Mobile IP Version 6 Route Optimization Security Design Background

Status of This Memo

Copyright Notice

Abstract

   This document is an account of the rationale behind the Mobile IPv6
   (MIPv6) Route Optimization security design. The purpose of this
   document is to present the thinking and to preserve the reasoning
   behind the Mobile IPv6 security design in 2001 - 2002.

   The document has two target audiences: (1) helping MIPv6 implementors
   to better understand the design choices in MIPv6 security procedures,
   and (2) allowing people dealing with mobility or multi-homing to
   avoid a number of potential security pitfalls in their designs.

Table of Contents

1. Introduction

   Mobile IPv4 is based on the idea of supporting mobility on top of
   existing IP infrastructure, without requiring any modifications to
   the routers, the applications, or the stationary end hosts. However,
   in Mobile IPv6 [6] (as opposed to Mobile IPv4), the stationary end
   hosts may provide support for mobility, i.e., route optimization. In
   route optimization, a correspondent node (CN) (i.e., a peer for a
   mobile node) learns a binding between the mobile node's stationary
   home address and its current temporary care-of address. This binding
   is then used to modify the handling of outgoing (as well as the
   processing of incoming) packets, leading to security risks. The
   purpose of this document is to provide a relatively compact source
   for the background assumptions, design choices, and other information
   needed to understand the route optimization security design. This
   document does not seek to compare the relative security of Mobile
   IPv6 and other mobility protocols, or to list all the alternative
   security mechanisms that were discussed during the Mobile IPv6 design
   process. For a summary of the latter, we refer the reader to [1].
   Even though incidental implementation suggestions are included for
   illustrative purposes, the goal of this document is not to provide a
   guide to implementors. Instead, it is to explain the design choices
   and rationale behind the current route optimization design. The
   authors participated in the design team that produced the design and
   hope, via this note, to capture some of the lessons and reasoning
   behind that effort.

   The authors' intent is to document the thinking behind that design
   effort as it was. Even though this note may incorporate more recent
   developments in order to illustrate the issues, it is not our intent
   to present a new design. Rather, along with the lessons learned,
   there is some effort to clarify differing opinions, questionable
   assumptions, or newly discovered vulnerabilities, should such new
   information be available today. This is also very important, because
   it may benefit the working group's hindsight as it revises or
   improves the Mobile IPv6 specification.

   To fully understand the security implications of the relevant design
   constraints, it is necessary to explore briefly the nature of the
   existing IP infrastructure, the problems Mobile IP aims to solve, and
   the design principles applied. In the light of this background, we
   can then explore IP-based mobility in more detail and have a brief
   look at the security problems. The background is given in the rest
   of this section, starting from Section 1.1.

   Although the introduction in Section 1.1 may appear redundant to
   readers who are already familiar with Mobile IPv6, it may be valuable
   to read it anyway. The approach taken in this document is very

different from that in the Mobile IPv6 specification. That is, we
have explicitly aimed to expose the implicit assumptions and design
choices made in the base Mobile IPv6 design, while the Mobile IPv6
specification aims to state the result of the design. By
understanding the background, it is much easier to understand the
source of some of the related security problems, and to understand
the limitations intrinsic to the provided solutions.

In particular, this document explains how the adopted design for
"Return Routability" (RR) protects against the identified threats
(Section 3). This is true except for attacks on the RR protocol
itself, which require other countermeasures based on heuristics and
judicious implementation (Section 3.3).

The rest of this document is organized as follows: after this
introductory section, we start by considering the avenues of attack
in Section 2. The security problems and countermeasures are studied
in detail in Section 3. Section 4 explains the overall operation and
design choices behind the current security design. Section 5
analyzes the design and discuss the remaining threats. Finally,
Section 6 concludes this document.

1.1. Assumptions about the Existing IP Infrastructure

One of the design goals in the Mobile IP design was to make mobility
possible without changing too much. This was especially important
for IPv4, with its large installed base, but the same design goals
were inherited by Mobile IPv6. Some alternative proposals take a
different approach and propose larger modifications to the Internet
architecture (see Section 1.4).

To understand Mobile IPv6, it is important to understand the MIPv6
design view of the base IPv6 protocol and infrastructure. The most
important base assumptions can be expressed as follows:

1. The routing prefixes available to a node are determined by its
    current location, and therefore the node must change its IP
    address as it moves.

2. The routing infrastructure is assumed to be secure and well
    functioning, delivering packets to their intended destinations as
    identified by destination address.

Although these assumptions may appear to be trivial, let us explore
them a little further. First, in current IPv6 operational practice
the IP address prefixes are distributed in a hierarchical manner.
This limits the number of routing table entries each individual
router needs to handle. An important implication is that the

topology determines what globally routable IP addresses are available
at a given location. That is, the nodes cannot freely decide what
globally routable IP address to use; they must rely on the routing
prefixes served by the local routers via Router Advertisements or by
a DHCP server. In other words, IP addresses are just what the name
says, addresses (i.e., locators).

Second, in the current Internet structure, the routers collectively
maintain a distributed database of the network topology and forward
each packet towards the location determined by the destination
address carried in the packet. To maintain the topology information,
the routers must trust each other, at least to a certain extent. The
routers learn the topology information from the other routers, and
they have no option but to trust their neighbor routers about distant
topology. At the borders of administrative domains, policy rules are
used to limit the amount of perhaps faulty routing table information
received from the peer domains. While this is mostly used to weed
out administrative mistakes, it also helps with security. The aim is
to maintain a reasonably accurate idea of the network topology even
if someone is feeding faulty information to the routing system.

In the current Mobile IPv6 design, it is explicitly assumed that the
routers and the policy rules are configured in a reasonable way, and
that the resulting routing infrastructure is trustworthy enough.
That is, it is assumed that the routing system maintains accurate
information of the network topology, and that it is therefore able to
route packets to their destination locations. If this assumption is
broken, the Internet itself is broken in the sense that packets go to
wrong locations. Such a fundamental malfunction of the Internet
would render hopeless any other effort to assure correct packet
delivery (e.g., any efforts due to Mobile IP security
considerations).

1.1.1. A Note on Source Addresses and Ingress Filtering

Some of the threats and attacks discussed in this document take
advantage of the ease of source address spoofing. That is, in the
current Internet it is possible to send packets with a false source
IP address. The eventual introduction of ingress filtering is
assumed to prevent this. When ingress filtering is used, traffic
with spoofed addresses is not forwarded. This filtering can be
applied at different network borders, such as those between an
Internet service provider (ISP) and its customers, between downstream
and upstream ISPs, or between peer ISPs [5]. Obviously, the
granularity of ingress filters specifies how much you can "spoof
inside a prefix". For example, if an ISP ingress filters a
customer's link but the customer does nothing, anything inside the
customer's /48 prefix could be spoofed. If the customer does

filtering at LAN subnets, anything inside the /64 prefixes could be
spoofed. Despite the limitations imposed by such "in-prefix
spoofing", in general, ingress filtering enables traffic to be
traceable to its real source network [5].

However, ingress filtering helps if and only if a large part of the
Internet uses it. Unfortunately, there are still some issues (e.g.,
in the presence of site multi-homing) that, although not
insurmountable, do require careful handling, and that are likely to
limit or delay its usefulness [5].

1.2. The Mobility Problem and the Mobile IPv6 Solution

The Mobile IP design aims to solve two problems at the same time.
First, it allows transport layer sessions (TCP connections, UDP-
based transactions) to continue even if the underlying host(s) move
and change their IP addresses. Second, it allows a node to be
reached through a static IP address, a home address (HoA).

The latter design choice can also be stated in other words: Mobile
IPv6 aims to preserve the identifier nature of IP addresses. That
is, Mobile IPv6 takes the view that IP addresses can be used as
natural identifiers of nodes, as they have been used since the
beginning of the Internet. This must be contrasted to proposed and
existing alternative designs where the identifier and locator natures
of the IP addresses have been separated (see Section 1.4).

The basic idea in Mobile IP is to allow a home agent (HA) to work as
a stationary proxy for a mobile node (MN). Whenever the mobile node
is away from its home network, the home agent intercepts packets
destined to the node and forwards the packets by tunneling them to
the node's current address, the care-of address (CoA). The transport
layer (e.g., TCP, UDP) uses the home address as a stationary
identifier for the mobile node. Figure 1 illustrates this basic
arrangement.

The basic solution requires tunneling through the home agent, thereby
leading to longer paths and degraded performance. This tunneling is
sometimes called triangular routing since it was originally planned
that the packets from the mobile node to its peer could still
traverse directly, bypassing the home agent.

```
 +----+                                           +----+
 | MN |=#=#=#=#=#=#=#=#=tunnel=#=#=#=#=#=#=#=#|#HA |
 +----+              _____              +-#--+
  | CoA     ___/               \_____              # Home Link
 -+-------/         Internet     * * *-*-*-*-#-#-#-#-----
            |                 * *       |    * Home Address
          \___          * *        ____/    + * -+
            \_____*_____/              | MN |
                  *                     + - -+
          +----+
          | CN |    Data path as      * * * *
          +----+    it appears to correspondent node

                     Real data path   # # # #
```
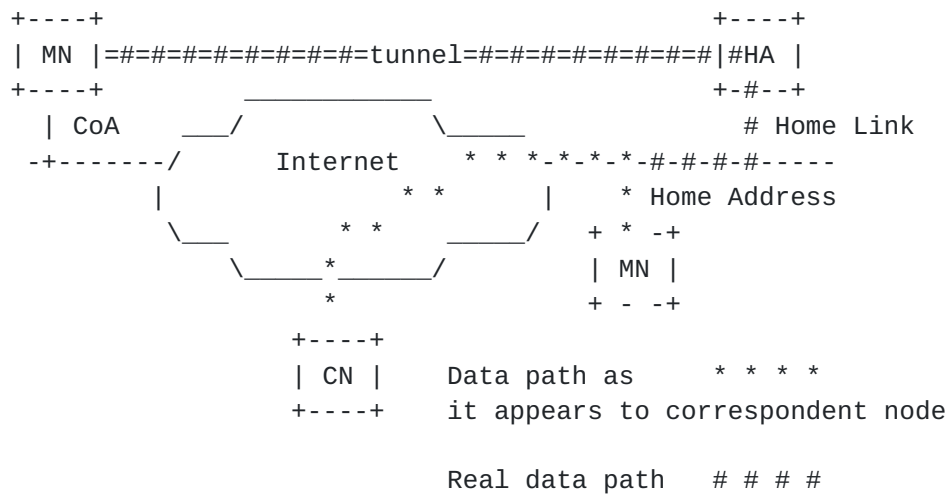
                Figure 1. Basic Mode of Operation in Mobile IPv6

To alleviate the performance penalty, Mobile IPv6 includes a mode of
operation that allows the mobile node and its peer, a correspondent
node (CN), to exchange packets directly, bypassing the home agent
completely after the initial setup phase. This mode of operation is
called route optimization (RO). When route optimization is used, the
mobile node sends its current care-of address to the correspondent
node, using binding update (BU) messages. The correspondent node
stores the binding between the home address and care-of address into
its Binding Cache.

Whenever MIPv6 route optimization is used, the correspondent node
effectively functions in two roles. Firstly, it is the source of the
packets it sends, as usual. Secondly, it acts as the first router
for the packets, effectively performing source routing. That is,
when the correspondent node is sending out packets, it consults its
MIPv6 route optimization data structures and reroutes the packets, if
necessary. A Binding Cache Entry (BCE) contains the home address and
the care-of address of the mobile node, and records the fact that
packets destined to the home address should now be sent to the
destination address. Thus, it represents a local routing exception.

The packets leaving the correspondent node are source routed to the
care-of address. Each packet includes a routing header that contains
the home address of the mobile node. Thus, logically, the packet is
first routed to the care-of address and then, virtually, from the
care-of address to the home address. In practice, of course, the
packet is consumed by the mobile node at the care-of address; the
header just allows the mobile node to select a socket associated with
the home address instead of one with the care-of address. However,
the mechanism resembles source routing, as there is routing state
involved at the correspondent node, and a routing header is used.

Nevertheless, this routing header is special (type 2) to avoid the
risks associated with using the more general (type 0) variant.

1.3. Design Principles and Goals

The MIPv6 design and security design aimed to follow the end-to-end
principle, to notice the differences in trust relationships between
the nodes, and to be explicit about delivering a practical (instead
of an over-ambitious) level of protection.

1.3.1. End-to-End Principle

Perhaps the leading design principle for Internet protocols is the
so-called end-to-end principle [4][11]. According to this principle,
it is beneficial to avoid polluting the network with state, and to
limit new state creation to the involved end nodes.

In the case of Mobile IPv6, the end-to-end principle is applied by
restricting mobility-related state primarily to the home agent.
Additionally, if route optimization is used, the correspondent nodes
also maintain a soft state relating to the mobile nodes' current
care-of addresses, the Binding Cache. This can be contrasted to an
approach that would use individual host routes within the basic
routing system. Such an approach would create state on a huge number
of routers around the network. In Mobile IPv6, only the home agent
and the communicating nodes need to create state.

1.3.2. Trust Assumptions

In the Mobile IPv6 security design, different approaches were chosen
for securing the communication between the mobile node and its home
agent and between the mobile node and its correspondent nodes. In
the home agent case, it was assumed that the mobile node and the home
agent know each other through a prior arrangement, e.g., due to a
business relationship. In contrast, it was strictly assumed that the
mobile node and the correspondent node do not need to have any prior
arrangement, thereby allowing Mobile IPv6 to function in a scalable
manner, without requiring any configuration at the correspondent
nodes.

1.3.3. Protection Level

As a security goal, Mobile IPv6 design aimed to be "as secure as the
(non-mobile) IPv4 Internet" was at the time of the design, in the
period 2001 - 2002. In particular, that means that there is little
protection against attackers that are able to attach themselves
between a correspondent node and a home agent. The rationale is
simple: in the 2001 Internet, if a node was able to attach itself to

the communication path between two arbitrary nodes, it was able to
disrupt, modify, and eavesdrop all the traffic between the two nodes,
unless IPsec protection was used. Even when IPsec was used, the
attacker was still able to block communication selectively by simply
dropping the packets. The attacker in control of a router between
the two nodes could also mount a flooding attack by redirecting the
data flows between the two nodes (or, more practically, an equivalent
flow of bogus data) to a third party.

1.4. About Mobile IPv6 Mobility and its Variations

Taking a more abstract angle, IPv6 mobility can be defined as a
mechanism for managing local exceptions to routing information in
order to direct packets that are sent to one address (the home
address) to another address (the care-of address). It is managing in
the sense that the local routing exceptions (source routes) are
created and deleted dynamically, according to instructions sent by
the mobile node. It is local in the sense that the routing
exceptions are valid only at the home agent, and in the correspondent
nodes if route optimization is used. The created pieces of state are
exceptions in the sense that they override the normal topological
routing information carried collectively by the routers.

Using the terminology introduced by J. Noel Chiappa [14], we can say
that the home address functions in the dual role of being an end-
point identifier (EID) and a permanent locator. The care-of address
is a pure, temporary locator, which identifies the current location
of the mobile node. The correspondent nodes effectively perform
source routing, redirecting traffic destined to the home address to
the care-of address. This is even reflected in the packet structure:
the packets carry an explicit routing header.

The relationship between EIDs and permanent locators has been
exploited by other proposals. Their technical merits and security
problems, however, are beyond the scope of this document.

2. Avenues of Attack

From the discussion above, it should now be clear that the dangers
that Mobile IPv6 must protect from lie in creation (or deletion) of
the local routing exceptions. In Mobile IPv6 terms, the danger is in
the possibility of unauthorized creation of Binding Cache Entries
(BCE). The effects of an attack differ depending on the target of
the attack, the timing of the attack, and the location of the
attacker.

2.1. Target

   Basically, the target of an attack can be any node or network in the
   Internet (stationary or mobile). The basic differences lie in the
   goals of the attack: does the attacker aim to divert (steal) the
   traffic destined to and/or sourced at the target node, or does it aim
   to cause denial-of-service to the target node or network? The target
   does not typically play much of an active role attack. As an
   example, an attacker may launch a denial-of-service attack on a given
   node, A, by contacting a large number of nodes, claiming to be A, and
   subsequently diverting the traffic at these other nodes so that A is
   no longer able to receive packets from those nodes. A itself need
   not be involved at all before its communications start to break.
   Furthermore, A is not necessarily a mobile node; it may well be
   stationary.

   Mobile IPv6 uses the same class of IP addresses for both mobile nodes
   (i.e., home and care-of addresses) and stationary nodes. That is,
   mobile and stationary addresses are indistinguishable from each
   other. Attackers can take advantage of this by taking any IP address
   and using it in a context where, normally, only mobile (home or
   care-of) addresses appear. This means that attacks that otherwise
   would only concern mobile nodes are, in fact, a threat to all IPv6
   nodes.

   In fact, a mobile node appears to be best protected, since a mobile
   node does not need to maintain state about the whereabouts of some
   remote nodes. Conversely, the role of being a correspondent node
   appears to be the weakest, since there are very few assumptions upon
   which it can base its state formation. That is, an attacker has a
   much easier task in fooling a correspondent node to believe that a
   presumably mobile node is somewhere it is not, than in fooling a
   mobile node itself into believing something similar. On the other
   hand, since it is possible to attack a node indirectly by first
   targeting its peers, all nodes are equally vulnerable in some sense.
   Furthermore, a (usually) mobile node often also plays the role of
   being a correspondent node, since it can exchange packets with other
   mobile nodes (see also Section 5.4).

2.2. Timing

   An important aspect in understanding Mobile IPv6-related dangers is
   timing. In a stationary IPv4 network, an attacker must be between
   the communication nodes at the same time as the nodes communicate.
   With the Mobile IPv6 ability of creating binding cache entries, the
   situation changes. A new danger is created. Without proper
   protection, an attacker could attach itself between the home agent
   and a correspondent node for a while, create a BCE at the

correspondent node, leave the position, and continuously update the
correspondent node about the mobile node's whereabouts. This would
make the correspondent node send packets destined to the mobile node
to an incorrect address as long as the BCE remained valid, i.e.,
typically until the correspondent node is rebooted. The converse
would also be possible: an attacker could also launch an attack by
first creating a BCE and then letting it expire at a carefully
selected time. If a large number of active BCEs carrying large
amounts of traffic expired at the same time, the result might be an
overload towards the home agent or the home network. (See Section
3.2.2 for a more detailed explanation.)

2.3. Location

In a static IPv4 Internet, an attacker can only receive packets
destined to a given address if it is able to attach itself to, or to
control, a node on the topological path between the sender and the
recipient. On the other hand, an attacker can easily send spoofed
packets from almost anywhere. If Mobile IPv6 allowed sending
unprotected Binding Updates, an attacker could create a BCE on any
correspondent node from anywhere in the Internet, simply by sending a
fraudulent Binding Update to the correspondent node. Instead of
being required to be between the two target nodes, the attacker could
act from anywhere in the Internet.

In summary, by introducing the new routing exception (binding cache)
at the correspondent nodes, Mobile IPv6 introduces the dangers of
time and space shifting. Without proper protection, Mobile IPv6
would allow an attacker to act from anywhere in the Internet and well
before the time of the actual attack. In contrast, in the static
IPv4 Internet, the attacking nodes must be present at the time of the
attack and they must be positioned in a suitable way, or the attack
would not be possible in the first place.

3. Threats and Limitations

This section describes attacks against Mobile IPv6 Route Optimization
and what protection mechanisms Mobile IPv6 applies against them. The
goal of the attacker can be to corrupt the correspondent node's
binding cache and to cause packets to be delivered to a wrong
address. This can compromise secrecy and integrity of communication
and cause denial-of-service (DoS) both at the communicating parties
and at the address that receives the unwanted packets. The attacker
may also exploit features of the Binding Update (BU) mechanism to
exhaust the resources of the mobile node, the home agent, or the
correspondent nodes. The aim of this section is to provide an
overview of the various protocol mechanisms and their limitations.
The details of the mechanisms are covered in Section 4.

It is essential to understand that some of the threats are more
serious than others, that some can be mitigated but not removed, that
some threats may represent acceptable risk, and that some threats may
be considered too expensive to the attacker to be worth preventing.

We consider only active attackers. The rationale behind this is that
in order to corrupt the binding cache, the attacker must sooner or
later send one or more messages. Thus, it makes little sense to
consider attackers that only observe messages but do not send any.
In fact, some active attacks are easier, for the average attacker, to
launch than a passive one would be. That is, in many active attacks
the attacker can initiate binding update processing at any time,
while most passive attacks require the attacker to wait for suitable
messages to be sent by the target nodes.

Nevertheless, an important class of passive attacks remains: attacks
on privacy. It is well known that simply by examining packets,
eavesdroppers can track the movements of individual nodes (and
potentially, users) [3]. Mobile IPv6 exacerbates the problem by
adding more potentially sensitive information into the packets (e.g.,
Binding Updates, routing headers or home address options). This
document does not address these attacks.

We first consider attacks against nodes that are supposed to have a
specified address (Section 3.1), continuing with flooding attacks
(Section 3.2) and attacks against the basic Binding Update protocol
(Section 3.3). After that, we present a classification of the
attacks (Section 3.4). Finally, we consider the applicability of
solutions relying on some kind of a global security infrastructure
(Section 3.5).

3.1. Attacks Against Address 'Owners' ("Address Stealing")

The most obvious danger in Mobile IPv6 is address "stealing", when an
attacker illegitimately claims to be a given node at a given address
and tries to "steal" traffic destined to that address. We first
describe the basic variant of this attack, follow with a description
of how the situation is affected if the target is a stationary node,
and continue with more complicated issues related to timing (so
called "future" attacks), confidentiality and integrity, and DoS
aspects.

3.1.1. Basic Address Stealing

If Binding Updates were not authenticated at all, an attacker could
fabricate and send spoofed binding updates from anywhere in the
Internet. All nodes that support the correspondent node
functionality would become unwitting accomplices to this attack. As

explained in Section 2.1, there is no way of telling which addresses
belong to mobile nodes that really could send binding updates and
which addresses belong to stationary nodes (see below), so
potentially any node (including "static" nodes) is vulnerable.

```
      +---+  original       +---+ new packet   +---+
      | B |<---------------| A |- - - - - - ->| C |
      +---+  packet flow    +---+ flow         +---+
                             ^
                             |
                             | False BU: B -> C
                             |
                   +----------+
                   | Attacker |
                   +----------+
```
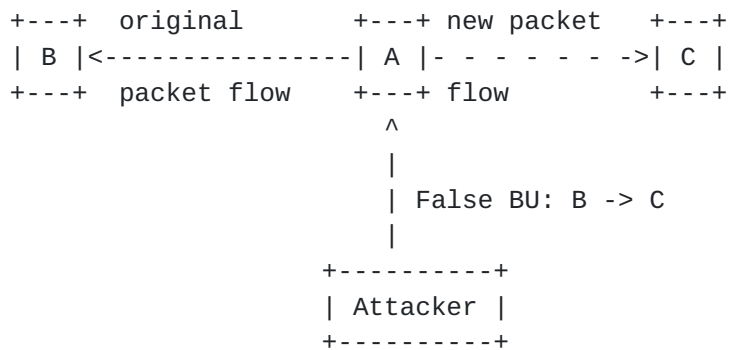
                  Figure 2. Basic Address Stealing

Consider an IP node, A, sending IP packets to another IP node, B.
The attacker could redirect the packets to an arbitrary address, C,
by sending a Binding Update to A. The home address (HoA) in the
binding update would be B and the care-of address (CoA) would be C.
After receiving this binding update, A would send all packets
intended for the node B to the address C. See Figure 2.

The attacker might select the care-of address to be either its own
current address, another address in its local network, or any other
IP address. If the attacker selected a local care-of address
allowing it to receive the packets, it would be able to send replies
to the correspondent node. Ingress filtering at the attacker's
local+ network does not prevent the spoofing of Binding Updates but
forces the attacker either to choose a care-of address from inside
its own network or to use the Alternate care-of address sub-option.

The binding update authorization mechanism used in the MIPv6 security
design is primarily intended to mitigate this threat, and to limit
the location of attackers to the path between a correspondent node
and the home agent.

3.1.2. Stealing Addresses of Stationary Nodes

The attacker needs to know or guess the IP addresses of both the
source of the packets to be diverted (A in the example above) and the
destination of the packets (B, above). This means that it is
difficult to redirect all packets to or from a specific node because
the attacker would need to know the IP addresses of all the nodes
with which it is communicating.

   Nodes with well-known addresses, such as servers and those using
   stateful configuration, are most vulnerable. Nodes that are a part
   of the network infrastructure, such as DNS servers, are particularly
   interesting targets for attackers and particularly easy to identify.

   Nodes that frequently change their address and use random addresses
   are relatively safe. However, if they register their address into
   Dynamic DNS, they become more exposed. Similarly, nodes that visit
   publicly accessible networks such as airport wireless LANs risk
   revealing their addresses. IPv6 addressing privacy features [3]
   mitigate these risks to an extent, but note that addresses cannot be
   completely recycled while there are still open sessions that use
   those addresses.

   Thus, it is not the mobile nodes that are most vulnerable to address
   stealing attacks; it is the well-known static servers. Furthermore,
   the servers often run old or heavily optimized operating systems and
   may not have any mobility related code at all. Thus, the security
   design cannot be based on the idea that mobile nodes might somehow be
   able to detect whether someone has stolen their address, and reset
   the state at the correspondent node. Instead, the security design
   must make reasonable measures to prevent the creation of fraudulent
   binding cache entries in the first place.

3.1.3. Future Address Sealing

   If an attacker knows an address that a node is likely to select in
   the future, it can launch a "future" address stealing attack. The
   attacker creates a Binding Cache Entry with the home address that it
   anticipates the target node will use. If the Home Agent allows
   dynamic home addresses, the attacker may be able to do this
   legitimately. That is, if the attacker is a client of the Home Agent
   and is able to acquire the home address temporarily, it may be able
   to do so and then to return the home address to the Home Agent once
   the BCE is in place.

   Now, if the BCE state had a long expiration time, the target node
   would acquire the same home address while the BCE is still effective,
   and the attacker would be able to launch a successful man-in-the-
   middle or denial-of-service attack. The mechanism applied in the
   MIPv6 security design is to limit the lifetime of Binding Cache
   Entries to a few minutes.

   Note that this attack applies only to fairly specific conditions.
   There are also some variations of this attack that are theoretically
   possible under some other conditions. However, all of these attacks
   are limited by the Binding Cache Entry lifetime, and therefore they
   are not a real concern with the current design.

3.1.4. Attacks against Secrecy and Integrity

   By spoofing Binding Updates, an attacker could redirect all packets
   between two IP nodes to itself. By sending a spoofed binding update
   to A, it could capture the data intended to B. That is, it could
   pretend to be B and highjack A's connections with B, or it could
   establish new spoofed connections. The attacker could also send
   spoofed binding updates to both A and B and insert itself in the
   middle of all connections between them (man-in-the-middle attack).
   Consequently, the attacker would be able to see and modify the
   packets sent between A and B. See Figure 3.

      Original data path, before man-in-the-middle attack

         +---+   +---+
         | A | | B |
         +---+   +---+
           _____/

      Modified data path, after the falsified binding updates

         +---+   +---+
         | A | | B |
         +---+   +---+
           \   /
            \   /
             \   +----------+   /
              \---------| Attacker |-------/
                        +----------+
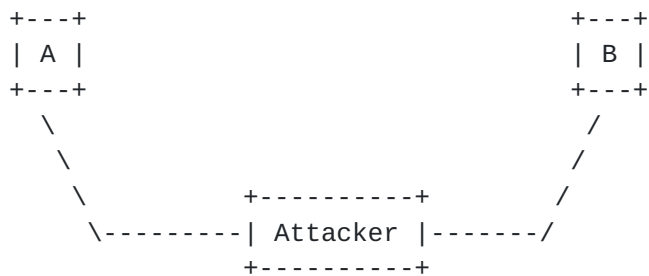
                    Figure 3. Man-in-the-Middle Attack

   Strong end-to-end encryption and integrity protection, such as
   authenticated IPsec, can prevent all the attacks against data secrecy
   and integrity. When the data is cryptographically protected, spoofed
   binding updates could result in denial of service (see below) but not
   in disclosure or corruption of sensitive data beyond revealing the
   existence of the traffic flows. Two fixed nodes could also protect
   communication between themselves by refusing to accept binding
   updates from each other. Ingress filtering, on the other hand, does
   not help, as the attacker is using its own address as the care-of
   address and is not spoofing source IP addresses.

   The protection adopted in MIPv6 Security Design is to authenticate
   (albeit weakly) the addresses by return routability (RR), which
   limits the topological locations from which the attack is possible
   (see Section 4.1).

3.1.5. Basic Denial-of-Service Attacks

   By sending spoofed binding updates, the attacker could redirect all
   packets sent between two IP nodes to a random or nonexistent address
   (or addresses). As a result, it might be able to stop or disrupt
   communication between the nodes. This attack is serious because any
   Internet node could be targeted, including fixed nodes belonging to
   the infrastructure (e.g., DNS servers) that are also vulnerable.
   Again, the selected protection mechanism is return routability (RR).

3.1.6. Replaying and Blocking Binding Updates

   Any protocol for authenticating binding updates has to consider
   replay attacks. That is, an attacker may be able to replay recently
   authenticated binding updates to the correspondent and, consequently,
   to direct packets to the mobile node's previous location. As with
   spoofed binding updates, this could be used both for capturing
   packets and for DoS. The attacker could capture the packets and
   impersonate the mobile node if it reserved the mobile's previous
   address after the mobile node has moved away and then replayed the
   previous binding update to redirect packets back to the previous
   location.

   In a related attack, the attacker blocks binding updates from the
   mobile at its new location, e.g., by jamming the radio link or by
   mounting a flooding attack. The attacker then takes over the
   mobile's connections at the old location. The attacker will be able
   to capture the packets sent to the mobile and to impersonate the
   mobile until the correspondent's Binding Cache entry expires.

   Both of the above attacks require that the attacker be on the same
   local network with the mobile, where it can relatively easily observe
   packets and block them even if the mobile does not move to a new
   location. Therefore, we believe that these attacks are not as
   serious as ones that can be mounted from remote locations. The
   limited lifetime of the Binding Cache entry and the associated nonces
   limit the time frame within which the replay attacks are possible.
   Replay protection is provided by the sequence number and MAC in the
   Binding Update. To not undermine this protection, correspondent
   nodes must exercise care upon deleting a binding cache entry, as per
   section 5.2.8 ("Preventing Replay Attacks") in [6].

3.2. Attacks Against Other Nodes and Networks (Flooding)

   By sending spoofed binding updates, an attacker could redirect
   traffic to an arbitrary IP address. This could be used to overload
   an arbitrary Internet address with an excessive volume of packets
   (known as a 'bombing attack'). The attacker could also target a

   network by redirecting data to one or more IP addresses within the
   network. There are two main variations of flooding: basic flooding
   and return-to-home flooding. We consider them separately.

3.2.1. **Basic Flooding**

   In the simplest attack, the attacker knows that there is a heavy data
   stream from node A to B and redirects this to the target address C.
   However, A would soon stop sending the data because it is not
   receiving acknowledgements from B.

```
        (B is attacker)

        +---+  original        +---+ flooding packet   +---+
        | B |<===============| A |=================>| C |
        +---+  packet flow     +---+ flow              +---+
         |                        ^
          \                      /
           _____/
          False binding update + false acknowledgements
```
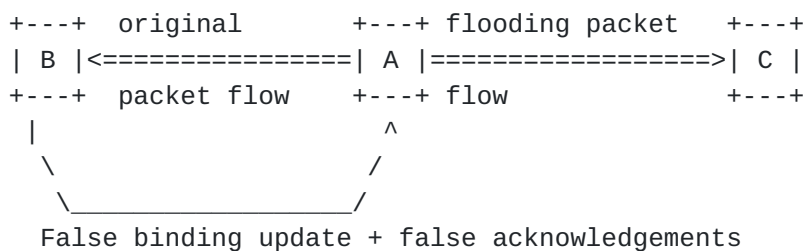
                   Figure 4. Basic Flooding Attack

   A more sophisticated attacker would act itself as B; see Figure 4.
   It would first subscribe to a data stream (e.g., a video stream) and
   redirect this stream to the target address C. The attacker would
   even be able to spoof the acknowledgements. For example, consider a
   TCP stream. The attacker would perform the TCP handshake itself and
   thus know the initial sequence numbers. After redirecting the data
   to C, the attacker would continue to send spoofed acknowledgements.
   It would even be able to accelerate the data rate by simulating a
   fatter pipe [12].

   This attack might be even easier with UDP/RTP. The attacker could
   create spoofed RTCP acknowledgements. Either way, the attacker would
   be able to redirect an increasing stream of unwanted data to the
   target address without doing much work itself. It could carry on
   opening more streams and refreshing the Binding Cache entries by
   sending a new binding update every few minutes. Thus, the limitation
   of BCE lifetime to a few minutes does not help here without
   additional measures.

   During the Mobile IPv6 design process, the effectiveness of this
   attack was debated. It was mistakenly assumed that the target node
   would send a TCP Reset to the source of the unwanted data stream,
   which would then stop sending. In reality, all practical TCP/IP
   implementations fail to send the Reset. The target node drops the
   unwanted packets at the IP layer because it does not have a Binding

Update List entry corresponding to the Routing Header on the incoming
packet. Thus, the flooding data is never processed at the TCP layer
of the target node, and no Reset is sent. This means that the attack
using TCP streams is more effective than was originally believed.

This attack is serious because the target can be any node or network,
not only a mobile one. What makes it particularly serious compared
to the other attacks is that the target itself cannot do anything to
prevent the attack. For example, it does not help if the target
network stops using Route Optimization. The damage is compounded if
these techniques are used to amplify the effect of other distributed
denial-of-service (DDoS) attacks. Ingress filtering in the
attacker's local network prevents the spoofing of source addresses
but the attack would still be possible by setting the Alternate
care-of address sub-option to the target address.

Again, the protection mechanism adopted for MIPv6 is return
routability. This time it is necessary to check that there is indeed
a node at the new care-of address, and that the node is the one that
requested redirecting packets to that very address (see Section
4.1.2).

3.2.2. Return-to-Home Flooding

A variation of the bombing attack would target the home address or
the home network instead of the care-of address or a visited network.
The attacker would claim to be a mobile with the home address equal
to the target address. While claiming to be away from home, the
attacker would start downloading a data stream. The attacker would
then send a binding update cancellation (i.e., a request to delete
the binding from the Binding Cache) or just allow the cache entry to
expire. Either would redirect the data stream to the home network.
As when bombing a care-of address, the attacker can keep the stream
alive and even increase the data rate by spoofing acknowledgements.
When successful, the bombing attack against the home network is just
as serious as that against a care-of address.

The basic protection mechanism adopted is return routability.
However, it is hard to fully protect against this attack; see Section
4.1.1.

3.3. Attacks against Binding Update Protocols

Security protocols that successfully protect the secrecy and
integrity of data can sometimes make the participants more vulnerable
to denial-of-service attacks. In fact, the stronger the
authentication, the easier it may be for an attacker to use the

protocol features to exhaust the mobile's or the correspondent's
resources.

3.3.1. Inducing Unnecessary Binding Updates

When a mobile node receives an IP packet from a new correspondent via
the home agent, it may initiate the binding update protocol. An
attacker can exploit this by sending the mobile node a spoofed IP
packet (e.g., ping or TCP SYN packet) that appears to come from a new
correspondent node. Since the packet arrives via the home agent, the
mobile node may start the binding update protocol with the
correspondent node. The decision as to whether to initiate the
binding update procedure may depend on several factors (including
heuristics, cross layer information, and configuration options) and
is not specified by Mobile IPv6. Not initiating the binding update
procedure automatically may alleviate these attacks, but it will not,
in general, prevent them completely.

In a real attack the attacker would induce the mobile node to
initiate binding update protocols with a large number of
correspondent nodes at the same time. If the correspondent addresses
are real addresses of existing IP nodes, then most instances of the
binding update protocol might even complete successfully. The
entries created in the Binding Cache are correct but useless. In
this way, the attacker can induce the mobile to execute the binding
update protocol unnecessarily, which can drain the mobile's
resources.

A correspondent node (i.e., any IP node) can also be attacked in a
similar way. The attacker sends spoofed IP packets to a large number
of mobiles, with the target node's address as the source address.
These mobiles will initiate the binding update protocol with the
target node. Again, most of the binding update protocol executions
will complete successfully. By inducing a large number of
unnecessary binding updates, the attacker is able to consume the
target node's resources.

This attack is possible against any binding update authentication
protocol. The more resources the binding update protocol consumes,
the more serious the attack. Therefore, strong cryptographic
authentication protocol is more vulnerable to the attack than a weak
one or unauthenticated binding updates. Ingress filtering helps a
little, since it makes it harder to forge the source address of the
spoofed packets, but it does not completely eliminate this threat.

A node should protect itself from the attack by setting a limit on
the amount of resources (i.e., processing time, memory, and
communications bandwidth) that it uses for processing binding

updates. When the limit is exceeded, the node can simply stop
attempting route optimization. Sometimes it is possible to process
some binding updates even when a node is under the attack. A mobile
node may have a local security policy listing a limited number of
addresses to which binding updates will be sent even when the mobile
node is under DoS attack. A correspondent node (i.e., any IP node)
may similarly have a local security policy listing a limited set of
addresses from which binding updates will be accepted even when the
correspondent is under a binding update DoS attack.

The node may also recognize addresses with it had meaningful
communication in the past and only send binding updates to, or accept
them from, those addresses. Since it may be impossible for the IP
layer to know about the protocol state in higher protocol layers, a
good measure of the meaningfulness of the past communication is
probably per-address packet counts. Alternatively, Neighbor
Discovery [2] (Section 5.1, Conceptual Data Structures) defines the
Destination Cache as a set of entries about destinations to which
traffic has been sent recently. Thus, implementors may wish to use
the information in the Destination Cache.

Section 11.7.2 ("Correspondent Registration") in [6] does not specify
when such a route optimization procedure should be initiated. It
does indicate when it may justifiable to do so, but these hints are
not enough. This remains an area where more work is needed.
Obviously, given that route optimization is optional, any node that
finds the processing load excessive or unjustified may simply turn it
off (either selectively or completely).

3.3.2. Forcing Non-Optimized Routing

As a variant of the previous attack, the attacker can prevent a
correspondent node from using route optimization by filling its
Binding Cache with unnecessary entries so that most entries for real
mobiles are dropped.

Any successful DoS attack against a mobile or correspondent node can
also prevent the processing of binding updates. We have previously
suggested that the target of a DoS attack may respond by stopping
route optimization for all or some communication. Obviously, an
attacker can exploit this fallback mechanism and force the target to
use the less efficient home agent-based routing. The attacker only
needs to mount a noticeable DoS attack against the mobile or
correspondent, and the target will default to non-optimized routing.

The target node can mitigate the effects of the attack by reserving
more space for the Binding Cache, by reverting to non-optimized
routing only when it cannot otherwise cope with the DoS attack, by

   trying aggressively to return to optimized routing, or by favoring
   mobiles with which it has an established relationship. This attack
   is not as serious as the ones described earlier, but applications
   that rely on Route Optimization could still be affected. For
   instance, conversational multimedia sessions can suffer drastically
   from the additional delays caused by triangle routing.

[3.3.3](#). Reflection and Amplification

   Attackers sometimes try to hide the source of a packet-flooding
   attack by reflecting the traffic from other nodes [[1](#)]. That is,
   instead of sending the flood of packets directly to the target, the
   attacker sends data to other nodes, tricking them to send the same
   number, or more, packets to the target. Such reflection can hide the
   attacker's address even when ingress filtering prevents source
   address spoofing. Reflection is particularly dangerous if the
   packets can be reflected multiple times, if they can be sent into a
   looping path, or if the nodes can be tricked into sending many more
   packets than they receive from the attacker, because such features
   can be used to amplify the traffic by a significant factor. When
   designing protocols, one should avoid creating services that can be
   used for reflection and amplification.

   Triangle routing would easily create opportunities for reflection: a
   correspondent node receives packets (e.g., TCP SYN) from the mobile
   node and replies to the home address given by the mobile node in the
   Home Address Option (HAO). The mobile might not really be a mobile
   and the home address could actually be the target address. The
   target would only see the packets sent by the correspondent and could
   not see the attacker's address (even if ingress filtering prevents
   the attacker from spoofing its source address).

```
        +----------+ TCP SYN with HAO    +-----------+
        | Attacker |-------------------->| Reflector |
        +----------+                     +-----------+
                                               |
                                               | TCP SYN-ACK to HoA
                                               V
                                         +-----------+
                                         | Flooding  |
                                         | target    |
                                         +-----------+
```
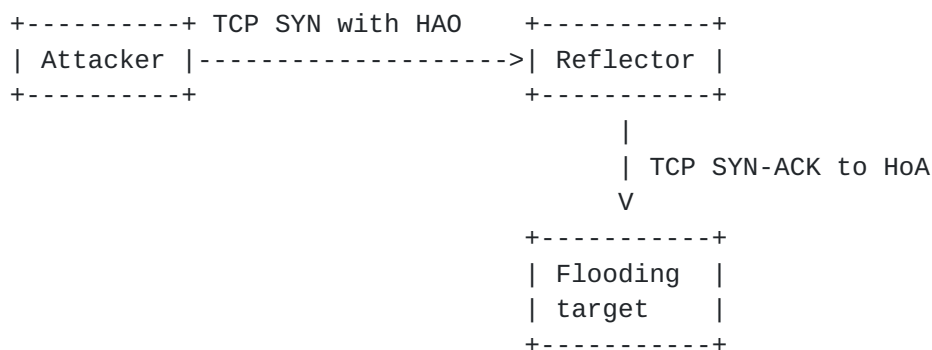
                        Figure 5. Reflection Attack

   A badly designed binding update protocol could also be used for
   reflection: the correspondent would respond to a data packet by
   initiating the binding update authentication protocol, which usually

involves sending a packet to the home address. In that case, the
reflection attack can be discouraged by copying the mobile's address
into the messages sent by the mobile to the correspondent. (The
mobile's source address is usually the same as the care-of address,
but an Alternative Care-of Address sub-option can specify a different
care-of address.) Some of the early proposals for MIPv6 security
used this approach and were prone to reflection attacks.

In some of the proposals for binding update authentication protocols,
the correspondent node responded to an initial message from the
mobile with two packets (one to the home address, one to the care-of
address). It would have been possible to use this to amplify a
flooding attack by a factor of two. Furthermore, with public-key
authentication, the packets sent by the correspondent might have been
significantly larger than the one that triggers them.

These types of reflection and amplification can be avoided by
ensuring that the correspondent only responds to the same address
from which it received a packet, and only with a single packet of the
same size. These principles have been applied to MIPv6 security
design.

[3.4](#). Classification of Attacks

```
Sect. Attack name                            Target Sev. Mitigation
---------------------------------------------------------------------
3.1.1 Basic address stealing                 MN     Med. RR
3.1.2 Stealing addresses of stationary nodes Any    High RR
3.1.3 Future address stealing                MN     Low  RR, lifetime
3.1.4 Attacks against secrecy and integrity  MN     Low  RR, IPsec
3.1.5 Basic denial-of-service attacks        Any    Med. RR
3.1.6 Replaying and blocking binding updates MN     Low  lifetime,
                                                         seq number,
                                                         MAC
3.2.1 Basic flooding                         Any    High RR
3.2.2 Return-to-home flooding                Any    High RR
3.3.1 Inducing unnecessary binding updates   MN, CN Med. heuristics
3.3.2 Forcing non-optimized routing          MN     Low  heuristics
3.3.3 Reflection and amplification           N/A    Med. BU design
```

                Figure 6. Summary of Discussed Attacks

Figure 6 gives a summary of the attacks discussed. As it stands at
the time of writing, the return-to-the-home flooding and the
induction of unnecessary binding updates look like the threats
against which we have the least amount of protection, compared to
their severity.

3.5. Problems with Infrastructure-Based Authorization

   Early in the MIPv6 design process, it was assumed that plain IPsec
   could be the default way to secure Binding Updates with arbitrary
   correspondent nodes. However, this turned out to be impossible.
   Plain IPsec relies on an infrastructure for key management, which, to
   be usable with any arbitrary pair of nodes, would need to be global
   in scope. Such a "global PKI" does not exist, nor is it expected to
   come into existence any time soon.

   More minor issues that also surfaced at the time were: (1)
   insufficient filtering granularity for the state of IPsec at the
   time, (2) cost to establish a security association (in terms of CPU
   and round trip times), and (3) expressing the proper authorization
   (as opposed to just authentication) for binding updates [13]. These
   issues are solvable, and, in particular, (1) and (3) have been
   addressed for IPsec usage with binding updates between the mobile
   node and the home agent [7].

   However, the lack of a global PKI remains unsolved.

   One way to provide a global key infrastructure for mobile IP could be
   DNSSEC. Such a scheme is not completely supported by the existing
   specifications, as it constitutes a new application of the KEY RR,
   something explicitly limited to DNSSEC [8] [9] [10]. Nevertheless,
   if one were to define it, one could proceed along the following
   lines: A secure reverse DNS that provided a public key for each IP
   address could be used to verify that a binding update is indeed
   signed by an authorized party. However, in order to be secure, each
   link in such a system must be secure. That is, there must be a chain
   of keys and signatures all the way down from the root (or at least
   starting from a trust anchor common to the mobile node and the
   correspondent node) to the given IP address. Furthermore, it is not
   enough that each key be signed by the key above it in the chain. It
   is also necessary that each signature explicitly authorize the lower
   key to manage the corresponding address block below.

   Even though it would be theoretically possible to build a secure
   reverse DNS infrastructure along the lines shown above, the practical
   problems would be daunting. Whereas the delegation and key signing
   might work close to the root of the tree, it would probably break
   down somewhere along the path to the individual nodes. Note that a
   similar delegation tree is currently being proposed for Secure
   Neighbor Discovery [15], although in this case only routers (not
   necessarily every single potential mobile node) need to secure such a
   certificate. Furthermore, checking all the signatures on the tree
   would place a considerable burden on the correspondent nodes, making
   route optimization prohibitive, or at least justifiable only in very

particular circumstances. Finally, it is not enough simply to check
whether the mobile node is authorized to send binding updates
containing a given home address, because to protect against flooding
attacks, the care-of address must also be verified.

Relying on this same secure DNS infrastructure to verify care-of
addresses would be even harder than verifying home addresses.
Instead, a different method would be required, e.g., a return
routability procedure. If so, the obvious question is whether the
gargantuan cost of deploying the global secure DNS infrastructure is
worth the additional protection it affords, as compared to simply
using return routability for both home address and care-of address
verification.

4. Solution Selected for Mobile IPv6

The current Mobile IPv6 route optimization security has been
carefully designed to prevent or mitigate the threats that were
discussed in Section 3. The goal has been to produce a design with a
level of security close to that of a static IPv4-based Internet, and
with an acceptable cost in terms of packets, delay, and processing.
The result is not what one would expect: it is definitely not a
traditional cryptographic protocol. Instead, the result relies
heavily on the assumption of an uncorrupted routing infrastructure
and builds upon the idea of checking that an alleged mobile node is
indeed reachable through both its home address and its care-of
address. Furthermore, the lifetime of the state created at the
corresponded nodes is deliberately restricted to a few minutes, in
order to limit the potential threat from time shifting.

This section describes the solution in reasonable detail (for further
details see the specification), starting from Return Routability
(Section 4.1), continuing with a discussion about state creation at
the correspondent node (Section 4.2), and completing the description
with a discussion about the lifetime of Binding Cache Entries
(Section 4.3).

4.1. Return Routability

Return Routability (RR) is the name of the basic mechanism deployed
by Mobile IPv6 route optimization security design. RR is based on
the idea that a node should be able to verify that there is a node
that is able to respond to packets sent to a given address. The
check yields false positives if the routing infrastructure is
compromised or if there is an attacker between the verifier and the
address to be verified. With these exceptions, it is assumed that a
successful reply indicates that there is indeed a node at the given

address, and that the node is willing to reply to the probes sent to
it.

The basic return routability mechanism consists of two checks, a Home
Address check (see Section 4.1.1) and a care-of-address check (see
Section 4.1.2). The packet flow is depicted in Figure 7. First, the
mobile node sends two packets to the correspondent node: a Home Test
Init (HoTI) packet is sent through the home agent, and a Care-of Test
Init (CoTI) directly. The correspondent node replies to both of
these independently by sending a Home Test (HoT) in response to the
Home Test Init and a Care-of Test (CoT) in response to the Care-of
Test Init. Finally, once the mobile node has received both the Home
Test and Care-of Test packets, it sends a Binding Update to the
correspondent node.

```
         +------+   1a) HoTI              +------+
         |      |---------------------->|      |
         |  MN  |    2a) HoT              |  HA  |
         |      |<---------------------|      |
         +------+                         +------+
1b) CoTI | ^  |                          /  ^
         | |2b|  CoT                    /  /
         | |  |                        /  /
         | |  | 3) BU                 /  /
          V |   V                    /  /
         +------+   1a) HoTI        /  /
         |      |<----------------/  /
         |  CN  |    2a) HoT         /
         |      |------------------/
         +------+
```
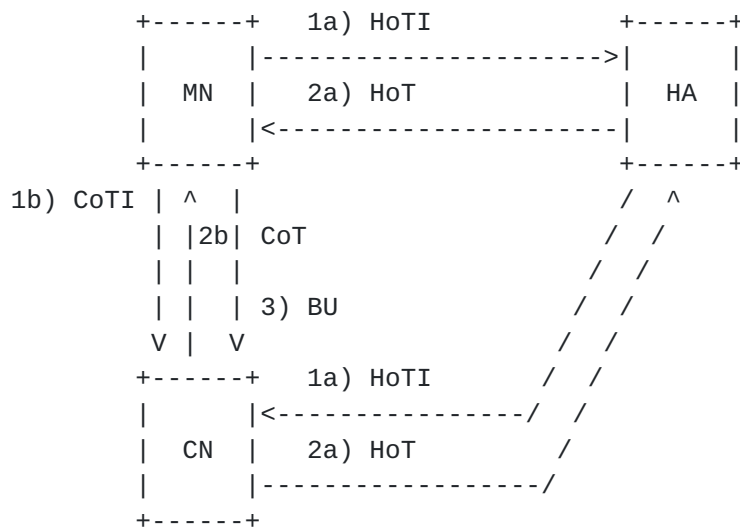
              Figure 7. Return Routability Packet Flow

It might appear that the actual design was somewhat convoluted. That
is, the real return routability checks are the message pairs < Home
Test, Binding Update > and < Care-of Test, Binding Update >. The
Home Test Init and Care-of Test Init packets are only needed to
trigger the test packets, and the Binding Update acts as a combined
routability response to both of the tests.

There are two main reasons behind this design:

o avoidance of reflection and amplification (see Section 3.3.3), and

o avoidance of state exhaustion DoS attacks (see Section 4.2).

The reason for sending two Init packets instead of one is to avoid
amplification. The correspondent node does not know anything about

the mobile node, and therefore it just receives an unsolicited IP
packet from some arbitrary IP address. In a way, this is similar to
a server receiving a TCP SYN from a previously unknown client. If
the correspondent node were to send two packets in response to an
initial trigger, that would provide the potential for a DoS
amplification effect, as discussed in Section 3.3.3.

This scheme also avoids providing for a potential reflection attack.
If the correspondent node were to reply to an address other than the
source address of the packet, that would create a reflection effect.
Thus, the only safe mechanism possible for a naive correspondent is
to reply to each received packet with just one packet, and to send
the reply to the source address of the received packet. Hence, two
initial triggers are needed instead of just one.

Let us now consider the two return routability tests separately. In
the following sections, the derivation of cryptographic material from
each of these is shown in a simplified manner. For the real formulas
and more detail, please refer to [6].

4.1.1. Home Address Check

The Home Address check consists of a Home Test (HoT) packet and a
subsequent Binding Update (BU). It is triggered by the arrival of a
Home Test Init (HoTI). A correspondent node replies to a Home Test
Init by sending a Home Test to the source address of the Home Test
Init. The source address is assumed to be the home address of a
mobile node, and therefore the Home Test is assumed to be tunneled by
the Home Agent to the mobile node. The Home Test contains a
cryptographically generated token, home keygen token, which is formed
by calculating a hash function over the concatenation of a secret
key, Kcn, known only by the correspondent node, the source address of
the Home Test Init packet, and a nonce.

    home keygen token = hash(Kcn | home address | nonce | 0)

An index to the nonce is also included in the Home Test packet,
allowing the correspondent node to find the appropriate nonce more
easily.

The token allows the correspondent node to make sure that any binding
update received subsequently has been created by a node that has seen
the Home Test packet; see Section 4.2.

In most cases, the Home Test packet is forwarded over two different
segments of the Internet. It first traverses from the correspondent
node to the Home Agent. On this trip, it is not protected and any
eavesdropper on the path can learn its contents. The Home Agent then

forwards the packet to the mobile node. This path is taken inside an
IPsec ESP protected tunnel, making it impossible for the outsiders to
learn the contents of the packet.

At first, it may sound unnecessary to protect the packet between the
home agent and the mobile node, since it travelled unprotected
between the correspondent node and the mobile node. If all links in
the Internet were equally insecure, the additional protection would
be unnecessary. However, in most practical settings the network is
likely to be more secure near the home agent than near the mobile
node. For example, if the home agent hosts a virtual home link and
the mobile nodes are never actually at home, an eavesdropper should
be close to the correspondent node or on the path between the
correspondent node and the home agent, since it could not eavesdrop
at the home agent. If the correspondent node is a major server, all
the links on the path between it and the home agent are likely to be
fairly secure. On the other hand, the Mobile Node is probably using
wireless access technology, making it sometimes trivial to eavesdrop
on its access link. Thus, it is fairly easy to eavesdrop on packets
that arrive at the mobile node. Consequently, protecting the HA-MN
path is likely to provide real security benefits even when the CN-HA
path remains unprotected.

4.1.2. Care-of-Address Check

From the correspondent node's point of view, the Care-of-Address
check is very similar to the home check. The only difference is that
now the source address of the received Care-of Test Init packet is
assumed to be the care-of address of the mobile node. Furthermore,
the token is created in a slightly different manner in order to make
it impossible to use home tokens for care-of tokens or vice versa.

     care-of keygen token = hash(Kcn | care-of address | nonce | 1)

The Care-of Test traverses only one leg, directly from the
correspondent node to the mobile node. It remains unprotected all
along the way, making it vulnerable to eavesdroppers near the
correspondent node, on the path from the correspondent node to the
mobile node, or near the mobile node.

4.1.3. Forming the First Binding Update

When the mobile node has received both the Home Test and Care-of Test
messages, it creates a binding key, Kbm, by computing a hash function
over the concatenation of the tokens received.

This key is used to protect the first and the subsequent binding
updates, as long as the key remains valid.

Note that the key Kbm is available to anyone who is able to receive
both the Care-of Test and Home Test messages. However, they are
normally routed by different routes through the network, and the Home
Test is transmitted over an encrypted tunnel from the home agent to
the mobile node (see also Section 5.4).

4.2. Creating State Safely

The correspondent node may remain stateless until it receives the
first Binding Update. That is, it does not need to record receiving
and replying to the Home Test Init and Care-of Test Init messages.
The Home Test Init/Home Test and Care-of Test Init/Care-of Test
exchanges take place in parallel but independently of each other.
Thus, the correspondent can respond to each message immediately, and
it does not need to remember doing that. This helps in potential
denial-of-service situations: no memory needs to be reserved for
processing Home Test Init and Care-of Test Init messages.
Furthermore, Home Test Init and Care-of Test Init processing is
designed to be lightweight, and it can be rate limited if necessary.

When receiving a first binding update, the correspondent node goes
through a rather complicated procedure. The purpose of this
procedure is to ensure that there is indeed a mobile node that has
recently received a Home Test and a Care-of Test that were sent to
the claimed home and care-of addresses, respectively, and to make
sure that the correspondent node does not unnecessarily spend CPU or
other resources while performing this check.

Since the correspondent node does not have any state when the binding
update arrives, the binding update itself must contain enough
information so that relevant state can be created. To that end, the
binding update contains the following pieces of information:

Source address: The care-of address specified in the Binding Update
   must be equal to the source address used in the Care-of Test Init
   message. Notice that this applies to the effective Care-of
   Address of the Binding Update. In particular, if the Binding
   Update includes an Alternate Care-of Address (AltCoA) [6], the
   effective CoA is, of course, this AltCoA. Thus, the Care-of Test
   Init must have originated from the AltCoA.

Home address: The home address specified in the Binding Update must
   be equal to the source address used in the Home Test Init message.

Two nonce indices: These are copied over from the Home Test and
   Care-of Test messages, and together with the other information
   they allow the correspondent node to re-create the tokens sent in
   the Home Test and Care-of Test messages and used for creating Kbm.

Without them, the correspondent node might need to try the 2-3
latest nonces, leading to unnecessary resource consumption.

Message Authentication Code (MAC): The binding update is
authenticated by computing a MAC function over the care-of
address, the correspondent node's address and the binding update
message itself. The MAC is keyed with the key Kbm.

Given the addresses, the nonce indices (and thereby the nonces) and
the key Kcn, the correspondent node can re-create the home and care-
of tokens at the cost of a few memory lookups and computation of one
MAC and one hash function.

Once the correspondent node has re-created the tokens, it hashes the
tokens together, giving the key Kbm. If the Binding Update is
authentic, Kbm is cached together with the binding. This key is then
used to verify the MAC that protects integrity and origin of the
actual Binding Update. Note that the same Kbm may be used for a
while, until the mobile node moves (and needs to get a new care-of-
address token), the care-of token expires, or the home token expires.

4.2.1. Retransmissions and State Machine

Note that since the correspondent node may remain stateless until it
receives a valid binding update, the mobile node is solely
responsible for retransmissions. That is, the mobile node should
keep sending the Home Test Init / Care-of Test Init messages until it
receives a Home Test / Care-of Test, respectively. Similarly, it may
need to send the binding update a few times in the case it is lost
while in transit.

4.3. Quick expiration of the Binding Cache Entries

A Binding Cache Entry, along with the key Kbm, represents the return
routability state of the network at the time when the Home Test and
Care-of Test messages were sent out. It is possible that a specific
attacker is able to eavesdrop a Home Test message at some point of
time, but not later. If the Home Test had an infinite or a long
lifetime, that would allow the attacker to perform a time shifting
attack (see Section 2.2). That is, in the current IPv4 architecture
an attacker on the path between the correspondent node and the home
agent is able to perform attacks only as long as the attacker is able
to eavesdrop (and possibly disrupt) communications on that particular
path. A long living Home Test, and consequently the ability to send
valid binding updates for a long time, would allow the attacker to
continue its attack even after the attacker is no longer able to
eavesdrop on the path.

To limit the seriousness of this and other similar time shifting
threats, the validity of the tokens is limited to a few minutes.
This effectively limits the validity of the key Kbm and the lifetime
of the resulting binding updates and binding cache entries.

Although short lifetimes are required by other aspects of the
security design and the goals, they are clearly detrimental for
efficiency and robustness. That is, a Home Test Init / Home Test
message pair must be exchanged through the home agent every few
minutes. These messages are unnecessary from a purely functional
point of view, thereby representing overhead. What is worse, though,
is that they make the home agent a single point of failure. That is,
if the Home Test Init / Home Test messages were not needed, the
existing connections from a mobile node to other nodes could continue
even when the home agent fails, but the current design forces the
bindings to expire after a few minutes.

This concludes our walk-through of the selected security design. The
cornerstones of the design were the employment of the return
routability idea in the Home Test, Care-of Test, and binding update
messages, the ability to remain stateless until a valid binding
update is received, and the limiting of the binding lifetimes to a
few minutes. Next we briefly discuss some of the remaining threats
and other problems inherent to the design.

5. **Security Considerations**

This section gives a brief analysis of the security design, mostly in
the light of what was known when the design was completed in Fall
2002. It should be noted that this section does not present a proper
security analysis of the protocol; it merely discusses a few issues
that were known at the time the design was completed.

It should be kept in mind that the MIPv6 RO security design was never
intended to be fully secure. Instead, as we stated earlier, the goal
was to be roughly as secure as non-mobile IPv4 was known to be at the
time of the design. As it turns out, the result is slightly less
secure than IPv4, but the difference is small and most likely
insignificant in real life.

The known residual threats as compared with IPv4 are discussed in
Section 5.1. Considerations related to the application of IPsec to
authorize route optimization are discussed in Section 5.2. Section
5.3 discusses an attack against neighboring nodes. Finally, Section
5.4 deals with the special case of two mobile nodes conversing and
performing the route optimization procedure with each other.

5.1. Residual Threats as Compared to IPv4

   As we mentioned in Section 4.2, the lifetime of a binding represents
   a potential time shift in an attack. That is, an attacker that is
   able to create a false binding is able to reap the benefits of the
   binding as long as the binding lasts. Alternatively, the attacker is
   able to delay a return-to-home flooding attack (Section 3.2.2) until
   the binding expires. This is different from IPv4, where an attacker
   may continue an attack only as long as it is on the path between the
   two hosts.

   Since the binding lifetimes are severely restricted in the current
   design, the ability to do a time shifting attack is equivalently
   restricted.

   Threats possible because of the introduction of route optimization
   are, of course, not present in a baseline IPv4 internet (Section
   3.3). In particular, inducing unnecessary binding updates could
   potentially be a severe attack, but this would be most likely due to
   faulty implementations. As an extreme measure, a correspondent node
   can protect against these attacks by turning off route optimization.
   If so, it becomes obvious that the only residual attack against which
   there is no clear-cut prevention (other than its severe limitation as
   currently specified) is the time shifting attack mentioned above.

5.2. Interaction with IPsec

   A major motivation behind the current binding update design was
   scalability, which implied the ability to run the protocol without
   any existing security infrastructure. An alternative would have been
   to rely on existing trust relationships, perhaps in the form of a
   special-purpose Public Key Infrastructure in conjunction with IPsec.
   That would have limited scalability, making route optimization
   available only in environments where it is possible to create
   appropriate IPsec security associations between the mobile nodes and
   the corresponding nodes.

   There clearly are situations where there exists an appropriate
   relationship between a mobile node and the correspondent node. For
   example, if the correspondent node is a server that has pre-
   established keys with the mobile node, that would be the case.
   However, entity authentication or an authenticated session key is not
   necessarily sufficient for accepting Binding Updates.

   Home Address Check: If one wants to replace the home address check
      with cryptographic credentials, these must carry proper
      authorization for the specific home address, and care must be
      taken to make sure that the issuer of the certificate is entitled

to express such authorization. At the time of the design work,
the route optimization security design team was not aware of
standardized certificate formats to do this, although more recent
efforts within the IETF are addressing this issue. Note that
there is plenty of motivation to do so, as any pre-existing
relationship with a correspondent node would involve the mobile
node's home address (instead of any of its possible care-of
addresses). Accordingly, the IKE exchange would most naturally
run between the correspondent node and the mobile node's home
address. This still leaves open the issue of checking the mobile
node's care-of address.

Care-of Address Check: As for the care-of-address check, in
practice, it seems highly unlikely that nodes could completely
replace the care-of-address check with credentials. Since the
care-of addresses are ephemeral, in general it is very difficult
for a mobile node to present credentials that taken at face value
(by an arbitrary correspondent node) guarantee no misuse for, say,
flooding attacks (Section 3.2). As discussed before, a
reachability check goes a long way to alleviate such attacks.
Notice that, as part of the normal protocol exchange, establishing
IPsec security associations via IKE includes one such reachability
test. However, as per the previous section, the natural IKE
protocol exchange runs between the correspondent node and the
mobile node's home address. Hence, another reachability check is
needed to check the care-of address at which the node is currently
reachable. If this address changes, such a reachability test is
likewise necessary, and it is included in ongoing work aimed at
securely updating the node's current address.

Nevertheless, the Mobile IPv6 base specification [6] does not specify
how to use IPsec together with the mobility procedures between the
mobile node and correspondent node. On the other hand, the
specification is carefully written to allow the creation of the
binding management key Kbm through some different means.
Accordingly, where an appropriate relationship exists between a
mobile node and a correspondent node, the use of IPsec is possible,
and is, in fact, being pursued in more recent work.

5.3. Pretending to Be One's Neighbor

One possible attack against the security design is to pretend to be a
neighboring node. To launch this attack, the mobile node establishes
route optimization with some arbitrary correspondent node. While
performing the return routability tests and creating the binding
management key Kbm, the attacker uses its real home address but a
faked care-of address. Indeed, the care-of address would be the
address of the neighboring node on the local link. The attacker is

able to create the binding since it receives a valid Home Test
normally, and it is able to eavesdrop on the Care-of Test, as it
appears on the local link.

This attack would allow the mobile node to divert unwanted traffic
towards the neighboring node, resulting in an flooding attack.

However, this attack is not very serious in practice. First, it is
limited in the terms of location, since it is only possible against
neighbors. Second, the attack works also against the attacker, since
it shares the local link with the target. Third, a similar attack is
possible with Neighbor Discovery spoofing.

5.4. Two Mobile Nodes Talking to Each Other

When two mobile nodes want to establish route optimization with each
other, some care must be exercised in order not to reveal the reverse
tokens to an attacker. In this situation, both mobile nodes act
simultaneously in the mobile node and the correspondent node roles.
In the correspondent node role, the nodes are vulnerable to attackers
that are co-located at the same link. Such an attacker is able to
learn both the Home Test and Care-of Test sent by the mobile node,
and therefore it is able to spoof the location of the other mobile
host to the neighboring one. What is worse is that the attacker can
obtain a valid Care-of Test itself, combine it with the Home Test,
and then claim to the neighboring node that the other node has just
arrived at the same link.

There is an easy way to avoid this attack. In the correspondent node
role, the mobile node should tunnel the Home Test messages that it
sends through its home agent. This prevents the co-located attacker
from learning any valid Home Test messages.

6. Conclusions

This document discussed the security design rationale for the Mobile
IPv6 Route Optimization. We have tried to describe the dangers
created by Mobile IP Route Optimization, the security goals and
background of the design, and the actual mechanisms employed.

We started the discussion with a background tour to the IP routing
architecture the definition of the mobility problem. After that, we
covered the avenues of attack: the targets, the time shifting
abilities, and the possible locations of an attacker. We outlined a
number of identified threat scenarios, and discussed how they are
mitigated in the current design. Finally, in Section 4 we gave an
overview of the actual mechanisms employed, and the rational behind
them.

As far as we know today, the only significant difference between the
security of an IPv4 Internet and that of an Internet with Mobile IPv6
(and route optimization) concerns time shifting attacks.
Nevertheless, these are severely restricted in the current design.

We have also briefly covered some of the known subtleties and
shortcomings, but that discussion cannot be exhaustive. It is quite
probable that new subtle problems will be discovered with the design.
As a consequence, it is most likely that the design needs to be
revised in the light of experience and insight.

7. Acknowledgements

We are grateful for: Hesham Soliman for reminding us about the threat
explained in Section 5.3, Francis Dupont for first discussing the
case of two mobile nodes talking to each other (Section 5.4) and for
sundry other comments, Pekka Savola for his help in Section 1.1.1,
and Elwyn Davies for his thorough editorial review.

8. Informative References

[1]   Aura, T., Roe, M., and J. Arkko, "Security of Internet Location
       Management", Proc. 18th Annual Computer Security Applications
       Conference, pages 78-87, Las Vegas, NV, USA, IEEE Press,
       December 2002.

[2]   Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery
       for IP Version 6 (IPv6)", RFC 2461, December 1998.

[3]   Narten, T. and R. Draves, "Privacy Extensions for Stateless
       Address Autoconfiguration in IPv6", RFC 3041, January 2001.

[4]   Bush, R. and D. Meyer, "Some Internet Architectural Guidelines
       and Philosophy", RFC 3439, December 2002.

[5]   Baker, F. and P. Savola, "Ingress Filtering for Multihomed
       Networks", BCP 84, RFC 3704, March 2004.

[6]   Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in
       IPv6", RFC 3775, June 2004.

[7]   Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to
       Protect Mobile IPv6 Signaling Between Mobile Nodes and Home
       Agents", RFC 3776, June 2004.

[8]   Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose,
       "DNS Security Introduction and Requirements", RFC 4033, March
       2005.

   [9]   Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose,
         "Resource Records for the DNS Security Extensions", RFC 4034,
         March 2005.

   [10] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose,
         "Protocol Modifications for the DNS Security Extensions", RFC
         4035, March 2005.

   [11] Chiappa, J., "Will The Real 'End-End Principle' Please Stand
         Up?", Private Communication, April 2002.

   [12] Savage, S., Cardwell, N., Wetherall, D., and T. Anderson, "TCP
         Congestion Control with a Misbehaving Receiver", ACM Computer
         Communication Review, 29:5, October 1999.

   [13] Nikander, P., "Denial-of-Service, Address Ownership, and Early
         Authentication in the IPv6 World", Security Protocols 9th
         International Workshop, Cambridge, UK, April 25-27 2001, LNCS
         2467, pages 12-26, Springer, 2002.

   [14] Chiappa, J., "Endpoints and Endpoint Names: A Proposed
         Enhancement to the Internet Architecture", Private
         Communication, 1999.

   [15] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure
         Neighbor Discovery (SEND)", RFC 3971, March 2005.

Authors' Addresses

    Pekka Nikander
    Ericsson Research NomadicLab
    JORVAS   FIN-02420
    FINLAND

    Phone: +358 9 299 1
    EMail: pekka.nikander@nomadiclab.com

    Jari Arkko
    Ericsson Research NomadicLab
    JORVAS   FIN-02420
    FINLAND

    EMail: jari.arkko@ericsson.com

    Tuomas Aura
    Microsoft Research Ltd.
    Roger Needham Building
    7   JJ Thomson Avenue
    Cambridge CB3 0FB
    United Kingdom

    EMail: Tuomaura@microsoft.com

    Gabriel Montenegro
    Microsoft Corporation
    One Microsoft Way
    Redmond, WA   98052
    USA

    EMail: gabriel_montenegro_2000@yahoo.com

    Erik Nordmark
    Sun Microsystems
    17 Network Circle
    Menlo Park, CA 94025
    USA

    EMail: erik.nordmark@sun.com

Full Copyright Statement

    Copyright (C) The Internet Society (2005).

    This document is subject to the rights, licenses and restrictions
    contained in [BCP 78](#), and except as set forth therein, the authors
    retain all their rights.

    This document and the information contained herein are provided on an
    "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS
    OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET
    ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED,
    INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
    INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
    WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

    The IETF takes no position regarding the validity or scope of any
    Intellectual Property Rights or other rights that might be claimed to
    pertain to the implementation or use of the technology described in
    this document or the extent to which any license under such rights
    might or might not be available; nor does it represent that it has
    made any independent effort to identify any such rights. Information
    on the procedures with respect to rights in RFC documents can be
    found in [BCP 78](#) and [BCP 79](#).

    Copies of IPR disclosures made to the IETF Secretariat and any
    assurances of licenses to be made available, or the result of an
    attempt made to obtain a general license or permission for the use of
    such proprietary rights by implementers or users of this
    specification can be obtained from the IETF on-line IPR repository at
    http://www.ietf.org/ipr.

    The IETF invites any interested party to bring to its attention any
    copyrights, patents or patent applications, or other proprietary
    rights that may cover technology that may be required to implement
    this standard. Please address the information to the IETF at ietf-
    ipr@ietf.org.

Acknowledgement

    Funding for the RFC Editor function is currently provided by the
    Internet Society.