Designing the Mobile IPv6 Security Protocol

Tuomas Aura, Michael Roe

April 2006

Technical Report
MSR ïTR ï2006 ï42

Microsoft Research
Roger Needham Building
7 J.J. Thomson Avenue
Cambridge, CB3 0FB
United Kingdom

This article will appear in Annales des télécommunications / Annals of telecommunications, Vol. 61 n°3-4, March-April 2006, Network and information systems security. Editors Frédéric Cuppens, Hervé Debar, and Elisa Bertino. Copyright 2006 Groupe des Ecoles des Telecommunication and Hermes Science Publications.

Designing the Mobile IPv6 Security Protocol

Tuomas Aura, Michael Roe
Microsoft Research
Roger Needham Building, 7 JJ Thomson Avenue, Cambridge, CB3 0FB, UK
{tuomaura,mroe}@microsoft.com

Abstract Mobile IPv6 is a network-layer mobility protocol for the IPv6 Internet. The protocol includes several security mechanisms, such as the return-routability tests for the mobileôs home address and care-of addresses. This paper explains the threat model and design principles that motivated the Mobile IPv6 security features. While many of the ideas have become parts of the standard toolkit for designing Internet mobility protocols, some details of the reasoning have not been previously documented.

Keywords: network security, authentication, denial of service, mobility protocol, IPv6, location management

I. Introduction

This paper describes the Mobile IPv6 security protocol. The focus is on the authentication of binding updates, i.e., location information sent by the mobile to its correspondents. We explain the security threats created by the introduction of mobility and the mechanisms that have been used to prevent the attacks. The protocol design is unusual and it would not be considered secure by the measures of traditional security protocol analysis. The security of the protocol depends on the partial reliability of the Internet routing infrastructure. The reason for using such techniques is that the protocol must work between any mobile node and any other Internet node even if they have no previous relationship, and we cannot assume the existence of a PKI or other global security infrastructure. On the other hand, the only security requirement was to counter any new threats created by mobility. The protocol does exactly that. This is a pragmatic way of thinking when introducing new technology such as mobility.

I.1 Related work

Mobile IPv6 is an IP-layer mobility protocol for the IPv6 Internet. It has been standardized by the IETF and published as the Request for Comments 3775 [14]. The design was based on the Mobile IP for IPv4 [27]. The Mobile IPv4 protocol follows the design principles outlined first by Ioannidis [13]: mobility is implemented in the network layer in such a way that it is transparent to the higher layers, mobile hosts retain their IP addresses over location changes, and the non-mobile host need not know about the mobility protocol. The main difference between Mobile IPv4 and Mobile IPv6 is that, in the latter, mobile hosts can perform mobility signaling directly with non-mobile correspondents. This enables more efficient routing of data to and from the mobile. Similar route optimization has been proposed for Mobile IPv4, as have been many other enhancements to both protocols. In this paper, however, we focus on the standard Mobile IPv6 protocol.

The Mobile IPv6 standardization process started in 1995. After about 3 years, the mobility mechanism itself had been fully specified, including an efficient and elegant protocol for location updates that enabled the optimized routing. Around this time, it became apparent that spoofed location updates posed a new security risk that had not been sufficiently taken into account in the design. Moreover, the lack of a global authentication infrastructure made it impossible to solve the problems with straightforward application of standard Internet security protocols, such as IPsec and IKE. As a result, the standardization was delayed.

The deadlock in the Mobile IPv6 standards process sparked a search for new types of security solutions that do not require special security infrastructure, possibly at the expense of providing slightly lower levels of security than is usually required from security protocols. Some new ideas were suggested in the CAM protocol by OôShea and Roe [25] and the BAKE protocol by Nikander and Perkins [22]. A threat analysis by Aura and Arkko [5] proved these to be insufficient. Based on the threat analysis, the current authors, together with O'Shea and Arkko, designed a family of location-update protocols [28] borrowing ideas from and improving on CAM and BAKE. The return-routability protocol that is now a part of the Mobile IPv6 standard appeared first as a part of this protocol family. The solution enabled the Mobile IPv6 standardization process to continue. Many details of the protocol were further refined by the IETF working group.

During the threat analysis, we discovered new attacks and introduced new defense mechanisms that have since been adopted by other mobility protocols. In particular, we identified a previously unknown class of attacks where the malicious node floods a victim with unwanted packets by using the location update protocol to redirect a data stream towards them. Similar attacks have later been found against other location-update and multi-addressing protocols [7]. The return-routability protocol has proven to be a general solution for this threat.

The protocol described in this paper is a slightly simplified version of the actual Mobile IPv6 protocol. We concentrate on the abstract design principles and avoid discussing packet formats, protocol state machines, and implementation details. This article is based in part on conference papers by Aura et al. [4][8]. Another paper by Nikander et al. [23] discusses the protocol architecture. A more detailed treatment of the security protocol is provided by Kempf et al. [16] and a readable explanation of the entire Mobile IPv6 standard by Soliman [30].

The paper is organized roughly to follow the design process. We introduce the Mobile IPv6 tunneling protocol and route optimization in Section II. Section III describes the basic binding-update authentication protocol. Section IV explains how even authenticated binding updates can be used for denial-of-service attacks and how these

BA	Binding acknowledgement	HoT	Home address test
BU	Binding update	HoTI	Home address test init
CN	Correspondent node	Kbm	Binding management key
CoA	Care-of address	MAC	Message authentication code
CoT	Care-of address test	MIPv6	Mobile IPv6
CoTI	Care-of address test init	MN	Mobile node
HA	Home agent	RH	Routing Header
HAO	Home address option	RO	Route optimization
HoA	Home address	RR	Return routability

Table I Mobile IPv6 acronyms

attacks are prevented. Section V considers some less serious threats and how the protocol was enhanced to mitigate them. Section VI concludes the paper.

II. Mobile IPv6

The basic idea behind Mobile IP is that, if mobility is implemented in the network layer, it needs to be implemented only once and will then be transparently available for all higher-layer protocols. It remains to be seen how well this promise is fulfilled in practice. There are, however, some applications like mobile VPN access, for which Mobile IP is obviously a good match. This section describes the Mobile IPv6 architecture and protocol in its early form, before the threat analysis and the addition of the security mechanisms. The later sections in the paper will cover the threats and security enhancements. The reader might want to take a peek at the final protocol in Figure 7. The protocol specification makes heavy use of acronyms, which have been collected in Table I.

II.1 Mobile network architecture

IP addresses serve a dual purpose in the Internet: they are used both to identify IP nodes and to route IP packets to them [17]. In particular, IPv6 addresses consist of two parts, a 64-bit subnet prefix and a 64-bit interface identifier. The subnet prefix is determined by the location (i.e., subnet) of the node in the IP routing infrastructure while the interface identifier distinguishes individual nodes within the same subnet. Together, the two halves of the address provide a globally unique identifier and a globally routable address for the IP node.

IP mobility means that an Internet node moves from one location in the IP routing infrastructure to another, either because it moves physically between network coverage areas or media types, or because its logical point of network access changes. The change in location implies a change in the subnet prefix and, therefore, in the mobile nodeôs IP address. This creates two kinds of problem: Firstly, existing connections, such as TCP connections and IPsec security associations, between the mobile and other hosts become invalid when the address (and, thus, identifier) of one endpoint changes. Secondly, the mobile is no longer reachable at its old address for new connections. The former problem is important for stateful protocols, but has less effect on stateless protocols such as HTTP. The latter problem typically concerns servers but not client computers, although instant messaging and voice over IP (VoIP) are making reachability more important for all computers. Mobile IPv6 aims to solve both kinds of problems created by mobility: all transport-layer and higher-layer connections and security associations between the mobile and its correspondents should survive the address change, and the mobile host should be reachable as long as it is connected to the Internet somewhere in the world.

A major assumption made in Mobile IPv6 is that every *mobile node (MN)* has a *home network*, i.e., a subnet where it has a permanent *home address (HoA)*. The home network can also provide infrastructure for implementing the mobility. This assumption has its origins in the time when mobility was an exception and most IP nodes were stationary. If one were to design a mobility protocol from scratch today, it almost certainly would not make such an assumption. In any case, Mobile IP solves the reachability problem by ensuring that the mobile is always able to receive packets sent to its home address.

The long-term contract between the mobile and its home network implies a mutual trust relationship that can be exploited in the security solution. Indeed, most proposed security protocols for Mobile IPv6, including the one in the standard, depend on the special relationship between the home network and the mobile. It is a completely open question, and not discussed further in this paper, what kind of security mechanisms would be needed if the home agent did not trust the information provided by the mobile. On the other hand, a *correspondent node (CN)* that communicates with the mobile can be any Internet node and it is not assumed to have any pre-existing relation with the mobile or its home.

II.2 Mobility protocol

Mobile IPv6 depends on two fundamental techniques for implementing mobility that are best understood by comparing them to the postal service. First, one can arrange traffic to be forwarded from a permanent address to a temporary one. Second, one can notify the correspondents about the address change. Forwarding has the advantage of being transparent to the correspondents while notifications result in direct and, thus, more efficient delivery of packets. There are other mobility techniques, such as directory lookups and dynamic routing, but these are not used in Mobile IPv6.

The transparent mode of Mobile IPv6 operation is shown in Figure 1(a). A router called the *home agent (HA)* at the home network acts as the mobile's trusted agent and forwards IP packets between the mobile's home network and its current location, called the *care-of address (CoA)*. The home agent intercepts packets sent by correspondents to the HoA and forwards them to the CoA over an IPIP tunnel, i.e., encapsulated in another IP packet. When the mobile wants to send packets to a correspondent, it sends them to the home agent over the reverse tunnel. The home agent decapsulates the packets and forwards them to the correspondent.

When the mobile moves to a new location, it tells the home agent its new care-of address by sending a *binding update (BU)* message. The binding update causes the

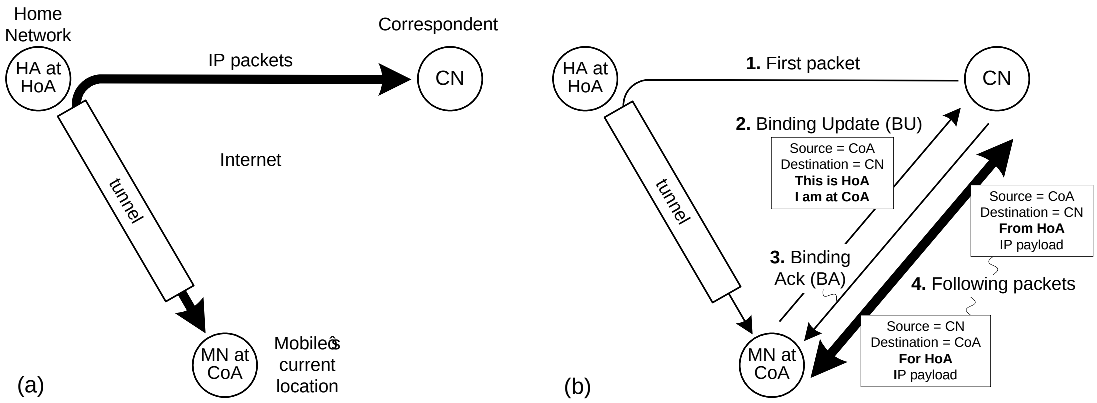

Figure 1 Mobile IPv6 tunneling (a) and route optimization (b)

home agent to update the IPIP tunnel in such a way that the tunneled packets are routed to and from the new CoA. The binding update and the following *binding acknowledgement (BA)* are authenticated using a preconfigured IPsec security association between the mobile and the home agent. This is the first point where we depend on the long-term trust relationship between the mobile and its home network. There are many possible ways of establishing the security association, such as IKE with certificate authentication [10]. In this paper, we assume that the security association exists and that the signaling messages sent between the mobile and its home agent are both authenticated and encrypted. The simplest way to implement this is to protect all packets between the HA and MN with tunnel-mode IPsec, including both the signaling and the tunneled data. To simplify the discussion in the following, we assume that all data between the HA and MN gets the same IPsec protection.

The bidirectional tunneling is sufficient to both enable reachability and to prevent the breaking of connections when the mobile moves. The routing is, however, far from optimal. Packets between the mobile and its correspondents have to travel via the home network, which may be far away. To rectify this problem, Mobile IPv6 defines a mechanism called *route optimization (RO)*. The optimization requires changes to the correspondent. It is optional to implement, although most non-mobile nodes are expected to eventually support it.

The route optimization protocol shown in Figure 1(b) also uses the binding update and binding acknowledgement messages. When the mobile changes its current address, it sends BUs to its correspondents to notify them about the new location. The binding update contains the mobile's home address and current care-of address. The correspondent acknowledges the binding update and stores the location information in a *binding cache*, which is effectively a routing table: it tells that packets destined to the HoA should instead be sent to the CoA. The binding needs to be refreshed every few minutes by sending a new BU even if the mobile stays at the same CoA. If the cache entry expires or if it is explicitly deleted (by sending a BU with zero lifetime), the correspondent reverts to sending packets to the mobileôs home address.It also stops accepting packets directly from the CoA. Note that we assume, for now, that the BU to the correspondent is sent unauthenticated. (Section III is dedicated entirely to discussing the need and mechanisms for the authentication.)

While the first binding update may be sent at any time, it is usually triggered either when the mobile has data to send to a new correspondent, or when the mobile receives a tunneled packet from a new correspondent. In practice, this means that if the correspondent supports Mobile IPv6, only one packet from the correspondent to the MN (often, a TCP SYN) is sent via the unoptimized route. After the binding has been created, the mobile and the correspondent can communicate directly.

The direct packets from the mobile to the correspondent have a header field called the *home-address destination option (HAO)*, which contains the HoA. The packets from the correspondent to the mobile contain the HoA in a type-2 *routing header (RH)*. When a correspondent node is sending a packet, it compares the destination address against the home addresses in its binding cache. If a binding exists, it replaces the destination IP address with the CoA and inserts the RH after the IP header. The mobile, receiving the packet, copies the HoA from the RH back into the destination address field and removes the RH, thus re-creating the original packet. Similarly, a mobile that is about to send a packet to a correspondent uses the CoA as the source IP address and inserts the HAO. When the correspondent receives the packet, it overwrites the source address with the HoA from the HAO, again re-creating the original packet. This way, the mobility is transparent to the upper protocol layers, including IPsec and the transport layer. The only address they see for the mobile is the HoA.

II.3 Design choices

Route optimization is voluntary in the sense that either the mobile or the correspondent can refuse to do it, in which case they continue to communicate via the home agent. The existence of the home agent guarantees that all IP nodes, including ones that do not support Mobile IPv6, can correspond with the mobile. An alternative would have been to make mobility support mandatory in all IPv6 nodes. In that case, the protocol could have been designed without home agents, or in such a way that a mobile node and its correspondent could continue to communicate when the HA was unreachable.

Mobile IP preserves the dual use of home addresses. The home address is an identifier for the mobile, as well as a routable location to which correspondents can send packets. The care-of address, on the other hand, is a pure location and serves no identification purpose.

In Mobile IPv6, any IPv6 address can be or become mobile and there is no way to distinguish a mobile node from a stationary one just by looking at its address. A mobile node can also return to its home network and become a stationary node. In retrospect, the protocol would have been significantly simpler if the home addresses had been allocated from a special class of IP addresses and if the home networks were logical networks to which the mobile could never physically return.

It is an important design consideration that all packets in the protocol are sent with topologically correct source IP addresses. That is, the source address of the outmost IP header always belongs to the subnet from which the packet is sent. This ensures that packets are never dropped by ingress filtering. The HAO and RH are, in a sense, degenerate tunnel headers. The Mobile IPv6 designers could have chosen to encapsulate the packets into a full IP header (i.e., IPIP tunnel) but that would have resulted in some redundant header fields. The HAO and RH are sufficient to convey all the information that is needed for the optimized routing.

One of the fundamental goals of Mobile IPv6 is to operate transparently to higher protocol layers. The advantage is that any upper-layer protocol that works in the stationary IPv6 network will work over Mobile IPv6 without modifications. There are, however, arguments why transport-layer and security protocols should be mobility-aware. We will point out in Section IV.4 that some security mechanism required by Mobile IPv6 would fit more naturally into the transport layer. Also, IPsec and Mobile IPv6 suffer from a kind of chicken-and-egg problem: On one hand, IPsec works best over Mobile IPv6 because the security associations are indexed by the endpoint IP addresses and Mobile IPv6 hides the address changes from IPsec. On the other hand, Mobile IPv6 depends on the IPsec tunnel between the mobile and its home agent. This means that mobility for the HA-MN tunnel has to be implemented as a special case. Therefore, it would be architecturally appealing to integrate mobility and security into one protocol, as has been done, for example, in the Host Identity Protocol (HIP) [24].

It should be noted that there are several alternative approaches to Internet mobility and that the attacks and protection mechanisms identified in this paper are general enough to be applicable to many such mechanisms. It is beyond the scope of this work to compare the relative merits of the alternative mobility protocols.

Another issue that we do not address in this paper is location privacy. Mobile IPv6 does nothing special to try to hide the mobile's home address or current location from others. Nevertheless, the protocol is relatively privacy-friendly: the mobile's current location is tracked by its own home agent but not by any global or centralized directory, the mobile is free to use temporary and multiple home addresses, and sending BUs to correspondents is a voluntary optimization for the mobile. That is, the MN can conceal its location from the CN by turning off the route optimization.

III. BU authentication

The binding update protocol, if implemented as described in the previous section, would create serious new security vulnerabilities. The binding updates are not authenticated and, therefore, can be spoofed. Unauthenticated location information makes it possible for an attacker to misinform correspondents about the mobile's location and, thus, to redirect packets intended for the mobile to a wrong destination. This can lead to the compromise of secrecy and integrity as well as to denial-of-service because the target nodes are unable to communicate. This section describes the basic attacks using unauthentic BUs and the potential BU authentication mechanisms.

The authentication mechanism that was selected for the Mobile IPv6 standard will be presented in Section III.4. We introduce a relatively weak routing-based authentication method that traditional network security thinking would consider to be insecure. Nevertheless, it provides an acceptable level of assurance in real networks and can complement or even replace the stronger methods. Instead of trying to prevent all attacks, the best strategy is often to limit the number of potential attackers that can attack a particular target, and to reduce the number of targets a potential attacker can threaten.

III.1 Connection hijacking

The connection-hijacking attack is shown in Figure 2. A, B and C are IPv6 addresses. The Internet nodes A and B are honest and communicating with each other. An attacker at the address C sends a false binding update to B, claiming to be a mobile with the home address A. If B, acting in the role of a correspondent, believes the binding update and creates a binding, it will redirect to C all packets that are intended for A. Thus, the attacker can intercept packets sent by B to A. The attacker can also spoof data packets from A by inserting a false home-address option into them. This way, it can hijack existing connections between A and B, and open new ones pretending to be A. The attacker can also redirect the packets to a random or non-existent care-of address in order to disrupt the communication between the honest nodes. It has to send a new binding update every few minutes to refresh the binding cache entry at the correspondent.

End-to-end encryption and integrity protection of payload data, e.g., with authenticated SSL or IPsec, can prevent the attacks against data secrecy and integrity but not denial-of service. This is because the binding update and route optimization happen transparently to IPsec and SSL. The attacker is able to redirect the encrypted data even though it cannot read it.

These attacks are serious because A, B and C can be any IPv6 addresses anywhere on the Internet. The only limitations are that the node at B must support route optimization and that the attacker needs to know the IPv6 addresses of A and B. Since there is no visible difference between a mobile home address and a stationary IPv6 address, the node at A can be stationary as well as mobile. The possibility of these attacks caused IETF to halt the Mobile IPv6 standardization process until a solution for authenticating the binding updates was found. It is believed that deployment of the protocol without security could have resulted in a break-down of the entire Internet.

In order to send false BUs, the attacker needs to know the IP addresses of both the communicating nodes. This means that nodes that have well-known or permanent addresses, such as public servers and those using stateless auto-configuration [31], are most vulnerable. They include nodes that are a part of the network infrastructure, such as DNS servers, which are particularly interesting targets for DoS attacks. Frequently changing random addresses, e.g., ones created using IPv6 addressing privacy [19], may mitigate the risks to some extent.

We have considered only active attackers because, in order to redirect packets, the attacker must sooner or later send one or more messages. In fact, the active attacks are easier for the average attacker than passive ones would be. In most active attacks, the attacker can initiate the BU protocol execution at any time while passive attacks would require the attacker to wait for suitable messages to be sent by the target nodes.

The vulnerability is, to some extent, a side effect of the effort to make mobility transparent. It can also be seen as a consequence of the desire to keep the binding update protocol simple and efficient. The addition of security mechanisms unavoidably makes the protocol slower and more complex.

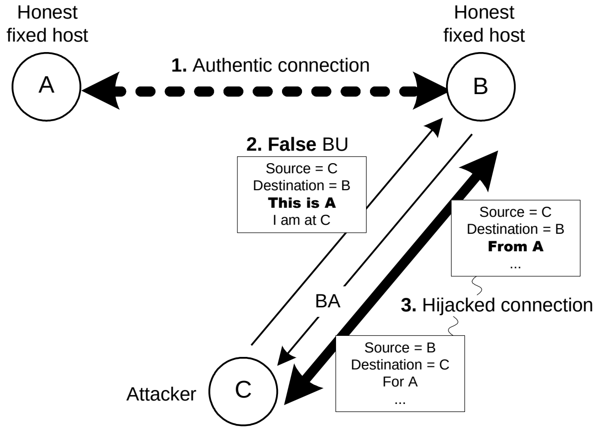

Figure 2 False binding update

III.2 Need for infrastructureless authentication

The obvious solution to the BU spoofing is to authenticate the binding updates. A typical authentication system would use a suite of strong cryptographic authentication protocols and a certification infrastructure, such as IPsec, IKE and an X.509-based PKI. The problem is that the authentication needs to work between any mobile Internet node and any correspondent. There does not currently exist any infrastructure that could be used to authenticate all IPv6 nodes. Neither is it realistic to suggest creating such a service for the needs of Mobile IPv6. This means that using the conventional authentication mechanism would confine route optimization to intra-organizational use where the required security services are in place. Moreover, the generic authentication protocols have usually been designed with general-purpose computers and application-level security requirements in mind. The overhead of these protocols can be too high for low-end mobile devices and for a network-layer signaling protocol. (There are, nevertheless, some situations where it is possible, and advisable, to apply the strong generic authentication solutions. In closed user groups and high-security environments, it may be possible to set up a PKI and to require the BU to be strongly authenticated between the group members.)

For the above reasons, we were forced to consider unconventional authentication methods that work without special security infrastructure. The advantage we had on our side is that the security requirements for BU authentication are unusually weak. The stated goal in the IETF working group was that the Mobile IPv6 protocol should be *at least as secure as the current non-mobile IPv4 Internet*. This means that we were not confined to designing a traditional strong security protocol. Our ambition was limited to making sure that Mobile IPv6 does not introduce any new major vulnerability to the Internet. The goal was not to create a strong general-purpose authentication protocol.

As mentioned earlier, the IP layer provides two kinds of services. First, the addressing architecture [11] provides Internet nodes with globally unique IPv6 addresses. Second, the routing infrastructure [12] delivers packets across the Internet to their destination address. It turns out that both the addressing and the routing can be used to bootstrap some form of authentication (see Sections III.3 and III.4). Although the authentication is not necessarily as strong as a PKI would enable in closed networks, it is, nevertheless, better than no authentication. Since these techniques do not require any special security infrastructure, they are called *infrastructureless authentication*. This label is somewhat inaccurate because, as explained above, the mechanisms depend on the existence of fundamental parts of the network infrastructure. The same security mechanisms are sometimes called *weak* because they do not satisfy the strict criteria the security community associates with strong authentication. A more general discussion of weak authentication can be found in [1].

III.3 Cryptographically Generated Addresses

There is a technique for the authentication of IPv6 addresses that provides an intermediate level of security below strong public-key authentication but above no authentication. The idea, first introduced in a BU authentication protocol CAM [25], is to select the least significant 64 bits of the IP address (the interface identifier) by computing a 64-bit one-way hash of the node's public signature key. The node signs its location information with the corresponding private key and sends the public key along with the signed data. The recipient hashes the public key and compares the hash to the address before verifying the signature on the location data. This prevents anyone except the node itself from sending location updates for its address. The attraction of this technique is that it provides public-key authentication of the IP address without any trusted third parties or PKI.

Several BU authentication protocols were proposed based on this idea [28][20][18]. While the authentication of the sender's IPv6 address would be of little value in most applications, it is exactly what is needed to authorize the binding update. The mobile signs the binding update and attaches its public key to the message. The correspondent can verify without any additional infrastructure that the binding update was signed by the owner of the home address. Nevertheless, this mechanism was rejected by the Mobile IPv6 designers in favor of an even simpler routing-based protocol, which will be covered in detail in the rest of the paper. (Our original protocol family [28] supported both types of infrastructureless authentication.) The addresses with an embedded public-key hash have since been standardized under the name *cryptographically generated addresses (CGA)* [2][3] for use in other security protocols.

III.4 Return routability test

The second infrastructureless authentication method is based on the fact that routing in the Internet is semi-reliable. It is difficult for a remote attacker to change the route of packets that do not travel via the attacker's network. Thus, in order to sniff or intercept a packet, the attacker needs to be on its route.

The first version of the routing-based BU authentication protocol is shown in Figure 3. The idea is that, after the mobile initiates the BU protocol (message 1), the correspondent sends a secret value as plaintext to the mobile's home address (message 2). The home agent intercepts the message and forwards it to the mobile via a secure tunnel. The mobile then uses the secret to compute a message authentication code for the binding update (message 3). This mechanism is called the *return-routability test for the home address (RR for HoA)* because the mobile must return to the correspondent (a function of) a secret value sent by the correspondent to the HoA. In effect, the correspondent verifies that the mobile is able to receive messages at the home address.

In the Mobile IPv6 standard terminology, message 2 is called the *home test message (HoT)* and the secret value is the *home keygen token*. Messages 3 and 4 are simply called the *binding update (BU)* and *binding acknowledgement (BA)*.

In order to break the protocol, the attacker needs to be on the route between the correspondent and the home agent. If it is on that route, it can intercept the HoT message and learn the secret that is necessary for spoofing the BU. Thus, the protocol is not secure against the standard network-security attacker model where the attacker can sniff and intercept all messages on the network. It is natural that most readers previously unfamiliar with the protocol will at this point object to the idea of sending a key in plaintext. There are, nevertheless, strong arguments in favor of the design.

First, the number of potential attackers and targets is dramatically reduced. Without authentication, any Internet node (e.g., C in Figure 2) could spoof binding updates to hijack connections between any two Internet nodes (A and B in Figure 2). In our new protocol, the attacker must be on the route of the hijacked connection (on the A-B route in Figure 2). There are typically only tens or hundreds nodes on this route, including both routers and hosts on the local networks of the end nodes. Compared with the situation where any malicious Internet node can mount an attack, it is much less likely that one of this limited set of nodes would do so. The fewer and the more local the potential attack sources are, the easier it is also to gain control of them if attacks sometimes occur. Another way to explain the same idea is that a malicious Internet node is able to target only the connections that pass though its local network. For a typical attacker, such as a compromised router or host, the number of such connections is small. This reduction in the scale of the potential damage alone means that deployment of the Mobile IPv6 would no longer be a danger to the Internet's stability.

Second, the protocol fulfills the explicit design goal of being as secure as the current Internet without mobility. Assume that the mobile node never leaves its home network and always communicates directly from the home address. In that case, an attacker on the route between the home address and the correspondent can spoof, intercept and sniff packets between them, and it can execute all the same attacks that are possible by exploiting the weaknesses of our BU authentication protocol. Therefore, we argue that the simple protocol of Figure 3 is sufficient for authenticating the sender of a binding update.

It should be noted that although the Mobile IPv6 standard does not absolutely require encryption of the HoT message from the HA to the mobile, the security of the BU authentication depends crucially on the secrecy of the HoT message on that path. If the tunnel is not encrypted, nodes in the local network of the CoA can intercept the home keygen token and, thus, can spoof a binding update. This would create a major vulnerability for mobiles that roam in untrusted access networks. Throughout this paper, we assume that all signaling messages between the mobile and its HA are

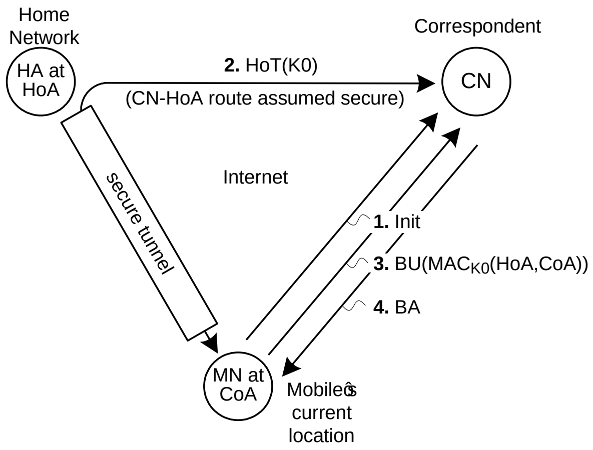

Figure 3 Return-routability test for HoA

encrypted and authenticated. (The reason for why the encryption is optional in the standard is that omitting it will make only the particular mobile node, and not the entire Internet, vulnerable to attacks. Thus, it can be left to the mobile and its home agent to decide whether they want the security.)

To summarize, the RR protocol protects against the spoofing of binding updates from the MN to the CN, except if the attacker is located on the CN-HoA route or at the local network of the CN. An attacker anywhere else in the Internet, for example, in the local network of the CoA, cannot spoof the binding updates.

III.5 The two-mobile case

The main remaining vulnerability in the RR protocol is that attackers in the same local network as the correspondent may be able to intercept and spoof all of the BU protocol messages. In particular, if the correspondent is a wireless node at an untrusted access network, it may be easy for an attacker to capture the keygen tokens and to spoof binding updates. This is, in fact, a quite likely scenario because the Mobile IPv6 protocol was specifically designed to support communication between two mobile nodes, i.e., a situation where each MN is the CN for the other.

This vulnerability can be mitigated if the correspondent is also a Mobile IPv6 mobile node. In that case, the correspondent should not send the HoT message directly to the other mobileôs HoA. Instead, it should tunnel the HoT via its own home agent.This prevents an attacker at the correspondentôs local network from sniffing the home keygen token. The other mobile needs to do the same in the opposite direction. That way, the two communicating mobiles can securely optimize the routing between their care-of addresses regardless of any potential attacker on the current access networks.

III.6 Some unsuccessful ideas

It is interesting to note how we originally arrived at the RR test idea. We were analyzing various proposals (e.g., BAKE [22]) for the BU authentication protocol.

The working hypothesis was that the CGA-based protocols were the only ones that provided some security without requiring a PKI or other global security infrastructure. This proved true in the sense that we could show all the other schemes to be no more secure than sending a key in plaintext, which for a security protocol is normally a death sentence. Figure 3 was first drawn to illustrate this failure. But, looking at the protocol in its most simple form, we could not help observing that it did, against our expectations, provide some security. This lead to the arguments outlined above.

BAKE and some other proposals were based on the idea of sending two secret values along two independent routes and hoping that the attacker is unable to sniff both. One would be tunneled to the mobile via the HoA and the other routed directly, thus exploiting the fact that the HA, MN and CN form a triangle. While this is an appealing idea in general, it does not work in Mobile IPv6. The reason is that although the routes usually form a triangle with two independent paths, a false mobile (i.e., the attacker) may be located so that the routes overlap. Specifically, if the attacker is located on the route between the CN and HoA, it can use its own address as the HoA. This causes both secret values to be sent via the attackerôs location. Thus, the two-secret protocol is no more secure than the protocol of Figure 3, which is also vulnerable to attackers on the CN-HoA route.

Another idea is the so called leap-of-faith authentication where the mobile sends a session key insecurely to the correspondent at the beginning of their correspondence and the key is used to authenticate subsequent messages. This does not work for BU authentication (unless they key is sent in a way that takes advantage of a relatively safe route) because the attacker can send its false key before the authentic mobile sends the authentic key. Furthermore, there must be a recovery mechanism for situations where the mobile or the correspondent loses its state, and the attacker can exploit this mechanism.

The leap-of-faith authentication is suitable for situations where a human user, or some other factor outside the attacker's control, at random times initiates the protocol execution. Perhaps the most successful use of the technique is in the secure shell (SSH) [32]. The party making the leap must always be the one that initiates the protocol. In such situations, it may be reasonable to argue that an attacker is unlikely to be present at the time of the unauthenticated key exchange. In BU authentication, the protocol is usually initiated by the mobile but the leap in faith should be made by the correspondent. Also, the attacker can trigger the BU protocol at any time by sending to the mobile's home address a spoofed packet that appears to come from the correspondent.

Ingress filtering [9] is another way of limiting the number of potential attackers and their targets. Ingress filtering means that a gateway router or firewall checks the source addresses of all packets that leave a local network and enter the Internet and drops ones that do not originate from the local network. This prevents nodes on the network from sending spoofed packets that appear to come from other networks. Since the mobile's new address in a Mobile IPv6 binding update is usually sent in the source address field of the IP packet header, ingress filtering seems to limit the choice of false addresses. There are, however, two weaknesses in this thinking. Firstly, there is the usual argument against ingress filtering: to be effective, it must be applied on the attacker's local network. Thus, it cannot be used by the attack targets to protect themselves. Secondly, Mobile IPv6 specifies a mechanism (Alternative Care-of Address sub-option) that can be used for sending a false care-of address without source spoofing. We conclude that, while it otherwise is advisable to apply ingress filtering, we cannot rely on this to stop the attacks against Mobile IPv6.

Any authentication protocol has to take into account replay attacks. The RR protocol uses a nonce (the home keygen token) to verify the freshness of the BU. Since the token is needed for the RR test anyway, it is appropriate to depend on it for freshness as well. Time stamps would be much more problematic because mobile devices may not be able to maintain sufficiently accurate clocks. Sequence-numbered BUs, on the other hand, could be intercepted and delayed for later attacks.

IV. Verifying the current location

The protocol described above is sufficient to authenticate the sender of the binding update and, thus, solves the problem that we originally set out to solve. There is, however, another major attack that exploits the binding updates and that is not prevented by the authentication. Even authenticated BUs can be used to redirect data to the target of a packet flooding attack. This section explains the attack in detail and the mechanism for preventing it.

IV.1 Bombing Attacks

The key observation is that a binding update contains two pieces of information, the HoA and CoA. The BU authentication provides a level of assurance that the HoA is not spoofed. On the other hand, it does nothing to check the CoA. Thus, the mobile could be lying about its own location. In this section, we explain how the mobile can exploit this vulnerability to mount a packet-flooding DoS attack.

Once we have made the above observation, it is easy to come up with an attack. Figure 4(a) shows a scenario where the attacker A tricks a public web site B into sending a flood of unwanted packets to a third party C. The attacker A first starts to download a stream of data, such as a long web page over TCP, from the public server B. It then sends an authenticated binding update to the server claiming to be at the care-of address C. (Details of the BU authentication have been left out of the figure.) The server accepts the binding update because A used an authentic home address. As a result, the server redirects the data stream to the false care-of address C.

The attacker does not need to be mobile. It can use its own stationary address A as the home address and act both as the home agent and as the mobile node in the binding update protocol. Also, the source IP address of the BU does not need to be spoofed (although it can be) because Mobile IPv6 allows the message to contain a CoA that is different from the source address.

The simplified attack description above ignores an important feature of transport-layer protocols: if the sender does not receive acknowledgements for the sent packets, it will stop transmitting the data stream. Unfortunately, this does not prevent the attack because the attacker can spoof the acknowledgments. The spoofing of TCP packets is usually prevented by the unpredictable initial sequence numbers but, in this case, the

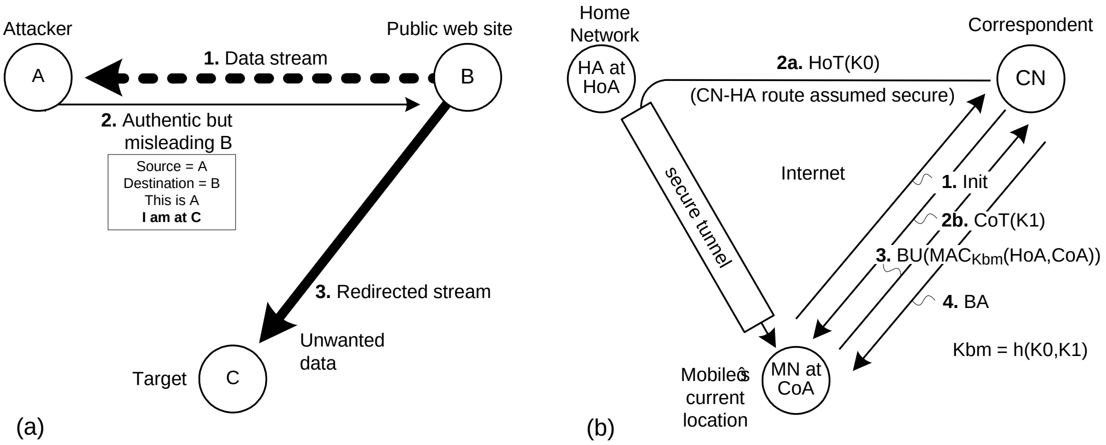

Figure 4 Bombing attack (a) and verifying the care-of address (b)

attacker itself performs the TCP handshake. This means that it knows the initial TCP sequence numbers and can spoof acknowledgments. Moreover, the attacker only needs to send one acknowledgment per TCP window, which means that by spoofing only a few packets it can get B to send a large data stream to C.

The attack is serious because it can be used to bomb any Internet node with data and the target node cannot do anything to stop the packet flow. If used in combination with distributed denial-of-service (DDoS) attacks, the bombing attack could seriously harm the reliability of the Internet.

The attacker needs to find a correspondent that is willing to send data streams to unauthenticated recipients. Many popular web sites provide such streams. The attacker also needs to know the target's IP address. If it does not know an individual address, it can target a network by redirecting data to one or more IP addresses within its address range.

The bombing attacks are by no means unique to Mobile IPv6. After we discovered the bombing attack against Mobile IPv6, several other mobility and multi-addressing protocols have been found to have similar vulnerabilities [7].

IV.2 Error notification

One suggested protection against the bombing attacks is that the target should send an error message to the sender of the unwanted data stream. In fact, TCP has a built-in mechanism for such notifications. The recipient of unwanted TCP packets usually sends a TCP Reset to the source of the packets, which immediately causes the sender to drop the connection. Thus, one might assume (as the current authors did for quite a while) that the target of the bombing attack would send a TCP Reset to the sender of the data. In practice, this does not quite work. In Figure 4(a), the packets sent by B to C have a routing header that says the packets are intended for A. When the IP layer at C processes the routing header, it encounters the strange address A and drops the packet without passing it to the transport layer. Thus, no TCP Reset will ever be sent.

On the other hand, if the target of the attacks is a non-existent IP address (in order to flood the network), the router on the subnet should send an ICMP Destination Unreachable message to the sender of the unwanted packets. This should stop the packet flow in the same way as a TCP Reset. However, not all routers and firewalls send ICMP error messages, and some TCP sender implementations will continue to send data if they receive both error messages and spoofed acknowledgments.

IV.3 Verifying the care-of address

A more reliable solution to the bombing attacks is to verify the care-of address before sending data to it. That is, the correspondent should check correctness of both the HoA and the CoA in the binding update. While it is almost impossible to securely verify the physical location of a mobile node in the Internet, the return routability test introduced in Section III.4 can be adapted to significantly mitigate the seriousness of the bombing attacks.

Figure 4(b) shows the improved BU protocol. In addition to sending a secret value to the HoA, the correspondent also sends a second secret directly to the care-of address. The correspondent uses both secrets to compute the MAC on the binding update. This proves to the correspondent that the mobile is able to receive messages sent to the care-of address.

In the Mobile IPv6 standard, this mechanism is called the *return-routability test for the care-of address (RR for CoA)*. The message sent by the correspondent to the CoA is called the *care-of test message (CoT)*. The HoT and CoT are sent in parallel. The secret value in the CoT message is the *care-of keygen token*. The BU is authenticated with a *binding management key (Kbm)* that is computed as a one-way hash of the two keygen tokens. (We have omitted some details of the key computation.)

The RR test does not strictly prove that the BU was sent by an honest mobile that is located at the new care-of address. The keygen token could be captured by an attacker who is located either at the CoA or on the CN-CoA route. The attacker could then spoof a BU from the CoA. But if that is the case, the stream of unwanted packets will flow to the attacker's own address or, at minimum, through a network where the attacker is present. Thus, the attacker is in a location where it could just as easily mount a DoS attack against the CoA without the help of any mobility protocol. It could do this by spoofing the connection establishment and acknowledgements. Effectively, the RR test for the CoA verifies that someone on the new route to which the correspondent is going to send data wants to receive the data. Nothing prevents the attacker from asking the server to send data to it in order to flood its own location, but that is already the case in the current Internet without mobility.

Although we have used the terms authentication and address verification, both of the return-routability tests can be seen as forms of authorization. The RR test for the HoA authorizes the sender of the binding update to change the binding for the home address. The RR test for the CoA authorizes the sender of the BU to request data to the care-of-address. These are quite different security goals and Mobile IPv6 achieves them using curiously symmetric mechanisms. This is perhaps best explained by viewing the two tests as a way to verify that the sender of the BU is authorized to control the use of the two addresses.

It should be noted that the RR test for the CoA is performed for a purpose that is independent of the RR test for the HoA. The two tests do not combine to provide stronger authentication of the mobileôs HoA, as is often mistakenly believed. This is because the routes along which the HoT and CoT are sent are not necessarily independent, as we explained in Section III.6.

We still need to consider a variant of the bombing attack that targets the home network instead of the care-of address. This attack is specific to mobility protocols like Mobile IPv6 where the mobile has a home address to which data will be sent when its current location is unknown. When the binding cache entry expires or the binding is explicitly deleted by a zero-lifetime BU sent by the mobile, the data flow is redirected to the HoA. An attacker could exploit this to bomb an address where it has once been. It could create the binding when it is located at the target address, acting itself as the home agent. It could then keep the binding alive by sending BUs when it moves away and later mount the attack by deleting the binding or by allowing it to expire. For the explicit binding deletion, the return-routability test for the HoA provides both authentication and bombing prevention. For the expiration, however, there is no perfect solution, because even an honest mobile may become temporarily unreachable. We could mark the cache entry as invalid instead of deleting it, and to stop sending data to the mobile until the RR test again succeeds. This could, however, mean that some cache entries are never deleted. Instead, the Mobile IPv6 standard requires the RR test for HoA to be performed every few minutes so that when the cache entry finally needs to be deleted for any reason, a successful RR test for the home address has always been performed recently. This limits the return-to-home variation of the bombing attack to target networks which the attacker has visited within the last few minutes.

It should be noted that while the RR test for the HoA could be replaced with alternative authentication methods, such as the CGA signatures, the need for the CoA verification still remains. There is currently no satisfactory alternative to the RR test for the CoA.

IV.4 Relation to Flow Control

It can be argued that the bombing attack is a flow-control issue and therefore should be taken care of in the transport layer rather than in the IP layer. That is, when sending a data flow into a new route, the correspondent should first verify that this route can accept the data. It could start by sending a single packet and gradually increase the transmission rate. For TCP streams, the natural solution would be to reset the TCP window size to one packet when the mobile moves. This would, in effect, test the return routability of the new route before sending large amounts of data into it.

From another viewpoint, the RR test can be seen as a variation of the cookie exchange, which has been used as part of the TCP handshake [29] and in authentication protocols, including Photuris [15]. We conclude that the cookie exchange should be performed once for each new destination IP address rather than once for each new connection. Before mobility and multi-addressing, there was no such distinction.

Unfortunately, adding a secure RR test to all transport protocols and changing the existing implementations would not be possible in practice. Moreover, many transport-layer protocols either do not practice TCP-compatible congestion control or allow spoofing of acknowledgments. Therefore, the most practical solution is to perform the return routability test in the IP layer.

V. Other security issues

Verification of the home and care-of addresses is sufficient to prevent most attacks that exploit weaknesses of the Mobile IPv6 route optimization. The return-routability protocol does this and, thus, protects the Internet from the new vulnerabilities introduced by the mobility mechanism. But like in all security protocols, there are a number of potential attacks against the security protocol itself that need to be considered. In this section, we make small changes to the protocol to prevent attacks like state-storage exhaustion and packet reflection.

V.1 CPU exhaustion

Authentication protocols are often vulnerable to flooding attacks that exploit the protocol features to consume the target node's computing power. This can be done by flooding the target with messages that cause it to perform expensive cryptographic operations. In order to exhaust the computing power of modern processors, the attacker needs to get them to perform public-key cryptographic operations. Symmetric encryption, hash functions and non-cryptographic computation are rarely the performance bottleneck.

Since the RR protocol only uses relatively inexpensive encryption and one-way hash functions, the consumption of CPU power is not a major concern. The CGA-based authentication, for example, would be much more likely to suffer from CPU-exhaustion attacks. For extreme low-end mobile devices, it might be necessary to trade security for performance by not encrypting the signaling messages between the mobile and the home agent.

V.2 State-storage exhaustion

It is well-known that stateful protocols expose the protocol participants to denial of service attacks. In particular, if a host stores a state as a result of an unauthenticated message, an attacker can initiate the protocol many times and cause the host to store a large number of unnecessary protocol states.

Figure 5(a) shows how this attack works against our protocol. The attacker sends a spoofed initial message with a false home address and false care-of address. The correspondent responds with two randomly chosen secret values, which it has to remember until it receives the authenticated BU. If the attacker repeats this many times, the correspondent may not be able to store all the state data and may drop some initial messages. This may prevent legitimate mobiles from using route optimization with the correspondent. The attack is similar to the SYN-flooding attack against the TCP protocol.

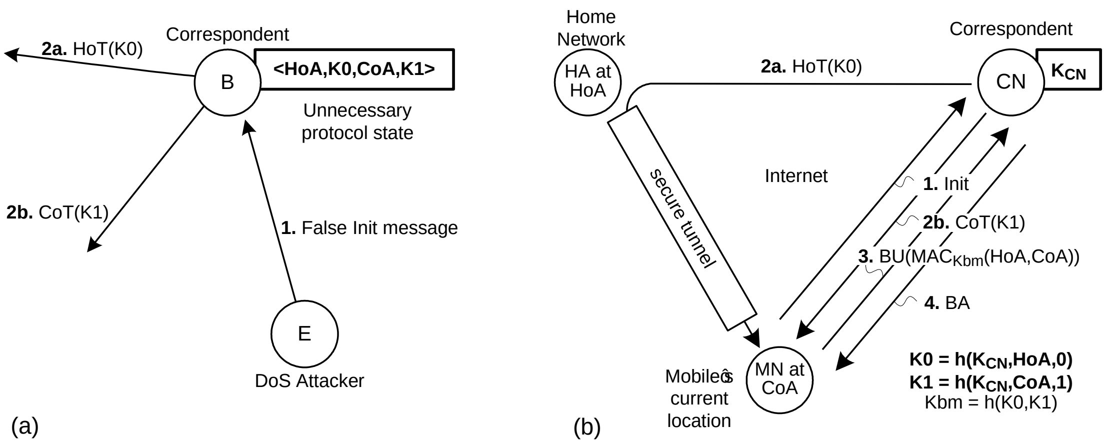

Figure 5 State-storage exhaustion attack (a) and stateless correspondent (b)

While the attack could be prevented by adding memory and managing the state storage carefully, it is much easier to design the protocol to be stateless. In Figure 5(b), the correspondent does not store a separate state for each mobile. Instead, it stores a single periodically-changing randomly-generated master secret (K_{CN}) and computes the two keygen tokens with a one-way function from the master secret and from HoA and CoA. After sending the HoT and CoT messages, the correspondent forgets the keygen token values. It recomputes them when it receives the authenticated binding update. This means that the correspondent remains stateless until it has authenticated the mobile. The stateless processing of the keygen tokens is similar to the statelessness during the TCP handshake when using the SYN cookies to prevent the SYN-flooding attack [29]. A more general discussion of statelessness in authentication protocols can be found in [6].

Careful readers might suggest a further improvement: binding the two keygen tokens together so that keys from different protocol runs cannot be mixed and matched. While this turns out to be unnecessary for the security of the current protocol, it might increase the robustness of the protocol if the assumptions or goals change in the future. The idea of binding the keys together was, nevertheless, rejected because it is useful to be able to reuse the home keygen token (K0 in the figure) within a short time if the care-of address changes frequently.

While the stateless handshake is now fairly standard in security protocols, some difficult decisions were made during the design process. The main problem is that only the responder can be stateless and it is not clear which party initiates the BU process and which one responds. Although the mobile initiates the BU protocol by sending the HoTI, this action is usually triggered by transport-protocol data that is either waiting to be sent to the correspondent or arrives from the correspondent via the HA-MN tunnel. Moreover, the transport layer connection may have been caused by an action of the other endpoint (e.g., a SIP INVITE). Thus, it is extremely difficult to tell which endpoint is the real initiator. Moreover, the IP layer is intended to be stateless and should not remember what packets have been sent previously. For simplicity, it was decided to make the correspondent stateless because it is more often a server and the mobile is more often a client.

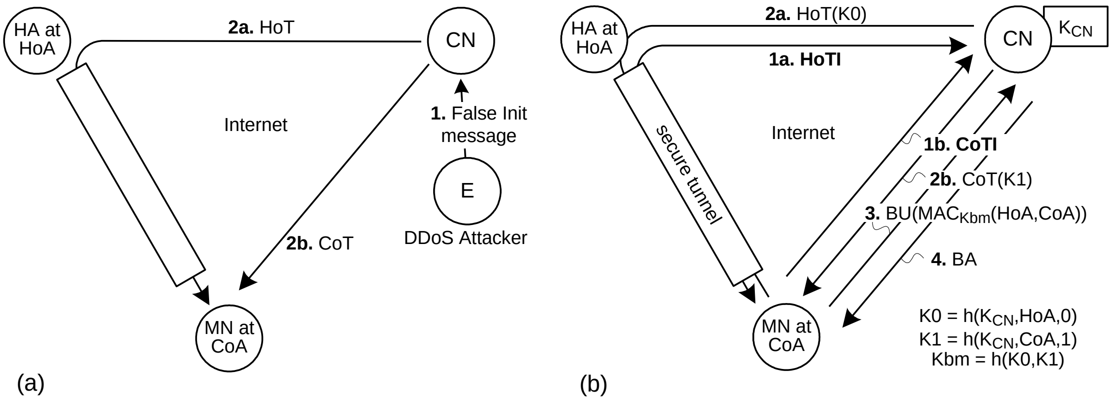

Figure 6 Reflection attack (a) and balanced messages (b)

V.3 Reflection and Amplification

Another attack that takes advantage of the BU security protocol is shown in Figure 6(a). The attacker E spoofs the initial message, which induces the correspondent to send two messages to the mobile. This causes two problems. First, the attacker sends only one packet but two arrive at the mobile. Thus, the attacker can use the binding-update authentication protocol to amplify a packet flooding attack against a mobile node by a factor of two. Second, the two messages arriving at the target of the flooding attacks have the correspondent's address as their source address. Any efficient mechanism for tracing the source of the packets probably won't be able to trace the attack back to its real origin. (For a detailed discussion of the problems caused by reflection, see [26].)

While these attacks may not seem very serious, it is hard to justify a security protocol that creates new vulnerabilities. The problem was solved by duplicating the initial message. Figure 6(b) shows an improved version of the BU-authentication protocol with one additional message to balance the message flows. The mobile sends one initial message via its home agent and another one directly to the correspondent. The Mobile IPv6 standard refers to the two initial messages as the *home test init (HoTI)* and *care-of test init (CoTI).*

The correspondent responds to each initial message independently by sending a secret value to the address from which the initial message came. The mobile needs to send both initial messages in order to receive both the HoT and CoT messages, which each contain one keygen token. The result is that the correspondent sends only as many messages as it receives, thus eliminating the amplification problem. The correspondent also responds always to the same address from which it receives a message, which may make it easier to trace the origin of the packets using standard methods.

The correspondent can still be stateless because it responds to a HoTI with a HoT and to a CoTI with a CoT and in no way associates the two exchanges to each other. The exchanges are parallel so that the total time taken by the protocol is not significantly increased. One consequence of the stateless correspondent is that, if the mobile moves

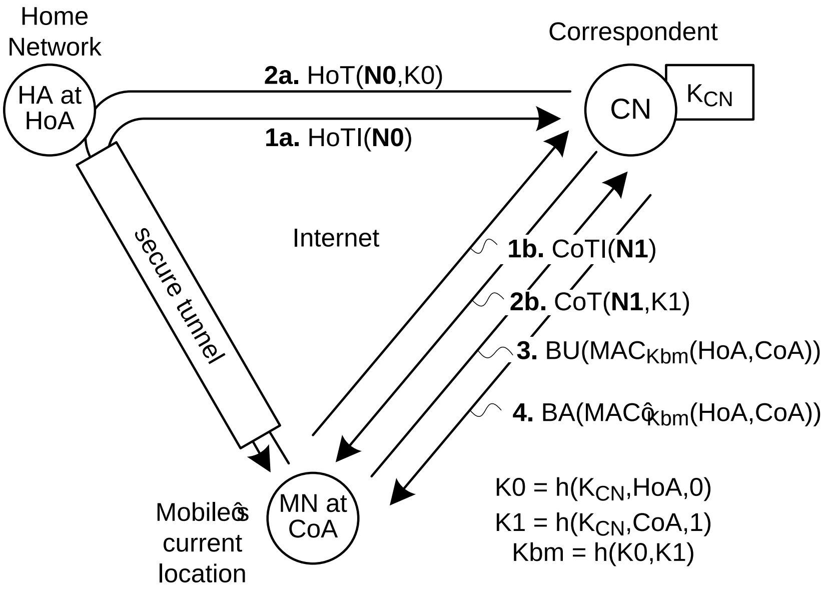

Figure 7 The complete BU protocol

frequently, it may not need to repeat the HoTI/HoT exchange after every move because it can reuse the recent home keygen token.

V.4 HoT, CoT and BA spoofing

There is one final adjustment that we will make to the RR protocol. As described so far, the protocol does not authenticate the HoT and CoT messages in any way. This leaves the protocol vulnerable to an attack where the attacker spoofs one of these messages and causes the mobile to use the wrong keygen token to authenticate the BU. Since the BU is authenticated with the wrong key, it will be rejected by the correspondent. If the attacker sends to the HoA a constant flow of HoT messages that appear be sent by the correspondent, it is likely that the first HoT to arrive at the mobile after it sends a HoTI is a spoofed one. This way, the attacker can prevent the mobile from sending any correctly authenticated BUs.

The simple solution is to include nonces in the HoTI and CoTI messages, which the correspondent copies to the HoT and CoT messages, respectively. The mobile can then reject the messages that do not have the correct nonce value. The final version of the protocol is shown in Figure 7. The standard names for the two nonces are the *home init cookie* and *care-of init cookie*.

Another potential problem, although much less serious, is spoofed binding acknowledgements. It was debated during the standardization process whether the BA needs to be authenticated or not. The argument for the authentication is that an attacker could somehow prevent a BU from reaching the correspondent and spoof a BA to convince the mobile that a binding was successfully created. The current authors would question the seriousness of this attack because, if the attacker is able to intercept the BU, it probably can mount a denial-of-service attack between the CoA and CN without resorting to BA spoofing. It was, nevertheless, decided in the IETF

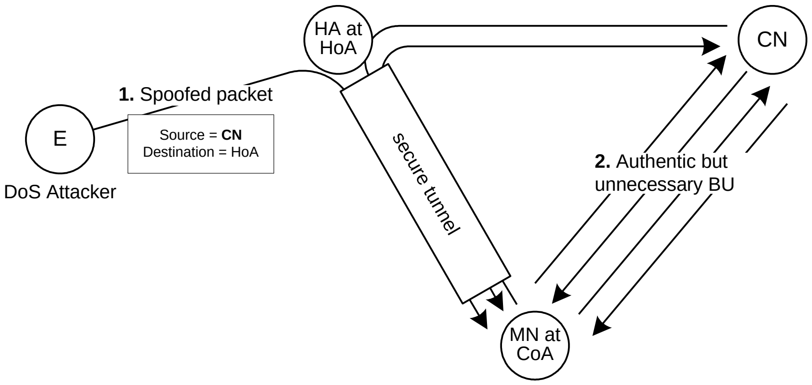

Figure 8 Inducing unnecessary BUs

working group to authenticate the BA using the same key as for the BU, as shown in Figure 7.

It is far from obvious that the binding management key is suitable for BA authentication. The key can be used to authenticate messages from the mobile to the correspondent because of the arguments we made in Section III.4. That is, in order to learn the key, the attacker must intercept the home keygen token on the CN-HoA route. This argument alone does not, in any way, make the key suitable for authenticating messages in the opposite direction, from the correspondent to the mobile. In fact, without the init cookies, anyone could spoof the HoT and CoT messages and, thus, determine the value of the binding management key.

The init cookies help because they prevent the spoofing of the HoT and CoT messages. In order to know the binding management key and to spoof a BA, an attacker must either sniff the keygen tokens or it must spoof them. In order to spoof a token, the attacker must sniff the corresponding init cookie. Which ever values the attacker aims to learn, it must be able to sniff traffic both on the CoA-CN route and on the CN-HoA route. This makes the spoofing of BAs more difficult than the spoofing of BUs, for which the attacker only needs to be on the CN-HoA route. This means that the BA is reasonably well authenticated. It remains an open question whether the authentication is at all necessary.

V.5 Unnecessary Authentication

There is one more flooding attack that needs to be considered. This attack is possible regardless of what kind of authentication is used for the binding updates. In fact, the stronger and the more expensive the authentication protocol, the more serious this attack becomes.

Figure 8 shows how the attacker can induce authentic but unnecessary binding updates. When a spoofed packet sent by the attacker is tunneled to the mobile, the mobile typically responds by executing the binding update protocol with the claimed correspondent. The correspondent will eventually accept the binding update because both the HoA and CoA are true. But the protocol execution is completely unnecessary. The attacker can repeat this with many different spoofed correspondent addresses to exhaust the resources of a single mobile, or with one spoofed correspondent address and many mobiles to attack a single correspondent.

Since the IP layer is stateless and BUs may be sent at any time, there is no practical way for the mobile or the correspondent to filter out the unnecessary binding updates without dropping also necessary ones. Therefore, the best defense against this attack is to limit the resources that the nodes allocate to processing binding updates. It also helps to prioritize the binding updates with known peers and to drop first the BUs to or from previously unknown hosts. It is possible to define a local security policy that lists specific high-priority peers for which route optimization is particularly important. A simpler policy, however, is to prioritize the refreshing of existing bindings at the expense of new ones. Although communication can continue unoptimized via the mobile's home network, it can suffer from low quality of service. The nodes should try to aggressively resume normal operation when they believe that the attack may be over.

VI. Conclusion

We have described the protection mechanisms used in the standard Mobile IPv6 binding-update protocol and the threats that they are intended to counter. In particular, the bombing attack that uses mobility signaling to redirect data to a target address had previously been ignored in many Internet protocols that update location or routing information. Some of the protection mechanisms used are unconventional: the security of the return routability protocol depends on the routing in the Internet being semi-reliable. This is because the protocol needs to work between any two Internet nodes without a PKI or other global security infrastructure. Without the infrastructureless operation, the Mobile IPv6 route optimization would have been confined to intra-organizational use.

Our protocol for securing the binding updates is included in the Mobile IPv6 standard [14] and it is now a part of various Mobile IPv6 implementations. The experiences from the Mobile IPv6 design process highlight the need to consider early the potential security threats created by new technology. The some of the design ideas described in this paper have been found to be applicable to other mobility and multi-addressing protocols. Sometimes, it helps to consider only the new threats rather than trying to design generic strong security solutions.

References

[1] Jari Arkko and Pekka Nikander, How to authenticate unknown principals without trusted parties. In Security Protocols, 10th International Workshop, volume 2845 of LNCS, pages 5-16, Cambridge, UK, April 2002. Springer.

[2] Tuomas Aura, Cryptographically generated addresses (CGA). RFC 3972, IETF. To appear.

[3] Tuomas Aura, Cryptographically generated addresses (CGA). In Proc. 6th Information Security Conference (ISC'03), volume 2851 of LNCS, pages 29-43, Bristol, UK, October 2003. Springer.

[4] Tuomas Aura, Mobile IPv6 security. In Proc. Security Protocols, 10th International Workshop, LNCS, Cambridge, UK, April 2002. Springer.

[5] Tuomas Aura and Jari Arkko, MIPv6 BU attacks and defenses. Internet Draft draft-aura-mipv6-bu-attacks-01, IETF Mobile IP Working Group, February 2002. Archived at http://www.watersprings.org/pub/id/draft-aura-mipv6-bu-attacks-01.txt.

[6] Tuomas Aura and Pekka Nikander, Stateless connections. In Proc. International Conference on Information and Communications Security (ICICS'97), volume 1334 of LNCS, pages 87-97, Beijing, China, November 1997. Springer.

[7] Tuomas Aura, Pekka Nikander and Gonzalo Camarillo. Effects of mobility and multihoming on transport-protocol security. In Proc. 2004 IEEE Symposium on Security and Privacy (SSP'04), Berkeley, CA USA, May 2004. IEEE Computer Society.

[8] Tuomas Aura, Michael Roe and Jari Arkko, Security of Internet location management. In Proc. 18th Annual Computer Security Applications Conference, Las Vegas, NV USA, December 2002. IEEE Press.

[9] Paul Ferguson and Daniel Senie, Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827, IETF, May 2000.

[10] Dan Harkins and Dave Carrel, The Internet key exchange (IKE). RFC 2409, IETF, November 1998.

[11] Robert M. Hinden and Stephen E. Deering, IP version 6 addressing architecture. RFC 2373, IETF, July 1998.

[12] Christian Huitema, Routing in the Internet. Prentice Hall, 1995.

[13] John Ioannidis, Protocols for Mobile Internetworking. PhD thesis, Columbia University in the City of New York, 1993.

[14] David B. Johnson, Charles Perkins, and Jari Arkko, Mobility support in IPv6. RFC 3775, IETF, June 2004.

[15] Philip Karn and William A. Simpson, Photuris: session-key management protocol. RFC 2522, IETF Network Working Group, March 1999.

[16] James Kempf, Jari Arkko, and Pekka Nikander, Mobile IPv6 security. Kluwer Wireless Personal Communications special issue on Security for Next Generation Communications, 29(3-4):389-414, June 2004.

[17] Brian E. Carpenter, Jon Crowcroft, and Yakov Rekhter. IPv4 address behaviour today. RFC 2101, IETF, February 1997.

[18] Gabriel Montenegro and Claude Castelluccia, SUCV identifiers and addresses. Internet Draft draft-montenegro-sucv-02, November 2001. Archived at http://www.watersprings.org/pub/id/draft-montenegro-sucv-02.txt.

[19] Thomas Narten and Richard Draves. Privacy extensions for stateless address autoconfiguration in IPv6. RFC 3041, IETF, January 2001.

[20] Pekka Nikander, A scaleable architecture for IPv6 address ownership. Internet-Draft draft-nikander-ipng-pbk-addresses-00, March 2001.

[21] Pekka Nikander, Denial-of-service, address ownership, and early authentication in the IPv6 world. In Proc. 9th International Workshop on Security Protocols, volume 2467 of LNCS, pages 12-21, Cambridge, UK, April 2001. Springer 2002.

[22] Pekka Nikander and Charles Perkins, Binding authentication key establishment protocol for Mobile IPv6. Internet Draft draft-perkins-bake-01, IETF Mobile IP Working Group, July 2001. Archived at http://www.watersprings.org/pub/id/draft-perkins-bake-01.txt.

[23] Pekka Nikander, Tuomas Aura, Jari Arkko and Gabriel Montenegro, Mobile IP version 6 (MIPv6) route optimization security design. In Proc. IEEE Vehicular Technology Conference Fall 2003, Orlando, FL USA, October 2003. IEEE Press.

[24] Pekka Nikander, Jukka Ylitalo and Jorma Wall, Integrating security, mobility, and multi-homing in a HIP way. In Proc. Network and Distributed Systems Security Symposium (NDSS'03), pages 87-99, San Diego, CA USA, February 2003.

[25] Greg O'Shea and Michael Roe, Child-proof authentication for MIPv6 (CAM). ACM Computer Communications Review, 31(2), April 2001.

[26] Vern Paxson, An analysis of using reflectors for distributed denial-of-service attacks. ACM Computer Communications Review (CCR), 31(3), July 2001.

[27] Charles Perkins, editor, IP mobility support for IPv4. RFC 3344, IETF, August 2002.

[28] Michael Roe, Tuomas Aura, Greg O'Shea and Jari Arkko, Authentication of Mobile IPv6 binding updates and acknowledgments. Internet Draft draft-roe-mobileip-updateauth-01, November 2001. Archived at http://www.watersprings.org/pub/id/draft-roe-mobileip-updateauth-01.txt.

[29] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. Spaffold, Aurobindo Sundaram, and Diego Zamboni. Analysis of a denial of service attack on TCP. In Proc. 1997 IEEE Symposium on Security and Privacy, pages 208-223, Oakland, CA USA, May 1997. IEEE Computer Society Press.

[30] Hesham Soliman. Mobile IPv6: Mobility in a Wireless Internet. Addison-Wesley, 2004.

[31] Susan Thomson and Thomas Narten, IPv6 stateless address autoconfiguration. RFC 2462, IETF, December 1998.

[32] Tatu Ylönen, SSH - secure login connections over the Internet. In Proc. 6th USENIX Security Symposium, pages 37-42, San Jose, CA USA, June 1996. USENIX Association.