

## Mobile IPv6 Binding Update - Return Routability Procedure

### Abstract

The Return Routability Procedure (RRP) attempts to provide basic authentication and integrity for Mobile IPv6 Binding Updates, without relying on Public Key Infrastructure (PKI). This project analyzes the RRP with the Murphi model checker and verifies protocol design decisions through a rational reconstruction approach. Two major attacks are discovered in the final version of the protocol and two solution ideas are discussed.

### I. Background

As a replacement to Mobile IPv4, Mobile IPv6 is a standard that presents a method for mobile nodes to maintain connectivity as they move across different networks (and hence change IP addresses). It is assumed that all mobile nodes (MN) have an associated “home” network with a corresponding permanent home IP address. Furthermore, each home network contains a home agent (HA) responsible for tracking these mobile nodes as they move around different networks. Once a MN moves into a foreign network, acquiring a new IP address - called a care-of address (CoA), the MN is required to register this address with its HA via a binding update. This binding update is issued over an IPsec tunnel, using an IPv6 security association, to protect its integrity and authenticity. Given this, a correspondent node (CN) can maintain communication with a MN even as the MN switches networks, using “indirect routing” with packets being relayed by the HA.

In order to optimize this and communicate directly with the CN, the MN can issue an additional binding update to the CN. The authenticity and integrity of this particular binding update cannot be secured using IPsec since it cannot be assumed that a common PKI exists between the two nodes. In order to have some assurance of the validity of this binding update, an alternate method is used: the Return Routability Procedure.

### II. Protocol Description

The protocol works as follows. The CN provides the MN with two keygen tokens sent through two different paths (one sent to the home address of the MN, and one sent to the care-of address of the MN), and these two tokens are used by the MN to compose a valid binding update to the CN. More specifically, these two tokens are used to create a key for an HMAC that is sent as a part of the final binding update message, which is verified by the CN. Figure 1 displays a diagram of the messages exchanged in the protocol. Please refer to Appendix A for a detailed description of these messages. [1]

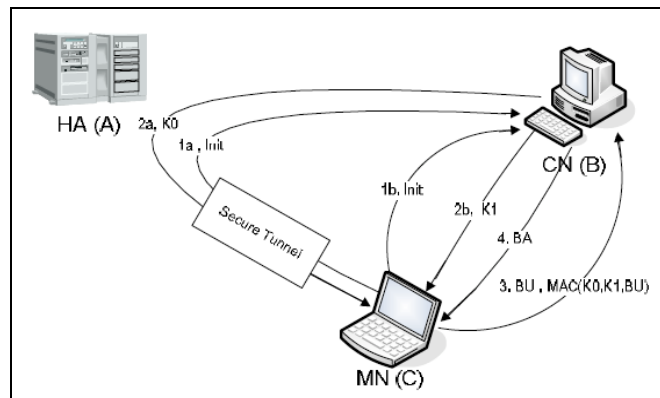


Figure 1: RRP Protocol Diagram, [2]

### III. Security Properties

The security properties of the RRP are all based on ensuring that Mobile IPv6 is as secure as IPv4 without mobility. This requirement is much more “relaxed” than other traditional security protocols. Many of these properties are based on preventing attacks that would have been possible if binding updates were not secure, and could thereby introduce security holes not present in IPv4. Below is a list of some of these attacks: [3]

- 1) Bombing attack: intruder provides a fake care-of address in a binding update and is able to bomb an innocent node with unwanted traffic.
- 2) Redirection/hijack attack: intruder provides his own address as the care-of address in a binding update to redirect traffic to himself.

- 3) Reflection and amplification attacks.
- 4) Binding update replay attacks.

#### Attacker scenarios

1. All nodes are honest; there is an outside attacker than can have access to one or more paths
2. MN is dishonest; the attacker acts as the MN
3. CN is dishonest (we do not consider this case to limit scope of our analysis)
4. HA is dishonest (we do not consider this case to limit scope of our analysis)

#### Security properties (Murphi invariants)

- 1) Legitimacy of the final binding known to the correspondent node: “Once a binding update has completed, the home and care-of address in the final binding known to the correspondent must refer to the same node.”
- 2) Keygen tokens secrecy: “An intruder may never obtain access to both the care-of and home keygen tokens unless the intruder is acting as an honest mobile node.”
- 3) Authenticity of MN: “Whenever a CN  $i$  completes a session (apparently with some MN  $j$ ), then it must be that  $j$  has completed a session with  $i$ ”. We need two invariants in this case, one for each of the addresses:
  - a) Authenticity of MN care-of address
  - b) Authenticity of MN home address
- 4) Authenticity of CN: “Whenever a MN  $i$  completes a session (apparently with some CN  $j$ ), then it must be that  $j$  has completed a session with  $i$ ”

### **IV. Rational Reconstruction**

We used a rational reconstruction approach to analyzing the RRP in Murphi. This involves building up the protocol from the bottom-up in incremental versions. Below we detail the incremental versions, each of which corresponds to a different Murphi model. The fifth and final version constitutes the complete protocol.

#### Version #1

- Initial version with only the care-of test init, home test, and binding update messages.
- Care-of init message must contain the home address of the mobile node (since we don't have a home-init).

**Invariants used:** 1, 3a, 3b

#### **Notable Attacks:**

- 1) Bombing attack without access to any paths: Intruder generates a valid binding update using his own home address and care-of address of a target node. This will redirect traffic to an unwanted host addressable at the given CoA. Violates invariant #1 and #3a.

#### Version #2

- Added the care-of test message sent from the CN to the MN; this was done to prevent the bombing attack (without access to any paths) we found in version #1.

**Invariants used:** 1, 2, 3a, 3b

#### **Notable Attacks:**

- 1) Bombing attack with access to CN-MN path: Intruder generates a valid binding update using his own home address and care-of address of a target node. Intruder receives the home keygen token, and must sniff/intercept the care-of keygen token on route from the CN to the target node. Violates invariants #1, #2, and #3a.
- 2) Redirection attack with access to CN-HA path: Intruder generates a valid binding update using the home address of a target node and his own care-of address. Intruder receives the care-of keygen token, and must sniff/intercept the home keygen token on route from the CN to the HA. This will redirect the target node's traffic to the intruder. Violates invariants #1, #2, and #3b.
- 3) Reflection and amplification attack (not found with Murphi).

#### Version #3

- Added the home test init message sent from the MN to the CN through the HA; this was done to prevent the reflection and amplification attacks found in version #2.
- Now, the care-of init message does not have to include the home address of the mobile node; the home address can now be conveyed to the correspondent node via the home init message.

**Invariants used:** 1, 2, 3a, 3b

### **Notable Attacks:**

- Bombing and redirection attacks from version #2 are still possible.
- 1) State exhaustion of the correspondent node (not found with Murphi)

### Version #4

- Added nonce/nonce index for CN; this was done to prevent CN state exhaustion attack found in version #3.

**Invariants used:** 1, 2, 3a, 3b, 4

### **Notable Attacks:**

- Bombing and redirection attacks from version #2 are still possible.
- 1) Binding update replay attack: An attacker can replay a valid binding update message that was previously used by an honest MN and honest CN. This would cause the most damage if done after the MN has moved to a new location (acquired a different CoA) and the attack replayed an older binding update to an outdated CoA, thereby interrupting communications between the MN and CN (not found with Murphi).
- 2) Binding update prevention attack: An attacker can spoof a fake care-of test and/or home test reply messages to the mobile node's init messages (before the honest correspondent does so) without access to any paths. MN then sends a binding update to the CN and the CN will not accept it because the tokens do not match its own, or it is not yet in a state to receive a binding update message. This attack basically prevents the MN from providing a valid binding update to the CN so it's not too major. Violates invariant #4.

### Version #5 (Final version - Complete protocol)

- Added cookies to the MN's init messages (cookies are basically just nonces) to prevent the binding update prevention attack in version #4. Cookies are used to verify that the home test or care-of test messages match the home test init or care-of test init messages, respectively. These cookies also serve to ensure that parties who have not seen the init requests cannot spoof responses.

- Added sequence numbers to the final binding update message to prevent replay attacks in version #5.

- Note: we do not model the final binding update ACK since this message is not required in the RRP!

**Invariant used:** 1, 2, 3a, 3b, 4

### **Attacks:**

- Bombing and redirection attacks from version #2 are still possible.
- There are other ways to perform the binding update prevention attack, but the attacker now needs access to one or more paths. Some examples are that the attacker can intercept one (or both) of the test messages and generate a corresponding valid reply, or prevent the final binding update message from going through. Both will prevent a binding update from occurring between the MN and CN, still fairly minor.

## **V. Assessment of Attacks**

We found no attacks in the final version of the RRP without the attacker intercepting on at least one path. This is good since it approximates the security properties of non-mobile IPv4. If the security goal of being at least as secure as regular IPv4 is taken to mean that an attacker must be able to intercept on at least one path to successfully execute an attack, then as far as our analysis has shown, the RRP meets its security goals. However, if the RRP security goals are understood to provide the extra assurance that an attacker would have to be able to intercept on two paths, then the following attacks are possible on the RRP.

- 1) Situation: Attacker can intercept between the CN and a Target Node

Attack: *Bombing attack* (reconstruction: version #2, attack #1)

This is a serious attack for three reasons:

- a) This attack can be used to direct a Denial of Service (DoS) or Distributed DoS (DDoS) packet flood attack from one or more chosen CN(s) to a target node or network. The target can be any IPv6 node (mobile or not).
- b) This attack is relatively easy to execute, since the attacker can choose the CN(s) and can therefore influence the path on which he/she must intercept.
- c) The RFC does not make any assumptions about the security of the CN – Target path. This attack holds even if all security assumptions in the RFC are in place.

Limitations:

- a) The default timeout for a MIPv6 binding is 5 minutes. This means the attacker must repeat the bombing attack to sustain a DoS attack for more than 5 minutes.

2) Situation: Attacker can intercept between the CN and HA

Attack: *Redirection / hijack attack* (reconstruction: version #2, attack #2)

This is a significant attack:

- a) This attack can be used to redirect the traffic from a given CN intended for a given MN to an arbitrary address (possibly the attacker). This may allow the attacker to receive sensitive information intended for the MN and also create a temporary DoS for the MN (until they can re-establish the binding or fall back on the indirect connection via the HA).

Limitations (why it is less of a concern to us than the bombing attack above):

- a) This attack is somewhat more difficult to execute. The target must be a MN who has an active binding with a CN (attacker cannot target any IPv6 node). The attacker must have prior knowledge of an existing HA-CN connection and the ability to intercept on this path. The attacker cannot choose the CN or the route on which he/she must intercept.
- b) The RFC notes that attacker interception on the HA-CN path is unsafe and says that an attacker intercepting on this path (HA-CN) is equivalent to what is possible in IPv4. [1]

3) Situation: Attacker can intercept between the CN and HA or between the CN and MN

Attack: *Binding Update Prevention attack* (reconstruction: version #5)

This is a minor attack:

- a) This is another DoS attack that does not require access to two paths. The form of this attack where the attacker intercepts an init cookie and then creates a corresponding valid test message containing an invalid token is somewhat more damaging than the other version where the attacker simply drops the BU message.

Limitations:

- a) The MN can retry the binding update immediately and in the worst case (if the attacker continues the attack on subsequent binding update attempts), the MN can simply fall back on the indirect connection via the HA. While this is certainly a real DoS attack, a network attacker is commonly assumed to have the ability to interfere an exchange in this way (e.g. by dropping a packet).

## VI. Possible Solutions to Attacks

### 1) Idea to minimize Bombing Attack damage

CN sends an additional message after the binding update (BU) is completed and it has started sending normal traffic to the MN via the new CoA. After 15 seconds (or some reasonable amount of time), it sends a Binding Update Verification (BUV) message with a new nonce to the MN. If the MN does not respond with an acknowledgement containing the verification nonce within another 15 seconds, the connection is terminated.

This idea is primarily designed to minimize damage from a bombing attack where the attacker redirects all his traffic to a target MN (or other node). The idea is to identify whether the node at the CoA (called MN here) really wants to receive this traffic and to do so in a way that is non-trivial for the attacker to circumvent.

Some cases we considered:

- 1) ***Bombing/(D)DoS attack***: Couldn't the attacker just intercept the BUV, since he is already between the CN and MN? If the binding update was initiated by an attacker who has executed a DoS attack against MN, then the network will be clogged and the BUV will not get through to where the intruder is most likely located (close to the MN, since he may want to intercept tokens from more than one CN to MN to increase the strength of the attack). This case is what motivated the BUV to be sent after starting to send traffic: so that the DoS traffic would be detectable by the BUV not getting through.
- 2) ***Not a DoS attack, but still invalid***: If the BUV makes it through to the MN and they did not initiate this BU, then they can cancel the unwanted traffic with a response, or by simply not responding.
- 3) ***Valid/honest BU***: If this was a valid BU initiated by the MN, then it can simply acknowledge the BUV. Note: This may create the possibility for a new DoS attack: intruder may intercept the BUV for a valid BU and cause the optimized connection to be terminated. However, this is much less serious, since invalidating the BU is normally possible for the intruder by simply blocking BU packets. The MN and CN can fall back to the un-optimized route and communicate indirectly via the HA.

### 2) Use Cryptographically Generated Addresses (CGA) [4] – Redirection Attack

Having the nodes involved in the RRP use CGAs makes message spoofing more difficult. More specifically, it has the following benefits:

- a) Makes it very difficult for an attacker to execute a redirection attack, since the attacker must now know the public/private key pair that matches the CGA for the MN's home address. This gives us some of the benefits of public-key signatures without infrastructure.
- b) Unlike the solution to minimize DoS damage proposed above, this idea does not require additional protocol messages. It does require some additional processing to replace the regular IPv6 addresses in the RRP messages with CGAs and additional fields to send the public key and signature.

## VII. Security Tradeoffs

The security vs. efficiency tradeoff is something we thought about a lot once we starting trying to think of viable solutions to the attacks we found. The first possible solution we considered was one proposed by Professor Mitchell and Arnab Roy at Stanford University to one of the primary designers of the RRP (Tuomas Aura [3]) in 2005. Interestingly, we also got to see Aura's response to the idea. His perspective was that although Roy's fix would have made the protocol more secure, he could not justify the associated implementation and efficiency cost (in this case, forcing all HAs/routers to check for a mobility header). Since increasing security often requires increased processing, additional messages, and/or the general overhead of implementing changes, this tradeoff is something that we decided to consider.

Cost of implementing the two solutions we proposed:

- 1) No changes to the existing messages of the protocol: Requires additional messages and associated processing at the end of the protocol. May be unacceptable if additional messages have too high an implementation cost.
- 2) No additional protocol messages: Requires additional fields within existing messages and processing of CGAs. May be more acceptable than solution 1 if CGA support is already available and overhead is low.

Another consideration that we looked into was the use of Public Key Infrastructure. Although we did not model this version of the protocol, it seems clear that PKI could prevent the attacks we found and provide strong security for MIPv6 Binding Updates (the MN and CN would be able to securely authenticate each other). In addition, it seems that using PKI would most likely reduce the number of messages in the protocol, since the two path-based authentication would no longer be required. However, using PKI would limit the use of MIPv6 Binding Updates to nodes that share access to such infrastructure. For example, this might limit widespread use of MIPv6 Binding Updates to corporate environments where PKI is already widely deployed. While this is not exactly a security vs. efficiency tradeoff, it is a similar situation where security guarantees must be weighed against implementation considerations. The designers of the RRP made the decision that efficiency and widespread deployability were more important than perfect security. [3]

## VIII. Conclusion

After reading the RFC and some related papers, we wrote out the protocol details and started thinking about them in a formalized way. This helped us to understand and start considering possible attacks. By coding this formalization in Murphi, we verified the known attacks that motivated the design of each stage of the protocol. We also discovered two significant attacks to which the final version of the protocol is vulnerable.

Finally, we consider two potential solutions and the associated costs. If found effective, our first idea for a solution could potentially be applied to minimize DoS attack damage in many other situations. Our second idea simply notices that CGAs allow us to apply some of the benefits of public-key signatures without infrastructure. In our last section, we look at some concerns that are central to both the original design of this protocol and to whether our solutions are viable.

## IX. References

- [1] D. Johnson, "Mobility Support in IPv6." RFC 3775, *IETF Network Working Group*, June 2004.
- [2] I. Ahmed, et al., "Binding Update Authentication Scheme for Mobile IPv6," *Information Assurance and Security 2007*, pages 109-114, August 2007.
- [3] T. Aura, "Mobile IPv6 Security." *In Proceedings of Security Protocols, 10th International Workshop*, volume 2845, pages 215-228, Cambridge, UK, April 2002.
- [4] T. Aura, "Cryptographically Generated Addresses (CGA)." RFC 3972, *IETF Network Working Group*, March 2005.

## Appendix A – RRP Protocol Messages

### General Descriptions:

*Kcn*: A “node” key generated by correspondent node that is a random number, 20 octets in length.

*Nonce*: A random number of any length (64 bits is recommended), generated at regular intervals, and may be stored in an array with the nonce index indicating array position

#### 1a: Home Test Init message

*Source*: Home address, *Destination*: correspondent

Contents:

Home init cookie – 64 bit random value

#### 1b: Care-of Test Init message

*Source*: Care-of address, *Destination*: correspondent

Contents:

Care-of cookie – 64 bit random value

#### 2a: Home Test message

*Source*: Correspondent, *Destination*: home address

Contents:

Home init cookie – received from mobile node

Home keygen token – First(64, HMAC\_SHA1(Kcn, (home address | nonce | 0)))

Home nonce index – identifies which nonce is being used in this message

#### 2b: Care-of Test message

*Source*: Correspondent, *Destination*: care-of address

Contents:

Care-of init cookie – received from mobile node

Care-of keygen token – First(64, HMAC\_SHA1(Kcn, (care-of address | nonce | 1)))

Care-of nonce index – identifies which nonce is being used in this message

(Note: nonces in 2a and 2b can be different)

→ Mobile node calculates  $K_{bm} = \text{SHA1}(\text{home keygen token} | \text{care-of keygen token})$

#### 3: Binding update message

*Source*: care-of address, *Destination*: correspondent

Contents:

Sequence number – 16-bit unsigned int

Home nonce index – received from correspondent

Care-of nonce index – received from correspondent

MAC = First(96, HMAC\_SHA1(K<sub>bm</sub>, (care-of address | correspondent | BU message)))

→ Correspondent node verifies the MAC and creates a Binding Cache entry for the mobile.

#### 4: Binding Acknowledgment message (optional)

*Source*: correspondent, *Destination*: care-of address

Contents:

Sequence number – 16-bit unsigned int; same as binding update received

MAC = First(96, HMAC\_SHA1(K<sub>bm</sub>, (care-of address | correspondent | BA message)))