and Sarah Mocas. A mechanized logic for secure key escrow protocol verification. *International Workshop on the HOL Theorem Proving System and its Applications*, September 1995.

[5] Paul F. Syverson and Paul C. van Oorschot. On unifying some cryptographic protocol logics. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, May 1994.

[6] P. vanOorschot. Extending cryptographic logics of cryptographic logics of belief to key agreement protocols. In *Proceedings of the First ACM Conference on Computers and Communications Security*, pages 232–243, 1993.

Appendix I

Quarterly report – Summer 1997

I.1 Project Status Overview

During the summer, substantial progress was made on a number of fronts as can be seen from the body of the report. At the same time, the project has suffered significantly from the loss of many of its key personnel. The Biggest loss has been the departure of Co-PI Jim Binkley who left to take a position at the Oregon Graduate Institute. Jim has been the primary researcher concerned with the networking aspects of the project. He will remain involved in the project, but at a lower level. In addition to Jim, the two full time staff members also left in September. Tom Schubert, who worked on protocol analysis left in September to join Intel. Remaining at PSU are John McHugh, Sarah Mocas, and three graduate students.

Uncertainties concerning FY 1998 funding prevent us from seeking to replace full time staff.

I.2 Summer Accomplishments

- In mid summer we made a combined release of Mobile-IP/IPSEC for FreeBSD version 2.2.1, which at that time was the latest version. Our code will however work with a subsequent bugfix release, 2.2.2. This release combined IPSEC bound to routes in a Virtual Private Network like fashion, where routes with IPSEC attributes (ESP) are automatically installed/changed/removed according to topological attributes. E.g., if a Mobile Node is at home, it might install security between itself and the Home Agent. If away at FAs, it might always install 2-way ESP tunnels between itself and the Home Agent. This work covers ad hoc mobile nodes as well, because a Mobile Node may, for selected other mobile nodes, install VPNs as link layer routes between itself and another Mobile Node.

- The Home Agent Redundancy Protocol was tested and demoed. We are currently in the process of finishing it and writing a draft-RFC that should appear in the next month or so. We are also planning on deploying HARP at PSU and at OGI during fall quarter in real mobile networks.

- Sarah Mocas and Tom Schubert presented a paper entitled "Formal Analysis of IP Layer Security" at the Workshop on Cryptographic Protocol Design and Verification which was conducted as a part of the DIMACS Special Year on Networks. The paper abstract is as follows:

  The use of formal methods during the development of a security protocol permits the designer to identify subtle assumptions that can lead to security flaws. Automated support for using formal methods can greatly assist this process, however, for many applications, correct formal modeling of a protocol and identification of its underlying assumptions is initially of greater importance than automated proofs. With this in mind, we model several different security mechanisms and protocols that are currently being developed for use at the IP network layer. Specifically we model the proposed IPSEC security extensions, AH and ESP, to determine the soundness of these propositions. In the context of Mobile-IP, we investigate the use of formal logics to examine IP layer security mechanisms but especially mobile node registration.

- John McHugh, Bill Trost, and Jim Binkley engaged in a successful test of long-distance use of the ISA-based WaveLAN cards (915mhz) between yagi antennae and more interesting, between yagi and laptop units. This is discussed in more detail below.

- Sarah Mocas and her graduate student developed a simple authentication protocol for Mobile-IP that uses digital signatures. This protocol is a replacement for the RFC2002 Mobile-IP shared secret system. At some point, this system could presumably be tied to digital signatures stored in the DNS, courtesy of DNSSEC. This is discussed in more detail below.

- One of our graduate students has finished implementation and testing of our multi-hop ad hoc routing protocol, called MADRP (Multicast Ad Hoc Routing Protocol). The protocol works between mobile nodes that are more than one hop apart. It also allows mobile nodes to talk to random hosts that are not part of the mobile infrastructure courtesy of any Mobile-IP agent. We are not aware of any other ad hoc work that has yet tied ad hoc routing directly to Mobile-IP so that nodes can still communicate with the Internet.

- Bill Trost demoed the use of ISAKMP bound to our Mobile-IP routing/IPSEC system. This is discussed in more detail, below.

- There has been a surprising amount of interest in our FreeBSD port of the ISA WaveLAN driver. Originally this driver was ported to FreeBSD by Jim after it was kindly given to us by Robert T. Morris, Jr. of Harvard. A number of companies and private individuals are using it to construct WAN T1 substitutes. Apparently Lucent has recently enhanced the 2.4 version so that it has sub-channels, and as a result, repeaters can be built with multiple 2.4 antennae. The price of WaveLAN equipment has also recently dropped by roughly one third.

- Jim Binkley has moved to Oregon Graduate Institute, but is still working with John McHugh, Sarah Mocas, and the PSU graduate students. Jim has deployed a small PSU/mobile network at Oregon Graduate Institute consisting of a Home Agent and two Foreign Agents. The mobile network is deployed in the vicinity of OGI administration and Computer Science and Engineering departmental buildings. Current users consist of the three network engineers (including Jim) at OGI. It is hoped that OGI will capitalize on this initial deployment and invest more heavily in mobile networking based on other OGI grant funds. Jim is exploring collaboration with David Steere, and Jon Walpole, who are fellow OGI/CSE faculty members.

- David Reeder has gone to work at Trusted Information Systems. We are sorry to see him go but we are glad that we were able to act as a stepping stone in his future network security career. We wish him well. Before departing, David finished the MIP/IPSEC work, so that the current system (not released yet) can use NRL's AH/AH+ESP, as well as the original ESP only mode.

- Tom Schubert has left PSU to join Intel where he will manage a group performing hardware verification on the next generation of microprocessors.

- Bill Trost has left PSU to devote more time to expert witness work. He will remain involved in the project, *pro-bono*, until the ISAKMP material is released.

I.3 Ad Hoc Routing

During the summer, we implemented and tested our multi-hop ad hoc routing protocol. Although there were a few problems on the way, it seems to work under the limited test conditions we can provide. We also were able to enhance it in a rather simple way so that nodes that are out of direct contact with a Mobile-IP agent can still access the Internet and still work with Mobile-IP.

To review, the protocol has the following basic ideas associated with it:

1. a node will multicast for another node.
2. intermediate systems according to a limited flooding algorithm; e.g., sequence numbers are used and the search is limited by ttl, may choose to forward the packet as long as it is "fresh" and authenticated.
3. an end system that is the proper receiver will reflect (reply) to the answer. According to network management setup, answers may be multicast (so others may learn) or unicast.
4. with either queries or replies, all participating systems setup host routes for either the IP source or IP destination.
5. a routing metric system exists that applies to the mesh as a whole (and in truth applies to one particular device). The metric consists of a weighted three-tuple of (signal strength, power, hop count). Individual nodes decide if they wish to forward packets or not. They may choose to drop out if for example, their own power reaches a minimum threshold.
6. agents also act as routing "sinks"; i.e., they return an answer in all cases that both says "I am an Agent", and "you may forward packets to me for that node". A Mobile Node that hears from both an agent and the real destination (who by definition must use MADRP), may choose between the two according to MADRP metrics.

This system is relatively simple and seems to work given a small population of nodes. It is certainly an "on-demand" routing scheme. There is no notion of routing convergence except between individuals. Oddly enough we found that TCP was actually quite happy in the FreeBSD implementation to assist us. Due to the path MTU implementation aspects in the current BSD TCP implementation, a end to end host route is dynamically created for every remote TCP/end-system (IP destination). TCP will complain to holders of this route if it finds it is not getting acks back from the end system. Thus this is a quite useful heuristic tool for timing out a defunct route. We also can get rid of routes quite easily if the next hop gateway disappears. With the current scheme, it may take a timeout to learn that a host route through a mobile node that is not on the current link is no longer viable. Of course, this information may be learned via snooping on someone else's multicast response.

Testing included basically two kinds of configuration: 1. linear, say, A to B to C to D, and 2. simple meshes where A talks to D and A can reach both B/C, which in turn can both reach D. We found that our metrics worked as expected and that linear "daisy chains" of Mobile Nodes simply imposed linear delays due to link speeds. For example, average link speed for single WaveLAN links is 7 milliseconds. Two such links would give a delay of 14 milliseconds.

A simple idea that occurred to us in late summer was how to enhance the basic protocol to provide Mobile-IP connectivity to a remote Home Agent. The idea is this: if a Mobile-Node does not have a direct link connection to an agent, it may launch a MADRP query for its Home Agent specifically. Because of the agent "sink" principle (#6 above), any agent be it Foreign or Home will thus answer. The Home Agent is treated as if it was any other possible MADRP destination. If for example, a Foreign Agent answers, we install a host route for the HA in the direction of the Foreign Agent. The MIP registration packet will arrive at the FA, be forwarded to the HA, which will as usual, tunnel packets back to the FA. The FA in turn has installed a MADRP source route back to the Mobile Node. As a result, Mobile-IP works. The Home Agent will tunnel Correspondent Host packets back to the Mobile Node.

During the fall quarter, we are going to hook up the ad hoc routing protocol to IPSEC in a manner similar to how it currently works with the link-level only ad hoc routing protocol. As a result, we will have Mobile Nodes that are both Mobile-IP capable across multiple links, and that can setup end to end IPSEC tunnels automatically to kin in their ad hoc multi-hop mesh. We will also begin to write a technical paper on our work. We hope to release the final work later this year in the release that will include HARP.

I.4 Home Agent Redundancy Work

During the summer we demoed and tested the Home Agent Redundancy Protocol, and it is about ready to release, although we want to first actually deploy it on at least one of the campuses and test it in real life. We need to resolve a couple of nagging problems, which will be discussed below. It is hard to gain operational experience with such a system without real deployment and everyday use. We are in the middle of constructing an Internet Draft on the protocol, and hope to release it soon. We also intend to include HARP in our next release.

The protocol has suffered through a couple of problems and questions. One problem was implementation dependent and is now fixed due to extensive rewriting of the Home Agent code over the summer. We need to test HARP in the presence of routed and determine if there are any interoperation problems. It is not yet clear if they will interoperate together. One problem may be called "how to deal with the issue of the Mobile-IP home link itself".

It has never been terribly clear how we might deal with the redundancy issues inherent in reachability issues concerning the Mobile-IP home link itself. We have felt until recently that those issues were implementation-dependent. HARP has been aimed primarily at solving the problems involved with Mobile-IP routing to Mobile Nodes when they are away, which can justly be claimed to constitute the core of the Mobile-IP idea. When Mobile Nodes are at home, they are believed in Mobile-IP theory, to be "at home", and thus indistinguishable from non-Mobile nodes. We could therefore claim that the issue of multiple Home Agents "on the same wire" is not our problem. Still the issue of Home Agent redundancy on the same link exists. We will attempt a summary of the issues and discuss possible solutions.

The bottom line here though is that probably a flexible implementation is what is needed and that in the end local network management must decide what they want, according to local resources and local tradeoffs.

The Home Mobile-IP subnet may be either:

1. not partitioned; i.e., the HARP Home Agents could reside on the same link.

2. partitioned; i.e., the HARP Agents might reside on different links.

In terms of the exterior Internet, it does not matter as dynamic interior routing protocols can deal with either case.

If the link is not-partitioned, we would assume that the HARP Home Agents could directly reach each other. It is probably not terribly sensible to have both HARP interfaces (which by definition share the same IP interface) active as confusion with nodes on the shared subnet might result. One could claim that it would not matter very much, but there is no point in having both systems race to answer ARP requests. Worse any node that attempted to contact the shared HARP subnet IP address directly might experience failure in a connection attempt. Thus it is reasonable to assume an implementation would provide a way to shut down one HARP interior interface and dynamically bring it up if failure of the other HARP node was detected. This can be done by the normal method of router election by taking the higher IP address and assuming that node will have the only running interface. HARP has a built-in PING as part of the protocol that is used to detect if a HARP peer has disappeared. We assume that the PINGS are done over the non-Mobile IP (and non-shared) IP address of course. Alternatively, Mobile-IP beacons might be used, but they are probably best avoided, due to the reasons involving questions of wireless interface reachability. The non-partitioned scheme as outlined here is probably reasonable and may in fact, be the default scheme that would be used.

We have one serious concern though about HARP peers on the same link, and that is the failure of an interior router or router interface could easily lead to the loss of all HARP agents. Thus we believe that if practicable (and this ultimately depends on per site network layout), it is better if the Mobile Network can be partitioned. In other words, the internal path to the HARP agents would still be done in terms of internal routing, but the agents themselves might best be on different sides of the moon (different buildings with different power supplies). This is certainly not ideal for non-Mobile nodes, as IP subnet reachability problems would result if non-Mobile nodes were placed on the Mobile subnet. There is a simple solution to this quandary. "Do not place non-Mobile nodes on the Mobile subnet". Even better: use our link-layer ad hoc routing protocol. As that practical lacks the side effects associated with ARP and the shared IP subnet routing notion, it is possible that Mobile Nodes can have two homes (a home and a beach condo?). Another simple solution is that one might statically (or dynamically) turn the Mobile-IP beacons off on one of the Home Agents. If the "real" Home was lost, the Mobile Nodes could still operate by finding a nearby Foreign Agent. One of the many lessons of Mobile-IP is that Mobile Nodes actually need never go home.

I.5 ISAKMP Progress

Cisco made another release of their free implementation of ISAKMP in July 1997. It contained the bug fixes we submitted prior to that release, along with our code to take advantage of FreeBSD's random number generator based on keyboard and other interrupt activity. We have submitted another implementation for multi-homed hosts, one that allows a host to serve ISAKMP on more than one interface, and a fix to the RSA key generation scheme that permitted anyone to predict another host's private key, but have not received any indication from Cisco as to whether or not these fixes will appear in the next release.

We have made the necessary operating system modifications so that routes can be created that rely on ISAKMP for security associations to be attached to the routes. These changes were done in such a way as to prevent the route from being used by most processes until the necessary security association(s) are present (the important exception being those privileged process which indicate that they require no security – the ISAKMP daemon does this so that it can exchange the packets to create the security association for the route over which it is exchanging packets).

To create a route with security associations that will be generated by ISAKMP, a SPI of zero is used when specifying the security association to be attached to the route:

```
route add destination gateway -esp 0 -itsrc ...
```

When a route is so created, or when a security association on an existing route is seen to have been deleted, the kernel first looks for a security association that can be used on the route. If no such security association exists, the key management daemon that implements ISAKMP is asked to create it. Packets bound for a destination that do not have the necessary security association(s) are dropped (this is the result of simplifications made when we ported the NRL IPSEC implementation to FreeBSD).

In addition to allowing for security associations to be dynamically created when the route is installed, the implementation allows for manual rekeying requests: A user can delete the security association(s) she wishes to be renegotiated, and the key management system will be informed that it needs to negotiate keys the next time the route is used. This could be done on a regular basis via a cron job, resulting in a system where security associations have limited lifetime.

We have integrated and tested the use of ISAKMP in combination with our implementation of Mobile IP, both between a mobile node and its home agent, and from a home agent to a foreign agent. The changes to the Mobile IP daemons to permit the use of ISAKMP-generated security associations were trivial. We set up ISAKMP with the necessary keys on the participating machines, installed security associations with SPI's of zero in the Mobile IP configuration files, and started the appropriate daemons. ISAKMP was automatically invoked when the route to the mobile node was established, causing security associations to be attached to the appropriate routes. (Note that, in the case of a security association from the home agent to the foreign agent, an unused security association from the foreign agent to the home agent is also created because ISAKMP negotiations are symmetric). These negotiations worked despite occasional dropped packets that result because foreign and home agents do not establish routes to the home agent until the end of the negotiation, but mobile nodes establish a route as soon as they send a registration request.

Unfortunately, the system based on Cisco's ISAKMP implementation is just as inconvenient to set up as the statically-keyed system that must be used if ISAKMP is not available. Since there is no certificate infrastructure available for ISAKMP, the public keys for each pair of hosts that wish to communicate must be exchanged and verified manually, just like when secret keys are being used. As noted above, however, there are security advantages to the ISAKMP-based system, and our kernel- and Mobile-IP-modifications will be able to take advantage of improvements in ISAKMP's key management without any changes to our code.

We will be making an alpha release containing all these latest changes in the near future. Our intent was to make this release shortly after Cisco's next release of their implementation of ISAKMP, but that release appears to have been delayed.

I.6 Distance Mobility

During the summer, we conducted a variety of experiments to determine the effective range of the mobile units that we are using. The WaveLAN radio units put out approximately 0.25W in an unlicensed band at 915MHz. Within campus buildings, the effective range is typically on the order of several hundred feet.

In July, we obtained a pair of multi-element directional antennas (Yagis) and managed to adapt them for use with the ISA bus WaveLAN cards which already use external antennas. We modified the MIP software for fixed nodes to measure and record the signal strength of the beacon signals and to record it in a file.

We conducted two sets of experiments, one on the PSU campus between a fixed unit with a directional antenna and a laptop with the integrated omnidirectional antenna. In this configuration, the laptop was able to receive usable signals at a distance of some 13 blocks. The path was clear, being roughly aligned with a straight street.

We then looked for a longer clear path. US highway 26 west of Portland contains a straight stretch of some 15 miles. At one end of this is a pedestrian overpass with reasonable access to electrical power. We placed the beaconing PC on the overpass along with a MN to ensure that the fixed unit was, in fact, transmitting continuously. The beaconing antenna was aligned with the highway, the center of its main lobe pointing down the median strip. Bill Trost monitored the fixed unit during the measurement period. The measuring unit was placed in the back of a station wagon and powered by a large UPS borrowed for the occasion. It's antenna was mounted on a 10' mast that could be slipped over a mount fastened to the station wagon's trailer hitch. Measurements were made by pulling the station wagon onto the median of the highway, setting up the antenna and aiming it back down the highway. Ten to twenty beacons were recorded at locations ranging from 1.5 to 12 miles from the transmitting location. The furthest locations were highway overpasses giving added elevation. The results were as follows:

Location	Distance (miles)	Signal Strength (Loss) Directional		Non-Directional	
Parking Lot	0.1	22	(0%)	9	(0%)
Murray Exit	1.8	5	(0%)	None	(100%)
Changs Grill	3.9	15	(6%)	4	(62%)
185th Exit	4.6	12	(0%)	3	(57%)
Low Clearance	6.0	9	(0%)	3	(88%)
Shute Overpass	7.9	8	(0%)	4	(0%)
Gordon Overpass	12.5	6	(0%)	2	(0%)

The numbers in the "Directional" and "Non-Directional" columns are signal strengths reported by the ISA WaveLAN card on the receiving unit. It is unclear exactly what these numbers represent, but experience shows that usable communications may be obtained down to values of 3 or so. The "Directional" column represents beacons from the transmitting unit equipped with a Yagi directional antenna , while the "Non-Directional" column represents beacons being sent by the laptop MN used to monitor the directional beacon. We were quite surprised to be able to see the later signal at all at these distances. The first number is signal strength while the second indicates the percentage of beacon loss from each source based on missing timestamps and a one per second beacon rate from both sources. Losses from the intermediate locations with the non-directional transmitting antenna are probably due to multipath distortion and similar effects as the line of sight signal path came quite close to numerous overpasses, power lines, and other structures. The two overpass locations provided an additional twenty feet of antenna elevation which apparently alleviated these effects despite the greater distance and lower signal strength. The Murray Exit location is substantially below the transmitting site, and we suspect that the vertical pattern of the directional antennas was such as to create a "null" in the pattern.

The significance of this experiment is that it demonstrates the feasibility of using the same radio technology that supports local mobility to provide point to point connections that could be used to link locally cohesive ad-hoc secure networks of mobile nodes to a fixed infrastructure some miles away. We suspect that with a reasonable number of hilltop sights, no location in the Portland metropolitan area would be unreachable.

I.7 Signed MN Registration

Over the summer Zhong Chen, under the supervision of Jim Binkley and Sarah Mocas, worked on adding digital signature extensions to Mobile IP registration messages. This was done specifically so that authentication of Foreign Agent messages during Mobile registration would be scalable. Since the destination of a Mobile Node may not be known in advance, it can not be assumed that shared keys exist between either a MN and FA or a HA and FA. On the other hand, authentication of registration messages between a Home Agent and a Mobile Node is easily achieved using a Mobile Security Association[2]. To implement authentication between Foreign Agents and Mobile Nodes or Home Agents we have used digital signature based authentication which has the advantage of allowing dynamic retrieval of public keys for mobile entities. A possible depository for certificates is a DNSSEC [1] domain name server.

Mobile IP authentication with digital signatures is inherently asymmetric, however, the mechanisms as defined in RFC 2002 [2] for mobile IP authentication are suitable mainly for symmetric authentication and have some shortcomings when an asymmetric mechanism is adopted. If Mobile-Foreign authentication and Foreign-Home authentication are required, the Mobile Node has to sign registration requests twice and the Home Agent has to sign registration replies twice also. The additional signature is unnecessary and adds a burden to mobile channels. For example, if a 512 bit RSA signature is used, total octets (type, length, SPI and signature) added to the registration packets will be 70 octets, which is a considerable when the bandwidth of a mobile channel is very limited.

Asymmetric authentication with digital signatures has another distinct feature which is not present in symmetric authentication: the mobile node can obtain public keys of Foreign Agents from its Home Agent. The Home Agent sends Foreign Agent public keys to its mobile nodes through a new mobile registration extension. Home Agents can also enforce security policies over all mobile nodes by sending mobile nodes a digital signature policy extension. The Home Agent will send Foreign Agent public keys to Mobile Nodes "inside of" a digital signature policy extension.

As defined by RFC 2002 [2], mobile IP registration uses three authentication extensions, they are:

- Mobile-Home Authentication Extension
- Mobile-Foreign Authentication Extension
- Foreign-Home Authentication Extension

The structures of these authentication extensions (TLVs) are:

```
+--------+--------+---------+---------------+
|  type  | length |   SPI   | authenticator |
+--------+--------+---------+---------------+
 (1 byte) (1 byte) (4 bytes)   (variable)
```

where,

type is

32 for Mobile-Home Authentication

33 for Mobile-Foreign Authentication

34 for Foreign-Home Authentication

length is "sizeof(SPI) + sizeof(authenticator)"

Positions of the authentication extensions inside a registration packet are illustrated by the following diagram:

```
+-------------+------~------+----------+------~------+---------+
| registration | HA-specific |   MN-HA   | FA-specific | MN-FA or |
|  request or  |  non-auth   |   auth    |  non-auth   |  FA-HA   |
|    reply     | extensions  | extension | extensions  | auth ext |
+-------------+------~------+----------+------~------+---------+
                              (required)              (optional)
```

where, Mobile-Home Authentication is always required except in the case when a registration request is denied by the FA, Mobile-Foreign Authentication and Foreign-Home Authentication are optional. FA-specific extensions, if present, MUST be striped off and interpreted by the FA when it tries to forward registration packets between MNs and HAs.

Due to the limited bandwidth of mobile channel, and the nature of asymmetric authentication with digital signatures, we have modified the protocol as follows:

```
+-------------+------~------+------~------+-----------+-----------+
| registration | HA-specific | FA-specific | MN or HA  |     FA    |
|  request or  |   non-auth  |   non-auth  | signature | signature |
|    reply     | extensions  | extensions  | extension | extension |
+-------------+------~------+------~------+-----------+-----------+
                                            (required)  (optional)
```

Major differences are:

- Packets sent from MN (HA) no longer carry two authentication extensions, they carry only one MN (HA) signature extension. This signature is supposed to be verified by both HA (MN) and the FA.
- The position of MN (HA) signature extension is at the end of the packet, before FA signature extension, if any, and after any non-authentication extensions.
- MN (HA) signature covers everything before the signature, up to the SPI value inside the signature extension. Note, the FA can not strip FA-specific extensions off when forwarding packets between MNs and HAs.

Note, that our design trades a reduces number of bits for the ability to nest extensions in a uniform manner.

We have three kinds of digital signature extensions used in the mobile registration protocol, they are:

- Mobile Node Signature Extension
- Home Agent Signature Extension
- Foreign Agent Signature Extension

The structure of these signature extensions is:

```
+--------+--------+---------+---------------+
|  type  | length |   SPI   |  signature    |
+--------+--------+---------+---------------+
 (1 byte) (1 byte) (4 bytes)   (variable)
```

where,

type is

160 for Mobile Signature Extension

161 for Home Agent Signature Extension

162 for Foreign Agent Signature Extension

length is "sizeof(SPI) + sizeof(signature)"

Due to the difference between asymmetric authentication and symmetric authentication, we have implemented some restrictions on the combination of asymmetric and symmetric authentication methods between MN-FA and FA-HA, they are:

- If MN-HA authentication is asymmetric, MN-FA and FA-HA authentication, if used, must also be asymmetric.

- If MN-HA authentication is symmetric, MN-FA and FA-HA authentication, if used, must also be symmetric.
- A MN can use either symmetric or asymmetric authentication with its HA, however, authentication with FA, if used, must be of the same type.
- A HA can use either symmetric or asymmetric authentication with any of its MNs, however, authentication with the particular FA a particular MN is currently attached to, if used, must be of the same type.
- FA authentication, if used, must be of the same type as the MN-HA authentication, this can be either asymmetric or symmetric.

I.7.1 Mobile Node Considerations about FA Signatures

How a MN obtains the FA's public key and verifies the FA's signature is the most intriguing part of asymmetric authentication. The MN might choose to ignore FA's signatures, but sometimes, from a security point of view, the HA may demand that the MN verify FA's signatures, thus we have a new TLV to let the HA sends its demand to MNs and ship public keys of FAs to MNs, the structure is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |D|P|       reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 IP address of public key owner                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              SPI                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Transform                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Keysize                 | BASE64 Encoded Public Key
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type 168

Length 2, if P = 0; 16 + sizeof(base64_encoded_public_key), if P = 1

D If set, HA demands that MN verifies FA's signatures

P If set, public key of FA enclosed in this extension, IP, SPI, Transform, Keysize and Public Key present if only P is set

IP IP address of public owner (FA's COA)

SPI Value used to match the SPI in FA signature extension

Transform Algorithm of FA signature

Keysize Keysize in bits, if RSA signature is used, this is the size of the RSA public key

Base64 _Encoded_Public_Key Base64 encoded public key

Bibliography

[1] D. Eastlake and C. Kaufman. Domain name system security extensions. Internet Draft?, January 1997.

[2] C. Perkins. Ip mobility support. RFC 2002, Internet Engineering Task Force, October 1996.

Appendix J

Quarterly report – Fall 1997

J.1 Project Status Overview

We operated during this quarter under the assumption that there might be no additional funds forthcoming for FY 98. Under this assumption, we would have had barely enough funding to continue support of our existing students through their scheduled graduation dates. In early January, we received an additional $75,000 (including overheads) which will allow completion of work currently under way and at least two additional releases of our system. It will also allow modest improvements in our equipment base.

In summary, we have slowed down quite a bit due to the programmers departing at the beginning of the quarter. We have three M.S. graduate students working on certain aspects of the mobile project at this point. Two of them will graduate at the end of the winter quarter and one will remain until the end of the spring quarter. It should be pointed out again that we made a combined release of Mobile-IP/IPSEC last summer and those files remain available at PSU and MIT.

Bill Trost and David Reeder left us at the beginning of the quarter, which was probably a good thing as we had no money to pay them. David Reeder is now employed at Trusted Information Systems. Jim Binkley is working as a combination faculty/network engineer job at Oregon Graduate Institute, and it still helping to author academic papers with John McHugh and Sarah Mocas and oversee our remaining project graduate students at PSU. The graduate students are Jennifer Ye, Zheng Chen, and Bjorn Chambless.

We have wrapped up work for now on our Multicast Ad Hoc Routing Protocol (MADRP) (Jennifer Ye is the graduate student). Jennifer tied the multi-hop ad hoc routing protocol to Mobile-IP and to IPSEC tunnels. A remote mobile node that is greater than one hop away from other mobile-node can setup 2-way ESP tunnels to other mobile nodes. It can also connect to the Internet via a Mobile-IP agent that is not directly hearable. This work is checked in and will be made available in a next release. At some point, we would like to write a paper on this effort, but are unsure as to when we might have time.

Bjorn Chambless has just about completed his work on the Home Agent Redundancy Protocol. The code has been checked into the source base. Bjorn still needs to successfully deploy the HARP protocol internal to both PSU and OGI mobile-IP systems; i.e., have two cooperating HAs in both cases. He also needs to demonstrate that a routing daemon (say routed running RIPv2) can co-exist with the PSU mobile-ip agent routing daemon (mipd). Bjorn and Jim Binkley co-authored a draft RFC that describes HARP (draft-chambless-mobileip-harp-00.txt). The document has been submitted to the IETF.

Zheng Chen has finished work on a simple authentication protocol for Mobile-IP that uses DNSSEC based public-key cryptography. The work is not checked in as of yet and it is not clear to us exactly how we might distribute it. He has helped to author a paper on the subject of public-key crypto and Mobile-IP with Sarah Mocas which was submitted to the Oakland IEEE Security conference.

Zeng Chen has begun work on supporting DHCP (Dynamic Host Configuration Protocol) in Mobile IP. Mobile Hosts use the protocol to obtain temporary addresses at foreign networks for routing purposes but still maintain "permanent" home addresses for identity purposes and keep continuous transport layer connectivity while moving. This way, the Mobile Node can acquired a "co-located-care-of-address" at the foreign network and can detunnel packets itself. This work will continue during the Winter quarter as noted below.

J.2 Next quarter plan

Bjorn will finish working on HARP and get it deployed with Jim's help so that we have dual HA's at PSU and OGI. It is important here to demonstrate that the HA can actually participate in an interior routing domain protocol and the implementation side of that statement is not straightforward. With the cooperation of network engineers in the engineering school at PSU, we will attempt to teach the local ciscos to accept route changes forwarded with RIPv2 from the FreeBSD based mobile routers. Thus HARP can interact (courtesy of the FreeBSD routed) with the interior gateway routing scheme in engineering. We are choosing RIPv2 for this experiment, because we must use class C based subnet masks, and because we might have to modify routed to teach it to ignore Mobile-IP tunnel routes.

Bjorn will be with us for two more terms. As a consequence, we have appointed him wireless system administrator with the task of keeping the wireless network system up including laptops and agents. He is also going to attempt a minor port of the mobile node daemon (the laptop Mobile-IP daemon) to linux. If he can do this, researchers at OGI (who use linux) will consider running our version of Mobile-IP.

Zheng Chen is going to use his remaining quarter to investigate the routing issues of combining DHCP support with our Mobile-IP routing setup (at laptops). He will also briefly investigate security considerations in this regard. We regard this as major redundancy work as the resulting laptop system should be able to work on either a Mobile-IP agent/beacon link or a DHCP (local IP address) link and dynamically switch between them when roaming. Our resulting Mobile Node daemon will have a simple DHCP client capability that will enable it to dynamically attempt DHCP hookup when Mobile-IP agent advertisements are not heard. We would then expect to tunnel back to our Home Agent as a very simple way to keep connections using the Mobile-IP "home" address alive.

Effectively such a mobile system becomes at least multi-homed; i.e., it will have a local and short-lived IP address used to talk to local systems and a global and long-lived IP address that is its Mobile-IP address. How to deal with such a situation is not clear in a routing sense (Please see the excellent "Internet Mobility 4x4" paper by Stuart Cheshire and Mary Baker under `http://rescomp.stanford.edu/ cheshire`) as fundamentally a policy mechanism that will allow transport-side applications (TCP/UDP) to choose a long-term or short-term IP address must exist. Further, security considerations are challenging in terms of possible policy. One simple mechanism might be to simply use the local IP (DHCP-gained) address to tunnel all "real" packets home so that they will appear on the Internet with the permanent home address. We will explore this latter option.

Jennifer Ye has started working on Foreign Agent redundancy. What we mean by this is the fundamental ability of a Mobile Node to use at least two agents as link attachments; .e.g., a wireless node might keep the best two agents in terms of signal strength and send packets to both of them. This work must be done in two stages:

1. A kernel routing mechanism must be created that allows a BSD route to have two gateways so that when IP encounters such a route, it will duplicate the packet and send it out the interface pointed to be the second gateway. Note that this gives us the extremely interesting capability to actually use two interfaces at once with possibly different links. This capability is not very different from the situation where a high-level routing protocol like OSPF must do packet load sharing to a remote but equal cost destination. Of course, in our case every outward bound packet will be duplicated. This is very selfish from the Mobile Node's point of view. We are effectively prioritizing its individual communication over the entire internal network bandwidth. Still, such a capability could be useful where links are stressed and command communication is important.

2. Mobile-IP must take advantage of this kernel capability. For example, we would take advantage of our existing ability to prioritize agents by signal strength and instead of using the best, use the top two. Home Agents must also support dual Mobile-IP registration.

If there is remaining money for one more graduate student, we might hire such a person to keep the project going. That person could conceivably take over Jennifer's work in the Foreign Agent redundancy area.

John McHugh, Sarah Mocas, and Jim Binkley are writing a high-level paper on mobile security policy. We will attempt to get this paper finished this quarter and submitted to a an appropriate journal. If time permits, the HARP draft will be converted into a paper and submitted to a wireless journal as well. Jim and David Reeder are going to attempt to collaborate with John Zao of BBN on a IETF draft that addresses certain Mobile-IP and IPSEC combinatory issues.

At the end of the winter quarter, we will bundle up another release of our software and make it. Minimally that software will include HARP and MADRP and may include other features if complete and in time.

J.3 Outreach

We have been contacted by a number of groups or people regarding our current efforts including Berkeley/South Carolina University/BBN and elsewhere and have attempted to provide information as possible within our time limitations. Lixia Zhang's research group at UCLA has made use of our tunnel driver code in FreeBSD. The latest release of FreeBSD contains the ISA wavelan driver that Jim Binkley ported to FreeBSD long ago.

We are attempting to locate an industrial partner in the cellular telephone field to partner on a proposal to PATH, the automated highway research group at Berkeley. The proposal would be to investigate extending Mobile-IP to provide cost effective internet access to mobile units on the highway. A key feature would be extension of FA beaconing to provide for local emergency and commercial notifications.

Jim Binkley and Bjorn Chamblis have written an internet draft on HARP which appears as an appendix to this report.

J.4 Equipment

Our equipment, especially laptops, is suffering from near obsolesence. Many of the laptops are over two years of age at this point and we note that the average laptop lifespan at this point seems to be (less than) a year. Our wireless network situation is made more complex by rumors that Lucent may stop selling the 915mhz equipment in order to centralize on IEEE 802.11 and 2.4 ghz equipment. This would effectively give us no way to keep our network up for research without replacing it. We hope that we might obtain a small amount of money to buy a few laptops, cheap desktops for agents, and some replacement 915mhz equipment.

J.5 Appendix - The HARP Draft


```
Internet Engineering Task Force                Bjorn Chambless
INTERNET DRAFT                                Portland State University
Jim Binkley
        Oregon Graduate Institute
        October 27, 1997
```


```
                  HARP - "Home Agent Redundancy Protocol"
   <draft-chambless-mobileip-harp-00.txt>
```

Status of This Memo

Distribution of this memo is unlimited.

This document is an Internet-Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months, and may be updated, replaced, or made obsolete by other documents at any time. It is not appropriate to use Internet Drafts as reference material, or to cite them other than as a ``working draft'' or ``work in progress.''

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this memo is unlimited.

Abstract

This document presents a protocol called the Home Agent Redundancy Protocol or HARP. HARP is an optional extension to Mobile-IP [RFC-2002]. Mobile-IP includes the notion of a Home Agent which is a host located on the home IP subnet for Mobile Nodes. Home Agents forward packets to Mobile Nodes that are away from home. Since Mobile Nodes are dependent on the Home Agent for connectivity when away from home, the Home Agent represents a possible single source of failure for the Mobile IP system.

HARP is a protocol which allows a pair (or set) of Home Agents to cooperate and share Mobile-IP Mobile Node registration information. If one of the HARP peers should fail, the Mobile-IP system will not necessarily fail, hence HARP introduces Home Agent redundancy into the Mobile-IP system. Mobile Nodes are not aware that HARP exists, so HARP requires no changes to Mobile-IP as a protocol. In this document, we present the HARP architecture and protocol.

Table of Contents

1. Introduction

Mobile-IP is designed to allow a Mobile Node (MN) to change its point of IP subnet attachment in the Internet at the network or IP layer. The MN is always identified by it home IP address regardless of its current network location. Its mobility is not limited by conventional IP network boundaries.

The Mobile-IP system consists of Mobile Nodes, and two kinds of agents, known as Home Agents (HA), and Foreign Agents (FA). Home Agents remain "home" and when the Mobile Node is not home, forward packets sent to the conventional IP subnet of the Mobile Node to a possibly distant point of attachment. The remote address is called a Care Of Address (COA), and may be at a Foreign Agent or a co-located Mobile Node. As a Mobile Node travels from one IP link to another, it determines possible COAs and uses the

Mobile-IP registration protocol to inform the Home Agent of its current location. The Home Agent then forwards packets addressed to the Mobile Node at its home network to the its current location.

In Mobile-IP, as currently specified, a single HA services an MN. The MN is reliant on this Home Agent for its connectivity. Thus the HA represents the possibility of a single point of failure for Mobile-IP. A Home Agent may be responsible for multiple Mobile Nodes on multiple home subnets. The failure of a single HA may then result in the loss of connectivity for numerous Mobile-IP Mobile Nodes located throughout the Internet. Thus the Home Agent and Mobile Node taken together have a shared fate. A Mobile Node cannot afford the loss of its Home Agent.

This vulnerability is inconsistent with the fault tolerant nature of the Internet. Additionally redundancy is needed. We have developed the Home Agent Redundancy Protocol (HARP) as an optional extension to Mobile-IP to address this problem.

HARP is a simple protocol based on the notion of one or more HARP peers that act as a single shared Home Agent. Each HARP peer is configured with information about its HARP peers and forwards any Mobile-IP registration messages it receives to its peers. HARP peers act in parallel to create or delete tunnels [RFC 2003] to the Mobile Node's remote COA according to the last registration message received. Although we speak of HARP peers as a set, in general, there will probably be only two cooperating systems in the HARP sub-system.

There are three major types of messages, 1. HARP TCP DUMP, 2. HARP UDP FORWARD., and 3. HARP UDP PING. At boot, a TCP connection may optionally be made to a remote HARP peer to exchange mobile routing information. At runtime, HARP UDP PING messages are exchanged to determine if remote HARP peers are up. At runtime, HARP UDP FORWARD messages are used to forward Mobile-IP registration messages from the receiving HARP agent to its HARP peers.

HARP assumes no changes to Mobile-IP proper; i.e., the existence of one or more HARP peers is kept hidden from Mobile Nodes. Therefore HARP will interoperate with existing Mobile-IP implementations. In routing terms, one may think of the HARP peers as advertising the existence of a common IP subnet into an interior routing domain. Externally, a MN's Mobile-IP authentication message is routed to the nearest (according to local routing metrics) HARP peer which, in turn, informs other HARP peers about the MN's location via a HARP FORWARD message. Packets are routed to HARP peer Home Agents via conventional routing, and since each HARP peer maintains Mobile Node COA information, packets are forwarded to the MN.

1.1 Design Goals

The Home Agent Redundancy Protocol (HARP) aims to remove the Home Agent as a single point of failure for Mobile-IP.

The protocol is implemented entirely through the enhancement of Home Agent functionality. There are no additional responsibilities or modifications required of either Mobile Nodes or Foreign Agents. Mobile Nodes and Foreign Agents have no knowledge of HARP and Mobile-IP will interoperate with HARP capable Home Agents.

The Home Agent Redundancy Protocol will be made secure, minimally with authentication, and optionally with authentication and privacy mechanisms.

Home Agent Redundancy makes no assumptions about the physical media utilized by the Mobile-IP environment. Therefore HARP does not limit the physical implementation of Mobile-IP.

The number of Mobile Nodes is not limited by HARP.

1.2 Terminology

Home Agent Redundancy Protocol terminology uses and expands on the Mobile-IP terminology presented in RFC 2002. The following terms are specific to the Home Agent Redundancy protocol:

HARP peers -
co-HAs -
co-Home Agents - A set of Home Agents acting in concert to provide connectivity to one or more Mobile Nodes. These hosts share an IP address on the Home Subnet but each has a uniquely identified interface outside of the Home Network. Co-HAs, a priori, know about a small set of cooperating Home Agents and exchange registration information regarding Mobile Nodes and periodically test peer co-HA reachability. One may assume that there are only two Home Agents in a set of co-HAs, but there is no inherent limit to the number of peers in a Co-HA set.

Primary-HA, Primary - The Home Agent of a co-HA set which is currently receiving registration information directly from a MN. The Primary Home Agent shares this information with its co-HAs (Secondaries) by forwarding registration packets to the peers. A HA is primary because the IP routing infrastructure is currently routing the Mobile-IP registration packet to it.

Secondary-HA, Secondary - A Home Agent of a HARP set which is receiving registration information about a given MN indirectly through its co-Home Agent which is acting as the Primary.

HARP - Home Agent Redundancy Protocol.

HARP PORT - The HARP port number which is the same for both the TCP and UDP ports. This port has not yet been allocated yet.

Home Network - Home Subnet - The subnet containing both Home Agents and the home addresses of the Mobile Nodes they are serving. This subnet may be partitioned in terms of the co-HAs or the co-HAs may be physically present on the same link.

Partitioned Subnet - A physically divided Home Subnet. Home Agents in a co-HA pair may be thought of as existing on a virtual subnet. Physically divided means that the co-HAs cannot use that link to communicate directly. Note that this is not a requirement of HARP, but an aspect of network design. We will discuss the network design aspects of HARP below.

AWAY(Mobile-IP state) - The state of a Mobile Node, with respect to its Home Agent(s), in which datagrams addressed to the MN arrive at its Home-Subnet and are tunneled to the MN's Care Of Address by one of the Mobile Node's Home Agents.

AT HOME (Mobile-IP state) - The state of a Mobile Node with respect to a Home Agent in which the MN's current point of attachment in the Internet is consistent with its IP address. In this state, the Mobile Node will receive packets directly.

At CoHA(Mobile-IP state) - The state of of Mobile Node, with respect to a Home Agent, in which the MN's home subnet is partitioned and a packet arrives for the MN at another link of the partitioned network. In this case, packets addressed to the home subnet may arrive on a portion of the home subnet to which the Mobile Node has no link layer attachment. These packets must then be forwarded (tunneled) by one HARP peer to another.

2. Protocol Overview

This section provides a protocol overview of HARP. We discuss HARP from a routing topological point of view, and provide a short discussion of redundancy issues.

2.1 Assumptions

The following are fundamental assumptions about the design of a HARP redundant agent network:

1. So that Mobile-IP and Mobile Nodes need not know about the HARP sub-system, we assume that the HARP peers share a single IP subnet and a single IP network address.

2. Due to assumption 1, and because the HARP agents must communicate amongst themselves, we assume they are multi-homed in the sense that there exist on one node at least two IP addresses. We will call them the "Mobile-IP subnet" address, and the "HARP peer" address. The former is shared. The latter is not shared. The HARP peer address is unique and is used by HARP systems to communicate amongst themselves. Note that this does not rule out an agent with only one physical interface.

3. We assume that the HARP peers exist within an interior routing domain that runs a IP interior routing protocol such as RIPv2 [RFC-1721], or OSPF [RFC-2178]. Thus packets addressed to the Mobile-IP subnet, including Mobile-IP authentication packets, are routed to the HARP peers according to the local metrics of the interior routing protocols. At any given time, any packet bound for a Mobile Host at HOME, will go to exactly one HARP peer. Ideally, if an interior link fails, an interior routing protocol will switch Mobile-IP packets to the other HARP system. This does not rule out the possibility of HARP being used with static routes in interior routers. External routes to the Mobile-IP subnet may always be changed by hand in the event of HARP agent failure.

Finally, it is NOT assumed that the HARP Mobile-IP subnet is partitioned. Partitioning may add useful redundancy attributes. But the subnet need not be partitioned. How this is handled is implementation and site specific and will be discussed more in the next section.

2.2 Protocol Overview

Diagram 1:

```
                         MN or CH

                            |
                            |  MN/CH path
                            |

Internet

                            |
                            |
```

```
                          |
                ------------------------
               R1    interior path      R2
               |                        |
H1                      H2
               |                        |
             ----- Mobile-IP subnet ------
```

Please refer to diagram 1 for the following discussion.

Assume we have a Mobile Node that is away from home. Assume that at home we have two routers R1, and R2, which are using a local interior routing protocol (for example, OSPF, or static routes). Behind them are two HARP peers which advertise a Mobile-IP subnet to the routers and to the network. Along with the Mobile Node on the exterior Internet, there exists a Correspondent Host (CH). The latter is any host interested in sending packets to the Mobile Node.

The Mobile Node is unaware of Home Agent redundancy and sends registration information only once to the shared HA address located on the Home Subnet. Due to the nature of unicast routing, this information may be received by either Home Agent, but is likely to be received by only one. Packets sent from the CH to the Mobile Node will likewise be routed to one or the other Home Agent (whichever is "closer", according to the metrics of the interior routing system). Note that any of these datagrams (Mobile-IP authentication or ordinary datagrams) may at any time go to either Home Agent.

When a Mobile-IP registration packet is received by a co-HA, it will encapsulate that registration packet and forward it to another co-HA via the HARP UDP port. Thus the peer co-HA will know that the other HA directly received the registration, and will also know the current MN Care Of Address; i.e., the forwarding address for CH packets.

Each HARP peer takes the same internal action whether it receives a Mobile-IP registration packet directly from a MN or whether it receives it from a HARP peer. For example, in parallel, the Mobile-IP vlist entry is created or refreshed, and tunnels are setup (if needed) from the Home Agent to the Care Of Address. In Mobile-IP terms, the Home Agents act in parallel.

Mobile-IP authentication (MN to HA, FA to HA, etc.) is performed once at the primary co-HA system. The secondary is expected to perform a separate HARP protocol authentication (HA to HA) on packets forwarded to it by the primary via the HARP UDP (or TCP in the case of table exchanges) ports.

If one Home Agent should fail, the interior routing structure should notice that it has failed as a router, and normal routing convergence will remove that path to the Mobile-IP subnet. Convergence may take manual route manipulation in some cases where static routes are used. As a result, the Mobile-IP system should be able to survive the loss of a Home Agent as packets will be routed to a surviving Home Agent.

HARP consists of three major packets type sent from one HARP peer to other HARP peers. (Note that some of the types have acknowledgements). We have already mentioned the HARP UDP FORWARD, which is a simple repackaging of the Mobile-IP registration packet itself. In addition there are two other kinds of messages used in the HARP system. There is a HARP PING and a HARP boottime table exchange. The former uses UDP and the later uses TCP.

The UDP HARP PING is used to determine if other HARP peers are up. If a PING fails (say 3 out of 5), a HARP agent knows that either its peer has failed or the path to it has been partitioned. It may then take implementation-specific actions which should include ways to attempt to notify local administrators that critical interior links or systems have failed. Other implementation actions may be taken as well if needed. For example, a dormant local link interface might be enabled.

HARP provides an optional boot protocol which uses TCP to exchange all HARP information in one connection. Thus if one HARP system reboots after failure, it can acquire Mobile-IP state information from another HARP peer. At boot, a HARP Home Agent will attempt to establish a TCP connections with its co-HAs at their respective TCP HARP PORTs. If successful, these connections are used to pass all current Mobile Node registration information from a running HA to a booting co-HA. When all relevant information has been passed, and the booting Home Agent is synchronized with respect to MN Registrations, the TCP connections are closed. If the peer system is not available, the sending system will timeout and proceed as the peer may be unavailable or rebooting.

2.3 Redundancy Considerations

There are a number of redundancy considerations regarding HARP that have driven its design which we will present in this section.

The failure of one HARP agent, or a network interface on that agent, or possibly a path to that agent should not necessarily cause the Mobile-IP network to fail. This is, of course, the goal of the protocol itself. In addition to simple node redundancy, redundancy may be possible if a path from the MN (CH) in question exists to another HARP agent even when an interior (or exterior) router (or associated interface) fails. Thus HARP may sometimes be able to take advantage of dynamic interior routing.

On the other hand, the interior path between the HARP agents should not be allowed to fail. If it does, it is possible that the Mobile-IP registration packets might go one way and datagram packets from a given CH might go another, thus leading to a (bizarre) partition. This is one of the reasons for the HARP PING protocol. Lack of connectivity between the agents should lead to a local management alert. Of course, fundamentally, an interior path failure might cut the Mobile Node off from important local services and should be taken seriously in any case.

As part of the overview, we should justify why we have chosen UDP for internal HARP forwarding as opposed to say TCP connections between HARP agents. We felt that the HARP protocol should internally match the design of Mobile-IP. Registration for Mobile Nodes while away from home is driven by Mobile-IP/UDP based packets. Since the Mobile Node drives the registration, we felt that forwarding these packets internally over presumably shorter channels (with higher bandwidth) was reasonable. As a result, Mobile-IP registration also drives the HARP sub-system. We also felt that given the goal of producing a reliable server with reliable software it made more sense to use a simple protocol without the enhanced complexity of a TCP state machine to deal with possible protocol errors. Thus agent-side implementation should be simpler. As a final point, even though the number of HARP peers in a HARP set is likely to be very small, at some future time, one might consider experimentally replacing the unicast UDP HARP (re)registration with reliable multicast registration. Of course, TCP lacks multicast capability.

Redundancy considerations for the Mobile-IP home link itself are discussed in the next section.

3. Mobile-IP Home Link Considerations

One might claim that the impact of HARP on the actual Mobile-IP home subnet link could be regarded as implementation dependent. This is because Mobile-IP itself is aimed not so much at dealing with how mobile nodes interact at the home subnet, but how they can take a home subnet IP address, move to other subnets, and retain connectivity. Thus HARP primarily seeks to address the problem of what might happen if the home forwarding agent is lost. Still, the issue of redundancy on the Mobile-IP home link exists and there is a small amount of protocol impact on HARP. In a very simple sense, having two possible "home" links can aid redundancy as well. If one home fails, a second home might be available. In this section, we will discuss this issue and make implementation suggestions.

In the end local network management must decide what they want, according to available local resources and local tradeoffs. Therefore we suggest that implementations remain flexible where home subnet design is concerned.

The Home Mobile-IP subnet may be either:

1. not partitioned; i.e., the HARP Home Agents could reside on the same link and would be able to reach each other on that link.

2. partitioned; i.e., the HARP Agents might reside on different links, and would NOT be able to reach each other on that link.

3.1 Non-Partitioned Home Subnet

If the link is not-partitioned, the HARP Home Agents can reach each other directly. It is not advisable to have both HARP IP interfaces (which by definition share the same IP interface) active as confusion with nodes on the shared subnet might result. If the systems act as routers only (and not as end systems), one might claim that it would not matter, but there is no point in having both systems race to answer ARP [RFC-826] requests. Worse, any node that attempted to directly connect to the shared HARP subnet IP address with a transport protocol may experience failure due to ARP cache overwrites.

Thus it is reasonable to assume an implementation might provide a way to shut down one HARP interior IP interface and dynamically bring another up if failure of the other HARP node is detected. This can be done by the normal method of designated router election. A set of HARP peers exchanging HARP PING messages

may choose the HARP peer with the highest IP address to be the only active peer on the interface. We assume that the PINGS are done over the non-Mobile-IP (and non-shared) IP address. If PINGS fail from that interface, (say after N failures, where N is configured in), the remaining HARP peers would again elect a designated router which will take over the "live" ARP function.

3.2 Partitioned Home Subnet

If HARP peers exist on the same link, it is possible that the failure of an interior router or router interface could lead to the loss of all HARP agents. Thus one may have replaced the Mobile-IP single point of failure mode with a more elaborate single failure mode. We suggest that if practicable (and this ultimately depends on per site network resources), it may be better to partition the home mobile link. In other words, the internal path to the HARP agents would still be an interior routing path, but the agents themselves might best be in widely dispersed locations. For example, HARP peers would be placed behind different interior routers, possibly in different buildings.

Such a partitioned network is not ideal for non-Mobile nodes, as IP subnet reachability problems would result if non-Mobile nodes were placed on the Mobile subnet. As a result, one might choose to only place Mobile-IP nodes on such partitioned links.

Where a partitioned scheme is used, it is necessary for the Home Agents to install tunnels between themselves. This mechanism is directly analogous to the Mobile-IP tunnel from the Home Agent to the Foreign Agent. A packet sent to a co-HA where the MN is not resident, would be forwarded to the other co-HA. Note that HA tunneling is not strictly necessary when the Mobile subnet is not partitioned, unless the Home Agents are only speaking one at a time to the home link.

A simple and less elegant solution would be to disable Mobile-IP router advertisements for Home Agents where actual physical residence is not desired. A dynamic scheme for election here could be used, or this function could be done manually. For example, one might choose to have only one Mobile-IP home subnet from the interior point of view. The other home might reside on a virtual interface that is not directly accessible except in the exterior routing sense. Mobile Nodes in such a system might never be able to directly visit the second home.

In summary, where interior router resources exist, partitioning may provide greater surviveablity.

4.1 Message Types and Functions

The Home Agent Redundancy Protocol has five messages:

HARP_FORWARD - This message consists of an encapsulated Mobile Node registration message which is tunneled from the receiving Home Agent to its co-HA. This information is used to update the co-HA's registration tables. This message type uses UDP and is sent to the HARP UDP PORT.

HARP_PING - This message is sent at configurable intervals from one co-HA to another to confirm connectivity. If the Home Agent receives no response from its co-HA peer, the co-HA is assumed to be unreachable. This message type uses UDP and is sent to the HARP UDP PORT.

HARP_ACK - Sent in response to a HARP_PING to acknowledge that the PING message has been received. This message type uses UDP and is sent to the HARP UDP PORT.

HARP_REG_REQ - A message requesting all Mobile Node registration information. This is the first message sent upon establishment of an inter-co-HA TCP connection. This message type utilizes TCP.

HARP_REG_DUMP - TCP message which contains Mobile Node registration information maintained by a Home Agent. This message is sent in response to a HARP_REG_REQ. Note that more than one DUMP message may be sent, but each DUMP message may contain more than one Mobile-IP node registration.

4.2. Message Formats

All HARP messages are structured in a Tag, Length, Value format.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |    Value ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type and Length occupy one unsigned short and are 16-bit values. They are assumed to be in network byte order. Messages are sent to either the HARP UDP or TCP PORT.

4.2.1. HARP Registration Forward (HARP_FORWARD)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     type=HARP_FORWARD         |     size                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Mobile-IP registration packet ...              |
|                                                               |
```

type: HARP_FORWARD = 1
size: in bytes of the value field.
value: Mobile-IP registration packet.

HARP Registration Forward messages are used to encapsulate and forward registration updates received from a Mobile Node. They are sent between co-HAs. The message consists of two bytes of type followed by two bytes indicating the length in bytes of a Mobile-IP registration packet. The Data field is a Mobile-IP registration packet of the size indicated by the size field. The registration packet has the same format as used with Mobile-IP. Optional (TLV) elements may be present and should be processed. However it is expected that the receiving Home Agent deals with all MN/HA and MN/FA authentication and may remove those elements from the registration packet.

4.2.2 HARP Ping (HARP_PING)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     type=HARP_PING            |       size=4                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     HARP peer IP address            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

type: HARP_PING = 2
size: 4
value: (reachable) IP address of HARP sender

HARP_PING messages are sent between all the members of a HARP co-HA set. They are sent with a configured periodicity and are sent within a configured mechanism for determining the loss of an interior routing path. For example, one might send one message a minute, and retry N times. If the sending agent determines that another agent is down, it may initiate implementation-dependent mechanisms including SNMP alerts, paging a network administrator, and/or taking up or down a local interface. We include the HARP peer IP address in order to limit possible problems caused by multi-homing.

4.2.3. HARP Ping Acknowledge (HARP_ACK)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     type=HARP_ACK             |        size=4                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     HARP peer IP address            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

type: HARP_ACK = 3
size: 4
value: HARP peer IP address

A HARP Ping Acknowledge message is sent in response to a HARP Ping. The peer IP address belongs to the sender of the acknowledgement.

4.2.4 HARP Dump Request (HARP_DUMP_REQ)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     type=HARP_DUMP_REQ      |        size=0                   |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

type: HARP_DUMP_REQ = 4
size: 0

The HARP Dump Request message consists of two bytes of type followed by two bytes giving the size of the data field, which is always zero.

A HARP_DUMP_REQ is passed from a Home Agent in Initializing State to its co-HA through the inter-co-HA TCP connection. If the receiving co-HA does not receive this message, it should shut down the connection and log an error. If the client's connection fails or no DUMP appears within a configured timeout interval, the client should disconnect and log an error.

4.2.5 HARP Registration Dump (HARP_REG_DUMP)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     type=HARP_REG_DUMP      |        size=8                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Number of Registrations                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Size of Registrations                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Mobile-IP registration field ...            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Mobile-IP registration field ...            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       etc.                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

type: HARP_REG_DUMP = 5
size: 8
value:
   Number of Registrations: a 32-bit unsigned integer in network byte order
   Size of Registrations: a 32-bit unsigned integer in network byte order. This field contains the size of all the Mobile-IP registration sections which must all have the same size.
   Some number of Mobile-IP registration requests.

A HARP Registration Dump message is sent in response to a TCP connect and HARP Dump Request. It contains current registration information for all Mobile Nodes serviced by a given Home Agent. "The number of registrations" field tells the receiver how many registration messages are being sent. The size of the registration message gives the size of each Mobile-IP registration message which as before, is assumed to be the same format as is used with Mobile-IP. Note we assume that all message sub-fields are of the same length. Also note that a Home Agent need not write all of the information with the same TCP write. Nor does the responding peer need to send all Mobile-IP registration messages in one DUMP message. Multiple DUMP messages may be sent over the same connection.

The channel must be closed by the receiver when it has written all of the messages.

5. Security Considerations

In this section, we briefly consider the security of the HARP protocol itself. Security might also be taken to mean "survivability" and we will discuss that notion first, and then return to HARP protocol network security.

In terms of survivability, HARP primarily addresses the problem that a Mobile-IP system, serving a given number of Mobile Nodes may fail due to the loss of a single Home Agent. Home Agent redundancy reduces the odds of a total Mobile-IP system outage due to failure of a Home Agent node, a network adapter on that node, or possibly an internal routing path to that node. Given that we may have multiple Home Agents that cooperate to emulate a single home agent there may also be additional security benefits. For example, a denial of service attack against one Home Agent may not apply to the other (perhaps if they are not on the same link). Hence the Mobile-IP network might continue to function.

For protocol security of HARP itself, we require the use of IP layer security [RFC 1825] between any given HARP pair in a set of HARP peers. The HARP pair may use a TCP connection between them at boot for route synchronization, and will use UDP to forward Mobile-IP registration packets. It is required that all of these end to end TCP and UDP channels be protected by at least IPSEC authentication [RFC 1826], and optionally by authentication and encryption [RFC 1827]. Authentication is a requirement. Thus security will be end to end for the HARP protocol between HARP peers.

References

[RFC-826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Addresses for Transmission on Ethernet Hardware", STD 37, November 1982.

[RFC-1721] Malkin, G., "RIP Version 2 Protocol Analysis", November 1994.

[RFC-1825] Atkinson, R., "Security Architecture for the Internet Protocol", Naval Research Laboratory, July 1995.

[RFC-1826] Atkinson, R., "IP Authentication Header", August 1995.

[RFC-1827] Atkinson, R., "IP Encapsulated Security Payload", August 1995.

[RFC-2002] Perkins, C., "IP Mobility Support", October 1996.

[RFC-2003] Perkins, C., "IP Encapsulation within IP", October 1996.

[RFC-2178] Moy, J., "OSPF Version 2", July 1997.

Contacts

Bjorn Chambless
Computer Science Department
Portland State University
Email: bjorn@cs.px.edu

Jim Binkley
Computer Science and Engineering Department
Oregon Graduate Institute
Email: jrb@cse.ogi.edu

Appendix K

Quarterly report – Winter 1998

K.1 Project Status Overview

We began this quarter under the assumption that there might be no additional funds forthcoming for FY 98. Under this assumption, we would have had barely enough funding to continue support of our existing students through their scheduled graduation dates. In early January, we received an additional $75,000 (including overheads) which will allow completion of work currently under way and at least two additional releases of our system. It will also allow modest improvements in our equipment base and support for the one student remaining for the spring and summer.

In summary, we have slowed down quite a bit due to the programmers departing at the beginning of the quarter. We have three M.S. graduate students working on certain aspects of the mobile project at this point. Two of them will graduate at the end of the winter quarter and one will remain until the end of the spring quarter. It should be pointed out again that we made a combined release of Mobile-IP/IPSEC last summer and those files remain available at PSU and MIT.

Graduate students Jennifer Ye and Zheng Chen graduated at the end of the winter quarter. Bjorn Chambless will stay on for the spring and probably the summer. Jim Binkley has accepted a full time, non faculty, position at PSU effective in September and will remain associated with the project through its end.

Zheng Chen has completed work on supporting DHCP (Dynamic Host Configuration Protocol) in Mobile IP. This work is reported more fully in the following section. Mobile Hosts use the protocol to obtain temporary addresses at foreign networks for routing purposes but still maintain "permanent" home addresses for identity purposes and keep continuous transport layer connectivity while moving. This way, the Mobile Node can acquired a "co-located-care-of-address" at the foreign network and can detunnel packets itself.

Jim Binkley has completed an internet draft on mobility and security which appears below.

K.2 Use of DHCP in Mobile IP – Zhong Chen

K.2.1 Architecture Overview

In addition to registering via a Foreign Agent, the Mobile Node can also register directly by obtaining a temporary IP address at the Foreign Agent. The temporary IP address is called a "co-located-care-of-address" in this situation. The "co-located-care-of-address" is the end point of tunnel and the Mobile Node detunnels data packets itself.

We decide to use the Dynamic Host Configuration Protocol (DHCP) for the purpose of address acquisition in our project. DHCP is a well deployed protocol for address allocation and host configuration. We use Internet Software Consortium's implementation of DHCP server and choose to write DHCP client code ourselves.

Although potentially in-efficient, we use DHCP itself as a mechanism of movement detection. If the Mobile Node DHCP client receives a DHCPNAK in response to DHCPREQUEST or no responses when renewing lease, the Mobile Node assumes itself has moved and tries to initialize a new round of DHCP session. The time interval in which the Mobile Node client renews leases is user configurable. The client tries to suggest a short lease time according to the time interval value with the DHCP server each time it sends a DHCPREQUEST.

We believe registering through Foreign Agent is preferable to registering directly using a "co-located-care-of-address". Thus, if the Mobile Node is configured to switch between modes automatically, the Mobile Node abides to the following rules:

- If the Mobile Node is in "NOWHERE" state and is not hearing from agents, switch to DHCP_MODE automatically
- If agent beacons are heard when the Mobile Node is in DHCP_MODE, switch to AGENT_MODE automatically

The Mobile Node can also be forced to work only in AGENT_MODE or DHCP_MODE.

A new state is defined for the Mobile Node: CO_LOCATED state. The Mobile Node is in this state after it has successfully obtained an IP address from the DHCP server.

In Foreign Agent mode, the reverse tunnel from Mobile Node to Home Agent is optional. In DHCP mode, this is required, because the existence of a new IP address makes outgoing packets carrying this new IP address as source address if not otherwise tunneled.

Security between DHCP client and server is not considered and left for future study.

K.2.2 Address Binding

Now the Mobile Node has two IP addresses, one home IP address and one temporary IP address, but only one physical interface (assumed), how can these two IP addresses be bound? Fortunately, we have a mvif virtual interface. We bind the addresses in this way:

- bind temporary address to physical interface
- bind home address to virtual interface

The binding of temporary address to physical interface is essential to the functioning of network services. DHCP servers rely on ARP (Address Resolution Protocol) to verify that an address is not occupied before it assigns the address to the client. IP routers sitting at the Foreign Network which will route packets for the Mobile Node also rely on ARP to resolve host IP address to link layer address.

K.2.3 Routing Table Configuration

As explained earlier, in order for the mobile host to appear to the outside world as if it communicates with its home address, we use reverse-tunneling technique. To achieve this goal, the routing table is configured this way:

- Default route is a tunneled route to Home Agent
  - inner source address is set to home address
  - outer source address is set to temporary address
  - inner destination address is the IP address of corresponding host
  - outer destination address is home agent
- An additional route is setup to redirect packets destined for Home Agent being routed through the IP router at the Foreign Network (this information is also given by DHCP server)

K.2.4 An example:

Suppose the name of physical interface is wlp0, virtual interface is mvif0, home address is 131.252.222.80/26, temporary address is 131.252.210.174/26, home agent is 131.252.222.65, IP router of the Foreign Network is 131.252.210.129, then the following is true when the Mobile Node works in DHCP_MODE:

- Address Binding:
  - mvif0 ¡- 131.252.222.80/26
  - wlp0 ¡- 131.252.210.174/26
- Tunnel Setting:
  - inner source address = 131.252.222.80
  - outer source address = 131.252.210.174
- Routing Table:
  - to default, next hop 131.252.222.65, tunneled, interface mvif0
  - to 131.252.222.65, next hop 131.252.210.129, interface wlp0

K.2.5 Configuration and Usage

In order to configure the Mobile Node to use DHCP, a new configuration directive is added. Here is the syntax and semantics:

`dhcp [mvif] [period]`

If this line is defined, the Mobile Node is DHCP capable. "mvif" is the name of the mobile virtual interface. "period" is the time interval in which the Mobile Node tries to renew lease with DHCP server. If this value is 0, then, the Mobile Node will not suggest lease time with DHCP server and will use whatever given by server (typically long, for example, 10 minutes). If this value is non-zero, the Mobile Node suggests a lease time which equals to 2 * period with DHCP server. The value defaults to 10 seconds and the minimum value is 5 seconds.

The Mobile Node switches between DHCP_MODE and AGENT_MODE automatically by default. However, this behavior can be changed by using "mnstat". The syntax is:

`mnstat -m auto|agent|dhcp`

where:

auto – automatic

agent – AGENT_MODE, manual

dhcp – DHCP_MODE, manual

The state information given by "mnstat -S" could be:
CO_LOCATED-ACKED
CO_LOCATED-PENDING
CO_LOCATED-DISCONNECTED?

K.2.6 IPSEC

Efforts are made to make IPSEC still functional in DHCP_MODE. We can setup an IPSEC tunnel from the Mobile Node to Home Agent. To do so, the default route via home agent is set to be secure using kernel IPSEC functions.

Registration messages themselves are not protected by IPSEC (because of the auxiliary route to home agent). Fortunately, they are protected by Mobile IP authentication extensions. For data communication, the IP packet has the following structure if IPSEC is turned on and ESP is used:

```
+--------+--------+--------+--------+---------+
| Outer  | IPSEC  |  ESP   | Inner  |         |
|   IP   |   IP   |        |   IP   | payload |
| header | header | header | header |         |
+--------+--------+--------+--------+---------+
                           |                  |
                           |<--- encrypted -->|
```

Note that now we have two tunnels (IPSEC tunnel and Mobile IP tunnel) and three IP headers. The addresses in these IP headers are:

- outer IP header: src = temporary address, dst = home agent
- IPSEC IP header: src = home address, dst = home agent
- Inner IP header: src = home address, dst = correspondent host

The outer IP header makes packets easily penetrate firewalls; the IPSEC IP header makes IPSEC security relationship (Mobile Node - Home Agent) work; the inner IP makes Mobile IP work (Mobile Node location transparent to correspondent host).

Use `mnstat -r on|off` to turn IPSEC on or off.

K.2.7 Conclusion

Using DHCP in Mobile IP is challenging because we have more network configurations to do. For example, we have two IP addresses instead of one; routing table could be more complex and reverse tunneling has to be used. When IPSEC is considered, we have to make sure that placement of protocol headers inside an IP packet is reasonable. Security problems of DHCP itself need to be future studied. Under any circumstance, the only purpose of DHCP is to obtaining temporary addresses. It should not be confused with Mobile IP, although the use of it can be integrated with Mobile IP, as in our implementation.

K.2.8 References:

1. C. Perkins, "IP Mobility Support", RFC 2002, Oct. 96
2. R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, Mar. 97
3. S. Alexander and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, Mar. 97
4. R. Atkinson, "Security Architecture for the Internet Protocol", RFC 1825, Aug. 95
5. R. Atkinson, "IP Authentication Header", RFC 1826, Aug. 95
6. R. Atkinson, "IP Encapsulating Security Payload", RFC 1827, Aug. 95

K.3 Security Considerations for Mobility and Firewalls

An Internet-Rough-Draft by Jim Binkley (jrb@cse.ogi.edu)

Abstract

In this paper we discuss various issues concerning Mobile Hosts using Mobile-IP or other mobility systems (DHCP) and firewalls. We first discuss some recent attacks on the Internet and what they might mean for such systems as Mobile-IP. We then consider tunnels as used in Mobile-IP (and elsewhere) as a security threat. We suggest how mobile systems may be made "less insecure" with current firewall technology and near-term packet firewall technology that will use Virtual Private Networks and IP layer security (IPSEC). The goal is to describe a security model wherein mobile systems can work across the I and not just as a interior routing protocol within one security and/or interior routing domain.

Introduction and Assumptions

The focus of this document is on how a secure enclave (firewall protected area) may tolerate Mobile-IP and remain secure. By secure enclave in the simplest sense we mean a conventional IP site with one management domain and centralized security administration typically behind one IP packet firewall [Cheswick]. Our focus here is on how a secure enclave can protect itself from foreign (non-local) Mobile Nodes. We also deal with IP spoofing issues.

The discussion here is focused on the network layer. We are not considering higher-level authentication or confidentiality services that might be part of an application-level system. When we talk about firewalls, we are mostly talking about network layer access, and such mechanisms as packet-level firewalls with access control, Virtual Private Networks implemented as IP tunnels, and IP layer security (IPSEC) [ipsec].

When we cite IP addresses in the text, we will use private IP addresses[cite]. These should be viewed as place holders for public ("real") IP addresses associated with an interior routing domain. They are not meant to be private as used "out there" and merely provide address anonymity for the sake of discussion.

The Problem

According to the BUGTRAQ list [cite] and recent CERT [cite] advisories the Internet has seen a set of attacks whereby attackers use various forms of address spoofing at the IP level. Some of the attacks are denial of service oriented. Some seek to cause the attacked system to crash or hang. Many of the attacks can be characterized as single packets wherein the IP source and destination addresses in the IP header appear to originate within the attacked systems site. We will mention three such attacks, TCP SYNC [CERT cite], smurf [CERT cite], and land [CERT cite].

In TCP SYN attacks, the attacker sends TCP initialization packets to a given site. The attacker system is tied up simply due to opening TCP control blocks, hence allocating internal memory. The attacker need only send one TCP SYNC packet. The attacker may choose to use a spoofed IP source address so that tracing the attack back to its originating system is difficult.

In smurf attacks, the attacker sends one or many ping packets to an IP directed-broadcast address with an IP source address that may also be in the site. For example, if a site had a site specific class C address along the lines of 192.168.1.0, the attacking IP destination might be 192.168.1.255 or 192.168.1.0 (0 broadcast addresses may be used as well). The IP source address may be external or internal. The result is that two systems (at least) may be attacked. The IP source itself is bombarded with ping reflections from all the systems at the directed broadcast address. Further the smurf vehicle may also be used for single packet "ping of death" attacks.

Land attacks involve one TCP packet in which the IP src is set to be the same as the IP destination. The attack may cause the victim to hang.

In general, note these attacks do not involve packets being returned, unless the packets are returned to another system that is being indirectly attacked itself (smurf). They may serve a denial of service function either by tying up network resources or causing systems to reboot. One general technique that can be used against them is to disallow IP spoofing for internal source addresses [cite]. Packets entering a network may not possess an IP source address internal to the network. Packets leaving the network must possess an internal IP source address.

We will briefly look at how this may be done with a Cisco router which we assume is the interface between a site having 172.16.*.* as its address space and the Internet at large. Note that the access list entries shown here may be part of a more complex firewall policy and/or access list, but we only show the part relevant to IP spoofing.

We apply the following access list to packets headed out on the external interface:

```
        access-list 111 ...
        access-list permit ip 172.16.0.0 0.0.255.255. any
        access-list deny ip any any log
```

This blocks packets headed out that do not have IP src addresses in the 172.16.*.* range and logs any internal attempts at spoofing. Thus spoofing attacks cannot originate in this site.

The following access list entry may be bound to an external router interface and would be applied to packets entering the site.

```
        access-list 101 ...

        access-list 101 172.16.0.0 0.0.255.255 any
```

The result is that a site firewall or border router will neither send or receive packets over an interface when the packets do not belong to the source IP routing domain. In addition, it has been suggested that firewalls might not accept unicast packets over a given interface when their IP source address would not be the unicast routing interface. [see cisco/firewall rfc].

The result of such anti-spoofing measures is that Mobile-IP packets headed in to "foreign" Mobile Nodes; i.e., Mobile-Nodes from some other site than 172.16..*.* will not be permitted to enter. Packets trying to leave from systems from another site, say 192.168.1.0, will not be permitted to leave. Mobile-IP within the same routing domain, which we might call "interior Mobile-IP" would be permitted. Mobile-IP between two interior routing domains would not be permitted.

There are a couple of considerations that should be pointed out at this stage. First the Mobile-IP registration protocol would still work, unless blocked (see below). Hence a foreign visitor at the 172.16.*.* site could run the Mobile-IP registration protocol to its Home Agent at say 192.168.1.1. This is because the Mobile-IP protocol may be viewed as an application level entity (UDP port 434) and a "Care Of Address" (a local non-mobile IP address) could be used as the IP source address for the registration packet itself. Hence it will reach the Home Agent across the firewall and not be blocked by the IP spoofing measures given above. See the figure below:

```
                                                     172.16.1.1
Home Agent: <-- MIP registration --> Firewall <---> Foreign Agent
192.168.1.1                                            |
                                                       |
                                                       |
                                                    Mobile Node
                                                    192.168.1.2
```

The Mobile Node would acquire the COA 172.16.1.1 from the Foreign Agent and use that to get its Mobile-IP UDP registration packet to/from the Home Agent. In the picture above, in security terms, the Foreign Agent is basically a application-level bastion host for Mobile IP.

The irony is that Mobile-IP will still work, but the Mobile Node will not be allowed to talk to home with ordinary data packets. Data packets tunneled from the Home Agent to the Foreign Agent will be allowed through. The Mobile Node may still engage in full 2-way traffic with nodes in the 172.16.*.* domain. It might also (if not blocked) attack systems behind the firewall itself with the one way attacks previously described.

Tunnels Considered Harmful

One mechanism that is part of Mobile-IP and in point of fact many other routing protocols are IP tunnels which might be implemented with IPIP[cite] or Cisco's [GRE]. It should be pointed out that IPIP tunnels are not peculiar to Mobile-IP. They are used in many routing protocols for many purposes including tunneling non-IETF protocols (e.g., Appletalk) or building virtual networks on top of the current Internet (MBONE, 6BONE).

Tunnels may mean encapsulated packets where one has one IP datagram inside another IP datagram and we will use IPIP (IP protocol 4) as our example here. Mobile-IP contains at least 3 forms of tunnels, IPIP, GRE, and so-called {TBD} direct encapsulation tunnels.

Mobile-IP uses tunnel mechanisms like IPIP to forward packets from the Home Agent to a remote "Care Of Address". The COA is a local site address that may represent a Mobile Node that has acquired a local IP address itself directly via DHCP or a router system that speaks Mobile-IP called a Foreign Agent. Any Mobile-IP system, including Mobile Nodes, Home Agents, or Foreign Agents, may source or sink tunnel packets. When a Home Agent forwards packets to a Mobile Node that is at a Foreign Agent, the use of IPIP may appear as follows:

```
IP outer                    IP inner                  IP datagram
ip src= Home Agent          ip src = peer end host
ip dst= Foreign Agent       ip dst = Mobile Node

|
| packets to MN
v
Home Agent ========= IPIP tunnel to COA ===>   FA/MN
```

One might ask if it is enough to simply use IPIP tunneling and tunnel somehow either from the Foreign Agent or Mobile Node back to the Home Agent and thus evade the anti-spoofing measures at a firewall? This is an insecure approach.

In the first place, it is not enough to simply tunnel over the IP spoofing firewall. This is simply "IPIP spoofing". The problem is that if one has a tunnel sink (be it any kind of agent or Mobile Node) that decapsulates packets and then forwards them, others can launch their spoofing attacks with IPIP too and thus have the spoofing emerge "inside" the firewall. For example, smurfing might simply be redirected through a tunnel. The inner IP header might be directed broadcast with an interior IP source named as a target.

Solutions here may involve secure network design, which is not a new idea where firewalls are concerned. We will suggest that one can in general block tunnels with current access list mechanisms and thus control tunnels so that tunneling from the outside can only be done to certain hosts that will be considered as bastion hosts. For example, with Cisco IOS one can add the following statements to access list 101:

```
access-list 101 deny ipinip any any
access-list 101 deny gre any any
```

and thus block any IPIP or GRE packets coming in over the router. One can further add a permit statement to force any IPIP coming in to land at a certain host and then treat that host as a bastion host; i.e., a nexus of security focus.

For example,

```
...
access-list 101 permit ipinip any host 172.16.1.3
access-list 101 deny ipinip any any
access-list 101 deny gre any any
...
```

in an input access list, means ipip will only be allowed to the host 172.16.1.3, which we will assume is a tunnel sink agent.

We have focused internal trust for IPIP on that one system. We next need to explore how to make sure that packets arriving at the Home Agent are *not* attacks that can be made on a tunnel sink. This can be done with IPSEC (site) tied to Virtual Private Network (secure tunnel) technology. We will explore this idea further in the next section.

Mobile Nodes Abroad

In this section we will consider the problems for a secure enclave if Mobile Nodes in that secure enclave go abroad; i.e., out to the Internet beyond the firewall. In essence, we are considering the problems for "our side". There are two sets of problems here. We must ask how the Mobile Node can secure its own traffic and in effect, take its security enclave with it. We must also ask how the enclave can secure traffic coming from that Mobile Node back inside, which will extend the thinking in the previous section.

We must point out that the security problems here are not terribly different from those encountered by current dialup clients into a secure enclave that have access to the enclave via a internal terminal multiplexor. One crucial difference is that data in our case will be assumed to be sent across the Internet, and not a telephone switching system. Thus some think that the data itself need not have authentication and/or confidentiality measures applied to it. All that is needed is a system along the lines of PPP with CHAP[cite] and RADIUS[cite] for one-time authentication. However commercial products are available from Cisco and Microsoft [cite] that at a high level are very similar to what we propose here. The exterior host tunnels into a secure enclave and an agent in the secure enclave applies crytographic measures to packets that have came in from the outside.

Glass and Gupta[cite] suggest that Mobile-IP Mobile Nodes abroad may use DHCP to acquire "local" IP addresses, thus they can get by the anti-spoofing measures in the firewall router. Further, the Mobile Nodes can use IPSEC with two-way tunnels between the Home Agent as a classic bastion host and the Mobile Node. (See [http://www.cs.pdx.edu/research/SMN for a combined Mobile-IP IPSEC system in which Mobile Nodes can do two-way MN/HA ESP tunnels). Please see the figure below:

```
                               Internet            172.16.1.1
Home Agent: <----> Firewall <-----------------> router/DHCP server
192.168.1.1                                           |
                                                      |
                                                      |
                                                   Mobile Node
                                                   172.16.1.2/
                                                   192.168.1.2
```

Packets sent from the Mobile Node to the Home Agent might have the structure:

IP(1) | IP(2) | <IPSEC> | IP(3)

Each IP header has its own purpose. The most external header, IP(1) exists to get the packets to the Home Agent with an IP source address == 172.16.1.2 acquired from a local DHCP server. The IP destination would be 192.168.1.1. Thus this header allows transit of the Internet and any firewalls. When the packet arrives at the Home Agent, that header is discarded and the IP(2) | <IPSEC> IPSEC tunnel portion is processed. Here we assume that the Mobile-IP address 192.168.1.2 is used for the source address and the Home Agent is again the destination. The fixed Mobile-IP address may be needed here as it allows a priori manual IPSEC keys to exist between the Mobile Node and the Home Agent. In effect, this is an IPSEC tunnel. The interior header would contain the Mobile Nodes fixed address (192.168.1.2) and the address of any destination to which it is allowed to send packets.

The above triple-header system may possibly be optimized by a higher-level protocol that could produce a dynamic binding between the local DHCP-acquired COA and the Home Agent's destination address. This would allow one header to be deleted. Such a system could be used by Mobile Nodes abroad that are not using Mobile-IP and are simply using DHCP for a simpler form of mobility.

We must also consider the threat to the secure enclave itself. The tunnel-sink agent must guarantee that all packets sent to it via tunnel are cryptographically verified; i.e., shared secret keys exist between it and the Mobile Node abroad and no packets that are leaked to it by the firewall will be internally forwarded until they have been verified. This might be done with an access list mechanism tied to IPSEC or by simpler means. For example, the PSU system mentioned above has a BSD sysctl(8) switch

```
sysctl -w net.inet.ip.mvifipsecinput=1
```

that if set forces the IPIP driver to only forward packets if and only if IPSEC authentication or decryption has successfully occured between the remote system and this system. This allows a Home Agent to only internally route packets post successful IPSEC processing.

If a system abroad can send a simpler tunneled packet into our enclave, it might take the form:

IP(1) | IPSEC | IP(3)

The local firewall might then permit only ESP (ip proto 50) or AH (51) to the Home Agent depending on local security policies:

```
        ...
        access-list 101 permit 50 any host 172.16.1.3
        access-list 101 permit 51 any host 172.16.1.3
        access-list 101 deny ipinip any any
        ...
```

Hosting Visitors From Abroad

Paranoia in the defence of internal security is no vice. Thus we could start from a position of not allowing visitors. However we wish to construct a system that will allow external visitors to visit our site securely. We suggest than an approach that can be made with current (or near-current) technology involves secure network design. A basic principle is: "design the network so that visitor packets are not allowed inside". We may observe that whatever is done will probably be similar to current systems that have two-level security enclaves.

For example, one might have a network designed as follows:

```
                      ^
                      |
                   Router(1)
                      |
             ------------------------- internal less secure DMZ
             |        |             |
           Firewall(2) term mux      DHCP/router for Mobile Visitors
             |
          ------------------------
          | bastion hosts       | Firewall(3) (secure enclave)
```

The above may be viewed as a logical (not necessarily physical) structure that may be optimized by careful design. We have a Router that simply serves to allow access (possibly serial) to the Internet. Inside the external router we have a less secure network that may serve to allow unfiltered access to the Internet. This network might include terminal multiplexors and local mobility servers. Behind it we would have at least one level of firewalls with bastion hosts (which might e on the less secure network as well).

The above may be viewed first as a physical system that would tolerate visitors. However it can probably be optimized with a number of techniques. Let us consider two methods that may be used to logically separate networks and thus remove the difficulties that may be imposed on a site by physical construction. Note that these techniques *require* careful network security design. The security onus here is certainly on the designer, who must be careful and have confidence in these techniques.

One might use Virtual Lans (VLANS) [cite CISCO] to isolate visitors to a particular subnet and then make sure there are no unintended local link systems that are accessible to a visiting Mobile Node. VLANS can be used 1-1 with IP subnets, hence packets sent from the visitor will have to visit a router and access lists can be applied to interior interfaces. In the above system, Router(1) or Firewall(2) (which might be the same system), could apply an access list on interior secure interfaces to prevent access to interior systems by the less secure network. Please see the following picture:

```
                ^
                | Internet  (vlan 2)
                Router --------------------- DHCP/server (insecure net)
                | (vlan 1)                    192.168.2.0
                secure net 192.168.1.0
```

Packets from 192.168.2.0 might not be permitted to access the secure network and could only go outside. Note that access lists on the Router above must be applied to the vlan 1 interface.

One might also use Virtual Private Networks in the guise say of tunnels between interior routers that disallow foreign packets from internal access. As a very simple possibility consider a combination DHCP server/router that would simply construct a 2-way IPSEC VPN between itself and a network that was "outside" the secure enclave:

```
  | --------------------------------- Internet accessible network
 Firewall(1)           VPN tunnel /router(2)
  |                             ^
  |                             |
  v                             |  VPN tunnel
secure enclave                  |
                                v
                       DHCP/server router (3)
```

In this case the VPN would make sure that any packets sent by visitors on a visitor link would be securely tunneled outside. We suggest that careful combination of access lists with VPN technology should allow the above picture to be collapsed in various ways. For example the Firewall(1) and router(2) systems might be the same.

Miscellaneous Considerations

Firewall Discovery

(This section might be burned if a reasonable citation cannot be found).

It has been suggested by some (cite) that some sort of Firewall Discovery system might allow Mobile Nodes to dynamically tunnel to and from firewalls. There are several problems with this notion:

1. It is complicated and probably unnecessary since our solution here will work with current firewall technology.

2. It is not very likely from a security point of view. Security people may not care for notions that poke holes through firewalls in the first place even when done dynamically. Complexities involved in cross-security domain certification are likely beyond near-term scope. The security folks "at-home" may not care for the notion of a Mobile Node somehow securing its packets between itself and a foreign Firewall. After all, that firewall might choose to store all data traffic, and enable a classic "man-in-the-middle" attack.

3. Traditional notions of IP Fate Sharing (considered bad) may apply here. Mobile-IP systems are already tied to the fate of their Home Agent. Additional tunnel ties between systems that are not related from interal routing or security enclave considerations may be complex. After all, it is hard to predict how many firewalls that rule out IP spoofing to/from a given cite may exist.

The Role of a Mobile-IP Foreign Agent

It would seem that there is little role for a Mobile-IP Foreign Agent in the system we have outlined above. However this is not necessarily the case. There is in general, no reason, why DHCP and Foreign Agents cannot be combined. Local nodes may perform some sort of security handshake with a local Foreign Agent [cite MOIPS] and avoid using DHCP for a local IP address. Once a Mobile Node has identified itself to a Foreign Agent as belonging to the agent's secure enclave, it could use a IPSEC VPN between itself and the agent. Any packets verified by the Foreign Agent as local could thus be delivered locally and not sent over the VPN from the Foreign Agent outside the secure enclave. Thus locals and visitors could both be tolerated at the same agent link.

Of course a local enclave might choose for policy reasons to force all visitors to be "local" or "foreign". "Locals" could be always treated as remote visitors and tunneled outside, thus having to use secure means to come back inside. Or foreigners might simply not be permitted entrance at a given agent. Both policy considerations are possible and should be considered in implementations.

Conclusions

In this document we have presented secure network proposals that will enable Mobile Nodes from abroad (or nearby) to securely access the Internet. We point out that such systems are not dissimilar from current dialup systems that involve a remote PPP-based dialup client and a local terminal multiplexor. VPN mechanisms may be used between the Mobile Node system and its home security companion. Very simply put, the Mobile Node is an extension of the local security domain. In addition, one must also consider how to make that security enclave secure in the face of foreign visitors.

In summary, we will make the following suggestions:

1. DHCP to acquire a local COA solves IP spoofing problems and may or maynot be combined with Mobile-IP.

2. Suitable two-way crytographic tunnels between a system abroad and a routing system at home will allow the Mobile Node's own traffic to be securely tunneled over the Internet.

3. IPIP tunnels sans cryptographic safeguards should be viewed with caution. If an IPIP tunnel sink does not guarantee cryptographically controlled access, an attacker may tunnel various attacks (land, etc.) into an enclave.

4. Flexibility in routing, access list mechanisms, and VPNs should be considered by implementors with the goal of allowing mobility.

5. security considerations must apply both to Mobile Nodes abroad, their implact on the home enclave, and also how an enclave itself might be designed to tolerate visitors.

biblio

mip rfc
ipsec
cert
bugtraq
cisco firewall design
TIS vpn document
firewall book
PSU web site

Appendix L

Quarterly report – Spring 1998

L.1 Project Status Overview

Progress during this quarter has been minimal as only a single graduate student, Bjorn Chambless, remains on the project. Bjorn is responsible for the HARP effort reported below. He will be working with Jim Binkley on the summer code release which will contain HARP as well as the MADRP and DHCP/IPSEC reported last quarter. John McHugh, Jim Binkley, and Sarah Mocas are working on papers describing the results of the project. One of these will be a paper on policy issues associated with mobility. A preliminary version of this paper will be presented by John McHugh as an invited talk the Naval Postgraduate School in August. Jim Binkley is currently working with John Richardson of Intel on a draft RFC on Mobile-IP and Firewall security considerations. We believe that we have an effective solution for the "Mobile Firewall" problem.

L.2 HARP – Bjorn Chambless

The Home Agent Redundancy Protocol (HARP) was completed and deployed during the Spring quarter. HARP provides additional security and robustness for the secure mobile IP system by allowing the transparent use of redundant Home Agents(HAs) by Mobile Nodes(MNs). A mobile node served by the HARP system will retain connectivity in spite of the failure of a Home Agent and/or a home subnet.

A pair (or more generally, a set) of active home agents share mobile node registration information through an IPSEC secured TCP tunnel so that each is able service any mobile host for which the group is responsible. In the event of a Home Agent failure, the route to the home network maintained by interior domain routing protocols (e.g. RIP, OSPF) is assumed to converge to one of the remaining hosts offering connectivity to the home network, i.e. a surviving home agent.

The HARP system was initially implemented and tested on a small three-host test network in which one host acted as a router and the other two acted as a home agent pair. The two HAs in turn provided connectivity to same virtual home subnet through their radio interfaces. Since the Home Agents shared registration information it was possible for either one to receive registration packets from a MN and forward packets addressed to a MN to an arbitrary Care of Address. This was tested by manually switching the route to the home subnet between the two Home Agents. Interoperability between the Mobile IP daemon and the FreeBSD routing daemon using the RIP protocol was also tested.

When deployed at PSU we sought to maximize survivability by locating the two Home Agents as far apart as possible. This was to insure that a local network disturbance, e.g. power failure, would have less likelihood of affecting the operation of the Mobile IP system. We further tested the system by repeatedly modifying the routing table on the local Cisco router and checking that connectivity for the Mobile Nodes being served was maintained.

HARP is currently in the final testing stages and will be contained in the Summer 98 code release along with MADRP and DHCP/IPSEC.

Appendix M

Quarterly report – Summer 1998

M.1 Project Status Overview

At this point our project has wound down. As a consequence this report will be short and in the form of a status report.

During Summer 98 Jim Binkley (faculty) did some pro bono project work and Bjorn Chambless (our last graduate student) remained employed as a M.S. level research assistant. We will briefly summarize our accomplishments for the summer in order of importance. The major accomplishment was our final source release which included student work from the previous year.

M.2 Status Items

Jim Binkley and Bjorn Chambless combined to produce our final major milestone, i.e., what we call the summer 1998 Secure Mobile Networks software release. This release is available on our project web page: http://www.cs.pdx.edu/research/SMN as the file mip-summer98.tar.gz. Please note that we reused the export-controlled bits at MIT, thus we did not have to re-release the kernel /netsec directory.

The release included the final MADRP, Multicast Adhoc Demand Routing Protocol work done by Jennifer Ye who graduated at the end of winter 1997/1998. It also included code done by Zhong Chen who made our Mobile-IP daemons work with DHCP as well with our simple IPSEC "away" policy scheme; i.e., we could combine DHCP/IP¡IPSEC¿IP tunnels so that a mobile node when away could tunnel home. This is one part of a "solution" for VPN-line mobile systems accessing home over anti IP-spoofing firewalls. Bjorn Chambless finished the Home Agent Redundancy Protocol and it was part of the release. More information on the release can be found in the summer release README file ftp://ftp.cs.pdx.edu/pub/mobile/mip-summer98.README. A copy of this appears as appendix M.3.

M.2.1 Wireless Network Real-Time Monitoring

Jim and Bjorn collaborated to deploy various server agents in order to instrument our wireless agent infrastructure. The infrastructure itself is now established in four buildings in the School of Engineering and Applied Sciences at PSU. It has a modest number of academic and staff users including system and network administrative staff as well as professors and graduate students in Computer Science and Electrical Engineering. The home page for the Seas Wireless Network can be found at:

http://guinness.cs.pdx.edu

The first link: "SEAS building agent map" shows currently deployed agents in the form of a very rough campus map (and is lacking information about a newly instrumented building).

The second link: "Wireless Agents in SEAS/Availablity - (via) Big Brother" shows the wireless agent infrastructure in terms of whether or not the mobile routers are operational. It uses the public domain system called Big Brother which displays graphical information about various system components on the web in "near" real-time. For example, if a system goes down, email will be sent to the adminstrators (Jim and Bjorn) in about five minutes time. Big Brother uses ping, disk checks, CPU process checks and other tests including testing the status of certain ASCII-oriented network servers like SMTP/sendmail. In our case, all our agents run snmpd and our mobile-ip daemon. The "procs" sub-field checks for the existance of those servers so we will be notified if they should crash. Big Brother is one of the two public domain applications we have installed across all of our agents (the other being the UCD snmp daemon).

The third link: "MRTG/SNMP - Network Traffic Display for SEAS Wireless Agents ...". Here we use the highly popular MRTG "Multi Router Traffic Grapher" tool which is essentially an SNMP-based system to monitor a few aspects of our more critical agents. We essentially use it to graph bytes in and bytes out on wireless and ethernet interfaces. MRTG[1] is a general tool and can actually be used to graph script-based applications as well as SNMP derived data variables. We have one link that is an attempt to graph our current Mobile-IP mobile node user count[2].

M.2.2 Linux Port

As a side project, Bjorn Chambless ported our mobile-node specific routing daemon code (not agents, just Mobile Nodes) to linux. He has made a release of this code on our home page. We are finding that many faculty members and students have linux on laptops and they may be able to make use of this code to join our Mobile-IP based wireless network. We have three candidate faculty members lined up and are in the process of making them "mobile".

M.2.3 Policy Work

As the work has come to a conclusion, we have spent a considerable amount of effort thinking about the policy implications of mobility. A journal publication is in progress, and several presentations have been given or are planned for the near future. Past presentations by John McHugh included ones at the Naval Postgraduate School in August and at Intel and Oregon State in November. A presentation at the CERT is planned for December.

Jim Binkley and John Richardson (of Intel) have prepared an IETF draft entitled "Security Considerations for Mobility and Firewalls" which appears as appendix M.4.

M.2.4 Just For Fun

As a demonstration of the mobile-IP linux laptop capability, we have hooked up a "quickcam" parallel-port based color camera to a linux laptop and have created a laptop/camera system that can be easily deployed anywhere within our wireless network infrastructure. We call the camera the "bjorncam"[3]. Of course, Bjorn may not actually be available at any given time. His bjorncam page records a few of his nearby mobile journeys and a few people encountered on his trips.

[1] (See http://ee-staff.ethz.ch/ oetiker/webtools/mrtg/mrtg.html)
[2] See http://guinness.cs.pdx.edu/mrtg/guinness.mip.html
[3] The official homepage for the bjorncam is: http://bjorncam.cs.pdx.edu/bjorncam.

M.3 Summer Release Information

```
A Mobile-IP Implementation based on rfc 2002
--------------------------------------------

Summer '98 Release for FreeBSD 2.2.1.

Jim Binkley,
John McHugh,
Sarah Mocas,
David Reeder,
Bill Trost,
Zhong Chen,
Bjorn Chambless,
Jennifer Ye

Portland State University
Computer Science Dept.  - Summer, 1998

email: mobileip@guinness.cs.pdx.edu

-----------------------------------------------------------------
Introduction

Please see the file README for the main documentation.  This
README.summer98 file only covers new features in the latest release.
Thus README.summer98 is a supplement.  README discusses how
to install the basic release.

In general the work here represents student work done on
the Secure Mobile Network project by M.S. students at PSU.
This release represents a feature upgrade for our release of summer 97.
It mostly adds only functionality, some of it experimental,
to the previous integrated IPSEC-Mobile-IP system.
A lot of the work here uses architectural features resident
in the code from summer 97 and is built on top of it.

The contents of the original README are still important
and it should be read first.  Then read this README.summer97
for more information.  this release is still based on the
FreeBSD 2.2.1 kernel, but we hope to make a minor release
based on the latest FreeBSD kernel RSN.

New features for summer 98 include master degree student
work and some features added in late summer 98.  The main
features which will be discussed briefly here are:

0. an internal improvement in mnd and mipd so that they use
the route(4) socket as opposed to route(8) and arp(8) calls.
Thus they are much more efficient.  We still supply modified
route(8) and arp(8) binaries and source, as they can be used to
manually setup Virtual Private Network connections along with IPSEC.
Given that these improvements are internal, there will be no further
documentation on this particular feature.

1. Jennifer Ye implemented a network-layer experimental ad hoc
routing protocol called MADRP, for "Multicast Adhoc Demand Routing
Protocol".  The code is included in mnd and mipd but it is NOT
```

compiled in by default (#ifdef MADRP). Functionally MADRP assumes that the previous ADHOC mode (ADHOC version ARP) is used for link-layer mobile communication. It allows Mobile Nodes that are separate from either Mobile-IP agents or other Mobile Nodes to set up host routes between them; i.e., packets are automatically forwarded across Mobile Nodes themselves. Connectivity between MNs and agents is also established; hence if a Mobile Node is separated from the wired infrastructure by other Mobile Nodes it can still find a routing path to Mobile-IP agents (and its own HA). See the MADRP section below for more information.

2. Zhong Chen has added DHCP capability for Mobile Nodes which may or may not be combined with the pre-existing IPSEC tunnel functionality between Mobile Node and Home Agent. DHCP is basically viewed as another link mode. We assume that DHCP servers are available on a link (and that ARP is used ...), thus Foreign Agents are not available on a link. This feature applies only to the mnd daemon. See the DHCP section below for more information.

3. IPSEC integration with Mobile-IP code. There are two features worthy of note. IPSEC AH and ESP may be used simultaneously for the route binding. Also IPSEC AH may be used with manual keys between Home Agents and "same security domain" Foreign Agents; i.e., instead of IPIP tunnels between HAs and FAs, one may use IPAHIP tunnels. There will be a brief discussion of this feature set below.

4. Bjorn Chambless (PSU CS/MS student) has implemented and released HARP, the Home Agent Redundancy Protocol. HARP allows for Mobile-IP Home Agent redundancy; i.e., one can have TWO Home Agents without the rest of the Mobile-IP systems (MNs/FAs) having to change or being aware of the co-HAs. The co-HAs cause no changes in the MIP protocol itself. They use the HARP protocol between themselves to communicate MIP routing information (where MNs are registered). See the HARP section below for a high level description, and configuration information.

Please note that for the supplied binaries and the source in mip.src, HARP, and DHCP are compiled in and turned on. The source for MADRP is included, but MADRP is experimental and the Makefile in mip.src has MADRP left out at this point. The MADRP code is not in the supplied binaries but of course they can be recompiled if you wish to experiment with it.

It should be possible to modify the Makefile and leave those #ifdef values in/out to produce binaries that have or do not have that functionality. Minimally we strongly urge that HARP and DHCP be left in the feature sets however. MADRP is interesting but is experimental.

----
1. MADRP - Jennifer Ye (programmer) and Xu Hao (designer)

MADRP is an experimental high-level ad hoc routing protocol. It may be viewed as an Interior Gateway Protocol. It allows Mobile Nodes to route packets over other Mobile Nodes which act as routers. It is very roughly based on or inspired by work done by Dave Johnson and Scott Corson in the distant past. MADRP contains certain design elements though of interest that include:

1. integration with Mobile-IP; i.e., Mobile Nodes that are more than one hop away from an agent can still use Mobile-IP to retain Internet connectivity.

2. the subnet-less nature of the previous link-layer ad hoc mechanism is still used; i.e., cooperating MNs do not assume they can only talk to other MNs of the same subnet.

3. MADRP hosts set up end to end host routes to other hosts they are talking to. Given that our IPSEC implementation is tied to routes in the routing table, this allows us to do end to end IPSEC between two participating laptops in so-called "tunnel mode".

Given that the Mobile Nodes are not gateways, such IPSEC tunnels are not subject to proposed plaintext attacks. For good or bad, all of the packets sent between the hosts would be covered by the same IPSEC security association. This is an interesting but experimental notion.

Note that MADRP assumes our link-level ad hoc scheme. It does not work with ARP. By definition, it uses any kernel routing mechanism involved with the link-level ad hoc scheme on which it is built. The protocol itself does not assume simple ad hoc or ARP, but the implementation certainly does.

Functionally MADRP uses UDP port 1066 and 224.0.0.11 as a multicast address. Given MADRP's experimental nature, these numbers are not cast in concrete and can be changed if desired (but at this time they are compiled in). See mip.src/sr_config.c for these and other configurable constants.

The protocol is actually fairly simple. If the kernel discovers that a route to a particular host does not exist (or mnd decides that a MADRP discovery protocol packet should be sent), e.g., due to a kernel route(4) socket upcall to the daemon ... mnd will send a MADRP multicast discovery packet. Participating hosts or agents may or may not forward the query decrementing the ttl (this depends on the limited flooding algorithm). Agents do not forward the query on internal (wired infrastructure) ports, but do forward it on wireless ports where the ad hoc link layer is used. The correct ip destination returns an answer (which may be multicast or unicast depending on configuration). Agents always return an answer (and thus in a sense act as a possible default route). Intermediate routing systems (other Mobile Nodes) setup host routes themselves typically back to the sender as MADRP queries (or multicast replies) are sent out and forwarded. As a result a routing path is setup.

One interesting trick is that Mobile-IP works simply because the mnd tries to find a host route to its Home Agent. If any agent can be found (possibly across intermediate mobile nodes), Internet access will thus be available. Another trick relies on TCP (the protocol) as an optimization. If TCP discovers that a round-trip "host route" fails due to the other end failing to return TCP ACKs, mnd is notified, and it will reinitiate the end host discovery mechanism.

MADRP at this time lacks much needed configuration and runtime status information.

Further MADRP documentation can be found in three places. The original ASCII specification as written by Xu Hao can be found in the papers directory. Jennifer Ye wrote a term paper on the implementation. Various descriptions of the protocol can be found in our project reports at http://www.cs.pdx.edu/research/SMN.

MADRP can only be compiled in or compiled out in the current implementation. See mip.src/Makefile and insert #ifdef MADRP as appropriate and recompile everything to add it.

----

2. DHCP/IPSEC - Zhong Chen

The DHCP mode causes mnd (the Mobile Node daemon) to be DHCP aware as well as Foreign Agent aware. This functionality only affects the Mobile Node, not Agents. DHCP is compiled in to the source base (#ifdef DHCP), but is not configured on by default. One and only one configuration line is necessary in /etc/mnd.conf in order to make mnd DHCP-aware.

```
dhcp mvif0 [lease-period-in-seconds]
```

The mvif device is used for "pushing" the Mobile-IP fixed node address (the home IP address) when a mnd system is resident at a DHCP server link. The lease-period-in-seconds value is optional. It defaults to 10 seconds. The MN will use the renewal period to try and detect movement; i.e., it will ask any local DHCP server for an address, and if the address is different, it will assume movement. A period of 60 seconds might be reasonable for casual mobility. Note that we assume that Foreign Agents are slightly better than DHCP-servers; i.e., if we don't have a FA, we will try and use DHCP (MN will make a DHCP query).

We should point out that the ISOC DHCP implementation was used on the servers and was of assistance in the creation of our DHCP code.

In addition to stock DHCP usage, mnd can also be IPSEC aware and tunnel packets to/from the Home Agent. When at a non-home (foreign) subnet, the security policy as specified with the mnd.conf line:

```
ipsec_mn bypass_fa ... etc
```

can be used; i.e., the MN will send IPSEC'ified packets to the Home Agent. The external IP header will use the DHCP local address as its IP source. The internal IP datagram will be shielded with ESP, hence the MN can send packets to/from home sans IP spoof filtering problems at nearby routers. HA IPSEC functionality is of course unchanged from the Foreign Agent (non-DHCP) IPSEC case).

As a security note, it should be pointed out that this mechanism
solves one of the so-called Mobile-IP firewall security problems;
i.e., it allows Mobile-IP to work in the presence of ip-spoof
filtering firewalls. It does *not* of course solve the problem
of protecting ones own security enclave from alien and presumed
hostile mobile nodes.

----

3. New IPSEC integration functionality (David Reeder)

3.1 two IPSEC bindings at once (combined AH/ESP)

Either route(8) or mnd may use combined AH+ESP headers
(note: we have no support for the newer form of ESP, just the older
separate forms) as opposed to (the older) ESP by itself.
For use with route(8), please see the IPSEC-related
modified man page on route (in our distribution tree, this
is found at ../sbin/route/route.8). nroff -man route.8 | more.
Note that route add can be used to add the first IPSEC association,
and route change is used to add a second IPSEC association.
One must a priori have two IPSEC associations (one for ESP
and one for AH) in /etc/keys which have been loaded with keyadmin(8),
and then one uses route(8) to tell the kernel to bind the IPSEC
associations to the routes in the routing table. Packets sent
through the routes will thus have the IPSEC modifications made to them.

This functionality may also be used with our Mobile-IP daemons.
As before, one must first setup /etc/keys and use keyadmin(8) to
load IPSEC associations into the kernel. The routing daemons (mipd)
will then use this information when routes are setup or torn down
automatically as Mobile-IP is used.

See the mnd.conf(5) man page for more details. The basic idea
is that e.g., with the mnd.conf configuration file line:

```
ipsec_mn at_ha ha_ip_address ipsec-protocol SPI-pairs [off]
```

instead of using ONLY esp, ah for the ipsec-protocol, one can
also specify ah+esp for the combined form. An additional SPI-pair
is needed here in the configuration line. See the mnd.conf(5)
man page for mnd functionality. See mipd.conf(5) for the related
HA-side functionality (again ah+esp is used). Note that functionality
may also be used with ipsec tunnels. Please see the mnd.conf
and mipd.conf man pages for more details.

3.2 IPSEC HA/FA tunnels

We have worked out the kernel details so that it was possible to
combine IPIP tunnels directly with IPSEC attributes. This
can be done manually with the route(8) command. We also
integrated this functionality directly with the mipd Mobile-IP
daemon. Of course, we still assume manual IPSEC keys (although
there is no reason a more general policy description could exist
that would allow ISAKMP (IKE) to automatically setup tunnels). The basic
idea is that if a priori configuration exists in both
(IPSEC spis) /etc/keys loaded at boot, and mipd.conf (control
for MIPD), then mipd will install per Mobile Node IPIP tunnels

so that a AH security association exists between a Home Agent
and a Foreign Agent. As a security PRO, this means that Foreign Agents
can be configured to
1. not accept plain IPIP packets and hence only accept IPAHIP
packets from hosts with which they share a security assocation.
(see the sysctl net.inet.ip.ipsecforwarding switch description in the
README file)

As a criticism, in truth, this feature sadly needs ISAKMP and
some one to centralize key storage in order to make the
IPAHIP tunnels dynamic in terms of configuration. With only manual
keys and a few agents, there are too many manual keys to keep straight.

We assume that the net.inet.ip.ipsecforwarding switch is used at
agents to prevent the acceptance of plain IPIP packets.

2. Configure IPAHIP tunnels as opposed to IPIP tunnels
between agents (or HA/MNs acting as their own FAs).

In order to configure a "ipsec_tun" tunnel which is a one
way tunnel between a HA and a FA, one does the following:

2.1 add an AH association in /etc/keys .
in both HA and FA files add a one-way AH association between
        the two in /etc/keys and of course make sure that is loaded
        with keyadmin(8).

2.2 /etc/mipd.conf ipsec_tun switch

In mipd.conf files make sure that relevant ipsec_tun ha (at the HA)
and ipsec_tun fa (at the FA) lines exist so that mipd knows to
use the security association. As a consequence IPIP packets will
be replaced with IPAHIP packets between the HA and FA.
See mipd.conf(5) for more details

----

4. HARP - Bjorn Chambless.

Overview:

The Home Agent Redundancy Protocol allows Mobile IP to maintain
Mobile Node (MN) connectivity in the event that a Home Agent (HA)
fails or some forms of network partition appear between the Home
Agent and the current care-of-address of a MN.

The protocol operates by extending Mobile IP so as to allow
multiple Home Agents to service a given set of MNs, thereby ensuring
that the loss of a single HA does not lead to loss of network connectivity.

Cooperating Home Agents share registration information so that either
HA may forward packets at any time to a Mobile Node's current
care-of-address at any time.

The protocol is fairly simple. HARP uses three kinds of messages:
1. HARP registration forwarding. A Mobile Node registration
at either HARP co-HA is internally forwarded as a HARP message
from one co-HA to the other. As a result, they setup tunnels
as necessary in parallel so that any received datagrams for the MN

will be forwarded from either co-HA. This mechanism uses the
HARP UDP port 1588 (look for HARP-PORT in mipconfig.h).

2. a boot time, HARP uses a TCP connection to exchange any
internal MIP registration info with the other co-HA. This is
an optimization and servers to speed up registration if one
co-HA reboots due a local failure.

3. a HARP internal "ping" is used. Periodically one HARP
co-HA will use the UDP HARP channel to determine if the other
co-HA exists. Actions taken on failure may include:
1. a page of a local network admin OR
2. bringing up a local interface to act as a HOME
for MNs if desired.

HARP is viewed as an Interior Gateway Protocol that ideally
acts in conjunction with another IGP like OSPF or RIPv2. It
may be used as static routes. Topologically we view co-HAs
architected as follows:

```
Internet
|
OSPF router1 -------------  OSPF router2
  | exterior link          | exterior link
HA1 HA2
  | mobile-IP subnet .......     |
```

co-HAs must have two (possibly virtual) interfaces. E.g.,
one could use an Ethernet and wireless interface. Or one
might use a single ethernet i/f with two ip addresses.
We assume that the co-HAs somehow share ONE Mobile-IP subnet
address, but that they also MUST have a separate unique IP
address. Calls these two ip addresses:

1. the exterior (unique) IP address
2. the MIP (shared) IP address

From the IGP (exterior) point of view, the co-HAs can function
as routers. They simply somehow inject the existance of the MIP
subnet into the local IGP routing domain. In our implementation,
we currently lack a dynamic routing daemon on FreeBSD (although
we have some experimental routed code included that is insufficiently
tested, but does show signs of co-existance with mipd). We recommend
that the "real" router/s simply use static routes for now. If
one co-HA fails, the static route path can be changed. Ideally
however the co-HAs would be part of an OSPF system and Mobile-IP
traffic could be load-balanced as appropriate through the routing
infrastructure.

From the Mobile-IP subnet point of view, one has a number of
topological possibilities. For example, one must act
whether the interior Mobile-IP subnet (home) is partitioned
or not. We recommend a partitioned subnet (ideal for wireless),
but the system should work with a normal non-partitioned link
(e.g., with ethernet). Keep in mind that the Mobile-IP subnet IP
at the Home Agents (for the HAs themselves) must be the same.

As a result, Mobile-IP does not change as a protocol and the MNs
do not need to know about "two" HAs.

The internal Mobile-IP link topology is important. Ironically as HARP is an IGP, we *could* claim that how we deal with that is not important (it has little to do with HARP as a routing protocol). However it is important in practical terms. Thus we support two possible internal Mobile-IP link topologies.

1. partitioned (the HAs can NOT reach each other on the Mobile-IP link)
2. non-partitioned (the HAs can reach each other on the Mobile-IP link)

At PSU we have two co-HAs deployed. Each has a wireless and ethernet interface. The Mobile-IP subnet is bound to the wireless interface. The co-HAs can NOT hear each other on the wireless interfaces. Other topologies are possible (e.g., two ethernet i/fs per HA or one ethernet i/f with two ip addresses).

Setup:

This implementation of HARP only allows for the deployment of two Home Agents. ie. A co-HA pair.

HA locations should be chosen so as to be as far separated as possible, both geographically and in terms of network topology. This is in order to avoid the loss of both Home Agents due to some localized disturbance like a power loss or a local router outage. This may not be possible, but it is good if it can be done.

At this point in time, we have experimented with a modified BSD routed on freebsd to make it coexist with mipd. This routing daemon would allow the FreeBSD systems to be part of a dynamic (but RIPv2) IGP infrastructure. Thus the infrastructure itself could determine if the interior routing path to the Mobile-IP subnet had gone down.

The experimental version of routed can be found in the usr.sbin/routed directory. We verified that it could work in conjunction with mipd using RIPv2. We do not feel that routed has been tested enough to be seriously deployed however and do not have the resources at the moment to do more testing.

Those wanting to actually use HARP can resort to static routes in local routers (e.g., Ciscos). We have tested changing static routes in neighborhood routers to make sure that HARP works; i.e., we take down one co-HA host to simulate loss of a Home Agent, and then change the Mobile-IP subnet route in a nearby Cisco by hand to point out the other HARP system to make sure HARP works. Although this is not the best model in terms of dynamic routing, it does provide proof of concept.

Re HARP configuration, first of all note that (almost) exactly the same mipd.conf file must appear on both HAs; i.e., /etc/mipd.conf. In general, key lines (e.g., for mip/mns) will most likely be the same. Of course other lines like coas/co-ha ip will be different.

There are three HARP config lines that can appear in /etc/mipd.conf

1. coha ... REQUIRED
2. harp_tunnel ... (optional)
3. harp2_arp_disable ... (optional)

(In addition there is an extra possible HARP protocol authentication line, which we will briefly discuss below.)

1. coha ip_address

Designates the IP address of the OTHER home agent in a co-HA pair. This line must be present for HARP to function. It must be present in both co-HAs and the IP address must be the "interior" (non-Mobile-IP) address that is unique per co-HA.

2. harp_tunnel < yes | no >

This configuration line applies to one of the two possible Mobile-IP link topologies. By default "no" is used (i.e., it is off). If a partitioned Mobile-IP link is desired, and you want Mobile Nodes to be able to be present at both Homes, set this value to "yes". As a result, the co-HA receiving packets for a MN present at the other co-HA will tunnels those packets to the other co-HA.

3. harp2_arp_disable <if_name>

This line should be used only when one desires to use a non-partitioned Mobile-IP home link. If used, harp_tunnel should be set to no (the default). It should only be set at one co-HA (not both). It causes that co-HA to dynamically enable ARP on the interface in question if and only if by using the HARP ping, that co-HA determines that the other co-HA is down. Hence confusion due to ARP and shared ip addresses will not occur.

HARP protocol authentication:

In terms of the implementation there are two possible mechanisms for HA to HA HARP protocol authentication. IPSEC AH at the application level may be used and in fact we assume that is available. mipd will simply try and use it with the current codebase. This means that an IPSEC/AH association must exist between the two Home Agent systems.

An optional mechanism that is like the Mobile-IP md5 authentication mechanism may be compiled into the code (this is not discussed in the HARP IETF draft). In order to use this, mipd must be recompiled. This form of authentication is compiled in with #ifdef MIPSEC for HARP. if IPSEC is not defined, then MIPSEC is compiled in by default and the haha_key item is used.

haha_key xxx.xxx.xxx.xxx md5 SPI Key

Again, haha_key must be compiled in and is only used as a stopgap so that some form of authentication always exists.

```
Additional documentation on HARP can be found in the ./papers
directory.  There is a student term paper written by Bjorn
and an Internet draft authored by Bjorn and Jim Binkley.

Final configuration note:

In our current implementation for HARP in /etc/mipd.conf we actually use:

co_ha <other ip address>
harp_tunnel yes

This is because our agents are configured with two real physical
interfaces (ethernet and wireless).  Thus it is possible for
our Mobile Nodes to be home in two places as well as take advantage
of the redundancy provided by HARP.

------------------------------------------------------------------
In Conclusion:

Again see the README for the main installation details.  This
release only adds functionality (and fixes a bug or two).

As a reminder, see our web page: http://www.cs.pdx.edu/research/SMN
for DARPA reports, which document the protocol and architectural aspects
of a lot of this work.

In general project documentation consists of:

1. README* in this release
2. papers/ ... various papers generated at various
times during the project that have bearing on
particular parts of the project.  Some of them
are simply student reports and others are rough
draft of either IETF documents or academic papers.
3. man/ ... man pages.  Note that ps versions are
                supplied as well.
4. http://www.cs.pdx.edu/research/SMN
```

M.4 Security Considerations for Mobility and Firewalls

Internet Engineering Task Force   Jim Binkley
INTERNET DRAFT   Portland State University
Category: Informational   John Richardson
                                                               Intel

Security Considerations for Mobility and Firewalls
<draft-binkrich-mobisec-00.txt>

Status of This Memo

Distribution of this memo is unlimited.

This document is an Internet-Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months, and may be updated, replaced, or made obsolete by other documents at any time. It is not appropriate to use Internet Drafts as reference material, or to cite them other than as a ''working draft'' or ''work in progress.''

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nic.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast) or ftp.isi.edu (US West Coast).

Distribution of this memo is unlimited.

Abstract

In this paper we discuss various security issues concerning Mobile Hosts using Mobile-IP or other mobility systems (DHCP standalone) and current firewall technology. We first present some recent attacks on the Internet and what they might mean for mobile systems like Mobile-IP that rely on tunneling technologies. We point out that tunnels are a security threat and suggest how mobile systems may be made "less insecure" with the use of IP layer security (IPSEC) as one means for creating Virtual Private Networks. The goal is to describe a security model wherein mobile systems can work across the Internet and not just as an interior routing protocol within one security and/or interior routing domain. Both the protection of Mobile Systems abroad and of Security Enclaves that tolerate mobile visitors must be considered.

Table of Contents

1. Introduction and Assumptions

With the rapid growth of Virtual Private Networks, tunneling protocols are assuming a high profile in the Internet. Our work with tunnels as applied to Mobile-IP has uncovered a vulnerability that most tunnels leave unprotected. Basically, while most of today's firewalls stop IP Spoofing attacks, tunnels "drilled" through those firewalls re-enable that class of attack.

Strong authentication of the remote systems by the tunnel endpoint, while necessary, is not sufficient to maintain the protection provided by the firewall complex.

More generally, if a tunnel server allows authenticated remote systems to become part of a "secure enclave", it must also provide the basic protection that the firewall provides for native hosts in that enclave.

The problem becomes more interesting if the secure enclave wishes to host "visiting" systems locally. For example, a company might wish to provide Internet connectivity in conference rooms and allow visitors to access the Internet (and not the secure enclave).

We will consider these problems below in more detail.

1.1 Assumptions

Networking, especially when done securely, has been developed from many different perspectives. Each community starts from a presumed base of common language and "normal" assumptions. To minimize confusion, we begin by stating our assumptions and provide a brief description of how commonly used terms are used in this document. We do not mean to imply anything about how they "should be used", just how we chose to use them here.

The focus of this document is on how a secure enclave (firewall protected area) may tolerate Mobile-IP [RFC-2002] or simpler mobility systems (for example, DHCP used standalone) and remain secure. By "secure enclave" we mean a conventional IP site with one management domain and a centralized security administration typically behind one IP firewall [Chapman]. By "firewall" we refer to one or more systems acting together to provide protection for a network. In particular, we assume that one (or more) endpoints of IP tunnels are part of the firewall complex.

Our focus here is on how a secure enclave can protect itself from foreign (non-local) Mobile Nodes. We also deal with IP spoofing issues and possible security problems that might occur due to naive implementations of IP tunneling [RFC-2003] when combined with such spoofing.

The discussion is focused on the network layer. We are not considering higher-level authentication or confidentiality services that might be part of an application-level system. When we talk about firewalls, we are mostly talking about network layer access, and such mechanisms as packet-level firewalls with access control, Virtual Private Networks implemented as IP tunnels, and IP layer security (IPSEC) [RFC-1825]. We do not mean to discourage application or transport layer security in any way (Please see [RFC-2316] for the latest IAB discussion on network security in general). It is simply not our focus in this document.

Regarding firewalls, we assume Cisco access lists as a rough lingua franca for access control on routers and will use access list examples suitable for Cisco routers. Please see [Ballew] for discussion of Cisco access list mechanisms. We assume packet filter technology simply because accidental holes may indeed be poked through such a router if its manager is not careful.

When we cite IP addresses as examples in the text, we will use private IP addresses as mentioned in [RFC 1918]. These should be viewed as surrogates for public ("real") IP addresses associated with an interior routing domain. We use these addresses because we do not want to cite "real" addresses in any examples.

2. The Problem Space

Firewalls are designed to separate "inside" from "outside". A naive approach to protection would use the source IP address to make the distinction. Unfortunately, IP header information is unreliable as it can be set by the source (or any intermediary) to any arbitrary value. The attacker community knows this well and it forms the foundation for an entire class of attacks known as "Spoofing".

Spoofing has been used as the basis for a whole set of recent attacks (for example, see [RFC-2267]). Some of the attacks are denial of service oriented. Some seek to cause the attacked system to crash or hang. Many of the attacks can be characterized as single packets wherein the IP source and destination addresses in the IP header appear to originate within the attacked systems site.

2.1 Spoofing Attack Examples

To highlight the importance of spoofing attacks, we will briefly discuss three such attacks, TCP SYN [CA-96.21], "smurf" [CA-97.28], and "land" [CA-97.28].

In TCP SYN attacks, the attacker sends TCP initialization packets to a given site. The attacked system is tied up simply due to opening too many TCP control blocks which cause allocation of precious kernel memory. The attacker need only send one TCP SYN packet. The attacker may choose to use a spoofed IP source address so that tracing the attack back to its originating system is difficult.

In "smurf" attacks, the attacker sends one or many ping packets to an IP directed-broadcast address with an IP source address that may also be at the destination site. For example, if a site had a site specific class C address along the lines of 192.168.1.0, the attacking IP destination might be 192.168.1.255 or 192.168.1.0 (0 broadcast addresses may be used as well) with an IP source of 192.168.1.1. The result is that two systems (at least) may be attacked. The IP source itself is bombarded with ping reflections from all the systems at the directed broadcast address. Further the smurf vehicle could also be used for single packet "ping of death" attacks.

"Land" attacks involve one TCP SYN packet in which the IP source is set to be the same as the IP destination. The attack may cause the receiving machine to hang.

In general, note these attacks do not involve packets being returned, unless the packets are returned to another system that is being indirectly attacked itself ("smurf").

2.2 Prevention of Spoofing Attacks

One general technique that can be used is to disallow IP spoofing for internal source addresses [RFC-2267]. Filters can be put in place so that packets arriving on an "outside" interface must have an "outside" source address (or must NOT have an "inside" source address). One may also filter out spoofing attacks attempting to leave from the "inside" of a network.

We will briefly look at how spoofing may be prevented with a Cisco router which we assume is the interface between a site having 172.16.*.* as its internal IP address space and the Internet at large. Note that the access list entries shown here may be part of a more complex firewall policy and/or access list, but we only show the part relevant to IP spoofing.

The following access list entry may be bound to an external router interface and would be applied to packets entering the site.

```
access-list 101 ...

access-list 101 172.16.0.0 0.0.255.255 any
```

Packets entering the site with 172.16.*.* addresses will be discarded.

We apply the following access list to packets headed out on the external interface:

```
access-list 111 ...
access-list permit ip 172.16.0.0 0.0.255.255. any
access-list deny ip any any log
```

This blocks packets headed out that do not have IP src addresses in the 172.16.*.* range and logs any internal attempts at spoofing. Thus spoofing attacks cannot originate at this site. The result is that a site firewall or border router will neither send or receive packets over an interface when the packets do not belong to the source IP routing domain.

2.3 Anti-Spoofing Measures and Mobile-IP

The result of such anti-spoofing measures is that packets headed into the enclave to "foreign" Mobile Nodes; i.e., Mobile-Nodes from some other site than 172.16..*.* will not be permitted to enter. Packets trying to leave the site from systems from another site, say 192.168.1.0, will not be permitted to leave. Mobile-IP within the same routing domain, which we might call "interior Mobile-IP" would be permitted. Mobile-IP ("exterior Mobile-IP") between two interior routing domains would not be permitted.

Now we must consider what happens to packets sent from a "visiting" foreign Mobile Node that is somehow operating within the secure enclave. Please refer to the picture below.

First the UDP-based Mobile-IP registration protocol would still work if Foreign Agents were used as Foreign Agents act as UDP proxies for Mobile-IP registration; i.e., they will replace a Mobile Nodes IP source address with their own (legal) source address. A Mobile Node using DHCP as a source of local addresses could also succeed if it used the DHCP-obtained local address.

Data packets sent directly out of the domain from the visiting Mobile Node (unless tunneled via a local IP source address) would be discarded at the border. Data packets that somehow escaped the local secure enclave's border router could also be discarded by the "home" border router's spoofing filter as well, as it would not permit packets to enter that have "local" IP source addresses. Data packets tunneled from the (exterior) Home Agent to the (interior) Foreign Agent would be allowed through because the external encapsulation would get past the spoofing filter; i.e., Home Agent to Foreign Agent IPIP packets would have the legal interior IP source for the Foreign Agent as the exterior IP source address.


```
Home Agent    <-------> Border Router (for 192.168.1.X domain)
(192.168.1.X)
      ^
                      |
                      |  the Internet
      |
      v
Border Router (secure enclave)
 ^
                 |
                 |  secure enclave (172.16.*.*)
 V
Foreign Agent (172.16.*.* address for Care Of Address)
 ^
 |
 V
                Visiting Mobile Node  (192.168.1.X)
```

In addition to the obvious problems that anti-spoofing raises for Mobile-IP, one must also ask if tunnels raise additional security concerns and how one might address both those concerns and security for both the Mobile Node itself, its home domain, and the "visited" domain too.

3. Tunnels Considered Harmful

One mechanism that is part of Mobile-IP and in point of fact many other routing protocols are IP tunnels which might be implemented with IPIP, IPSEC Tunnel Mode, or Cisco's Generic Routing Encapsulation [RFC-1701]. It should be pointed out that IPIP tunnels are not peculiar to Mobile-IP. They are used in many routing protocols for many purposes including tunneling non-IETF protocols (e.g., Appletalk) or building virtual networks on top of the current Internet (MBONE, 6BONE).

Tunnels may mean encapsulated packets where one has one IP datagram inside another IP datagram and we will use IPIP (IP protocol 4) as our example here.

Mobile-IP uses tunnel mechanisms like IPIP to forward packets from the Home Agent to a remote "Care Of Address". The COA is a local site IP address that may represent a Mobile Node that has acquired a local IP address itself directly via DHCP or a router system that understands Mobile-IP called a Foreign Agent.

Any Mobile-IP system, including Mobile Nodes, Home Agents, or Foreign Agents, may source or sink tunnel packets. When a Home Agent forwards packets to a Mobile Node that is at a Foreign Agent, the use of IPIP in a datagram may appear as follows:

```
IP outer header         IP inner                     IP datagram
--------                --------
ip src= Home Agent     ip src = peer end host     |  TCP, etc.
ip dst= Foreign Agent   ip dst = Mobile Node       |

|
| packets to MN
v
Home Agent ========= IPIP tunnel to COA ===>   FA and Mobile Node
```

One might ask if it is enough to simply use IPIP tunneling and somehow tunnel either from the Foreign Agent or Mobile Node back to the Home Agent and thus evade the anti-spoofing measures at a firewall? Unfortunately, this is an insecure approach.

In the first place, it is not enough to simply tunnel over the IP spoofing firewall. This is simply a new form of spoofing which we might call: "IPIP spoofing". The problem is that if one has a tunnel sink (be it any kind of agent or Mobile Node) that decapsulates packets and then forwards them, others can launch their spoofing attacks with IPIP too and thus have the spoofing emerge "inside" the enclave firewall. For example, smurfing might simply be redirected through a tunnel. The inner IP header might be directed broadcast with an interior IP source named as a target. All the previous attacks (TCP SYN, "smurf", "land") can thus be done through the firewall.

We suggest that one can block tunnels with current access list mechanisms and thus control tunnels so that tunneling from the outside can only be done to certain hosts that will be considered as "network-layer" bastion hosts.

For example, with Cisco IOS one can add the following statements to access list 101:

```
access-list 101 deny ipinip any any
access-list 101 deny gre any any
```

and thus block any IPIP or GRE packets coming in over the router. One may further add a permit statement to force any IPIP packets coming in to land at a certain host and then treat that host as a bastion host; i.e., a nexus of security focus.

For example,

```
...
access-list 101 permit ipinip any host 172.16.1.3
access-list 101 deny ipinip any any
access-list 101 deny gre any any
...
```

in a Cisco input access list, means IPIP will only be allowed to the host 172.16.1.3, which we will assume is a tunnel sink agent.

We have focused internal trust for IPIP on that one system. We next need to explore how to make sure that packets arriving at the tunnel sink agent are *not* attacks that can be made via a tunnel sink. This can be done with "tunnel-mode" IPSEC tied to IPIP. We will explore this idea further in the next section.

4. Problem Solution Space

We will discuss how to solve these problems from two topological points of view. First we look at the situation from the Mobile Node abroad's point of view. We assume it actually wants to get packets home and not compromise home security. Thus this point of view must necessarily include the Mobile Node's home enclave. We then look at the situation from the "foreign" security enclave's point of view. We assume that the foreign enclave wants to allow mobile service but protect itself. We also must consider the question of how both security enclaves (home and away) in general protect themselves from any tunnel-based attacks.

In this discussion we also try to contrast the use of Mobile-IP versus a simpler form of DHCP-only mobility that does not use Mobile-IP. Keep in mind that the main semantic for the use of Mobile-IP is that the Mobile Node retains at least one fixed IP address that is non-local for the subnet it is visiting.

A Mobile-IP system may have two IP addresses associated with it, a fixed permanent Mobile-IP address (call it the "Mobile-IP address"), and a locally acquired address (call it the "DHCP address"). A DHCP-only system would only have one locally acquired IP address.

4.1 Mobile Nodes Abroad Point of View

In this section we will consider the problems for a secure enclave if Mobile Nodes in that secure enclave go abroad; i.e., out to the Internet beyond the firewall. We must ask how the Mobile Node can secure its own traffic and in effect, take its security enclave with it. We must also ask how the home enclave can secure traffic coming from that Mobile Node back inside, which will extend the thinking about tunnels in the previous section.

Glass and Gupta [Gupta-draft] suggest that Mobile-IP Mobile Nodes abroad may use DHCP to acquire "local" IP addresses, Thus they can get by the anti-spoofing measures in the firewall router. This is indeed a reasonable possibility. Further, the Mobile Nodes can use IPSEC with two-way tunnels between the Home Agent as a classic bastion host and the Mobile Node. (See [http://www.cs.pdx.edu/research/SMN for a combined Mobile-IP IPSEC system in which Mobile Nodes can do two-way MN/HA ESP tunnels).

Please see the figure below:

```
                              Internet           172.16.1.1
Home Agent: <----> Firewall <-----------------> router/DHCP server
192.168.1.1                                     |
                                                    |
                                                    |
                                                 Mobile Node
   172.16.1.2/
                                                 192.168.1.2
```

4.1.1 Packets from the Mobile Node Out

If we assume Mobile-IP and two addresses in use by the Mobile Node, packets tunneled from the Mobile Node to the Home Agent might have the structure:

IP(1) | IP(2) | <IPSEC> | IP(3)

Each IP header has its own purpose. The most external header, IP(1) exists to get the packets to the Home Agent with the DHCP acquired address == 172.16.1.2. The IP destination would be 192.168.1.1. Thus header(1) allows transit of the Internet and any anti-spoofing firewalls. When the packet arrives at the Home Agent, that header is discarded and header(2), consisting of IP(2) combined with an IPSEC header is processed. Here we assume that the Mobile-IP address 192.168.1.2 is used for the source address and the Home Agent is again the destination.

The fixed Mobile-IP address may be needed here as it allows a-priori manual IPSEC keys to exist between the Mobile Node and the Home Agent. In effect, this is an IPSEC tunnel between the Mobile Node and the Home Agent. The interior header would contain the Mobile Nodes fixed address (192.168.1.2) as IP source and the address of any destination to which it is allowed to send packets.

The above triple-header system could be optimized by a higher-level protocol that could produce a dynamic binding between the local DHCP-acquired COA and the Home Agent's destination address.
There is no reason Internet Key Exchange protocols [IKE] where non-IP naming schemes are used could not be deployed here. This would allow one header to be deleted.

For a DHCP-only form of mobility, the packet layout situation would be simpler. The Mobile Node would use a non-IP naming scheme with IKE to form a security association with a Home Security Agent.
IP header (2) would not be needed.

4.1.2 Packets Coming From Home to the Mobile Node

For Mobile-IP, we must now consider packets coming back to the Mobile Node via a tunnel from the Home Agent. By definition these packets are tunneled to a local IP address and are not subject to problems caused by anti-spoofing filtering. However IPIP unadorned is a security threat to the receiving enclave. And of course, the Mobile Node may choose to have IPSEC-based security between itself and its home enclave.

One possible encapsulation scheme might take this form:

IP(1) | IP(2) | IPSEC | IP datagram

IP(1) exists to get the packets from the Home Agent to the remote Care Of Address which might be a Foreign Agent or a Mobile Node that has acquired a local IP address. The inner IP header would exist where manual keying is needed with IPSEC and the IP source would be the Home Agent. The IP destination would be the Mobile Node itself. Note that again IKE could be used to optimize out an IP header as long as IP addresses are not part of a manual configuration scheme.

It is highly likely that from a security policy point of view, one would not form security associations (especially confidentiality-based security associations) between random Home Agents and random enterprise-external Foreign Agents. As a policy consideration, unsecured IPIP might simply not be allowed to Foreign Agents. Foreign Agents might insist that all IPIP packets be sent to them from internal Home Agents with which they share an a priori security association. Alternatively Foreign Agents might exist "outside" a secure enclave, or unadorned IPIP packets when decapsulated might only be allowed to go "outside".

4.1.3 Tunnel Security at Tunnel-Exit Agents

We suggest that a tunnel-sink agent like a Mobile-IP Home Agent may want to guarantee that all packets sent to it via a tunnel are cryptographically verified; e.g., shared secret keys might exist between it and the Mobile Node abroad. No packets forwarded to the tunnel-sink agent by the firewall will be internally decapsulated and forwarded until they have been cryptographically verified. This might be done with an access list mechanism tied to IPSEC or by simpler means. For example, the PSU system mentioned above has a BSD sysctl(8) switch:

```
# sysctl -w net.inet.ip.mvifipsecinput=1
```

that if set forces the IPIP driver to only forward packets if and only if IPSEC authentication or decryption has successfully occured between the remote system and this system. As a consequence, one may be sure that a Mobile-IP Home or Foreign Agent or any tunnel sink only forwards IPIP packets that have successfully passed IPSEC processing. Put another way, a security association must exist between the tunnel sink and the tunnel source system.

Packets coming from remote security-aware Mobile Nodes might have several forms:

IP(1) | IPSEC | IP datagram

or possibly

IP(1) | IP(2) | IPSEC | IP datagram

For example, the former packet architecture might occur with a remote Mobile Node that is only using DHCP and wants to securely tunnel home. The latter might be used by a remote Mobile Node that is using Mobile-IP and has also used DHCP to acquire a local COA.

The local anti-IP-spoofing firewall might then be configured in a number of possible manners depending on local security policies and the structure of external but acceptable packets. For example, with current Cisco access list technology, we could permit IP | IPSEC packets using ESP (ip proto 50) or AH (51) to the Home Agent:

```
...
access-list 101 permit 50 any host 172.16.1.3
access-list 101 permit 51 any host 172.16.1.3
access-list 101 deny ipinip any any
...
```

As in our previous example, the firewall might simply allow IPIP but only to a Home Agent. This would apply to the second IP | IP | IPSEC example.

We must point out that the security problems here are not terribly different from those encountered by current dialup clients into a secure enclave that access the enclave via an internal terminal multiplexor. The exterior host tunnels into a secure enclave and an agent in the secure enclave applies cryptographic measures to packets that have come in from the outside.

4.2 Hosting Visitors From Abroad

In the previous section we have focused on how to protect the Mobile Node abroad and also discussed the problem of how to make tunnel (exits) more secure. One must also worry about the security of the "other" enclave, else enclaves may not desire to host foreign Mobile Nodes. It makes little sense for a firewall-protected enclave to allow visitors to penetrate the enclave at will and thus enable possible attacks on internal systems by visitors.

Of course, we could start with a security policy that does not allow visitors to penetrate the firewall. In effect, that is the current security policy for many sites. However it is our goal here to discuss how we might tolerate "less trusted" visitors, not define them out of existance.

We suggest a topological approach based on network design measures that can be made with current (or near-current) technology and that should allow a secure enclave to remain secure.

Our basic principle is: "design the network so that visitor packets are not allowed inside". We observe that whatever is done to implement

this goal will probably be similar to current systems that have
two-level security enclaves.

For example, one might have a network designed as follows:

```
                           ^
                           |
Router(1)
   |
------------------------- internal less secure DMZ
        |          |             |
      Firewall(2) term mux DHCP/router for Mobile Visitors
|
     ------------------------
     | secure enclave       |
```


The above may be viewed as a logical (not necessarily physical) structure. We have a Router(1) that simply serves to allow access to the Internet. Inside the external router we have a less secure DMZ network that may serve to allow unfiltered access to the Internet. This network might include terminal multiplexors and local mobility servers. Behind it we could have at least one level of firewalls with bastion hosts (which may or may not be on the DMZ network). Firewall(2) would serve the function of the traditional firewall machine. One might observe that the above scheme does not force visiting laptops to acquire local addresses to bypass IP-spoofing filters. True, but unfortunately there may be other firewalls on the way home that possess such filters. Certainly the firewall at home may possess such a filter.

The above may be viewed as a physical network design that would tolerate visitors. However it is very likely that such a design may not be very convenient to implement. We suggest that various virtual network techniques may be used to approximate the physical structure above. Let us briefly consider two methods that may be used to logically separate networks and thus remove construction difficulties and make the deployment of mobile networks easier. Note that these techniques *require* careful network security design. The security onus here is certainly on both the network designer, and on the creators of both Mobile Agents and security gateways. Caution needs to be employed in the design of such systems and functionality must be clearly communicated by implementors to potential network designers.

First one may use a combination of simple tunneling sans IPSEC and/or authenticated tunnel packets (e.g., IPSEC AH) between a mobile router (MIP foreign agent or DHCP router/server) and the "outside" network. The basic idea would be that a router (e.g., a Mobile-IP Foreign Agent) could take all packets presented to an "exterior" interface and tunnel them (using IPIP or GRE) (possibly with additional IPSEC to alleviate paranoia) to a firewall-exterior interface on a border router. As a result, visitor packets would have no opportunity to access interior hosts. They would be tunneled "outside" and would be treated as external packets coming back through any existing firewall mechanism.

```
  | --------------------------------- Internet accessible network
 Firewall(1)           exit tunnel router (2)
  |                             ^
  |                             |
  v                             |  tunnel to outside
secure enclave                  |
 v
DHCP server router or MIP agent (3)
                                ^
                                | external visitor link (4)
 v
                            visiting mobile node (5)
```

Note that we assume here that the agent(3) and the exit tunnel router(2) are under the control of the same network administration.
We suggest that careful combination of access lists with tunnel technology should allow the above picture to be collapsed in various ways. For example the Firewall(1) and exit router(2) systems could be the same system.

In addition, the router(3) that enables mobility could potentially optimize packet delivery. If IPSEC security associations existed at that router between a Mobile Node and the router itself, it might choose to NOT forward IPSEC-verified packets that show up via the external visitor link(4) over the internal tunnel. Thus IPSEC packets from "local" mobile systems that belong to the enclave itself could be allowed direct access to the local enclave. Of course, packets that lacked a security association with the mobile agent router would be forced over the tunnel to the "outside" world.

We will not go into details, but link-layer switching technologies can also be of use here. For example, Virtual Lans [3com] when assumed to be 1-1 with IP subnets could be used as a way to funnel visitor packets back to a router that might apply access list technology to packets trying to cross from an "exterior" subnet to an "interior" subnet.

5. Miscellaneous Considerations

5.1 Firewall Discovery

Although we cannot attribute such discussion, some have suggested that some sort of Firewall Discovery system might allow Mobile Nodes to dynamically tunnel to and from firewalls. There are several problems with this notion:

1. It is unnecessary since our solution here will work with current or near-current firewall technology.

2. It is not very likely from a security point of view. Security people and network managers may not care for notions that involve poking holes dynamically through firewalls. Complexities involved in cross-security domain certification are likely beyond near-term

scope. Further the security folks "at-home" may not care for schemes that involve key exchange with strangers; i.e., a Mobile Node from home somehow secures packets between itself and a foreign firewall at a different enterprise. After all, that firewall might choose to store all data traffic, and enable a classic "man-in-the-middle" attack.

3. Traditional notions of IP fate sharing (considered bad) may apply here. Mobile-IP systems are already tied to the fate of their Home Agent. Additional ties between systems that are not related from interal routing or security enclave considerations may be complex. After all, it is hard to predict how many firewalls that rule out IP spoofing to/from a given site may exist.

Schemes that allow trusted locals to poke holes through firewalls are perilous by definition since "untrusted" people may crack the scheme. It is unlikely that dynamic mechanisms that allow random visitors access will prove widely acceptable.

5.2 The Role of a Mobile-IP Foreign Agent

In the previous discussion, we suggested that DHCP can be used to simply allow Mobile Nodes abroad to obtain a local address. Using that address they can then send packets wherever they choose. As a result, it might seem that there is little role for a Mobile-IP Foreign Agent in a security system. Ultimately the roles that mobility systems play depend upon policy considerations. One could suggest a policy wherein Mobile Nodes abroad are not allowed to speak directly to (as opposed to through) or exchange cryptographic material with "foreign agents". This is certainly a reasonable policy. However the focus of such a policy is on the Mobile Node. We need to also consider Foreign Agent oriented policy and how a Foreign Agent might serve as a border router for a secure enclave.

Foreign Agents may serve as routers that simply do not allow foreign visitors any access to an internal enclave and only allow authorized local Mobile Nodes entrance. Many techniques exist for such screening including the pre-existing Mobile-IP manually keyed registration that can secure Mobile Node access via a given Foreign Agent. However, security techniques should apply to all packets and not just Mobile-IP registration packets.

As another possibility (and there are probably others), IPSEC-aware Foreign Agents can discriminate between locals (who hold security credentials) and non-locals (who lack security credentials). Once a Mobile Node has identified itself to a Foreign Agent as belonging to the agent's secure enclave, it could use an IPSEC security tunnel between itself and the Foreign Agent. Any packets verified by the Foreign Agent as belonging to the local secure enclave could thus be delivered locally and not pushed out of the routing domain via a tunnel. Non-local visitor packets might be unceremoniously escorted off the premises via another kind of tunnel and would have no access to internal resources. Thus local mobile systems and visitors could both be tolerated at the same agent link.

Of course a paranoid enclave might choose for policy reasons to force all wireless visitors to be "foreign". "Locals" could be always treated as remote visitors and tunneled outside, thus having to use secure means to come back inside. Or foreigners might simply not be permitted entrance at a given agent. Both policy considerations are possible and should be considered in implementations.

6. Security Considerations

The entire document is about security, mobility, and dangers inherent in IP tunnels. We focus specifically on issues arising out of the interaction between firewalls and any tunneled protocol and highlight security concerns regarding Mobile-IP or simple DHCP for foreign visitors "beyond" the home firewall.

We should point out at least one more specific security consideration for tunnel entrances. If IPSEC is used in "tunnel-mode" at a router or forwarding system that is neither the IP source or IP destination, it is possible that the security system may be subject to "proposed plaintext attacks". Please refer to the following figure:

```
Home Agent (1) (or tunnel entrance)
|                        ^
|                        |    plaintext packets
|  attacking system (3) ---
|
| cryptotext packets to Mobile Node
               V
remote Mobile Node (2) (tunnel exit)
```

If a attacking system(3) can present plaintext packets to (1), and then read them back after encryption in the tunnel between (1) and (2), the potential for proposed plaintext attacks exists. This liability exists for a number of proposed combined tunnel and security systems, as long as network-layer forwarding combined with IPSEC (or cryptography) is part of the architecture. Solutions for the problem include session keys [IKE] and possible restriction of communication between the Home Agent(1) and the Mobile Node(2) to exclude IP sources that do not lie within the home enclave. By definition, this problem is found only with network-layer forwarding (i.e., at IPSEC gateways), and is not present in any end-system to end-system IPSEC.

7. Conclusions

In this document we have presented proposals that will enable Mobile Nodes from abroad or nearby to less insecurely access the Internet. Such systems are not dissimilar from current dialup systems that involve a remote PPP-based dialup client and a local terminal multiplexor. IPSEC-enabled tunnel mechanisms may be used between the Mobile Node system and its home security companion. Very simply put, the Mobile Node is an extension of the local security domain. However, in addition to securing the Mobile Node and its home enclave, one must also give thought both to the dangers of tunnels and to how a local enclave may enable its own security and still tolerate visitors.

In summary, we will make the following suggestions:

1. DHCP to acquire a local COA solves problems caused by IP spoofing prevention for visiting Mobile Nodes abroad and may or may not be combined with Mobile-IP.

2. Suitable two-way cryptographic tunnels between a system abroad and a routing system at home will allow a Mobile Node's own traffic to be securely tunneled over the Internet.

3. IPIP tunnels sans cryptographic safeguards should be viewed with caution. If an IPIP tunnel sink does not guarantee cryptographically controlled access, an attacker may tunnel various one-way attacks (land, etc.) into an enclave. The tunnel sink may be logically regarded as an extension of the firewall itself. It may be co-located. If it is *not* co-located, firewall filtering mechanisms may need to be duplicated at the tunnel-exit point.

4. Flexibility in routing, access list mechanisms, and encapsulation possibly with authentication should be considered by implementors so that a secure enclave can securely escort visitor packets off-site without threat to secured systems within the site.

5. Security considerations must apply both to Mobile Nodes abroad, their own home enclave itself, and also to how enclaves may be designed to tolerate visitors.

8. Acknowledgements

We would like to thank David Reeder of Trusted Information Systems and Mark Morrissey of Oregon Graduate Institute for their comments.

9. References

[Ballew] Ballew, Scott, "Managing IP Networks", O'Reilly and Associates, Inc., 1997; ISBN 1-56592-320-0

[Chapman] Chapman, D.B., and Zwicky, E.D., "Building Internet Firewalls", O'Reilly and Associates, Inc., 1995; ISBN 1-56592-124-0

[RFC-1701] Hanks, S., Li, T., Farinacci, D., Traina, P., "Generic Routing Encapsulation", October 1994.

[RFC-1825] Atkinson, R., "Security Architecture for the Internet Protocol", August 1995.

[RFC-1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", February 1996.

[RFC-2002] Perkins, C., "IP Mobility Support", October 1996.

[RFC-2003] Perkins, C., "IP Encapsulation within IP", October 1996.

[RFC-2267] Ferguson, P., and Senie, D., "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", January 1998.

[RFC-2316] Bellovin, Steve, "Report of the IAB Security Architecture Workshop", April 1998.

[IKE drafts] Internet Key Exchange (ISAKMP/Oakley). Works in progress.

[CA-96.21] CERT Advisory CA-96.21; TCP SYN Flooding and IP Spoofing Attacks; September 24, 1996.
http://www.cert.org/advisories/CA-96.21.tcp_syn_flooding.html

[CA-97.28] CERT Advisory CA-97.28; "Teardrop/Land" IP Denial-of-Service Attacks; December 16, 1997.
http://www.cert.org/advisories/CA-97.28.Teardrop_Land.html

[CA-98.01] CERT Advisory CA-98.01; "smurf" IP Denial-of-Service Attacks; January 5, 1998. http://www.cert.org/advisories/CA-98.01.smurf.html

[3com] http://www.3com.com/nsc/200374.html;
Passmore, David, and Freeman, John. "The Virtual Lan Technology". 3com Inc.

[Gupta-draft]
Gupta, V., Glass, S., "Firewall Traversal for Mobile IP: Guidelines for Firewalls and Mobile Ip entities", draft-ietf-mobileip-firewall-trav-00.txt, work in progress, March 17, 1997.

10. Contact Information

Jim Binkley
Computer Science Department
Portland State University
Email: jrb@cs.px.edu

John Richardson
Intel
Email: jwr@intel.com