Authenticated Binding Update in Mobile IPv6 Networks

Qiu Ying  
Institute for Infocomm Research  
Singapore  
qiuying@i2r.a-star.edu.sg

Bao Feng  
Institute for Infocomm Research  
Singapore  
baofeng@i2r.a-star.edu.sg

Abstract—Mobile IPv6 allows mobile nodes to remain reachable while moving around in the Internet. In Mobile IPv6, each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a care-of address, which provides information about the mobile node’s current location. This chapter reviews security threats and solutions in Mobile IPv6 with particular emphasis on security of binding update operation – a central and mandatory operation to achieve efficient mobility support in Mobile IPv6. The binding update operation allows a mobile node to inform its home agent and its correspondent nodes about its new care-of address. Unauthenticated or malicious binding updates, however, open the door for intruders to perform redirect attacks - malicious acts which redirect traffic from home agent or correspondent nodes to locations chosen by intruders. We present various methods for securing the binding update operation, analyze their security and performance, and point out possible future research directions.

Keywords: Authentication, mobile IP, redirect attacks, binding update

I. INTRODUCTION

Mobile networking technologies, along with the proliferation of numerous portable and wireless devices, promise to change people’s perceptions of the Internet. In mobile networking, communications activities are not disrupted when a user changes his/her device’s point of attachment to the Internet - all the network reconnections occur automatically and transparently to the user.

In today’s Internet, an IP address typically carries with it information that specifies the IP node’s point of attachment to the Internet. As a mobile node roams in the Internet, it needs to change its IP address every time it moves to a new location. On the other hand, however, to maintain existing transport-layer connections as a mobile node moves from one place to another, it must keep its IP address the same, changing the IP address will cause the existing transport layer connections to be disrupted and lost.

The Mobile IPv6 specifications RFC 3775 [1] supports mobile networking by allowing a mobile node to be addressed by two IP addresses, a home address and a care-of address. The former is an IP address assigned to the mobile node within its subnet prefix on its home subnet and the latter is a temporary address acquired by the mobile node while visiting a foreign subnet. The dual address mechanism in Mobile IP network allows packets to be routed to the mobile node regardless of its current point of attachment and the movement of the mobile node away from its home subnet is transparent to transport and higher-layer protocols.

One of the major features in Mobile IPv6 is the support for “Route Optimization” as a built-in fundamental part of the Mobile IPv6 protocol. The integration of route optimization functionality allows direct routing from any correspondent node (CN) to any mobile node (MN), without needing to pass through the mobile node’s home subnet and be forwarded by its home agent (HA), and thus eliminates the problem of “triangle routing”. Route optimization in Mobile IPv6 requires that the MN, HA and the CNs maintain a Binding Cache. A binding is the association of a MN’s home address (HoA) with a care-of address (CoA) for that mobile node, along with the remaining lifetime of that association. A mobile node uses Binding Update (BU) messages to notify its CNs or its HA of its current binding. Unfortunately, unauthenticated binding update messages provide intruders with an easy means to launch “Redirect Attacks”, i.e., malicious acts which redirect traffic from the correspondent nodes to destinations chosen by intruders. Therefore, security of the binding update messages is of paramount importance for Mobile IPv6 to meet its basic security requirements.

In this paper, we would introduce and compare some authenticated binding update protocols proposed in MIPv6 networks. The rest of the paper is organized as follows. In Section 2, we give a short overview of Return Routability protocol with emphasis on route optimization and binding update operations. We also detail the types of redirect attacks and state the security assumptions in MIPv6. In Section 3, 4 and 5, we introduce an improved RR protocol, the Cryptographically Generated Addresses (CGA) protocol and the Certificated Binding Update protocol, respectively. Finally, Section 6 contains our concluding and comparing remarks.

II. RETURN ROUTABILITY PROTOCOL

In IETF RFC 3775 [1], the Return Routability protocol (RR) is deployed to secure binding updates from MN to CNs. The basic RR mechanism consists of two checks, a home address check and a care-of-address check.

CN keeps a secret key kCN and generates a nonce at regular intervals, say every few minutes. CN uses the same key kCN and nonce with all the mobile nodes it is in communication with, so that it does not need to generate and store a new nonce when a new mobile node contacts it. Each nonce is identified by a nonce index. When a new nonce is
generated, it must be associated with a new nonce index, e.g., $j$. CN keeps both the current value of $N_j$ and a small set of previous nonce values, $N_{j-1}, N_{j-2}, \ldots$. Older values are discarded, and messages using them will be rejected as replays. Message exchanges in the RR protocol are shown in Figure 1, where the HoTI (Home Test Init) and CoTI (Care-of Test Init) messages are sent to CN by a mobile node MN simultaneously. The HoT (Home Test) and CoT (Care-of Test) are replies from CN. All RR protocol messages are sent as IPv6 “Mobility Header” in IPv6 packets.

MN  HA  CN

![Figure 1. Return Routability protocol.](image)

When MN wants to perform route optimization, it sends

$$\text{HoTI} = \{ \text{src} = \text{HoA}, \text{dst} = \text{CN}, r_{ni} \}$$

and

$$\text{CoTI} = \{ \text{src} = \text{CoA}, \text{dst} = \text{CN}, r_c \}$$

to CN, where $r_{ni}$ and $r_c$ are random values used to match responses with requests. HoTI tells MN’s home address HoA to CN. It is reverse tunneled through the home agent HA, while CoTI informs MN’s care-of address CoA and is sent directly to CN.

When CN receives HoTI, it takes the source IP address of HoTI as input and generates a home keygen token

$$K_{TH} = \text{prf}(k_{CN}, \text{HoA}|N_j|0)$$

and replies MN with

$$\text{HoT} = \{ \text{src} = \text{CN}, \text{dst} = \text{HoA}, \text{rH}, K_{TH}, j \}.$$  

where $|$ denotes concatenation and the final “0” inside the pseudo random function is a single zero octet, used to distinguish home and care-of cookies from each other. The index $j$ is carried along to allow CN later efficiently finding the nonce value $N_j$ that it used in creating the token $K_{TH}$. Similarly, when CN receives CoTI, it takes the source IP address of CoTI as input and generates a care-of keygen token

$$K_{TC} = \text{prf}(k_{CN}, \text{CoA}|N_i|1)$$

and sends

$$\text{CoT} = \{ \text{src} = \text{CN}, \text{dst} = \text{CoA}, r_c, K_{TC}, i \}$$

to MN, where the final “1” inside the pseudo random function is a single octet “0x01”. Note that HoT is sent via MN’s home agent HA while CoT is delivered directly to MN.

When MN receives both HoT and CoT, it hashes together the two tokens to form a session key

$$k_{BU} = h(K_{TH}, K_{TC}),$$

which is then used to authenticate the correspondent binding update message to CN:

$$\text{BU} = \{ \text{src} = \text{CoA}, \text{dst} = \text{CN}, \text{HoA}, \text{Seq#}, i, j, MAC_{BU} \},$$

where Seq# is a sequence number used to detect replay attack and

$$MAC_{BU} = \text{prf}(k_{BU}, \text{CoA}|\text{CN}|\text{HoA}|\text{Seq#}(ij))$$

is a message authentication code (MAC) protected by the session key $k_{BU}$. $MAC_{BU}$ is used to ensure that BU was sent by the same node which received both HoT and CoT. The message BU contains $j$ and $i$, so that CN knows which nonce values $N_j$ and $N_i$ to use to first re-compute $K_{TH}$ and $K_{TC}$ and then the session key $k_{BU}$. Note that CN is stateless until it receives BU and verifies MAC. If MAC is verified positive, CN may reply with a binding acknowledgement message

$$\text{BA} = \{ \text{src} = \text{CN}, \text{dst} = \text{CoA}, \text{HoA}, \text{Seq#}, MAC_{CA} \},$$

where Seq# is copied from the BU message and

$$MAC_{CA} = \text{prf}(k_{BU}, \text{CN}|\text{CoA}|\text{HoA}|\text{Seq#})$$

is a MAC generated using $k_{BU}$ to authenticate the BA message. CN then creates a binding cache entry for the mobile node MN. The binding cache entry binds HoA with CoA which allows future packets to MN be sent to CoA directly.

**Analysis:** The RR protocol are easily attacked by many threats, such as Session Hijacking Attacks, Movement Halting Attacks, Traffic Permutation Attacks, etc.

It is considerably easy to launch a session hijacking attack to break the RR protocol. Assume that $MN_1$ and CN are having an on-going communication session and the intruder wants to redirect CN’s traffic to his collaborator $MN_2$. The intruder monitors the CN-HA path (i.e., anywhere from $MN_1$’s home network to CN’s network) to obtain HoT, extracts the home keygen token $K_{TH}$ and sends it to $MN_2$. Upon receiving $K_{TH}$, $MN_2$ sends a CoTI to CN and CN will reply with a care-of keygen token $K_{TC}$. $MN_2$ simply hashes the two keygen tokens to obtain a valid binding key, and uses the key to send a binding update message to CN on behalf of $MN_1$. The binding update will be accepted by CN which will in turn direct its traffic to $MN_2$.

Another related attack is Movement Halting Attacks. When a mobile node MN rapidly moves from one care-of address CoA to another CoA’. Since MN runs the RR protocol whenever it moves to a new location, an intruder can intercept the care-of keygen token $K_{TC}$ in the current RR session and the home keygen token $K_{TH}$ in the next RR session, hash the two keygen tokens to get a valid binding key, and then send a binding update message with the CoA in the current session to the correspondent node. The correspondent node will still send its traffic back to CoA. Hence, MN, which has moved to CoA’, will not receive data from the correspondent node. Note that in this attack the attacker does not have to intercept the two keygen tokens at the "same time".

The RR protocol is also subject to a Traffic Permutation Attack. Consider a correspondent node which provides online services to many mobile clients. An intruder can simply eavesdrop on the RR protocol messages to collect keygen
tokens on the border between the correspondent node and the Internet. The intruder then hashes random pairs of keygen tokens to form binding keys, and sends binding update messages to the correspondent node. This will cause redirection of traffic to randomly selected mobile clients and eventually bring down the services of the correspondent node.

III. IMPROVED RETURN ROUTABILITY PROTOCOL

The attacks against RR protocol outlined in the above section are due to the decoupling of HoA and CoA in RR messages. In the original RR protocol, the home keygen token

\[ K_{T_H} = \text{prf}(k_{CN}, \text{HoA}|N|0) \]

and the care-of keygen token

\[ K_{T_C} = \text{prf}(k_{CN}, \text{CoA}|N|1) \]

are delivered without any stated relationship. Any pair of home keygen token and care-of keygen token can generate a valid binding key

\[ k_{BU} = h(K_{T_H}/K_{T_C}) \]

as long as the indexes, i and j, are still valid.

However, the attacks described in the above section can be prevented by modifying the RR protocol to include both CoA and HoA in the generation of home keygen token and care-of keygen token, respectively. In the improved RR protocol [2], HoA and CoA are bound together. A mobile node sends

\[ \text{HoTI'} = \{ \text{src}=\text{HoA}, \text{dst}=\text{CN}, \text{CoA}, r_{H} \} \]

and

\[ \text{CoTI'} = \{ \text{src}=\text{CoA}, \text{dst}=\text{CN}, \text{HoA}, r_{C} \} \]

to a CN, which replies with the home keygen token

\[ K'_{T_H} = \text{prf}(k_{CN}, \text{HoA}|N|\text{CoA}|0) \]

and the care-of keygen token

\[ K'_{T_C} = \text{prf}(k_{CN}, \text{CoA}|N|\text{HoA}|1). \]

Then the new binding key

\[ K'_{BU} = h(K'_{T_H}/K'_{T_C}) \]

is valid only for the pair of HoA and its claimed CoA. Therefore the misuse of keygen tokens can be avoided.

IV. CRYPTOGRAPHICALLY GENERATED ADDRESSES PROTOCOL

An IPv6 address consists of 128 bits and is divided into two portions: a subnet prefix and an interface identifier. The home addresses of all the mobile nodes associated with a home link share the same home link subnet prefix and are differentiated by their unique interface identifiers. The CGA protocol [3, 4] generates an IPv6 home address for a mobile node where the interface identifier portion is created from a one-way hash \( h() \) of the mobile node’s public key. The mobile node uses the corresponding private key to sign correspondent binding update messages. Each mobile node MN has a public/private key pair \( P_{MN} \) and \( S_{MN} \) in a digital signature scheme. MN’s home address is given by

\[ \text{HoA} = \{n\text{-bit home link subnet prefix} \mid \text{interface identifier} \} \]

The interface identifier field is obtained by taking the left-most \((128-n)\) bits of the hash function output \( h(P_{MN}) \). A binding update message from \( MN \) to a correspondent node \( CN \) is given by

\[ \text{BU} = \{ \text{src}=\text{CN}, \text{dst}=\text{CoA}, \text{HoA}, \text{Seq#}, P_{MN}, 128-n, \text{SIG}_{MN}\} \]

where

\[ \text{SIG}_{MN} = \sigma(S_{MN}, \text{CoA} | \text{CN} | \text{HoA} | \text{Seq#} | P_{MN} | 128-n) \]

is MN’s digital signature generated using its private key \( S_{MN} \). Upon receiving the BU, CN computes \( h(P_{MN}) \), compares the left most \((128-n)\) bits of \( h(P_{MN}) \) with the right most \((128-n)\)-bit in HoA, and verifies the signature using the public key \( P_{MN} \). If the hash value matches the related value in HoA and if the signature verification is positive, CN accepts the binding update message.

Analysis: The hash function \( h() \) here acts as a “one-to-one” mapping from a public key value to an interface identifier, it binds a public key value with an interface identifier. Since it is computationally hard to either find the private key or forge a digital signature given the public key, a match of \( h(P_{MN}) \) with interface identifier in HoA as well as positive verification of the signature on BU proves that BU was generated by the mobile node whose interface identifier portion is interface identifier and who knows the private key \( S_{MN} \). This is the only assurance a correspondent node gets from BU. As a consequence, the protocol is able to provide good protection against the session hijacking attack provided the number of bits in interface identifier, \((128-n)\), is large enough. If \((128-n)\) is small, an intruder can randomly generate pairs of public and private keys, hash the public keys and look for a match to a target node’s II. Once a match is found, the intruder is able to impersonate the target node and forge binding updates. The computational complexity of this brute force attack is on the order of \( o(2^{(128-n)}) \). A clever method was presented in [3] which effectively removes \((128-n)\)-bit limit on the hash length. This is achieved by artificially increasing both the cost of generating a new CGA address and the cost of a brute-force attack while keeping the cost of CGA based authentication constant.

On the other hand, since this protocol does not provide any proof on the authorization of MN to use the particular HoA, it is not able to protect against the malicious mobile node flooding attacks. Actually, an intruder can just generate a public/private key pair, hashes the public key to form a home address, signs a binding update message which contains a victim’s address as CoA, and sends it to a correspondent node. The correspondent node will accept the binding update and start sending traffic to flood the victim node.

Compared with the RR protocol, the CGA protocol is computational intensive since every binding update message requires the mobile node to generate a digital signature and the correspondent node perform a verification of digital signature.
V. CERTIFICATED BINDING UPDATE PROTOCOL

The Certificated Binding Update (CBU) protocol [5] employs public key cryptosystems in order to provide strong security and good scalability.

Similar in the RR protocol, all the protocol messages in CBU are carried within IPv6 “Mobility Header” which allows protocol messages to be piggybacked on any existing IPv6 packets. The protocol messages exchanged among a mobile node MN, its home agent HA and its correspondent node CN are shown in Fig 2. In the protocol, the existence of and operations performed by HA are transparent to both MN and CN. As far as MN is concerned, it sends message REQ to and receives REP from CN. Similarly, from CN’s point of view, it receives COOKIE0, EXCH0 and CONFIRM from and sends COOKIE1 and EXCH1 to MN.

\[
\begin{array}{c|c|c}
MN & HA & CN \\
\hline
REQ & COOKIE0 & \\
& COOKIE1 & \\
& EXCH0 & \\
& EXCH1 & \\
& CONFIRM & \\
REP & 
\end{array}
\]

Figure 2. Message exchange in the proposed protocol

The use of cookies during the key exchange is a weak form of protection against an intruder who generates a series of request packets, each with a different spoofed source IP address and sends them to a protocol party. For each request, the protocol party will first validate cookies before performing computationally expensive public key cryptographic operations.

When MN wants to start route optimization operation with CN, it sends a route optimization request

\[
REQ = \{ src=HoA, dst=CN, n0 \}
\]

to CN via reserve tunneling, where \( n0 \) is a nonce value used to match the reply message REP. Message REQ is sent to MN’s home subnet via the IPsec protected secure tunnel. IPsec provides replay protection only when dynamic security association establishment is used. This may not always be possible and manual keying might be preferred in certain circumstances. For this reason, \( n0 \) is used to counter message replay. Upon arriving at the home link, REQ is intercepted by HA using IPv6 Neighbor Discovery [1, 6]. HA will not forward REQ to CN, instead, it creates a cookie \( C_0 \) and sends

\[
COOKIE0 = \{ src=HoA, dst=CN, C_0 \}
\]
to CN. In reply, CN creates a nonce \( n_1 \) and a cookie \( C_1 \), and sends

\[
COOKIE1 = \{ src=CN, dst=HoA, C_0, C_1, n_1 \}
\]
to MN. Note that the destination address in COOKIE1 is MN’s home address HoA. As a result, this message is delivered to MN’s home subnet and intercepted by HA using IPv6 Neighbor Discovery. After receiving COOKIE1, HA checks on the validity of \( C_0 \), generates a nonce \( n_2 \) and a Diffie-Hellman secret value \( x < p \), computes its Diffie-Hellman public value \( g^x \) and its signature

\[
SIG_H = \sigma(S_H, HoA|CN|g^x|n_1|n_2|TS)
\]

using home link’s private key \( S_H \), where \( TS \) is a time stamp. This time stamp does not have to be checked by the recipient during the message exchange. It will be used to trace back the culprit should a malicious mobile node flooding attack have occurred. This point will be made clearer later. Finally, HA replies CN with

\[
EXCH0 = \{ src=HoA, dst=CN, C_0, C_1, n_1, n_2, g^x, TS, SIG_H, Cert_H \}
\]

where \( Cert_H = \{ HS, P_H, VI, SIG_{C_{VI}} \} \) is the public key certificate of the home subnet as defined before. Note that the values of \( n_1 \) and \( n_2 \) are included in the signature \( SIG_H \) in order to counter replay of old signatures and to resist chosen message attacks to the signature scheme, respectively.

When CN receives EXCH0, it validates the cookies, the home link’s public key certificate \( Cert_H \) and the signature and importantly, checks for equality of the home subnet prefix strings embedded in both \( Cert_H \) and HoA. If all the validations and checking are positive, CN can be confident that the home address \( HoA \) of MN is authorized by its home subnet and the Diffie-Hellman public value \( g^x \) is freshly generated by MN’s home subnet. CN next generates its Diffie-Hellman secret value \( y < p \). It then computes its Diffie-Hellman public value \( g^y \), the Diffie-Hellman key \( k_{DH} = (g^y)^x \), a session key

\[
k_{BU} = prf(k_{DH}, n_1|n_2)
\]

and a MAC

\[
MAC_1 = prf(k_{BU}, g^x|EXCH0),
\]

and sends

\[
EXCH1 = \{ src=CN, dst=HoA, C_0, C_1, g^x, MAC_1 \}
\]
to MN. Again, this message is intercepted by HA, which first validates the cookies, calculates the Diffie-Hellman key \( k_{DH} = (g^y)^x \) and the session key \( k_{BU} = prf(k_{DH}, n_1|n_2) \). HA then computes

\[
MAC_2 = prf(k_{BU}, EXCH1),
\]

and sends

\[
CONFIRM = \{ src=HoA, dst=CN, MAC_2 \}
\]
to CN. The validity of \( MAC_2 \) is checked by CN and if it is valid, CN creates a cache entry for HoA and the session key \( k_{BU} \), which will be used for authenticating binding update messages from MN.

Upon positive verification of \( MAC_1 \), HA also sends

\[
REP = \{ src=CN, dst=HoA, n_0, k_{BU} \}
\]
to MN through the secure IPsec ESP protected tunnel. After receiving REP, MN checks that \( n_0 \) is the same as the one it sent out in REQ. If so, MN proceeds to send CN binding update messages protected using \( k_{BU} \), as in the RR protocol. It should be noted that the CONFIRM message serves to confirm the key to CN and hence is optional.
Analysis: The trust model and the design principle of the CBU protocol follow those of the SSL. In the CBU, a CN is the equivalent of a web browser and a HA is the equivalent of a web server. CAs issue public key certificates directly to HAs. The CBU performs a strong one-way authentication of MHoA to CN and provides CN with the confidence that it shares a secret session key with MN. Here we would like to point out that the most important message is EXCH0. Recall that after receiving EXCH0, CN checks on the equality of the home subnet prefix contained in both CertMN and HoA. This check is critical to detect man-in-the-middle attack. The signature $SI_{H} = S_{H}(HoA/CN|g^{|n|}n_{1}/n_{2}|TS)$ serves two purposes. First, it certifies that the Diffie-Hellman value $g^{|n|}$ was originated by $MN$’s home agent $HA$ on behalf of $MN$ and second, it testifies that $HoA$ is under $HA$’s (or equivalently the home link’s) jurisdiction and is a legitimate home address for its mobile node $MN$. This authenticates $MN$’s $HoA$ to $CN$.

Since a successful completion of the protocol allows $CN$ to authenticate $MN$’s $HoA$ as well as allows the two nodes to set up a secret session key for securing binding updates, the protocol prevents the session hijacking attack shown. This protocol, as any other protocols, is not able to completely prevent malicious mobile node flooding attacks. However, if a correspondent node were blamed to have bombarded a network service or site, it could present the signature $SI_{H} = S_{H}(HoA/CN|g^{|n|}n_{1}/n_{2}|TS)$ and point its fingers at the home agent $HA$. $HA$ can subsequently nail down the mobile node $MN$ which had a home address $HoA$ and performed a binding update at the time specified by $TS$.

In the CBU protocol, mobile nodes are not required to perform any public key cryptographic operations but correspondent nodes are. Public key cryptographic operations may not be a great concern if a correspondent node is a server machine. However, a correspondent node can also be a mobile node with limited computational power and battery life. In this case, public key operations supposed to be performed by the correspondent node can be off-loaded to its home agent.

Since the CBU protocol uses strong cryptosystems, the secret session key $k_{BU}$ established from the protocol could be used for a long period of time.

VI. CONCLUDING AND COMPARING REMARKS

In this paper, we introduced 4 protocols on authenticated binding update in mobile IPv6 networks.

The original Return Routability protocol is light and do not assume the existence of an Internet wide public key infrastructure (PKI). However, it cannot protect against Session Hijacking Attacks, Movement Halting Attacks, Traffic Permutation Attacks, etc.

Hence, the improved RR protocol is developed. By coupling of $HoA$ and $CoA$ in RR messages, the protocol can provide much stronger security than the original RR protocol without changing its architecture.

The Cryptographically Generated Addresses protocol also does not need the existence of PKI, but it is computational intensive and is not able to protect against the malicious mobile node flooding attacks.

The 4th protocol is Certificated Binding Update that makes use of a digital signature scheme and the Diffie-Hellman key exchange algorithm, where public key certificates are not issued for each and every mobile node, but are issued for home subnets based on home subnet prefixes. Such an approach makes certificate issuing, tracking, and revocation much more practical and manageable. In CBU, a home agent functions as security proxy for its mobile nodes and testifies the legitimacy of a mobile node’s home address to a correspondent node during protocol execution. Recognizing that most mobile nodes are constrained in processing power and battery life and that home agents can be easily equipped with increasingly low cost yet powerful cryptographic processing hardware accelerators, the protocol was designed to off load all the expensive public key cryptosystem operations from mobile nodes to their home agents.