Mobility Optimizations
Internet-Draft
Expires: November 19, 2004

J. Arkko
Ericsson Research NomadicLab
C. Vogt
University of Karlsruhe

May 21, 2004

Credit-Based Authorization for Binding Lifetime Extension
draft-arkko-mipv6-binding-lifetime-extension-00

Status of this Memo

Copyright Notice

Abstract

Mobile IPv6 return routability mechanisms require home and care-of address keygen tokens to be used to authorize a binding update to correspondent nodes. The current rules dictate that such authorization be performed every seven minutes, using tokens at most three and half minutes old. This requirement results in an average signaling traffic of around 7 bits per second when the hosts are not moving around. This traffic load by itself is neglible, but can be problematic for hosts in standby mode. We present a secure and lightweight extension of return routability that can reduce this

signaling load to around 0.1 bits per second, and require hosts to wake up much less frequently.

Table of Contents

1. Introduction

This document focuses on mobile nodes that stay long periods on the same care-of address, and wish to retain efficient routing throughout these periods. In order to keep Mobile IPv6 route optimization state alive, periodic signaling is needed. Such periodic signaling consumes a small amount of bandwidth from the point of view of the participants, but the total amount of signaling load in a commercial network can be large.

More importantly, typical implementations of hand-held devices employ a standby mode in order to conserve battery energy. Periodic signaling causes a need to wake up for such nodes, and can consume extra energy. While it is possible to re-establish route optimization at the time when the mobile node has some actual traffic to send, this will cause additional signaling and delay before actual payload traffic can flow efficiently.

Current Mobile IPv6 return routability mechanisms require home and care-of keygen tokens to be used to authorize a Binding Update to correspondent nodes. According to the current rules, such tokens may be used at most MAX_TOKEN_LIFETIME seconds (3.5 minutes) after they have been acquired in an address test procedure. Section 5.2.7 of the base specification [2] states:

> A fast moving mobile node MAY reuse a recent home keygen token from a correspondent node when moving to a new location, and just acquire a new care-of keygen token to show routability in the new location.

Vogt et al defined the Early Binding Updates [6] procedure to expand on this approach a little bit by suggesting that a pre-emptive home test exchange be used to ensure that a home keygen token is available when it is needed. Early Binding Updates could also be used to perform a care-of address test in parallel with sending payload

traffic. In this application the Early Binding Updates do not fully protect against flooding attacks. This has since then been corrected in [12].

The problem with the base mechanism and the Early Binding Update scheme is, however, that both require extra signaling. A number of proposals have been made to reduce this, see for instance [13], [7], and [8]. Some of these mechanisms are extremely efficient, such as the preconfigured binding keys [13] which adds no signaling at all. Other proposals such as [8] employ techniques such as the Cryptographically Generated Addresses (CGAs) that can achieve both high security and efficiency, particularly for testing home address ownership.

This document explores the design space into a new direction, namely stretching the signaling frequency limits of the current return routability mechanism without introducing new vulnerabilities. Our design is based on explicitly addressing the costs and benefits of attacks. This proposal is not necessarily intended to be a standalone proposal for Mobile IPv6 optimization. Rather, it is intended as a research idea that could be used as a possible component in a solution that addresses all aspects of the optimization problem. The proposal is in its early stages, however, so additional discussion is warranted.

This document is organized as follows. In Section 3 we discuss the performance of the current return routability mechanism. Section 4 describes our solution. Finally, Section 5 evaluates the performance of this solution, and Section 6 analyzes its security implications.

2. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in [1].

3. The Problem

The performance of the current return routability mechanism can be evaluated according to its impact on handover delay, the amount of bandwidth it uses per movement, and the amount of bandwith it uses

when not moving. In this document we focuse only on the last aspect.

Current specifications require a periodic return routability test and the re-establishment of the binding at the correspondent node. One round of a full return routability procedure requires the following messages:

HOTI: This needs 40 + 8 + 8 or 56 bytes.

HOT: This needs 40 + 8 + 16 or 64 bytes.

COTI: This needs 40 + 8 + 8 or 56 bytes.

COT: This needs 40 + 8 + 16 or 64 bytes.

BU: This needs 40 + 6 + 6 + 6 + 2 + 12 or 72 bytes.

BA: This needs 40 + 6 + 6 + 12 or 64 bytes.

Taken together, this results in 376 bytes, or on the average about

7.16 bits/second, if performed every seven minutes. While this is an insignificant bandwidth for nodes that are actually communicating, it can still represent a burden for hosts that just have the bindings ready for a possible packet but are not currently communicating. This can be problematic for hosts in standby mode, for instance.

When two mobile nodes are communicating with each other, these numbers double, i.e., 14.32 bits/second. This is because both nodes need to act as receivers and senders of the messages.

The bandwidth itself may also be an issue in some scenarios. For instance, a correspondent node such as a SIP proxy may have a large number of mobile nodes, and the sum of the small bandwidth from each of them becomes considerable. The traffic load for a home agent may also be significant. In a network of a 100 million hosts, keeping route optimization up all the time between the hosts and some other node would require 220 Mbits/second of traffic through the home agent(s), and 716 Mbit/second for the other nodes altogether.

Note that when evaluating the impact of signaling on performance, one should take into account the whole stack and not inspect just one

layer or task. For instance, if the mobile node actually moved, the above signaling would have to be compared to the link layer signaling, access control and authentication signaling, IPv6 neighbor discovery signaling, and so forth. Such other signaling introduces quite significant delays, but is not relevant for discussion in this draft as we assume a stable mobile node.

4. Protocol Definition

4.1 Overview

We take an approach similar to Credit Based Authorization (CBA) [12], developed from Early Binding Updates [6] and a suggestion to track packet counts instead of a fixed time [9] into a fully flegded credit-based scheme [10, 11, 12]. These techniques are specific instances of microeconomics-based solutions to security problems [3, 4].

CBA for Early Binding Updates [12] focuses on the care-of address tests; the objective is to weigh the effort the mobile node has spent in communicating with the correspondent node, and to ensure that any unverified communication sent to a claimed new address causes less damage than this effort. The rationale is that even without Mobile IPv6, attackers can easily cause similar flooding attacks by sending packets to the victim directly or via a reflector; the real issue is avoiding amplification. By ensuring that the amplification achieved via Early Binding Updates is smaller than the amplification achieved

via these other attacks, the overall seriousness of vulnerabilities is not increased.

In our approach, the same idea is used to reduce the amount of signaling required for address tests. The primary reason why address tests are needed is to ensure that a mobile node "owns" its home address and is reachable at the care-of addresses in question. In basic Mobile IPv6 design, this is achieved by a testing that the mobile node can receive packets at the given address, i.e., is on the routing path to that address. If this test was not regularly done, attackers who only visited the path for a short time could claim address ownership for a long time. This vulnerability does not exist in the Internet today, as was hence considered as something that should be avoided [5].

However, the effort-damage considerations from the Early Binding Updates can be applied also to the frequency of address tests. Assuming at least one test is performed, the frequency of the tests is not related to flooding. Instead, the efforts and damage must be counted in different units than in Early Binding Updates. Specifically, we can track the amount of time the mobile node has been reachable at its home address, and we can set a limit on how long the mobile node can continue to claim this without performing a new test. So, instead of the current fixed limit, the mobile node can, for instance, continue to use an existing address test 30% longer than the time it has already been reachable at this address. The only thing that is needed is that the mobile node can prove it is still the same node than it has been at the time of previous tests.

4.2 Behaviour

This section shows the steps of the protocol. For brevity, we omit the description of packet formats.

First, the mobile node establishes a binding cache entry:

Step 1. MN->HA->CN: Home test init

Step 2. CN->HA->MN: Home test

Step 3. MN->CN: Care-of test init

Step 4. MN->CN: Care-of test

Step 5. MN->CN: Binding Update + Lifetime Credit Authorization option

Step 6. MN->CN: Binding Acknowledgement + Lifetime Credit Authorization option

These steps are equivalent to a standard correspondent binding update procedure, except that the initial lifetime specified MUST be less than or equal to 30% of MAX_RR_BINDING_LIFETIME, and that the Lifetime Credit Authorization option is included in the Binding

Update and Acknowledgement. The contents of the option is based on the Kbm, and it will be discussed in more detail later.

After time t, the mobile node needs to re-establish the binding, as its lifetime is about to run out. (Note that the binding may have been redone several times during time t; the important factor is the total amount of time the binding has existed.)

Step 7. MN->HA->CN: Home test init

Step 8. CN->HA->MN: Home test

Step 9. MN->CN: Care-of test init

Step 10. MN->CN: Care-of test

A new return routability procedure is run here, again in the standard manner.

Step 11. MN->CN: Binding update + Lifetime Credit Authorization option

The requested lifetime in the Binding Update is not limited to MAX_RR_BINDING_LIFETIME (7 minutes) but rather t * 0.3. To authorize this, the mobile node has to provide a keyed hash using the key Kcredit, proving that it has participated in all the binding updates between Step 5 and Step 10. Kcredit is calculated as follows:

Kcredit = hash(KbmN | hash(KbmN-1 | hash(KbmN-2 | ...Kbm1)))

Here Kbm1 through KbmN represent the Kbm used to calculate the Binding Authorization Data option in the Step 5 binding update and all subsequent binding updates. Note that neither the mobile nor the correspondent node needs to remember the whole sequence, as they can calculate the next Kcredit value based on the previous Kcredit value and the latest Kbm. However, in order to know Kcredit one has to have had knowledge of all Kbm values.

We set an upper bound for these lifetimes in order to ensure that the system can still recover. The upper bound is 8 hours.

Finally, the correspondent node responds:

Step 12. CN->MN: Binding Acknowledgement + Lifetime Credit Authorization option

The returned Lifetime Credit Authorization option assures the mobile node that the correspondent node is also still the same node it has been in the past. It also informs the mobile node that it supports this extension. The returned lifetime is set according to the correspondent node's calculation of the time t.

Note that we did not propose any modifications to the actual return routability test, binding updates, or the timing of these events with respect to data packet flows. The solution outlined here can be used in conjunction with other optimizations, such as those defined in [12].

4.3 State Requirements

Both mobile and correspondent nodes hold some state in the Binding Cache Entries, related to the credit authorization. The following conceptual information MUST be kept:

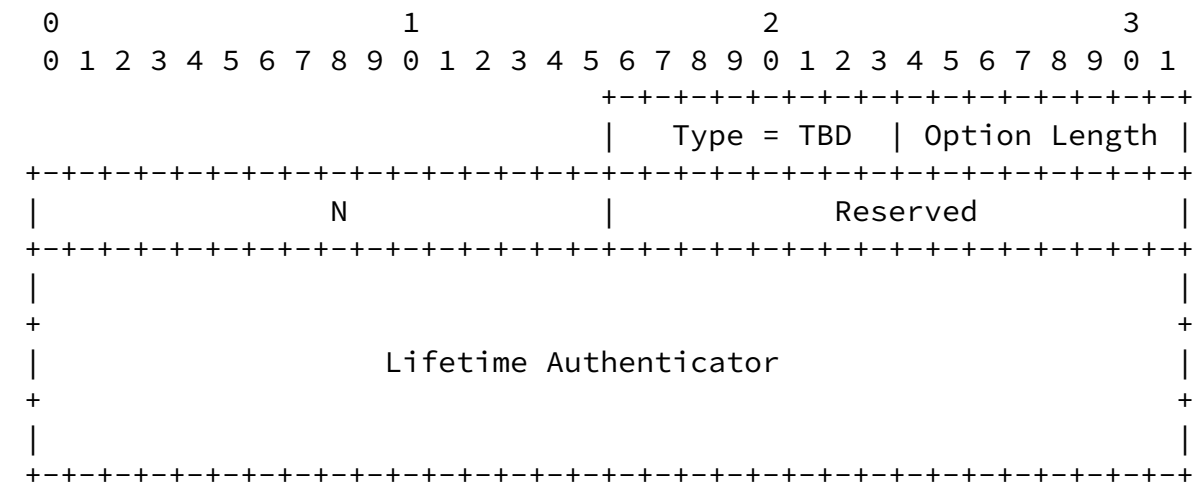
- The total time there has been a binding for this home address.
- The current Kcredit value.
- The number of Kbm values included in the Kcredit value.

4.4 Lifetime Credit Authorization Option

This extension introduces one new mobility option, the Lifetime Credit Authorization option.

This option is similar to the Binding Authorization Data option, but uses Kcredit as the key instead of Kbm.

The Lifetime Credit Authorization option does not have alignment requirements. The format of this option is as follows:



This option is valid in the Binding Update and Binding Acknowledgement messages. Its field are filled as follows:

Type

TBD < To Be Assigned By IANA >

Option Length

This field contains the length of the authenticator in octets.

N

16-bit counter, representing the number of Kbm keys included in Kcredit so far. This value is used by the correspondent node to ensure that it is synchronized with the mobile node regarding the keying material. If the value sent by the mobile node differs from the expected value, the correspondent node MUST send a value of 1 in the Binding Acknowledgement and reset the lifetime to the initial value (see Section Section 4.2).

Reserved

16-bit reserved field. The value MUST be initialized to zero by the sender, and MUST be ignored by the receiver.

Lifetime Authenticator

This field contains a keyed MAC calculated as follows:

```
Mobility Data = care-of address | correspondent | MH Data
Lifetime Authenticator = First (96, HMAC_SHA1 (Kcredit,
                                                Mobility Data))
```


Where | denotes concatenation and "correspondent" is the IPv6 address of the correspondent node. Note that, if the message is sent to a destination which is itself mobile, the "correspondent" address may not be the address found in the Destination Address field of the IPv6 header; instead the home address from the type 2 Routing header should be used.

"MH Data" is the content of the Mobility Header, excluding the Lifetime Authenticator field itself and the Authenticator field from the Binding Authorization Data option. The Lifetime Authenticator value is calculated as if the Checksum field in the Mobility Header was zero. The Checksum in the transmitted packet is still calculated in the usual manner, with the calculated Authenticator being a part of the packet protected by the Checksum.

The first 96 bits from the MAC result are used as the Lifetime Authenticator field.

5. Performance Considerations

Initially, the token and BCE lifetimes provided by this scheme are smaller than those in the current return routability method. This provides additional security against attackers that just came on the link. However, after a while the lifetimes become higher and there's a significant reduction in the need for signaling. For instance, after the binding has been up for an hour, home address tests can be performed as infrequently as once every eighteen minutes compared to the standard seven minutes.

For a connection that stays up continuously, the lifetimes approach the maximum lifetimes (eight hours), which implies that at least three return routability protocol runs have to be performed per day. The signaling load this of this is around 0.104 bits per second, or

68 times less than in the baseline method.

6. Security Considerations

The security of this approach is based on the following principles:

- The bindings resulting from running this method are not permanent, i.e., can be overridden at any time by a new run of the return routability procedure and binding procedures. This avoids problems associated with attackers grabbing a binding before legitimate nodes.

- With the timing formula that is used, it is guaranteed that whatever exposure there is for on-path attackers, this method increases this exposure by a known amount (30%). It is already known that the only vulnerability in the original return routability mechanism is a slight, constant, increase in exposure to on-path attackers. This problem is called the time shifting vulnerability in [5]. The difference between original return routability and this method is that the exposure increase is variable instead of a constant. In both cases it is, however, limited and quantifiable.

- Depending on the amount of time the node has been on the link, this method provides either a smaller or larger window of vulnerability. We argue that this is at least as reasonable as the constant windows in RR.

Attackers who are temporarily on the path between the mobile and correspondent nodes (and simultaneously also on the path between the home agent and correspondent node) can fraudulently represent the mobile node in the return routability procedure. This implies that the attacker can get one Kbm value. With this value, it can register a false binding, de-register the existing binding and zero the credit collected by the mobile node, or introduce a new Kbm into the Kcredit value, making it impossible for the mobile node to use its credit. These are denial-of-service attacks; our method is incapable of ensuring that the credit can be retained in the presence of on-path attackers. On the other hand, base Mobile IPv6 mechanisms have similar limitations, and even basic IPv6 is vulnerable to on-path denial-of-service attacks.

7. IANA Considerations

One new mobility option number has to be allocated for this protocol.

8. Conclusions

This approach can reduce the amount of signaling needed for home address tests, care-of address tests, and binding updates, all without exposing nodes to significant new vulnerabilities. It does not eliminate all the signaling, however, and works best when the mobile node stays a long time at the same location.

This approach is one possible component in further optimizations of Mobile IPv6. As its primary purpose is to reduce signaling, it can be used together with other approaches such as [13] to assist in reducing the frequency of care-of address tests and expand the applicability of the solution, with [12] to provide both reduced signaling and reduced latency upon movements, or with [8] to provide

reduced signaling while still performing care-of address tests.

Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

Informative References

[3] Anderson, R., "Why information security is hard - an economic perspective", Proceedings of the 17th Annual Computer Security Applications Conference, December 2001.

[4] Arkko, J. and P. Nikander, "Weak Authentication: How to Authenticate Unknown Principals without Trusted Parties", Proceedings of Security Protocols Workshop 2002, Cambridge, UK, April 16-19, 2002.

[5] Nikander, P., "Mobile IP version 6 Route Optimization Security

Design Background", draft-ietf-mip6-ro-sec-00 (work in progress), April 2004.

[6] Vogt, C., Bless, R., Doll, M. and T. Kuefner, "Early Binding Updates for Mobile IPv6", draft-vogt-mip6-early-binding-updates-00 (work in progress), February 2004.

[7] Haddad, W., Dupont, F., Madour, L., Krishnan, S. and S. Park, "Optimizing Mobile IPv6 (OMIPv6)", draft-haddad-mipv6-omipv6-01 (work in progress), February 2004.

[8] Haddad, W., Madour, L., Arkko, J. and F. Dupont, "Applying Cryptographically Generated Addresses to OMIPv6 (OMIPv6+)", draft-haddad-mip6-cga-omipv6-00 (work in progress), April 2004.

[9] Arkko, J., "Comments on draft-vogt-mip6-early-binding-updates-00", E-mail discussion on the mip6 list, February 2004.

[10] Vogt, C., "Response to comments on draft-vogt-mip6-early-binding-updates-00", E-mail discussion on the mip6 list, February 2004.

[11] Vogt, C., "Early Binding Updates for Mobile IPv6", Presentation in the MOBOPTS IRTF WG, March 2004.

[12] Vogt, C., Arkko, J., Bless, R., Doll, M. and T. Kuefner, "Credit-Based Authorization for Mobile IPv6 Early Binding Updates", draft-vogt-mipv6-credit-based-authorization-00 (work in progress), May 2004.

[13] Perkins, C., "Preconfigured Binding Management Keys for Mobile IPv6", draft-ietf-mip6-precfgKbm-00 (work in progress), April 2004.

Authors' Addresses

Jari Arkko
Ericsson Research NomadicLab

FI-02420 Jorvas
Finland

EMail: jari.arkko@ericsson.com

Christian Vogt
Institute of Telematics
University of Karlsruhe (TH)
P.O. Box 6980
76128 Karlsruhe
Germany

Phone: +49-721-608-8282
Fax: +49-721-388-097
EMail: chvogt@tm.uka.de
URI: http://www.tm.uka.de/~chvogt/

Appendix A. Acknowledgments

The authors would like to thank Wassim Haddad, Lila Madour, Roland Bless, Mark Doll, and Tobias Kuefner for interesting discussions in this problem space.

Intellectual Property Statement

on the IETF's procedures with respect to rights in IETF Documents can be found in BCP 78 and BCP 79.

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

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

Disclaimer of Validity

Copyright Statement

Acknowledgment

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