Network Working Group   C. Vogt
Internet-Draft   Univ. of Karlsruhe, Germany
Expires: August 18, 2005   J. Arkko
                                              Ericsson Research NomadicLab
                                                         February 14, 2005

Credit-Based Authorization for Mobile IPv6 Early Binding Updates
draft-vogt-mobopts-credit-based-authorization-00.txt

Abstract

The latency associated with Mobile IPv6's Return Routability test can have an adverse impact on delay-sensitive applications. Early Binding Updates mitigate this issue by already using a new care-of address in parallel with testing it. We propose and analyze a credit-based mechanism that prevents misuse of Early Binding Updates

for amplified flooding attacks and discourages such misuse for non-amplified flooding attacks.

Table of Contents

1. Introduction

Along with the introduction of mobility support in the Internet comes the concern about a new family of previously unknown attacks. The potential misuse of mobility-management protocols is multifold, ranging from resource-exhaustion attacks and redirection-based flooding attacks to masquerading attacks and attacks against data confidentiality or integrity [4]. Mobility-related attacks are of great concern because any node in the Internet is a potential victim. A victim does not even have to be mobile. The problem is that the victim has little means to protect itself, whereas the party that is in a position to build an effective security mechanism does not directly benefit from its investments into this mechanism [8].

There is hence a strong need that a mobility-management protocol, like Mobile IPv6, incorporates the required security mechanisms in the first place. Mobile IPv6 has been carefully designed with regard to this. Malicious binding updates with home agents are discouraged by a mandatory security and trust relationship between a mobile node and its home agent. Since neither a security nor a trust relationship can be presupposed to exist between a mobile node and a correspondent node, Mobile IPv6 prevents malicious binding updates with correspondent nodes, when using Route Optimization, through a home-address test and a care-of-address test. A mobile node must pass both tests in order to authenticate its home address and care-of address during the binding-update phase. Home-address authentication verifies that the mobile node is the legitimate owner of the home address. It prevents masquerading attacks and attacks against data confidentiality or integrity. Care-of-address authentication verifies that the mobile node is actually present at the new care-of address. It prevents a malicious node from flooding a third party with a stream of redirected packets.

A disadvantage of Mobile IPv6's home-address and care-of-address tests is that both tests have to be accomplished before a mobile node can initiate a binding update. The latency of registering a new care-of address with a correspondent node can therefore be significant, and it can have an adverse impact on delay-sensitive applications. Early Binding Updates for Mobile IPv6 [2], an optional extension to Mobile IPv6 Route Optimization, reduce this latency by moving both tests out of the critical period during which a new care-of address cannot yet be used: A "prophylactic home-address test" is accomplished before the mobile node changes its point of network attachment, and a "concurrent care-of-address test" runs in parallel with data transfer to and from the new care-of address. The new care-of address is said to be "unconfirmed" until the mobile node authenticates the care-of address to the correspondent node, and it is called "confirmed" thereafter.

Section 7 of [2] furnishes evidence that the security of Early Binding Updates is equivalent to that of standard Mobile IPv6 except for a new vulnerability to flooding attacks based on packet redirection towards an unconfirmed care-of address. We show in Section 3 that the data volume and rate of a redirection-based flooding attack can be substantially higher than the data volume and rate generated by the attacker itself. No such "amplification" can be achieved through flooding attacks that exist in today's Internet. With the objective that a mobility-management protocol does not introduce new threats in addition to those that already exist, flooding-attack amplification, be it with respect to data volume or data rate, is certainly unacceptable. In this document, we propose a mechanism, Credit-Based Authorization, that prevents misuse of Early Binding Updates for amplified, redirection-based flooding attacks. Through proper parameterization, Credit-Based Authorization can discourage misuse of Early Binding Updates even for non-amplified, redirection-based flooding attacks.

Credit-Based Authorization limits the data volume that a correspondent node can send to a mobile node's unconfirmed care-of address, and it limits the data rate at which the correspondent node can send this data. The maximum data volume and rate depend on how much data the mobile node has sent to, or received from, the correspondent node during the last few seconds. With Credit-Based Authorization, a correspondent node measures the "effort" that a mobile node spends for sending or receiving packets, and it grants "credit" for this effort. The mobile node needs credit during the binding-update phase: If it has enough credit, the mobile node can use a new, unconfirmed care-of address for both sending and receiving packets. Without sufficient credit, the mobile node can use a new, unconfirmed care-of address only for packets that it sends to the correspondent node. The correspondent node then sends packets for the mobile node to the mobile node's home address until the mobile node authenticates its care-of address, and the care-of address becomes confirmed. We show that Credit-Based Authorization can be realized at reasonable deployment costs.

The applicability of Credit-Based Authorization goes beyond the protection against misuse of Early Binding Updates. In [5], a credit-based approach is used to incrementally extend binding lifetimes. This reduces the signaling overhead required for binding refreshes. We are also working on an extended version of Early Binding Updates which allows a mobile node to already register a new care-of address from its old point of network attachment. Such "Anticipated Binding Updates" would naturally depend on something similar to Mobile IPv6's Alternate Care-of-Address option. We believe that Credit-Based Authorization can help to make Anticipated Binding Updates a secure and, thus, realistic option.

The rest of this document is structured as follows: We define a set of key terms in Section 2. In Section 3, we compare the types of flooding attacks that already exist in today's Internet to those that could become possible by the introduction of mobility support if no preventive measures are taken. In Section 4, we describe Early Binding Updates in a nutshell, and we sketch the idea of Credit-Based Authorization. We show that there are two variants of Credit-Based Authorization: The first variant measures a mobile node's effort for sending packets to the correspondent node, the second variant measures a mobile node's effort for receiving packets from the correspondent node. Either variant comes with its particular advantages and drawbacks. We detail both variants in Section 5. In section Section 6, we discuss potential security concerns which one might have with measuring a mobile node's effort for receiving packets from the correspondent node. We propose a mechanism, periodic care-of-address spot checks, which can disperse these concerns. Security considerations follow in Section 7. We conclude in Section 8.

Credit-Based Authorization for Mobile IPv6 Early Binding Updates was first presented at the 59th IETF meeting in Seoul, Republic of Korea, in February 2004.

2. Terminology

Effort
   Effort is a mobile node's investment in terms of bandwidth, processing power, and memory. There are two types of effort which a mobile node typically spends: effort for sending packets to the correspondent node and effort for receiving packets from the correspondent node.

Credit
   Credit is a unit used by a correspondent node to determine how much data it can send to a particular mobile node. A mobile node earns credit by investing effort.

Amplification
   Amplification describes the severity of a flooding attack. It is defined as the ratio between the data volume and rate that the attacked victim is exposed to and the data volume and rate that the attacker itself generates.

Early Binding Update message

An Early Binding Update message equals a standard Binding Update message except that it only authenticates the mobile node's home address. When a mobile node changes its point of network attachment and configures a new care-of address, the mobile node sends to the correspondent node an Early Binding Update message in order to tentatively register the new care-of address with the correspondent node [2].

Early Binding Acknowledgement message
A correspondent node sends to a mobile node an Early Binding Acknowledgement message in order to indicate successful reception of an Early Binding Update message [2].

Prophylactic home-address test
A prophylactic home-address test is a home-address test which a mobile node performs before it changes its point of network attachment. The mobile node executes a prophylactic home-address test whenever a handover is imminent. Alternatively, if handovers cannot be anticipated, the mobile node should do a prophylactic home-address test periodically [2].

Concurrent care-of-address test
A concurrent care-of-address test is a care-of-address test which is performed in parallel with data transfer to and from the tested care-of address [2].

Confirmed care-of address
A confirmed care-of address is a care-of address which a mobile node has registered with a correspondent node by means of a properly authenticated standard Binding Update message [2].

Unconfirmed care-of address
An unconfirmed care-of address is a care-of address which a mobile node has registered with a correspondent node by means of a properly authenticated Early Binding Update message and which has not yet been confirmed by means of a properly authenticated standard Binding Update message [2].

3. Flooding Attacks

Flooding attacks are a variant of denial-of-service attacks. They are characterized by a victim being bombarded with unwanted packets at a rate that the victim, and possibly the victim's access network, cannot handle. In this section, we compare the types of flooding attacks that already exist in today's Internet to those that could become possible by the introduction of mobility support if no preventive measures are taken. We show that the latter are potentially much more severe due to a significantly higher amplification. "Amplification" is the ratio between the data volume and rate that the victim is exposed to and the data volume and rate that the attacker itself generates.

There are essentially two types of flooding attacks that are already possible in today's Internet: direct flooding attacks and reflection attacks. In a direct flooding attack, the attacker itself sends unwanted packets to the victim. In an indirect reflection attack, a third node, the reflection point, sends the packets. The attacker typically uses a known protocol vulnerability to make the reflection point generate these packets [7]. One example is that the attacker sends ICMP Echo Request packets to the reflection point with the packets' source addresses set to the victim's IP address. The reflection point, in turn, sends ICMP Echo Reply packets "back" to the victim. Another example is that the attacker sends TCP SYN packets, again with false source addresses, to the reflection point, which in turn sends TCP SYN-ACK packets to someone who does not expect these packets. Since most TCP servers are configured so that they re-send a TCP SYN packet multiple times when failing to receive an acknowledgement, this reflection attack can even produce a small amplification. Gaining more significant amplification in today's Internet typically requires more complex attack strategies like taking over control of other nodes by spreading compromised code. Such attacks are of a different quality because they change the behavior of infected nodes. We do not discuss these attacks in this document. Instead, we focus on attacks that are based on misuse of vulnerable, but uncompromised protocol behavior. Note that vulnerable protocol behavior could be misused by infected nodes to substantially increase the damage each of them causes.

Given uncompromised behavior of Internet protocols, we make the following two observations: First, both in a direct flooding attack and in a reflection attack, the attacker can conceal its identity by spoofing its packets' source addresses. Reflection attacks are purely based on source-address spoofing. Ingress filtering [6] in the attacker's access network may prevent source-address spoofing, but more than a few access networks do not employ ingress filtering. Second, what limits a flooding attack in today's Internet is the minimum bandwidth along the path from the attacker to the victim, possibly through a reflection point. The data volume and rate that the flooding attack comes to is by and large determined by the data volume and rate that the attacker itself generates, i.e., there is no significant amplification.

The situation can change when we introduce mobility. Suppose a node is allowed to change its IP address without having to evidence that

it is present at the new IP address. Then, an attacker can subscribe, through its own IP address, to a large data flow (e.g., a video stream) offered by some server on the Internet. The attacker can easily accomplish the initial handshake procedure with the server while it uses its own IP address. Once data is flowing, the attacker can redirect the flow to the IP address of an arbitrary third party, the victim. The attacker can use the sequence numbers learned during the initial handshake procedure in order to spoof acknowledgements for packets that it assumes the server has sent to the victim.

The unprecedented severity of redirection-based flooding attacks is that not the attacker, but a faithful server on the Internet can be made generate packets used for an attack. The server does not have to be infected with compromised code, and neither the victim nor the server has to be mobile. The attacker produces as little as spoofed feedback information to keep the data flow alive. To make matters worse, the attacker can redirect to the victim data flows from multiple servers. We observe that the scope of a redirection-based flooding attack is only limited by the data volume and rate that one, or even multiple, servers can generate, as well as the cumulative minimum bandwidths of the paths from each server to the victim. The associated amplification is much higher than in the discussed ICMP and TCP flooding attacks. Hence, there is a strong need that a mobility-management protocol cannot be misused for amplified, redirection-based flooding attacks. In fact, since non-amplified flooding attacks are already possible in today's Internet, protection against non-amplified, redirection-based flooding attacks is less important.

4. Overview

In this section, we provide a brief description of Early Binding Updates for Mobile IPv6, and we sketch the idea of Credit-Based Authorization. We explain how Credit-Based Authorization can discourage misuse of Early Binding Updates for redirection-based flooding attacks, and how it can prevent such misuse for flooding-attack amplification. We show that there are two variants of Credit-Based Authorization: The first variant measures a mobile node's effort for sending packets to the correspondent node, the second variant measures a mobile node's effort for receiving packets from the correspondent node. We discuss the particular advantages and drawbacks of either variant.

We refer to Section 5 for more details on Credit-Based Authorization, and we refer to [2] for a thorough specification of Early Binding Updates.

4.1 Early Binding Updates

When a mobile node recognizes that it is about to change its point of network attachment, the mobile node initiates a prophylactic home-address test. Alternatively, if the mobile node cannot anticipate handovers, it should periodically repeat the home-address test. When the mobile node eventually moves, the mobile node configures a new care-of address and sends to the correspondent node an Early Binding Update message. The prophylactic home-address test allows the mobile node to authenticate its home address in this message.

The mobile node initiates a concurrent care-of-address test as soon as it has dispatched the Early Binding Update message. The mobile node can start using its new care-of address as of then. The correspondent node starts using the mobile node's new care-of address as soon as it receives the Early Binding Update message from the mobile node. Hence, both the mobile node and the correspondent node use the mobile node's new care-of address in parallel with the concurrent care-of-address test being executed.

When the correspondent node receives the Early Binding Update message from the mobile node, it can be confident that the mobile node is the legitimate owner of the home address advertised in this message due to the home-address authentication. Since there is no care-of-address authentication in the Early Binding Update message, the correspondent node does not know at this point whether the mobile node is actually present at its new care-of address. The new care-of address is therefore said to be "unconfirmed". The binding lifetime for an unconfirmed care-of address is limited to a few seconds, TENTATIVE_RR_BINDING_LIFETIME [2].

When the concurrent care-of-address test concludes, the mobile node sends to the correspondent node a standard Binding Update message. This message obeys the formats and semantics defined in [1], i.e., it authenticates both the mobile node's home address and the mobile node's new care-of address. Hence, when the correspondent node receives the standard Binding Update message, it can be confident that the mobile node uses the correct home address, and that the mobile node is actually present at the new care-of address. The correspondent node then changes the status of the new care-of address from "unconfirmed" to "confirmed", and it extends the lifetime of the associated binding to the smaller of the lifetime requested by the mobile node and the maximum lifetime, MAX_RR_BINDING_LIFETIME, according to Section 9.5.1 and Section 9.5.2 of [1].

Due to the lack of care-of-address authentication in the Early Binding Update message, there needs to be additional protection while a mobile node's care-of address is unconfirmed. If no such

protection is provided, a malicious node may misuse Early Binding Updates to register, as an unconfirmed care-of address, the IP address of a third party. The correspondent node would then redirect the attacker's packets to the third party without knowing it. Credit-Based Authorization provides the additional protection that is needed while a mobile node's care-of address is unconfirmed.

4.2 Credit-Based Authorization

Credit-Based Authorization limits the data volume that a correspondent node can send to a mobile node's unconfirmed care-of address, and it limits the rate at which the correspondent node can send this data. Credit-Based Authorization is transparent to the mobile node. There are two variants of Credit-Based Authorization, which determine the maximum data volume and rate in different ways: The first variant limits the data volume and rate according to how much data the mobile node has sent to the correspondent node during the last few seconds. The second variant limits the data volume and rate according to how much data the mobile node has received from the correspondent node during the last few seconds. We will compare both variants in Section 4.3.

The correspondent node measures the "effort" that a mobile node spends for sending or receiving packets, depending on which variant of Credit-Based Authorization is used. Effort is measured in terms of the size of packets recently sent or received. The product of effort and a factor, EFFORT_QUENCH_FACTOR, of less than 1.0 yields the mobile node's "credit". Credit, in turn, authorizes the mobile node to receive packets at an unconfirmed care-of address. For each packet that the correspondent node sends to an unconfirmed care-of address of the mobile node, the correspondent node charges the mobile node credit in the amount of the size of the packet. Through exponential aging, we ensure that the mobile node's credit is at all times determined by the effort that the mobile node has spent throughout the last few seconds.

When the correspondent node has a packet for the mobile node, the correspondent node first checks the state of the mobile node's care-of address. If the care-of address is confirmed, the care-of address has recently been authenticated by the mobile node, and the correspondent node can be confident that the mobile node is actually present at the care-of address. In this case, the correspondent node sends the packet to the mobile node's care-of address as usual, without touching the mobile node's credit. Otherwise, if the mobile node's care-of address is unconfirmed, the correspondent node determines the packet's destination according to how much credit the mobile node has. If the mobile node's credit is more than, or equal

to, the size of the outgoing packet, the correspondent node sends the packet directly to the mobile node's care-of address, and it reduces the mobile node's credit by the size of this packet. If the mobile node's credit is less than the size of the outgoing packet, the correspondent node sends the packet to the mobile node's home address. The packet is forwarded to the mobile node by the mobile node's home agent in this case. Since the mobile node's home address has been authenticated in the Early Binding Update message, the correspondent node can assume that the mobile node is the legitimate owner of the home address [2]. Thus, the correspondent node can send packets to the mobile node's home address without security concerns, and it does not need to reduce the mobile node's credit in this case.

We believe that a faithful mobile node, communicating with a correspondent node in a typical manner, automatically expends effort for sending packets to the correspondent node as well as for receiving packets from the correspondent node. A faithful mobile node hence automatically earns credit due to its normal behavior such that it can receive packets at an unconfirmed care-of address during the binding-update phase. As a consequence, the mobile node does not have to be aware that Credit-Based Authorization is applied at the correspondent node.

4.3 Variants of Credit-Based Authorization

A mobile node spends effort--in terms of bandwidth, processing power, and memory--for sending packets to the correspondent node as well as for receiving packets from the correspondent node. A correspondent node may credit either type of effort. Effort for sending packets can be credited, at the correspondent node, by measuring packets received from the mobile node. Effort for receiving packets can be credited by measuring packets sent to the mobile node while the mobile node's care-of address is confirmed. Both variants have their particular advantages and drawbacks.

The advantage of crediting a mobile node's effort for sending packets is that it is exactly this type of effort that an attacker would have to invest for a direct flooding attack or a reflection attack, both of which are already possible in today's Internet. Through proper selection of the EFFORT_QUENCH_FACTOR protocol-configuration variable, the crediting function can easily be parameterized such that misuse of Early Binding Updates for a redirection-based flooding attack becomes more expensive than a direct flooding attack or a reflection attack. Thus, requiring a node, whether faithful or malicious, to send packets in order to earn credit is a straightforward way to discourage misuse of Early Binding Updates.

The motivation for crediting a mobile node's effort for receiving packets is different. Consider a mobile node running an application which receives much more data from the correspondent node than it sends back to the correspondent node. Streaming applications are probably the most dominant example where user data is transferred one-way only. The correspondent node is typically a server originating a lot of data, whereas the mobile node generates little more than acknowledgements and status information. The huge data stream originating from the correspondent node leads to excessive credit consumption during the binding-update phase. The mobile node, however, can be expected to have a hard time gathering this credit if its credit depends on the packets that it sends to the correspondent node. We believe that crediting the mobile node's effort for receiving packets is the most reasonable approach in this case. Then, assuming that the volume and rate of data originating from the correspondent node does not fluctuate too much--particularly, that there is no significant increase in the generated data volume and rate during the binding-update phase--applications with asymmetric traffic will work fine. We emphasize that only while the mobile node's care-of address is confirmed should the correspondent node grant credit for packets sent to the mobile node. Only then can the correspondent node expect that the mobile node actually receives these packets. Note that this crediting function also works fine for applications with symmetric traffic.

Packet loss on the path from the correspondent node to the mobile node can be an issue when the correspondent node measures a mobile node's effort for receiving packets based on the packets that it has sent to the mobile node. We discuss this issue in Section 6. We show how the correspondent node can use care-of-address spot checks to take packet loss into consideration when it grants credit for packets the mobile node was supposed to receive.

The first variant of Credit-Based Authorization, crediting a mobile node's effort for sending packets, is transparent to the mobile node. The second variant of Credit-Based Authorization, crediting a mobile node's effort for receiving packets, is transparent to the mobile node unless care-of-address spot checks are used.

A correspondent node may credit a mobile node's effort for both, sending packets and receiving packets. After all, the mobile node invests resources for traffic in both directions. The combination of the two variants of Credit-Based Authorization is straightforward. One simply needs to make sure that the total credit a mobile node can earn does not exceed the total effort the mobile node has spent on sending and receiving packets. To simplify matters, we do not explicitly consider the case of crediting traffic in both directions in this document.

5. Detailed Description

In this section, we describe independent implementations for two variants of Credit-Based Authorization. The first variant measures a mobile node's effort for sending packets to the correspondent node, the other variant measures a mobile node's effort for receiving packets from the correspondent node. Both implementations affect the correspondent node only; no changes are needed at the mobile node.

5.1 State at the Correspondent Node

Credit-Based Authorization requires a correspondent node to be aware of the state, confirmed or unconfirmed, of all registered care-of addresses. The state of a care-of address is important to decide whether or not a mobile node needs credit for packets sent to its care-of address. We propose that the correspondent node maintains, for each binding, an additional one-bit flag, CONFIRMED_CARE_OF_ADDRESS, indicating whether or not the respective care-of address is confirmed. CONFIRMED_CARE_OF_ADDRESS equals 1 if the care-of address is confirmed. CONFIRMED_CARE_OF_ADDRESS equals 0 if the care-of address is unconfirmed. Figure 1 shows the two care-of-address states as well as the state transitions that a care-of address can be subject to.

```
   Reordered +------+             Early
       Early |      |             Binding
     Binding |    +-------------+ Update    +-------------+
      Update +--> |             | --------> |             | ---+ Early
                  |  Confirmed  |           | Unconfirmed |    | Binding
    Standard +--> |             | <-------- |             | <--+ Update
     Binding |    +-------------+  Standard +-------------+
      Update |      |               Binding
             +------+                Update
```

Figure 1: Care-of Address States

When a mobile node changes its point of network attachment, it sends to the correspondent node an Early Binding Update message in order to tentatively register its new care-of address. Having sent the Early Binding Update message, the mobile node initiates a concurrent care-of-address test and, once the care-of-address test concludes, sends to the correspondent node a standard Binding Update message in order to confirm its care-of address [2].

When the correspondent node receives the Early Binding Update message from the mobile node, the correspondent node registers the mobile node's new care-of address, and it sets the lifetime of the mobile node's binding to TENTATIVE_RR_BINDING_LIFETIME, according to Section 4.1 of [2]. Beyond this, the correspondent node clears the CONFIRMED_CARE_OF_ADDRESS flag in the mobile node's binding in order to indicate that the new care-of address is unconfirmed at this point. When the correspondent node receives the standard Binding Update message from the mobile node, the correspondent node extends the lifetime of the mobile node's binding to the smaller of the lifetime requested by the mobile node and the maximum lifetime, MAX_RR_BINDING_LIFETIME, according to Section 9.5.1 and Section 9.5.2 of [1]. Moreover, the correspondent node sets the CONFIRMED_CARE_OF_ADDRESS flag in the mobile node's binding in order to indicate that the care-of address is now confirmed. When the mobile node changes its point of network attachment another time, it again sends to the correspondent node an Early Binding Update message, and the procedure repeats itself.

The correspondent node may receive from the mobile node multiple Early Binding Update messages in a row. The messages may carry the same care-of address, for instance, if the concurrent care-of-address test fails for some reason, and the mobile node recognizes, due to a timeout, that either the Care-of Test Init message or the Care-of Test message got lost under way. The mobile node may then re-send both, the Early Binding Update message in order to refresh the tentative binding at the correspondent node, and the Care-of Test Init message in order to re-initiate the concurrent care-of-address test. The Early Binding Update messages may carry different care-of addresses, for instance, if the mobile node changes its point of network attachment two times shortly one after the other, and there is no sufficient time to let the first concurrent care-of-address test conclude.

When the correspondent node receives multiple Early Binding Update messages in a row, the correspondent node handles each message according to the procedure defined in Section 4.1 of [2], regardless of whether this message carries a new or an already registered care-of address. I.e., the correspondent node registers the care-of address and resets the binding lifetime to TENTATIVE_RR_BINDING_LIFETIME. The state of the care-of address remains unconfirmed.

When the correspondent node receives multiple standard Binding Update messages in a row, the correspondent node handles each message according to the procedure defined in Section 9.5.1 and Section 9.5.2 of [1], regardless of whether this message carries a new or an already registered care-of address. I.e., the correspondent node registers the care-of address and resets the binding lifetime to the smaller of the lifetime requested by the mobile node and the maximum

lifetime, MAX_RR_BINDING_LIFETIME. The state of the care-of address remains confirmed.

A mobile node's Early Binding Update message and subsequent standard Binding Update message might get reordered on the way to the correspondent node. In this case, the correspondent node receives from the mobile node a standard Binding Update message before it receives from the same mobile node an Early Binding Update message carrying the same care-of address. Since the two messages carry the same care-of address, they obviously belong together. Thus, when the correspondent node receives an Early Binding Update message carrying an already registered, confirmed care-of address, the correspondent node must ignore the message, keeping both the care-of-address state and the binding lifetime unchanged.

5.2 Exponential Aging

We feel that a mobile node's credit should represent the mobile node's most recently spent effort only. Otherwise, a mobile node may collect credit over a potentially very long period and consume this credit during a very short period. Though unlikely, a malicious node may exploit this property by accumulating a large amount of credit with a relatively slow data stream, and by using this credit, all at once, for a short but potent data burst against an arbitrary victim.

We propose exponential aging to gradually reduce a mobile node's old credit over time. With exponential aging, a mobile node's credit diminishes while the mobile node does not use it. Accumulating credit over a very long period, as well as collecting an indefinite amount of credit, is thus impossible. An implication of this is that exponential aging can limit the rate of packets which are being sent to an unconfirmed care-of address of a mobile node to approximately the rate of packets which have recently been sent to a confirmed care-of address of the same mobile node. In other words, the implication is that exponential aging can limit the rate of packets which are consuming a mobile node's credit to approximately the rate of packets based on which the same mobile node has recently earned this credit. See Section 7.3 for a more detailed explanation on why exponential aging can limit the rate of packets sent to an unconfirmed care-of address.

A precise aging function ages a mobile node's credit whenever this credit is about to be increased or reduced, and it accommodates the time passed since the mobile node's credit was changed before. A precise aging function is very complex, though. We thus propose to simply re-calculate a mobile node's credit in equidistant "crediting intervals" of TENTATIVE_RR_BINDING_LIFETIME length. The effort that

a mobile node has invested throughout a crediting interval is turned into credit after having exponentially aged the mobile node's old credit at the end of the crediting interval. We thus ensure that new credit does not age before it has been around for at least one crediting interval. The correspondent node may synchronize the crediting intervals for all entries in its binding cache. A single clock is therefore sufficient.

Note that TENTATIVE_RR_BINDING_LIFETIME defines both the length of a crediting interval and the binding lifetime for an unconfirmed care-of address [2]. Since the binding lifetime for an unconfirmed care-of address is exactly the time for which the mobile node's credit should be good during the binding-update phase, it makes sense to let a mobile node collect credit during this time and to age the mobile node's credit only when it is older than this time.

An alternative to exponential aging is to simply limit a mobile node's credit by a constant upper bound. A limit certainly prevents a mobile node from collecting an indefinite amount of credit. However, a limit does not prevent a mobile node from keeping collected credit for a long time before using the credit. A limit is also insensitive to throughput conditions on the path between the mobile node and the correspondent node. I.e., a limit for a low-bandwidth connection is certainly inappropriate for a high-bandwidth connection, and vice versa. We thus feel that exponential aging is a more appropriate approach than limiting a mobile node's credit by a constant upper bound.

5.3 Sending Packets to a Mobile Node

Suppose a correspondent node has a packet for a mobile node, and the mobile node's care-of address is confirmed. If the correspondent node credits the mobile node's effort for sending packets to the correspondent node, it does not need to do anything related to Credit-Based Authorization in this case. The correspondent node simply sends the packet to the care-of address (A.1) without changing the mobile node's credit:

```
   (A)   Sending a packet to a confirmed care-of address
         (Credit is given for sending packets)
   ------------------------------------------------------------
   (A.1) send(PACKET,CARE_OF_ADDRESS)
```

If the correspondent node credits the mobile node's effort for receiving packets from the correspondent node, the correspondent node proceeds according to the following algorithm.

```
(A')   Sending a packet to a confirmed care-of address
       (Credit is given for receiving packets)
------------------------------------------------------------
(A'.1) EFFORT := EFFORT + size(PACKET)
(A'.2) send(PACKET,CARE_OF_ADDRESS)
```

The correspondent node measures the mobile node's effort for receiving a packet from the correspondent node in terms of the size of the packet in bytes (A'.1). Effort invested during one crediting interval is accumulated in a variable, EFFORT, which should be sufficiently long not to roll over. The correspondent node maintains this variable for each entry in its binding cache. New credit will be calculated, based on the value of EFFORT, at the end the crediting interval in step (D), after having exponentially aged any old credit. We thus ensure that new credit does not age before it has been around for at least one crediting interval. In step (A'.2), the packet is sent to the mobile node's care-of address.

Now suppose that the correspondent node has a packet for a mobile node, and the mobile node's care-of address is unconfirmed. In this case, the correspondent node determines the packet's destination according to the following steps, independently of whether the correspondent node credits the mobile node's effort for sending or for receiving packets.

```
(B)   Sending a packet to an unconfirmed care-of address
      (Credit is given for sending or receiving packets)
------------------------------------------------------------
(B.1) If CREDIT >= size(PACKET)
      Then
(B.2)    CREDIT := CREDIT - size(PACKET)
(B.3)    send(PACKET,CARE_OF_ADDRESS)
(B.4) Else
(B.5)    CREDIT := 0
(B.6)    send(PACKET,HOME_ADDRESS)
      EndIf
```

The correspondent node keeps the mobile node's credit in a variable, CREDIT, which should be sufficiently long not to roll over. The correspondent node maintains this variable for each entry in its binding cache. Provided that CREDIT is greater than or equal to the size of the outgoing packet in bytes (B.1), CREDIT is reduced by this packet size (B.2), and the packet is sent to the mobile node's care-of address (B.3). Otherwise, if CREDIT is smaller than the size of the outgoing packet in bytes (B.4), any remaining credit is eliminated (B.5) in order not to switch between the mobile node's home address and current care-of address multiple times during a single binding-update phase in case packet sizes differ. The packet

is then sent to the mobile node's home address (B.6). Note that CREDIT never becomes a negative value.

Note that, even when the mobile node's care-of address is unconfirmed, the correspondent node can assume that the mobile node is the legitimate owner of the registered home address, and it can send packets to this home address without security concerns. This is because the correspondent node has recently received from the mobile node an Early Binding Update message in which the mobile node's home address has been authenticated [2]. However, differences in propagation delay between packets directly relayed between the mobile node and the correspondent node and packets passing the mobile node's home agent may have an adverse performance impact on transport-layer protocols and application behavior. It is subject to further research how significant this impact can be.

The amount of available credit does not impact the route taken by those packets which the mobile node sends to the correspondent node. The correspondent node can safely accept a packet sent from the mobile node's care-of address even if this care-of address is unconfirmed and CREDIT is smaller than the size of the packet. The reason is that packets sent from the mobile node to the correspondent node cannot leverage a flooding attack other than a direct flooding attack or a reflection attack, both of which are already possible in today's Internet (Section 3). This implies that a mobile node can send packets to the correspondent node from a new care-of address immediately after having dispatched an Early Binding Update message for this care-of address. The mobile node does not need to be aware that the correspondent node uses Credit-Based Authorization, and it does not need to know how much credit it has.

5.4 Receiving Packets from a Mobile Node

Suppose a correspondent node receives a packet from a mobile node. If the correspondent node credits the mobile node's effort for sending packets to the correspondent node, the correspondent node executes the following algorithm independently of whether the mobile node's care-of address is confirmed or unconfirmed.

```
(C)   Receiving a packet
      (Credit is given for receiving packets)
------------------------------------------------------------
(C.1) EFFORT := EFFORT + size(PACKET)
(C.2) deliver(PACKET)
```

The correspondent node measures the mobile node's effort for sending a packet to the correspondent node in terms of the size of the packet

in bytes (C.1). The packet is delivered to the application in step (C.2).

If the correspondent node credits the mobile node's effort for receiving packets from the correspondent node, the correspondent node does not need to do anything specific to Credit-Based Authorization. It simply delivers the packet to the application (C'.1) without changing the mobile node's credit:

```
(C')   Receiving a packet
       (Credit is given for sending packets)
------------------------------------------------------------
(C'.1) deliver(PACKET)
```

5.5 Handling Clock Ticks

The correspondent node may synchronize the crediting intervals for all entries in its binding cache. A single clock marking the end of a crediting interval is therefore sufficient. Upon a clock tick, the correspondent node re-calculates the credit for each entry in its binding cache according to the following formula. This formula is used independently of whether the correspondent node credits the mobile node's effort for sending or for receiving packets. There is no distinction between bindings with a confirmed care-of address and bindings with an unconfirmed care-of address.

```
(D)   Handling a clock tick
      (Credit is given for sending or receiving packets)
------------------------------------------------------------
(D.1) For Each Binding Do
(D.2)    NEW_CREDIT := EFFORT_QUENCH_FACTOR * EFFORT
(D.3)    CREDIT := CREDIT * CREDIT_AGING_FACTOR \
                   + NEW_CREDIT
(D.4)    EFFORT := 0
      EndFor
```

The correspondent node re-computes the credit for each entry in its binding cache (D.1). A particular mobile node's new credit, NEW_CREDIT, equals EFFORT multiplied by EFFORT_QUENCH_FACTOR (D.2). NEW_CREDIT is a helper variable used in this specification for clarity purposes only. NEW_CREDIT is added to the aged value of CREDIT in step (D.3). We use the aging factor, CREDIT_AGING_FACTOR, proposed in Section 9. In step (D.4), the correspondent node resets EFFORT to zero.

EFFORT_QUENCH_FACTOR is intended to optionally choke the increase of a mobile node's credit. If EFFORT_QUENCH_FACTOR is smaller than 1.0, a mobile node is forced to invest more effort, through sending or receiving packets, than it gets back from a correspondent node while its care-of address is unconfirmed. Therefore, the smaller EFFORT_QUENCH_FACTOR is chosen, the less efficiently can Early Binding Updates be misused for a flooding attack. If EFFORT_QUENCH_FACTOR is set to 0.0, the correspondent node does not send any packets to an unconfirmed care-of address, but it sends these packets to the mobile node's home agent. Obviously, EFFORT_QUENCH_FACTOR is ineffective if set to 1.0. We propose the value given in Section 9.

We believe that the following observation is worthwhile mentioning: If the values of EFFORT_QUENCH_FACTOR and CREDIT_AGING_FACTOR are 0.5 or less, the theory of infinite series tells that, assuming EFFORT is constant throughout all crediting intervals, CREDIT approaches, but never exceeds, the value of EFFORT.

6. Care-of-Address Spot Checks

A correspondent node may credit a mobile node's effort for sending packets to the correspondent node, or it may credit a mobile node's effort for receiving packets from the correspondent node (Section 4.3). In this section, we discuss security concerns that one might have with the latter approach. Although we provide rationale that such security concerns are of less importance than they may seem to be, we propose an optional mechanism, care-of-address spot checks, that can disperse these security concerns.

6.1 Introduction

The question with crediting a mobile node's effort for receiving packets from a correspondent node is how the correspondent node can measure this effort. Limiting effort measurements to confirmed care-of addresses may not be sufficient in case of packet loss along the path from the correspondent node to the mobile node. If a router drops packets due to congestion, the mobile node receives less data than the correspondent node has originally sent, even though the mobile node's care-of address is confirmed. Thus, the mobile node may earn credit for data that was lost and that it has never received. A malicious node may even be able to position itself behind a bottleneck router on purpose, and it may spoof acknowledgements to encourage the correspondent node to send more data.

Fortunately, there is an argument that the described attack scenario is less likely to occur than it may seem: As explained in Section 5.2 and Section 7.3, exponential aging ensures that packets used to collect credit must occur at approximately the same rate as packets that consume this credit. Consequently, if an attacker hid behind a bottleneck router, illegitimately collecting credit for a later flooding attack against a third party, the workload exposed to the bottleneck router would have to be comparable with the magnitude of the flooding attack to come upon the third party. Moreover, a bottleneck router is typically located at the edge of the Internet, presumably in the access network itself. Abnormal workload of a router should draw the attention of the access network's administrator, who could easily identify the perpetrator for two reasons: First, the attacker's care-of address would show up as the destination address of all packets causing the high workload. The care-of address would reveal the attacker's identity because it would have to be confirmed in this case. Second, the attacker's home address would show up in the Type-2 Routing extension header mandatorily attached to all packets causing the high workload. The home address, too, would reveal the attacker's identity independently of the care-of-address state, because it would have been authenticated in all recently transmitted Early Binding Update messages and standard Binding Update messages.

Note that the attacker may spoof acknowledgements in order to accelerate the packet stream originating from the correspondent node. The resulting packet flood might go beyond the capacity not only of the attacker's access network, but also of some link deeper in the Internet. It can be expected, though, that congestion inside the Internet does not mitigate the flood upon the attacker's access network due to a higher available bandwidth inside the Internet. Hence, the attacker should be identified even in this situation.

These observations suggest that no attractive new type of flooding attack is introduced when a correspondent node credits a mobile node's effort for receiving packets from the correspondent node. On the other hand, we recognize the usefulness of some sort of feedback mechanism that can help the correspondent node determine how much data a mobile node really receives. Transport- or application-layer protocols may provide this feedback. We propose a network-layer approach, periodic care-of-address spot checks, which can disperse the security concerns explained above. Our work on care-of-address spot checks is in its very early stages. We decided to present care-of-address spot checks in this document in order to spark a discussion on their practicality and sufficiency.

6.2 Overview

Care-of-address spot checks periodically probe a mobile node's presence at a confirmed care-of address. They are used to approximately determine the congestion on the path between the correspondent node and the mobile node. The correspondent node calculates the mobile node's credit based on the packets sent to the mobile node and the measured congestion. The more congestion the correspondent node perceives, the fewer packets is the mobile node expected to have received, and the less credit is given to the mobile node. Thus, care-of-address spot checks provide reasonable certainty that a mobile node earns credit only for packets that it actually receives. Spot-check-related signaling data is piggybacked to user-data packets to reduce the bandwidth required for care-of-address spot checks to a minimum. Care-of-address spot checks are, by definition, unnecessary if there are no user-data packets to which spot-check-related signaling data can be attached.

Care-of-address spot checks are initiated by the correspondent node. When a care-of-address spot check is due for a particular mobile node, the correspondent node attaches a random string to the next packet that it sends to the mobile node. In case the packet gets through to the mobile node, the mobile node retrieves the random string from the received packet, and it sends the random string back with the next packet addressed to the correspondent node. The mobile node is said to "pass" the care-of-address spot check if the correspondent node finds that the returned random string matches the random string originally sent to the mobile node.

An important rule in the security design for Mobile IPv6 is that all signaling is initiated by the mobile node, and that the correspondent node only responds to the source address from which it has received a request. This rule ensures that a malicious node cannot redirect a correspondent node's response except by spoofing a request's source address. Mobile IPv6's home-address and care-of-address tests, both of which are initiated by the mobile node, are a good example where this rule has left a mark. See Section 3.3.3 of [3] for details on this.

One may argue that care-of-address spot checks violate the described security-design rule since care-of-address spot checks are initiated by the correspondent node. This, however, is not the case for two reasons. First, random strings are attached to packets which would anyway be sent. The increase in packet size, which is due to an attached random string, should be acceptable. Second, care-of-address spot checks are exclusively initiated for confirmed care-of addresses. Therefore, the correspondent node can rest assured that packets carrying a random string cannot be redirected to

spoofed care-of addresses, even though some of these packets may be dropped under way.

6.3 Generating Random Strings

There are multiple ways in which a correspondent node can generate the random strings that it needs for care-of-address spot checks. For instance, the correspondent node may generate a new random string for each care-of-address spot check. An obvious disadvantage of this approach, though, is that the number of random strings which the correspondent node must remember grows with the number of pending care-of-address spot checks.

Mobile IPv6 uses Care-of Keygen Tokens [1] for care-of-address tests, which can be handled in a more economic way. The beauty of Care-of Keygen Tokens is that the correspondent node does not have to explicitly store them. Instead, the correspondent node can re-compute a mobile node's Care-of Keygen Token on demand given a nonce, which can be re-used for multiple mobile nodes, and the mobile node's care-of address. All the correspondent node needs to remember is a set of nonces, the size of which is independent of the number of pending care-of-address tests. This kind of resource preservation is an important precaution against denial-of-service attacks that aim at exhausting the correspondent node's resources [3][4].

One may argue that protection against denial-of-service attacks is less important in the case of care-of-address spot checks than it is in the case of care-of-address tests. After all, both a mobile node's home address and care-of address are already authenticated during a care-of-address spot check. Computational efficiency may therefore be more important than storage efficiency. This is a fundamental difference to standard care-of-address tests. Nonetheless, in order to simplify our proposed implementation of care-of-address spot checks, we re-use existing Mobile IPv6 code for handling Care-of Keygen Tokens.

Note that nonces used for care-of-address spot checks typically have a much shorter lifetime, and are more frequently re-produced, than nonces used for care-of-address tests. We therefore use separate nonce sets for care-of-address tests and care-of-address spot checks. Henceforth, when we mention nonces or Care-of Keygen Tokens, we are referring to those used for care-of-address spot checks rather than care-of-address tests.

We propose a new option, a Spot Check option, for the IPv6 Destination Options extension header. When a correspondent node initiates a spot check of a mobile node's care-of address, it uses a

Spot Check option to carry to the mobile node a Care-of Keygen Token and the index of the nonce used to produce this Care-of Keygen Token. Likewise, the mobile node uses a Spot Check option to return the received Care-of Keygen Token and nonce index to the correspondent node. The mobile node may use a single Spot Check option to return multiple pairs of Care-of Keygen Tokens and nonce indices to the correspondent node. This can be helpful in case the rate at which the mobile node sends packets to the correspondent node is less than the frequency of care-of-address spot checks.

One may deploy a mechanism other than Care-of Keygen Tokens to generate and verify random strings for care-of-address spot checks. Although we do not discuss any such mechanism in this document, we emphasize that a random string, regardless of how it is created, must be sufficiently long to discourage a malicious node from guessing the string. Moreover, random-string generation must not be predictable, and there must be different random strings for different mobile nodes.

6.4 Spot Check Frequency

A correspondent node calculates a mobile node's new credit at the end of each crediting interval. An important design choice is hence how many care-of-address spot checks the correspondent node should initiate for a particular mobile node during one crediting interval. Obviously, there is a trade-off: On one hand, the higher the frequency of spot checks, the more accurately can the correspondent node determine the level of congestion on the path towards a mobile node. The frequency of care-of-address spot checks should therefore be high. On the other hand, each care-of-address spot check has an associated processing overhead--not so much at the mobile node as at the correspondent node. The frequency of care-of-address spot checks should therefore be limited in order to contain this overhead. We propose a maximum number of MAXIMUM_SPOT_CHECKS spot checks per crediting interval (Section 9). The correspondent node then initiates at most one care-of-address spot check per mobile node every TENTATIVE_RR_BINDING_LIFETIME/MAXIMUM_SPOT_CHECKS time. The correspondent node may initiate less care-of-address spot checks for a particular mobile node if there are, at times, no packets to be sent to the mobile node, or if the mobile node's care-of address is unconfirmed during part of the crediting interval.

6.5 State at the Correspondent Node

The time required to complete a care-of-address spot check not only depends on the round-trip time between a correspondent node and a mobile node, but also on the availability of packets which can carry a Spot Check option. However, neither the round-trip time nor the availability of packets is known in advance. Since a mobile node must have enough time to return a received Spot Check option, the correspondent node should be able to re-produce Care-of Keygen Tokens for a sufficiently long time. We propose that the correspondent node keeps the MAXIMUM_SPOT_CHECKS recently generated nonces. This allows it to re-produce one crediting interval worth of Care-of Keygen Tokens. We store these nonces in an array, NONCE[], with MAXIMUM_SPOT_CHECKS slots. The currently oldest nonce is replaced by a new nonce every TENTATIVE_RR_BINDING_LIFETIME/MAXIMUM_SPOT_CHECKS time. Consequently, any nonce is valid for a period of TENTATIVE_RR_BINDING_LIFETIME length, and a care-of-address spot check should not take longer than this time to complete. An index, CURRENT_NONCE_INDEX, points to the most recently generated nonce in the NONCE[] array.

The correspondent node needs to remember which nonces it has used for spot checks of a particular mobile node's care-of address. Therefore, we extend each entry in the correspondent node's binding cache by an array, INITIATED[], of MAXIMUM_SPOT_CHECKS bits' length. Given a binding for a particular mobile node, INITIATED[i] is one if and only if NONCE[i] was used for a spot check of this mobile node's care-of address. When NONCE[CURRENT_NONCE_INDEX] is replaced by a new value, INITIATED[CURRENT_NONCE_INDEX] is set to zero for all entries in the correspondent node's binding cache.

Moreover, the correspondent node needs to remember whether an initiated care-of-address spot check has been passed by a particular mobile node. This information is needed to compute the mobile node's credit at the end of a crediting interval. For this, we extend each entry in the correspondent node's binding cache by another array, PASSED[], of MAXIMUM_SPOT_CHECKS bits' length. Given a binding for a particular mobile node, PASSED[i] is one if and only if INITIATED[i] is one and the mobile node has passed the care-of-address spot check for which NONCE[i] was used. When NONCE[CURRENT_NONCE_INDEX] is replaced by a new value, PASSED[CURRENT_NONCE_INDEX] is set to zero for all entries in the correspondent node's binding cache.

Care-of-address spot checks for multiple mobile nodes may coincide because each mobile node is spot-checked with a different Care-of Keygen Token. A single clock is thus sufficient to trigger spot checks of all registered confirmed care-of addresses. The same clock can also trigger the crediting function at the end of a crediting interval, i.e., after each series of MAXIMUM_SPOT_CHECKS clock ticks.

A mobile node may attach the same Spot Check option to multiple packets sent to a correspondent node in order to compensate for

potential packet loss. Of course, the correspondent node must not attach the same Spot Check option to multiple packets sent to the mobile node, because this would increase the likelihood that the mobile node receives the Spot Check option. An increased likelihood, in turn, would distort the measured congestion on the path towards the mobile node. For the same reason, a nonce must not be used for more than one spot check of a single care-of address, although it may be re-used for spot checks of multiple different care-of addresses.

6.6 Sending Packets to a Mobile Node

When a correspondent node has a packet for a mobile node, and the mobile node's care-of address is confirmed, the correspondent node's operation looks like this:

```
(A")   Sending a packet to a confirmed care-of address
       (Credit is given for receiving packets)
------------------------------------------------------------
(A".1) EFFORT := EFFORT + size(PACKET)
(A".2) If INITIATED[CURRENT_NONCE_INDEX] == 0
       Then
(A".3)    CoKT := careOfKeygenToken(NONCES[CURRENT_NONCE_INDEX])
(A".4)    attachSpotCheckOption(PACKET,CoKT,CURRENT_NONCE_INDEX)
(A".5)    INITIATED[CURRENT_NONCE_INDEX] := 1
       EndIf
(A".6) send(PACKET,CARE_OF_ADDRESS)
```

The correspondent node measures the mobile node's effort for receiving a packet in terms of the size of the packet in bytes (A".1). Effort invested during one crediting interval is accumulated in the variable EFFORT, which we have introduced in Section 5.3. New credit will be calculated at the end the crediting interval in step (D"), after having exponentially aged any old credit. The height of new credit depends on the value of EFFORT as well as the ratio of initiated to passed spot checks of the mobile node's care-of address.

In step (A".2), the correspondent node checks whether a new spot check of the mobile node's care-of address is due. This is the case if INITIATED[CURRENT_NONCE_INDEX] is zero, which means that the most recently generated nonce, NONCE[CURRENT_NONCE_INDEX], has not yet been used for a spot check of the mobile node's care-of address. In step (A".3), the correspondent node produces a Care-of Keygen Token based on the mobile node's care-of address and NONCE[CURRENT_NONCE_INDEX] as defined in Section 5.2.5 of [1]. The correspondent node attaches a Spot Check option carrying this Care-of Keygen Token and CURRENT_NONCE_INDEX to the outgoing packet in step (A".4). The correspondent node then sets

INITIATED[CURRENT_NONCE_INDEX] to one (A".5) in order not to do another care-of-address spot check with the same nonce for the same mobile node. If, at step (A".2), the correspondent node finds that INITIATED[CURRENT_NONCE_INDEX] is one, NONCE[CURRENT_NONCE_INDEX] has already been used for a spot check of the mobile node's care-of address. In this case, no care-of-address spot check is due, and steps (A".3) through (A".5) are skipped. Finally, the correspondent node sends the packet to the mobile node's care-of address (A".6).

When the correspondent node has a packet for a mobile node, and the mobile node's care-of address is unconfirmed, the correspondent node determines the packet's destination according to step (B) of Section 5.3.

6.7 Receiving Packets from a Mobile Node

When the correspondent node receives a packet from a mobile node, the correspondent node executes the following algorithm independently of whether the mobile node's care-of address is confirmed or unconfirmed.

```
(C")   Receiving a packet
       (Credit is given for receiving packets)
------------------------------------------------------------
(C".1) If hasSpotCheckOption(PACKET)
       Then
(C".2)    CoKT := getCareOfKeygenToken(PACKET)
(C".3)    NONCE_INDEX := getNonceIndex(PACKET)
(C".4)    If (INITIATED[NONCE_INDEX] == 1) \
             and (PASSED[NONCE_INDEX] == 0)
          Then
(C".5)       myCoKT := careOfKeygenToken(NONCES[NONCE_INDEX])
(C".6)       If CoKT == myCoKT
             Then
(C".7)          PASSED[NONCE_INDEX] := 1
             EndIf
          EndIf
       EndIf
(C".8) deliver(PACKET)
```

In step (C".1), the correspondent node checks whether the packet has a Spot Check option attached to it. If there is no Spot Check option, the packet is simply delivered to the application in step (C".8). If a Spot Check option exists, the correspondent node determines as follows whether this Spot Check option is the correct response to a recently initiated care-of-address spot check. Let CoKT be the Care-of Keygen Token advertised in the Spot Check option

(C".2), and let NONCE_INDEX be the nonce index advertised in the Spot Check option (C".3). NONCE_INDEX identifies the nonce that was allegedly used to create CoKT.

The correspondent node then ensures that INITIATED[NONCE_INDEX] is one and PASSED[NONCE_INDEX] is zero (C".4). If INITIATED[NONCE_INDEX] is zero, NONCE[NONCE_INDEX] has not yet been used for a spot check of the mobile node's care-of address. This may happen if the received Spot Check option pertains to a nonce that has been replaced in the meantime. The correspondent node can safely ignore the Spot Check option in this case. If PASSED[NONCE_INDEX] is one, the mobile node has already passed the care-of-address spot check for which NONCE[NONCE_INDEX] was used. This may happen if the packet carrying the Spot Check option gets duplicated during transmission, or if the mobile node attaches the same Spot Check option to multiple packets in order to mitigate the potential of packet loss. Although it would not harm if the correspondent node set PASSED[NONCE_INDEX] to 1 again in step (C".7), the correspondent node can avoid re-producing the Care-of Keygen Token from the Spot Check option in step (C".5) in this case.

Given that INITIATED[NONCE_INDEX] is one and PASSED[NONCE_INDEX] is zero, the correspondent node re-produces the Care-of Keygen Token from the Spot Check option based on the mobile node's care-of address and NONCE[NONCE_INDEX] (C".5). If the re-produced value matches the Care-of Keygen Token from the Spot Check option (C".6), the correspondent node sets PASSED[NONCE_INDEX] to one (C".7). In case of a mismatch, the correspondent node ignores the Spot Check option. The correspondent node should not punish the mobile node for returning an incorrect Care-of Keygen Token because a malicious node may otherwise fake false Spot Check options in order to prevent a faithful mobile node from gathering credit. Finally, the correspondent node delivers the packet to the application in step (C".8).

Care-of-address spot checks are needed only when a mobile node collects credit, i.e., when the mobile node's care-of address is confirmed. On the other hand, the correspondent node should accept a Spot Check option coming back from a mobile node even if the mobile node's care-of address is unconfirmed. The reason is that the mobile node may receive a Spot Check option at a confirmed care-of address shortly before changing its point of network attachment. The mobile node may not be able to immediately return the Spot Check option due to a lack of outgoing packets. If the mobile node changes its network-attachment point at that time, chances are that the mobile node returns the received Spot Check option through an unconfirmed care-of address.

6.8 Handling Clock Ticks

As mentioned, the correspondent node may synchronize the care-of-address spot checks for all entries in its binding cache. A single clock is thus sufficient to trigger care-of-address spot checks, nonce replacements, and crediting. With spot check, the clock ticks MAXIMUM_SPOT_CHECKS times per crediting interval, i.e., in intervals of TENTATIVE_RR_BINDING_LIFETIME/MAXIMUM_SPOT_CHECKS length. The currently oldest nonce is replaced upon every clock tick; crediting is done only once per crediting interval, i.e., every MAXIMUM_SPOT_CHECKS clock ticks.

Nonce replacement and crediting are united in the following algorithm, which the correspondent node runs upon every clock tick. The algorithm also prepares the next spot check of all confirmed care-of addresses. Note that the initiation time for a particular care-of-address spot check depends on when the correspondent node sends to the mobile node the next packet. This packet is then used to carry a Spot Check option.

```
(D")   Handling a clock tick
       (Credit is given for receiving packets)
------------------------------------------------------------
(D".1) CURRENT_NONCE_INDEX := (CURRENT_NONCE_INDEX + 1) \
                              modulo MAXIMUM_SPOT_CHECKS
(D".2) NONCE[CURRENT_NONCE_INDEX] := random(2**64-1)
(D".3) For Each Binding Do
          PASSED[CURRENT_NONCE_INDEX] := 0
          INITIATED[CURRENT_NONCE_INDEX] := 0
       EndFor
(D".4) If CURRENT_NONCE_INDEX == 0
       Then
(D".5)    For Each Binding Do
(D".6)       NEW_CREDIT := EFFORT * EFFORT_QUENCH_FACTOR \
                           * sum(PASSED)/sum(INITIATED)
(D".7)       CREDIT := CREDIT * CREDIT_AGING_FACTOR \
                       + NEW_CREDIT
(D".8)       EFFORT := 0
          EndFor
       EndIf
```

In step (D".1), the correspondent node increments CURRENT_NONCE_INDEX, which rolls over to zero when it reaches MAXIMUM_SPOT_CHECKS. The correspondent node then replaces NONCE[CURRENT_NONCE_INDEX] by a new random string (D".2), and it sets INITIATED[CURRENT_NONCE_INDEX] and PASSED[CURRENT_NONCE_INDEX] to zero for each entry in its binding cache (D".3). There is no distinction between bindings with a confirmed care-of address and

bindings with an unconfirmed care-of address. For a particular mobile node with a confirmed care-of address, INITIATED[CURRENT_NONCE_INDEX] and PASSED[CURRENT_NONCE_INDEX] help the correspondent node remember that a new care-of-address spot check is due, and that a Spot Check option should be attached to the next packet sent to the mobile node.

In step (D".4), the correspondent node checks whether the current crediting interval is over. This is defined to be the case when CURRENT_NONCE_INDEX rolls over to zero in step (D".1). CURRENT_NONCE_INDEX then points to the first nonce in the NONCE[] array. If so, the correspondent node re-computes the credit for each entry in its binding cache (D".5). A particular mobile node's new credit, NEW_CREDIT, is the product of EFFORT, EFFORT_QUENCH_FACTOR, and the ratio of the number of passed to the number of initiated care-of-address spot checks during the crediting interval (D".6). NEW_CREDIT is added to the aged value of CREDIT in step (D".7). Finally, the correspondent node resets EFFORT to zero (D".8).

7. Security Considerations

Credit-Based Authorization can prevent misuse of Early Binding Updates for amplified, redirection-based flooding attacks, and it can discourage such misuse for non-amplified, redirection-based flooding attacks. In this section, we explain why this is exactly the scope of protection that is needed to securely employ Early Binding Updates. We also discuss the design of Credit-Based Authorization itself with respect to security.

7.1 Scope of Protection

We emphasize that the objective of Credit-Based Authorization is not to prevent flooding attacks in general: This would probably be a hopeless venture given that we already have direct flooding attacks and reflection attacks in today's Internet (Section 3). The objective of Credit-Based Authorization is rather to prohibit misuse of Early Binding Updates for a type of flooding attack that is more potent than those types of flooding attacks already possible today. With this aim, Credit-Based Authorization prevents misuse of Early Binding Updates for amplified, redirection-based flooding attacks. As shown in Section 3, these attacks constitute to the Internet a significant threat which does not exist today. Credit-Based Authorization prevents them by limiting the data volume that a correspondent node can send to a mobile node's care-of address while this care-of address is unconfirmed. We show in Section 5.2 and

Section 7.3 that, through exponentially aging a mobile node's old credit, there is also a limitation on the correspondent node's data rate when sending to an unconfirmed care-of address.

Although Credit-Based Authorization does not prevent misuse of Early Binding Updates for non-amplified, redirection-based flooding attacks, it still makes such misuse unattractive enough to render it practically negligible. The reason is that the correspondent node can control, through the EFFORT_QUENCH_FACTOR protocol-configuration variable, how much effort a potential attacker must spend to gain the credit it needs for a flooding attack. Provided that EFFORT_QUENCH_FACTOR is below 1.0, the maximum data volume and rate that a flooding attack can amount to is less than the data volume and rate that the attacker has either previously sent to the correspondent node or previously received from the correspondent node. This makes misuse of Early Binding Updates for non-amplified, redirection-based flooding attacks very ineffective.

From an attacker's point of view, misuse of Early Binding Updates for non-amplified, redirection-based flooding attacks has another important disadvantage compared to a conventional flooding attack: The attacker's home address shows up in the Type-2 Routing extension header mandatorily attached to all redirected packets. The home address reveals the attacker's identity independently of the care-of-address state, because it has been authenticated in all recently transmitted Early Binding Update messages and standard Binding Update messages.

7.2 Attacks on the Routing Infrastructure

Suppose an attacker has access to the path between a mobile node's home agent and the correspondent node. In this position, the attacker can impersonate the mobile node by spoofing an Early Binding Update message or a standard Binding Update message. For this, the attacker has to obtain a valid Home Keygen Token, which it can obtain in one of two ways. First, the attacker can eavesdrop on the Home Test Init and Home Test messages being exchanged between the mobile node and the correspondent node. Second, the attacker can itself inject into the network a Home Test Init message on behalf of the mobile node, and it can intercept the Home Test message coming back from the correspondent node.

The difference between spoofing an Early Binding Update message and a standard Binding Update message is that the attacker has to have a valid Care-of Keygen Token in addition to the Home Keygen Token in the latter case. The attacker can produce a Care-of Keygen Token for a particular care-of address if it is present, or has a recorder, on

the path from the correspondent node to that care-of address. For this, the attacker injects into the network a spoofed Care-of Test Init message, and it intercepts the Care-of Test message, along with the Care-of Keygen Token, returning from the correspondent node--either by itself or through the recorder. In a special case, the attacker is located in the correspondent node's network, where it can acquire a Care-of Keygen Token for an arbitrary care-of address.

With the standard binding-update procedure, there is no way by which the attacker can redirect the mobile node's packets to a care-of address unless it is present, or has a recorder, on the path from the correspondent node to that care-of address. Things are different if the correspondent node supports Early Binding Updates and Credit-Based Authorization. In this case, an attacker on the path between the mobile node's home agent and the correspondent node can redirect the mobile node's packets to an arbitrary care-of address without having to show presence at that care-of address. Of course, the accumulated data volume of the redirected packets is limited by the mobile node's credit.

We argue that the described type of attack is unlikely to occur for two reasons. First, a fundamental assumption in the design of the Mobile IPv6 security design [3] is that the routing infrastructure is secure and functioning. As this specifically includes the path between the mobile node's home agent and the correspondent node, it is reasonably unlikely that an attacker can get access to this path.

Second, an attacker would have to invest a significant amount of effort in order to wage the described attack. Apart from having to get access to the routing infrastructure between the mobile node's home agent and the correspondent node, the attacker would have to synchronize with the mobile node in order to determine a point of time at which the mobile node's credit is high enough to make the attack worthwhile. We believe that these investments are unprofitable given that the yield of the described attack is limited by the mobile node's credit anyway.

Though unlikely to occur, this potential attack ought to be borne in mind during experimental deployments of Early Binding Updates and Credit-Based Authorization.

7.3 Limiting Packet Rate

We repeatedly mention in this document that Credit-Based Authorization limits both the volume and the rate of packets that a correspondent node can send to a mobile node's unconfirmed care-of address. According to the specification in Section 5, however, there

is an explicit limitation only of the data volume, whereas the limitation of the data rate is an implicit consequence of the application of exponential aging. In this section, we explain how exponential aging effectively enforces a limitation of the rate of packets that a correspondent node can send to a mobile node's unconfirmed care-of address.

Exponential aging ensures that credit older than one crediting interval rapidly looses its value. Consequently, a mobile node's available credit is, at any time, for the most part determined by the effort that the mobile node has spent during the preceding crediting interval. A crediting interval has the same length as a tentative binding's lifetime, TENTATIVE_RR_BINDING_LIFETIME, which, in turn, is the time period for which the mobile node needs its credit during a binding-update phase. These things considered, we observe that the time period during which credit can be most effectively collected is as long as the time period during which this credit can be consumed, i.e., TENTATIVE_RR_BINDING_LIFETIME time. The consequence is as follows.

Suppose, first, that the correspondent node measures the mobile node's effort for sending packets to the correspondent node. Since credit translates into data volume, the data volume that the correspondent node can send to a mobile node's unconfirmed care-of address during TENTATIVE_RR_BINDING_LIFETIME time is limited by the data volume that the mobile node has recently sent to the correspondent node during a period of the same length. Hence, the mean rate at which the correspondent node can send data to a mobile node's unconfirmed care-of address, averaged over a period of TENTATIVE_RR_BINDING_LIFETIME length, is limited by the mean rate at which the mobile node has recently sent data to the correspondent node during a period of the same length. Provided the value of TENTATIVE_RR_BINDING_LIFETIME is reasonably small, the actual data rates are bound to be close together as well.

Things are similar when the correspondent node measures the mobile node's effort for receiving packets from the correspondent node. In this case, the data volume that the correspondent node can send to a mobile node's unconfirmed care-of address during TENTATIVE_RR_BINDING_LIFETIME time is limited by the data volume that the mobile node has recently received from the correspondent node during a period of the same length. Provided the value of TENTATIVE_RR_BINDING_LIFETIME is reasonably small, the actual rate at which the correspondent node can send data to a mobile node's unconfirmed care-of address is bound to be close to the actual rate at which the mobile node has recently received data from the correspondent node.

In Section 3, we show that the power of a conventional flooding attack is by and large determined by the data volume and rate generated by the attacker itself. As explained above, Credit-Based Authorization transfers this property from conventional flooding attacks to flooding attacks realized through misuse of Early Binding Updates if and only if the correspondent node measures the mobile node's effort for sending packets to the correspondent node. The situation is different if the correspondent node measures the mobile node's effort for receiving packets from the correspondent node. See Section 7.4 for more details on this.

7.4 Asymmetric Network Attachment

In Section 4.3, we show that there are two variants of Credit-Based Authorization: The first variant measures a mobile node's effort for sending packets to a correspondent node; the second variant measures the mobile node's effort for receiving packets from the correspondent node. The first variant is based on the observation that, in a direct flooding attack or a reflection attack (Section 3), the attacker needs to spend resources in terms of sending, rather than receiving, packets. Thus, requiring a node, whether faithful or malicious, to send packets in order to earn credit is a straightforward way to discourage misuse of Early Binding Updates for redirection-based flooding attacks. The second variant accommodates a mobile node which receives much more data from the correspondent node than it sends back to the correspondent node. If the first variant was used in this case, this mobile node might not be able to collect the credit it needs, during a binding-update phase, to sustain the data stream received from the correspondent node.

In most scenarios, crediting a mobile node's effort both for sending packets and for receiving packets provides comparable protection against misuse of Early Binding Updates. This even though, in a direct flooding attack or a reflection attack, the attacker's investments are in terms of packets which the attacker sends rather than in terms of packets which the attacker receives. The reason is that a node, whether faithful or malicious, typically spends comparable effort--in terms of bandwidth, processing power, and memory--for both sending and receiving packets.

Things are different, however, if a node is asymmetrically attached to the Internet through, for instance, ADSL. An attacker could collect much more credit for receiving packets than it could collect for sending packets in this situation. As a consequence, if credit is given for packets which the attacker receives, misuse of Early Binding Updates for a redirection-based flooding attack might still be lucrative. Whether the existence of asymmetric network attachment

prohibits the second variant of Credit-Based Authorization is subject to further discussion.

As an aside, there is an alternative to the second variant of Credit-Based Authorization which likewise accommodates applications with asymmetric traffic patterns: Here, the correspondent node extends the length of a crediting interval such that the mobile node--even though it may not send as much data to the correspondent node as it receives from the correspondent node--has a chance to collect the amount credit that it needs, during a binding-update phase, to sustain the data stream received from the correspondent node. The advantage of this approach is that it accommodates applications with asymmetric traffic patterns without posing a new threat due to the existence of asymmetric network attachment. The drawback is that extending the length of a crediting interval allows a mobile node to collect credit over a longer period of time. A malicious node may misuse this property to by-pass the rate-controlling function of exponential aging, which we describe in Section 5.2 and Section 7.3. For this reason, we do not deepen the idea of extending the length of a crediting interval in this document.

7.5 Resource Exhaustion Attacks

In Section 6.3, we discuss how a correspondent node may produce and check the random strings needed for care-of-address spot checks. We propose to re-use the standard formula from Section 5.2.5 of [1], which is used to produce and check Care-of Keygen Tokens for care-of-address tests. This formula is an HMAC_SHA1 function.

Mobile IPv6's Care-of Keygen Tokens help the correspondent node to efficiently manage its resources in terms of memory storage [3][4]. However, when Care-of Keygen Tokens are used for care-of-address spot checks, care has to be taken by the correspondent node not to fall victim to denial-of-service attacks that aim at exhausting its resources in terms of processing power. This can happen if a malicious node sends to the correspondent node Spot Check options with bogus Care-of Keygen Tokens. The correspondent node would have to check the validity of each of the received tokens in this case. The processing capacity required to validate Care-of Keygen Tokens in normal operation should be acceptable, but the processing capacity needed to invalidate a large number of bogus Care-of Keygen Tokens may be substantial.

On the other hand, both a mobile node's home address and care-of address are already authenticated during a care-of-address spot check. One may argue that this feature is likely to discourage an

attacker from waging a denial-of-service attack against the correspondent node.

Still, it remains to be discussed how susceptible care-of-address spot checks are to denial-of-service attacks. One could mitigate the threat of such attacks by using a more efficient function to produce and check Care-of Keygen Tokens for care-of-address spot checks, while keeping the standard function used in [1] to produce and check Care-of Keygen Tokens for care-of-address test. For instance, one may calculate Care-of Keygen Tokens for care-of-address spot checks as a simple SHA1 hash rather than a keyed HMAC_SHA1.

7.6 Alternatives to Credit-Based Authorization

There are alternatives to Credit-Based Authorization which can protect against misuse of Early Binding Updates for redirection-based flooding attacks. One alternative is to grant the use of Early Binding Updates to a trusted community only. Recall that, when the correspondent node receives an Early Binding Update message from the mobile node, it can be confident that the mobile node is the legitimate owner of the home address advertised in this message due to the home-address authentication. This, in turn, allows the correspondent node to use the mobile node's home address as a criterion for deciding whether or not to install a tentative binding for the mobile node's new care-of address. For instance, the correspondent node may be a corporate server which grants Early Binding Updates to mobile nodes from the corporate network only. [9] goes a step further; it uses pre-configured binding-management keys shared between mutually trusting nodes. These approaches have a rather limited application scope, however.

Another alternative to Credit-Based Authorization is a combination of ingress filtering [6] and a ban on the Alternate Care-of-Address option in the Early Binding Update message. Ingress filtering is a function in the access network, preventing packets with spoofed source addresses to leave the network. The disadvantage of ingress filtering is that it is not universally deployed, and as such cannot be relied upon.

A third alternative to Credit-Based Authorization is to identify and blacklist malicious nodes based on their observed behavior. Credit-Based Authorization is a proactive strategy, whereas behavior-based blacklisting is a reactive one. The advantage of a reactive approach is that a faithful mobile node does not need to invest effort (i.e., collect credit) before it can receive packets at an unconfirmed care-of address during the binding-update period. This can accommodate a mobile node having trouble to collect

sufficient credit--be it because the mobile node's care-of address changes too frequently, or because the correspondent node measures the mobile node's effort for sending packets to the correspondent node, but the mobile node's application generates only very little data. On the other hand, a reactive approach is by definition unable to prevent misuse of Early Binding Updates in advance. What it can do is contain the damage of such misuse and punish the perpetrator. Punishment, however, will be negligible in most cases, since an administrative relationship between a mobile node and a correspondent node does usually not exist.

Behavior-based blacklisting requires a heuristic to determine what behavior is considered "ill". Choosing the right heuristic, however, is far from trivial. If carelessly chosen, a heuristic may punish a faithful mobile node or overlook an evil one. For instance, a simple heuristic may assume that a faithful mobile node sends a standard Binding Update message soon after having sent an Early Binding Update message. At first glance, this heuristic seems to work well: Keeping alive an unconfirmed care-of address by repeatedly sending Early Binding Update messages is no longer possible. On the other hand, there is an easy way to trick this heuristic: An attacker can send to the correspondent node a forged Early Binding Update message using a victim's IP address as its care-of address. The attacker can cause the correspondent node to send packets to the victim as long as the heuristic allows. The attacker can then update the care-of address to its own IP address by means of a standard Binding Update message--thereby fulfilling what the heuristic expects it to do--and immediately re-update the care-of address to the victim's IP address by means of an Early Binding Update message. This procedure can be repeated indefinitely.

While a heuristic may fail to identify a malicious node, it may also wrongly accuse a faithful mobile node. For example, a mobile node may be subject to two handovers immediately following each other. After the first handover, the mobile node may be able to send an Early Binding Update message, but it may not have enough time to complete the concurrent care-of address test. If the mobile node attempted another Early Binding Update after the second handover, the above-described heuristic would wrongly blacklist it.

8. Conclusion

Early Binding Updates for Mobile IPv6 have been proposed as an optional optimization for Mobile IPv6. Early Binding Updates allow a mobile node to use a new care-of address while a care-of-address test is in progress. Protective measures are necessary to rule out misuse

of this concurrency for redirection-based flooding attacks. We propose a mechanism, Credit-Based Authorization, that prevents misuse of Early Binding Updates for amplified, redirection-based flooding attacks. Through proper parameterization, Credit-Based Authorization can discourage misuse of Early Binding Updates even for non-amplified, redirection-based flooding attacks.

With Credit-Based Authorization, a correspondent node monitors a mobile node's effort for sending or receiving packets. The correspondent node acknowledges invested effort based on the size of packets that the mobile node sends to the correspondent node or receives from the correspondent node. Invested effort is turned into credit. This credit is consumed by each packet that the correspondent node sends to an unconfirmed care-of address of the mobile node during the binding-update phase. The invoiced credit is such that a malicious node would have to invest more effort to misuse Early Binding Updates for a redirection-based flooding attack than it would have to invest for a conventional flooding attack. It stands to reason that this property will make the malicious node change its mind about misusing Early Binding Updates.

9. Protocol Configuration Variables

In this section, we recommend default values for the protocol-configuration variables used in this document. We believe that the proposed values are reasonable. A protocol implementer may choose different values nonetheless.

CREDIT_AGING_FACTOR	0.5
CREDIT_AGING_FACTOR	0.5
EFFORT_QUENCH_FACTOR	0.5
MAXIMUM_SPOT_CHECKS	15
TENTATIVE_RR_BINDING_LIFETIME	3 seconds

TENTATIVE_RR_BINDING_LIFETIME is originally defined in [2]. It is used as the length of a crediting interval in this document. When care-of-address spot checks are used, we propose a maximum of MAXIMUM_SPOT_CHECKS spot checks per crediting interval, i.e., at most one care-of-address spot check every 200 milliseconds.

10. Acknowledgements

The necessity for a mechanism to prevent or discourage misuse of Early Binding Updates for flooding attacks was sparked by a fruitful discussion on the MIP6 and MOBOPTS mailing lists. Credit-Based Authorization has been developed as a candidate solution, and a first presentation was given at the 59th IETF meeting. For their interest and valuable feedback, the authors thank the MIP6 and MOBOPTS communities, in particular Roland Bless, Mark Doll, Francis Dupont, Gregory Daley, Lars Eggert, James Kempf, Tobias Kuefner, Marco Liebsch, Gabriel Montenegro, Nick (Sharkey) Moore, Pekka Nikander, Erik Nordmark, Charles Perkins, and Kilian Weniger (listed in alphabetical order). Thanks are also due to Keiichi Shima and his colleagues from the Kame-Shisa development team.

11. References

11.1 Normative References

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

[2] Vogt, C., "Early Binding Updates for Mobile IPv6", Internet-Draft draft-vogt-mobopts-early-binding-updates (work in progress).

11.2 Informative References

[3] Nikander, P., Arkko, J., Aura, T., Montenegro, G. and E. Nordmark, "Mobile IP version 6 Route Optimization Security Design Background", Internet-Draft draft-ietf-mip6-ro-sec (work in progress).

[4] Aura, T., Roe, M. and J. Arkko, "Security of Internet Location Management", In Proceedings of the 18th Annual Computer Security Applications Conference, Las Vegas, Nevada, USA., December 2002.

[5] Arkko, J. and C. Vogt, "Credit-Based Authorization for Binding Lifetime Extension", Internet-Draft draft-arkko-mipv6-binding-lifetime-extension (work in progress).

[6] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", RFC 2827, May 2000.

[7] Paxson, V., "An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks", Computer Communication Review 31(3)., July 2001.

[8] Anderson, R., "Why Information Security is Hard -- An Economic Perspective", In Proceedings of the 17th Annual Computer Security Applications Conference, New Orleans, Louisiana, USA., December 2001.

[9] Perkins, C., "Precomputable Binding Management Key Kbm for Mobile IPv6", Internet-Draft draft-ietf-mip6-precfgkbm (work in progress).

Authors' Addresses

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

Email: chvogt@tm.uka.de
URI: http://www.tm.uka.de/~chvogt/

Jari Arkko
Ericsson Research NomadicLab
FI-02420 Jorvas
Finland

Email: jari.arkko@ericsson.com

Acknowledgment

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