IETF Mobile IP Working Group Pekka Nikander
INTERNET-DRAFT Ericsson Research NomadicLab
Charles Perkins
Nokia Research Center
2 July 2001

Binding Authentication Key Establishment Protocol for Mobile IPv6
<draft-perkins-bake-01.txt>

Status of This Memo

Abstract

A method is proposed for providing key information for use between a mobile node and correspondent node, to be used for the purpose of authenticating Mobile IPv6 Binding Updates. The key distribution is secure except against certain man-in-the-middle attacks, when the attacker resides along the routing path between the correspondent node and the mobile node's home agent. The key can be used for authenticating subsequent Binding Updates from the same mobile node, substantially reducing the number of key establishment cycles needed for maintaining efficient communication paths between the mobile node and correspondent node.

Contents

1. Introduction

Mobile IPv6 specifies the use of a Binding Update destination option for use between a correspondent node and a mobile node. This Binding Update associates a "care-of address" to the mobile node's "home address" enabling direct routing of packets to the current point of attachment in use by the mobile node. Unfortunately, there may be many instances where no pre-existing security association is available for use by the correspondent node to authenticate a Binding Update from the mobile node.

In order to solve this problem, we propose a method by which a correspondent node supplies some random data for use by the mobile node as input to an algorithm for creating a security association. This security association, once established between the mobile node and the correspondent node, is then used to create the authentication data that is required to be supplied as one of the security fields of the Binding Update destination option.

Without introducing reliance upon use of certifiable public keys or trusted third party based security infrastructure, it is quite difficult to avoid all attacks against the correspondent node that might allow some malicious third part to masquerade as the mobile node. However, we can at least make sure that any such malicious node would have to reside along the routing path between the correspondent node and the home agent. This substantially reduces to the vulnerability to such attacks, since the specific routing path required amounts to an extremely small proportion of the Internet.

Some of the security features of this protocol rely on requiring that the correspondent node must send the Binding Key Requests to the home network. This gives some measure of assurance that subsequent protocol actions could avoid interference by nodes not along the natural routing path between the correspondent node and the home network.

> DISCUSSION. If the security associations created are AH security associations, most IPsec implementations would need to change their policy engine. However, we would consider AH better than a special purpose protection since using AH would allow IKE/AAA/whatever to be used to create the SAs between the MN and the HA without too many changes. On the other hand, nothing prevents from specifying a new DOI/whatever to create security associations with IKE/AAA/whatever. Thus, from that point of view, this protocol is just a special purpose key management protocol, able to create only security associations for securing Binding Updates.

In this protocol specification, several new messages are introduced:

Binding Warning	which is sent by a mobile node to a correspondent node as an indication that a new care-of address should be obtained for the one of the mobile node's addresses.
Binding Key Request	which is sent by a correspondent node to the home address of the mobile node in order to elicit a Binding Update together with a Binding Key Establishment Extension.
Binding Key Establishment Extension	which supplies a key to be used by the correspondent node when verifying authentication data supplied by the mobile node to ensure the integrity of the Binding Update data.

Subsequent sections provide an overview of the operation of the key establishment mechanism, discussion about the cryptological analysis motivating the design of the protocol, the format of the protocol messages, and detailed specification for the message formats. Although a specific authentication key establishment algorithm is proposed, it should be noted that other key establishment algorithms may be proposed in the future. Nevertheless, for interoperability, all correspondent nodes MUST implement the algorithm specified in this document.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1]. Furthermore, this document makes use of many general terms defined in the protocol specification for Mobile IPv6 [2].

2.1. Specific Document Terminology

Other specific terms are defined as follows.

Cookie	a tasty morsel with random bits of goodness
Hash	a way to scramble tasty morsels of data; often used as a message digest or message authentication code (MAC)
Key	a secret number that enables operation of a cryptographic algorithm.

Binding Warning Packet
: any packet containing a Binding Warning destination option

BKE Packet
: any packet containing a Binding Key Establishment destination option

Binding Key Request Packet
: any packet containing a Binding Key Request destination option

Binding Key
: a key used for authenticating Binding Update messages.

Security Association
: a security object shared between two nodes which includes the data mutually agreed on for operation of some cryptographic algorithm (typically including a key, as defined above).

Binding Security Association (BSA)
: a security association established specifically for the purpose of producing and verifying authentication data passed with a Binding Update destination option.

Tunnel Protection
: protecting tunneled data against attackers, either by encryption, inserting additional integrity checking, or by modifying the data in some unforgeable fashion. See section 6.5.

2.2. Abbreviations and Symbolic Names

The protocol messages defined in this specification use some opaque data generated according to the needs of various cryptographic algorithms. These data are referred to by symbolic names. In order to reduce confusion, we attempt to use the same symbolic names throughout the document. These names are gathered together here for easy reference. We also include various abbreviations for completeness.

CN	The correspondent node
HA	The home agent
MN	The mobile node

HoA	The home address of the mobile node

CNA	The (IPv6) address of the correspondent node
KeyMat	Data computed for use in generating the Binding Key which defines the BSA
BK	The binding key
BKE	The Binding Key Establishment destination option
K_CN	A secret value maintained (and reselected periodically) by the correspondent node for use in generating nonce N2
K_(MN,HA)	A key shared between a mobile node and its home agent.
PCB	Protocol Control Block (used by TCP)
N1	A nonce created by the mobile node for use in the Binding Warning, and eventually used to identify the Binding Key Request
N2	A nonce, reproducible from K_CN and T1, for use by the correspondent node when producing token T2. It is used to validate an eventual Binding Key Establishment message.
N_BK	A random number created by the mobile node after reception of the Binding Key Request message, and used when creating BK.
T0	A token computed in order to assure data integrity, and to provide a temporarily hidden input for token T1.
T1	A token computed from T0, and made publicly visible in protocol messages, in order to assure data integrity.
T2	The first part of the key generation material, calculated by the correspondent node, and delivered to the mobile node in the Binding Key Request message.
HASH_T0	A cryptographic algorithm used to produce the token T0 (see section 8.1)
HASH_T1	A cryptographic algorithm used to produce the token T1 (see section 8.2)
HASH_N2	A cryptographic algorithm used to produce the nonce N2 (see section 8.3)
HASH_T2	A cryptographic algorithm used to produce the token T2

(see section 8.4)

HASH_BK	A cryptographic algorithm used to produce BK, the desired Binding Key (see section 8.5)

3. Design goals

The protocol MUST fulfill a set of defined goals. The goals can be expressed in terms of beliefs that the parties may hold after a successful protocol run. These beliefs are outlined in Sections 3.1 through 3.3, below.

The primary goal of the protocol is to create unauthenticated but appropriately authorized keying material that may be reasonably used to secure future Binding Updates between a MN and a CN. By "unauthenticated" we mean that the protocol does not give any assurance about any identity of the MN to the CN or vice versa. By "appropriately authorized" we mean that the CN has reasonable bases to believe that it is sharing the keying material with the MN, i.e., with a party that is reachable through the home address that it claims to "own", and that the MN has reasonable bases to believe that it is sharing the keying material with the CN, i.e., with the party who is reachable through the address that the MN expects the CN to have.

By "reasonable bases" we mean that the protocol does not aim to be fully secure in any stronger means of the term "secure" but that it does protect against attacks that some unrelated third party might be able to launch from an arbitrary location in the Internet. In particular, the protocol does not aim to protect the parties against attackers that are able to eavesdrop all traffic flowing between the MN, CN and HA.

3.1. Beliefs held by the CN

In this section, we list the beliefs that the correspondent node is expected to act upon during its part of the protocol operation.

- The CN is able to believe that the MN has the given home address, since it has checked that MN eventually receives packets sent to the Home Address, and that the MN is (was) able to reply to them.

- The CN is able to believe that the MN wants to provide the given CoA for use in routing headers for data to be sent to mobile node.

- The CN believes that the key it holds with an MN is only known to the MN unless there is some third party who is able to either

* eavesdrop all traffic sent and received by the CN, or

* eavesdrop all traffic sent by both the CN and MN, or

* eavesdrop all traffic sent and received by the MN, and the traffic between the MN and the HA is in clear.

No other passive or active attack scenarios are believed to be tolerated by this protocol.

3.2. Beliefs held by the MN

In this section, we list the beliefs that the mobile node is expected to act upon during its part of the protocol operation.

- The MN believes that packets sent to the CN address are (eventually) received by the CN, and that the CN is able to reply to them.

- The MN believes that the key it holds with the CN is only known to the CN unless there is some third party who is able to eavesdrop traffic as specified in section 3.1.

3.3. Optional Beliefs

[XXX: Should we allow the HA have optional beliefs?]

[XXX: Should we allow the MN have optional beliefs about endorsement by the HA?]

4. Design principles

In this section, we discuss the following design principles which have been used in the construction of this protocol.

- Low computational overhead

- Minimal messaging: MN->CN, CN->HA->MN, MN->CN

- Resistant to DoS attacks

  -- CN does not need to create state before the third message

- Allows active participation of HA but does not require it

- No single message alone gives keying material out

- Randomness included by both MN and correspondent node

- Allows inband establishment for the Binding Security Association (BSA)

- Protection against "future address stealing"

- Allows initation by the correspondent node.

4.1. Low computational overhead

The protocol should have low computational overhead. In particular, it must not require any heavy computation by the CN before that node has some assurance that the MN is actually there and that the MN is able to receive messages sent to the Home Address. Furthermore, it should not require any heavy computation by the MN as the MN may have significant processing limitations.

4.2. Minimal Messaging

The protocol is conveyed by three messages, namely:

1. Binding Warning: sent by a MN to a CN (MN->CN)

2. Binding Key Request: sent by a CN to the MN through the HA (CN->HA->MN)

3. Binding Key Establishment: sent by a MN to the CN (MN->CN)

A two message variant of this, initiated by the CN, is also needed; this is defined in Section 7.14.

As a particular design principle, we want to make it possible for a MN to send the first message to the CN before any prior communication between the MN and the CN (see section 4.7. For example, this first message could be included in a TCP SYN sent by the MN to the CN.

4.3. DoS resistance

The protocol can be made resistant to "denial of service" (DoS) attacks by using the following principles.

1. The involved parties must minimize state creation before they obtain assurance that the alleged peer is on-line. That is, state is created only when a such a message is received that the party creating the state can verify that it has itself recently been involved in sending a message to which the received message is a reply.

2. The amount of computation performed by a responding party should be limited until it is able to check that the initiating party has been performed an equal or larger amount of computation.

In order not to create a possibility for a new TCP SYN flooding type of attack, the protocol MUST NOT require the CN to create any state before it receives its final protocol message. This is a particular instance of the first DoS design principle, in the context of the three-message protocol we specify in this document. Additionally, we would like to design the protocol in such a way that the CN could optionally defer even TCP state creation until it receives the third protocol message. Specifying such mechanism is beyond the scope of this document; for here, it suffices to merely create a transparent mechanism that could be used for that and other similar purposes.

4.4. Optional Active Participation of HA

We envision a future with various relationships between the MN and the HA, and therefore very different kinds of HAs. Some of the HAs may be mere packet reflectors, tunneling all packets sent to them to the MN. For example, HAs used for temporary packet forwarding are likely to be like this. However, some HAs may be much more intelligent, and even provide computation services to the MNs.

In order not to rely on any specific properties of the HAs, but allow space for utilizing them, we want to design the protocol in such a way that it allows participation of HAs but does not require it. We also expect that some HAs will encrypt encapsulated packets when tunneling them to the mobile nodes, providing additional security.

4.5. No single message alone gives keying material out

Since we do not want to assume any existing security associations between the MN and the CN or between the HA and the CN, the protocol cannot create properly authenticated keying material out of nothing. That is, in the light of currently established wisdom, the protocol cannot create keying material in such a way that the participating parties would have enough grounds to believe that only they and no one else possess the keying material.

DISCUSSION: For the purposes of appropriate authorization (but not authentication), something like the protocol presented in draft-nikander-ipng-pbk-addresses-00.txt could be used to create keying material in such a way that it would be protected against passive attackers and even (at least most if not all) active attackers. However, the protocol requires public key computations.

Thus, the protocol has to be designed in such a way that an attacker that is only able to eavesdrop any single message is unable to create the keying material. This gives reasonable protection against passive attackers. In practice, an attacker would have to be on the same link with the CN, on the same link with the MN and the traffic between the HA and the MN be unprotected, or be able to eavesdrop traffic between both the MN and CN and CN and HA. This drastically reduces the number of locations from which a malicious node can mount an attack.

DISCUSSION: Some people have suggested to use (unauthenticated) Diffie-Hellman (D-H) instead of simply combining two independent random numbers through a one-way hash. From the security point of view, that would have the benefit of protecting the generated key against those passive attackers that are able to eavesdrop both halves of the key. However, unless properly authenticated, it would NOT provide any more security against active attackers. Thus, given the additional complexity and computational overhead, i.e. the requirement of implementing a big integer library and calculating big integer exponetiations, we do not consider the added security worth the added trouble. (See also section B.)

4.6. Randomness included by both MN and CN

To protect against possible bad random number generators, the keying material includes randomness generated both by the MN and the CN.

4.7. Inband creation for BSA

We wish to avoid requiring extensive changes to the current Mobile IPv6 specification. Therefore, the protocol is designed in such a way that the MN creates a Binding Key Security Association (BSA) after receiving the message from the correspondent node, and that the CN is able to create the same BSA after receiving its final message. This makes it possible to send an authenticated BU along with the third message. Furthermore, in this way the normal data flow between the two endpoints (mobile node and correspondent node) experiences minimal disruption.

4.8. Protection against Future Address Stealing

Future address stealing [8] is a vulnerability whereby an adversary may be able to usurp ownership of an address which would likely be claimed by a mobile node some time in the future. To protect against this threat", the mobile nodes SHOULD use random CoAs [7].

5. Protocol overview

This section gives an overview of the protocol from an implementation (engineering) point of view. A corresponding cryptographic (security point of view) description is given in Section 6. We first outline the message exchanges, and then discuss the protocol steps one by one. Section 7 specifies the processing rules, and Section 9 the exact message formats.

5.1. Protocol Messages

The protocol uses the Binding Warning, Binding Key Request, and the Binding Key Establishment (BKE) destination options, as follows.

1. If no security association between a mobile node and a correspondent node exists for the purpose of authenticating a Binding Update to that correspondent, the mobile node SHOULD send a Binding Warning to the correspondent node, asking it to start the process of establishing the needed security association with the indicated address of the mobile node. This packet MUST contain the Home Address Destination Option, informing the CN about the alleged Home Address (HoA) of the MN.

2. If the Correspondent Node wants to continue the exchange, it sends a Binding Key Request to the Home Address of the MN. The packet MUST be sent to the Home Address independent on any possibly existing Bindings. As the default action, the Home Agent SHOULD tunnel the packet to the Mobile Node. The tunnel SHOULD be protected using ESP [3], or some other means.

3. When the Mobile Node receives a Binding Key Request, passed via the tunnel and originated by the Correspondent Node, it creates a BSA that will be used to secure Binding Updates.

   The Binding Key Establishment Destination Option allows the Correspondent Node to perform checks to reduce the risk that the peer is not the right Mobile Node "owning" the Home Address to a reasonable level, and to create the same BSA that the Mobile Node created after the Binding Key Request message.

At the conclusion of the protocol, a BSA will have been established, for use to create and verify the data containing in Binding Update destination options.

5.2. Prerequisites

We assume that certain conditions are in place before the actual protocol is run. This is a standard practice in security protocols, which usually build upon existing security relationships. However, in the case of this protocol those relationships are relatively weak, at least if compared to typical security protocols.

Briefly, there are two requirements:

- There is a relationship between the Mobile Node and the Home Agent.

- The Correspondent Node has to generate random keys, K_CN, known only to itself.

In the simplest form, the relationship between the Mobile Node and the Home Agent may be the implicit agreement that the Home Agent will tunnel received packets to the Mobile Node. Such an agreement could be arranged without having any explicit security association between the MN and the HA; for example, a temporary HA might just forward packets to a fixed address for a period of time. Using Mobile IPv6 [2], a mobile node can update the tunnel endpoint (in the Binding Cache) which the home agent maintains for it, using a pre-established BSA. Often, the Mobile Node and Home Agent have an IPsec ESP security association, and the Home Agent is able to encrypt and integrity protect the tunneled packets. The protocol in this document also allows more advanced security relationships between the Home Agent and the Mobile Node, enabling some of the crypto processing to be performed by the Home Agent instead of the Mobile Node. The specification of such processing is outside this specification.

The present protocol enables the use of a key, K_(MN,HA), shared between the MN and the HA, whose sole purpose SHOULD be to be used within this protocol. Before using the key for any other purposes, the possibility of unwanted interactions must be eliminated.

With some loss of security, it is also possible to run a variation of the protocol where the key K_(MN,HA) belongs solely to the Mobile Node, and is not known by the HA or any other party.

The random keys, K_CN, generated by the Correspondent Node are assumed to be short lived; a typical lifetime of such a key might be e.g. 10 seconds or one minute. As the key is local to the CN, the exact policy and mechanism for generating such keys is not specified. For example, one possibility is to generate a completely new key only every few minutes or hours, and just keep incrementing the key until it is the time to create a completely new key. To add robustness, the Correspondent Node MUST remember a number of previous K_CN values. The exact number of remembered values is a matter of local policy.

6. Cryptographic Design Analysis

From the cryptographic point of view, we have a three-party protocol where one of the parties may be a passive forwarder, only optionally taking part of the protocol run in any other way than reflecting messages. To emphasize the difference between the parties and their addresses, we denote the Mobile Node's current address (care-of-address) as CoA, its Home Address as HoA, and the Correspondent Node's address as CNA. The parties are the Mobile Node (MN), the Corresponding Node (CN) and the Home Agent (HA), assumed connected in a triangular manner:

```
                               HA
                              /  ^
                             /    \
                            /      \
                           V        \
                         MN <-----> CN
```

The MN and CN are able to directly exchange messages. However, for the purposes of this protocol we assume that the links between the CN and the HA, and between the HA and MN are unidirectional. By default, we assume only that the HA can act as a passive reflector, tunneling any packets sent to the mobile node's home address (HoA) to the MN's care-of address (CoA).

The messages tunneled by the HA to the MN may or may not be encrypted. If they are encrypted, the encryption is assumed to effectively defeat eavesdropping on the links along the tunnel's path. The links between the MN and the CN and leading from CN to HA are assumed to be clear and vulnerable to eavesdropping.

From the security point of view, the purpose of the protocol can be described as follows. The protocol starts from an initial situation where the MN and HA have a security association with each other. More formally, the MN knows that it is currently using the address CoA and that its home address is HoA, and that there is the Home Agent HA at the home address HoA. Likewise, the HA knows that it is currently reflecting packets sent to HoA to the care-of-address CoA, and that the MN is assumed to be currently reachable through CoA. Furthermore, they may share a secret key K_(MN,HA). Furthermore, the MN has (recently) learned an address CNA that it assumes to belong to some corresponding node CN. The HA does not have any information about the CN. The CN does not have any information about the MN or HA.This initial state can be described as the following knowledge sets.

```
   MN: { CoA, HoA, K_(MN,HA), CNA }
```

```
      HA: { CoA, HoA, K_(MN,HA) }
      CN: { K_CN, CNA }
```

6.1. Message 1: Binding Warning

The protocol is triggered by the MN deciding that it wants to exchange a binding key with whatever CN happens to currently be at CNA. To ensure message freshness the MN creates a nonce, N1, and adds it to its knowledge set.

```
MN: { CoA, HoA, K_(MN,HA), N1 }
HA: { CoA, HoA, K_(MN,HA) }
CN: { CNA, K_CN }
```

After that, it uses the algorithms in sections 8.1 and 8.2 to create cryptographic tokens T0 and T1. T1 may be optionally authenticated by the HA if the HA knows the key K_(MN,HA). The reason for this two step process of creating the token T1 is that later, in the third message, T0 is used to authenticate the MN as the original source of T1.

The first message contains the addresses, the nonce N1 and the token T1. (Note that the addresses are already available in the IP headers, and they SHOULD NOT be repeated in the payload.)

```
MN->CN: < CoA, CNA, HoA, N1, T1 >
```

When the CN receives the message, it learns the addresses CoA, HoA, the nonce N1 and the token T1. These are tabulated as a separate knowledge set since the CN should forget them after it has sent the reply. The reason why we want the CN to forget these data is denial-of-service resistance. Basically, we want the CN to handle the packet effectively and fast, and then forget about it.

```
MN: { CoA, HoA, K_(MN,HA), N1 }
HA: { CoA, HoA, K_(MN,HA) }
CN: { CNA, K_CN } + { CoA, HoA, N1, T1 }
```

6.2. Message 2: Binding Key Request

For the next message, the CN prepares a structured nonce N2 and an authentication token T2. The first of these, N2, is sent to the MN through the HA, and used by the MN as a part of creating the new keying material. The second token, T2, is passed by the HA and MN back to the CN, allowing it to authenticate the third message as it arrives. Furthermore, the CN must be able to reconstruct the nonce N2, based on the authenticated third message, so that it can create the same keying material as the MN does.

```
N2 := HASH_N2 ( K_CN ; 0 || T1 ) (see section 8.3)
```

```
      T2 := HASH_T2 ( K_CN ; T1 || CoA || CNA || HoA)
      (section 8.4)
```

When creating the tokens, CN uses only information that will be available to it when it receives the third message. The authentication token T2 must include all information whose integrity must be protected. Including T1 helps to block active attacks later in the protocol. On the other hand, the content of the nonce N2 is not that critical; the only requirements are that it is hard to predict (hence K_CN) and it can be easily recomputed upon arrival of the third message. The (relative) freshness of the key K_CN supplies freshness to the nonce N2 and the token T2.

In the second message, CN sends its address, the nonces N1 and N2 and the tokens T1 and T2 to the home network, by way of the Home Address HoA.

```
   CN->HA: < CNA, HoA, N1, T1, N2, T2 >
```

6.2.1. Optional Processing by the Home Agent

A simple HA will simply forward the second message to the CN through tunneling (ESP or non-ESP).

A more intelligent HA may want to authenticate the packet, and perhaps also perform other functions. After the HA receives the message its knowledge set is as follows:

```
   HA: { CoA, HoA, K_(MN,HA), CNA, N1, T1, N2, T2 }
```

By receiving the message, the HA has already implicitly checked that the HoA in the message matches a home address for a known mobile node. It MAY now proceed to check validity of T1.

```
   T1 =?  HASH_T1( K_(MN,HA) ; N1 || CoA || CNA || HoA )
```

This check allows the HA to verify that the token T1 was generated by the MN, and that MN meant it to be used for a protocol run between CoA, CNA and HoA.

The HA may also perform other activities, like logging some of the information to an audit trail. Additionally, based on a mutual agreement between the MN and HA, it MAY also change the contents of N1 and T1 to something that allows MN to determine that the message has been processed by the HA. For example, the HA might want to replace N1 and T1 with NewN1 and NewT1, where

```
   NewN1 := N1 + 1
   NewT1 := HASH_T1( K_(MN,HA) ; NewN1 || CoA || CNA || HoA )
```

However, such practices are beyond the scope of this protocol.

In any case, the HA forwards (through a tunnel) a message that has the same format as the message it received.

```
   HA->MN: TUNNEL < CNA, HoA, N1, T1, N2, T2 >
```

After processing the message, the home agent SHOULD reduce its knowledge set by discarding information that it no longer needs.

```
   HA: { CoA, HoA, K_(MN,HA) }
```

6.2.2. Processing by the Mobile Node

Before the mobile node receives the Binding Key Request, it has the following knowledge set:

```
   MN: { CoA, HoA, CNA, K_(MN,HA), N1, T0 }
```

The mobile node must first check that the addresses match. The CoA is implicitly checked by receiving the message. The mobile node MUST explicitly check that the message was received through a tunnel from the HA, and that the inner header in the tunneled packet contains CNA as the source address and HoA as the destination address. After that, it SHOULD check that the nonce N1' received with the Binding Key Request equals the nonce N1 generated in section 6.1. This protects the MN against replay attacks. Finally, it SHOULD check that the received token T1' authenticates correctly.

```
   T0' := HASH_T0 ( K_(MN,HA) ; N1' || CoA || CNA || HoA ) )
   T1' =?  HASH_T1 ( T0')
```

These checks allow the the MN to verify the following.

- The initial message was received by some party that is able to receive messages sent by the MN to CNA.

- If the tunneling check was strong (the MN-HA tunnel is ESP) or if the <N1,T1> pair was modified according to a mutual agreement between the MN and the HA, the party that received the initial message sent a reply to the correct HA, who forwarded the message back to the MN.

Any message may have been intercepted and modified by an active attacker. If the active attacker was able to intercept the Binding Warning message, the above statements still hold since it then acts as "some party that is able to receive messages sent by the MN to CNA." On the other hand, if the active attacker is only able to intercept messages at the CN->HA or HA->MN link, it still cannot modify the <N1,T1> pair without knowing the key K_(MN,HA), and

therefore T1 authenticates the address triple. Consequently, a
securely tunneled packet or a correctly transformed <N1,T1> pair

indicates that the message was indeed processed by the HA, and therefore that the message was received and forwarded by the HA. On the other hand, an active attacker may have changed the nonce N2 and token T2.

Within the design principles, no simple means that we are aware of allows the MN or the CN to be protected against an active attacker present in the same network with the MN. That is, an active attacker can play the role of CN towards the MN, and therefore also the role of MN towards the CN, and even send packets directly to the HA using the CN address as the source address without having the CN involved at all. On the other hand, having the HA involved (either through secure tunneling or explicitly) does provide a level of protection for the CN against passive attackers close to the MN. [XXX: More work is needed to determine the real limitations of the approach.]

After the checks, the MN proceeds to create the keying material BK. To do that, it first creates a fresh random number N_BK, and then combines the nonce N2 with it, according to the algorithm HASH_BK:

```
BK := HASH_BK( N_BK || N2 )      (see section 8.5)
```

Even though neither N2 nor N_BK are hidden (since that they have to be sent in clear over some unsecure link) they are never sent together or over the same link. N2 is only sent through CN->HA->MN, where even the HA->MN link may be encrypted, and N_BK is sent through MN->CN. As a result, the only viable points where eavesdropping is easy are the local link where the CN is located and (if the tunnel is unprotected) the local link where the MN is located.

If the CN is a stationary server, getting access to its local link is probably hard. On the other hand, if the CN is a mobile node itself, the MN is most probably sending the first and third messages to the CN through the CN's Home Agent, and therefore the first and third messages may well be encrypted as they arrive at the CN.

Perhaps the weakest point lies near to the MN, since the local link of the MN is likely to be wireless. If the HA uses the protected tunnel between HA and MN, this link is effectively secured. [XXX: but we need more security analysis on this point.]

6.3. **Message 3:** Binding Key Establishment (with Binding Update)

In the third message, the MN sends the pre-image of the token T1, the authentication token T2 and the new random number N_BK to the CN.

```
MN->CN: < CoA, CNA, HoA, T0, T2, N_BK >
```

When the correspondent node receives the message, it first generates
T1 according to the algorithm in section 8.2. Then CN verifies the

message's authenticity and freshness by verifying the token T2, using the algorithm in section 8.4. This verification allows the CN to establish the following beliefs:

- Some party received the second message that it sent to HoA, and is now replying to that message. That party also allegedly received N2 and generated N_BK, and therefore may be expected to have calculated and stored the keying material BK.

- Second, that same party also knew T0, which the MN is not supposed to reveal before it sends the third message.

Thus, to fool the check the attacker must have intercepted the BKE option sent by the MN and replaced it with its own. Furthermore, in order to possess the BK it must have been able to eavesdrop the Binding Key Request in order to learn N2.

6.4. Discussion

Unless there are holes (but taking into account how new this protocol is, we aren't so confident yet), the protocol seems fairly secure from the CN's point of view. Basically, there are two ways an attacker can break the protocol.

1. If the attacker, acting alone, is able to break the home agent check in the sense that no packets flow through the home agent, then the attacker is apparently able to create bindings for whatever address.

2. If the attacker, attacking while there is an MN running the protocol, is able either to learn the newly created keying material (BK) or to replace the correct keying material with something else, the the attacker will be able to later replace the Binding for that particular MN with something else.

In order to break the home agent check, the attacker must be able to eavesdrop packets flowing from the CN to the HA. If it can do that, then it can play the part of the MN even when it does not receive the packets forwarded by the HA. The only remedy against this would be to enhance the protocol in such a way that the CN can actually check that the real HA was involved. Unfortunately, this seems to be impossible since there we do not presuppose any security associations between the MN and CN.

DISCUSSION: It might be possible, though, to use the Home Address as a pre-established security context, as indirectly suggested in [8] and directly in [6]. There are two problems with such practice. First, there may be IPR

problems. Second, the practice goes beyond the currently
established security mechanisms, and requires rigorous

peer review before its security can be considered known. However, maybe a mechanism based on that idea should be an optional extension? Such an extension would not do any harm (other than slightly add complexity), but it might provide better protection for those hosts that do implement it.

To passively learn the BK, an attacker must be able to eavesdrop both the second packet containing N2 and the third packet containing N_BK. If the attacker is able to do this, it could most probably play the part of CN as well.

To replace the real BK with falsified one that it knows, an active attacker must be able to eavesdrop the second packet containing N2, and then produce a new third packet containing falsified N_BK. However, producing falsified third packet is hard unless the attacker is able to intercept the real third packet. Otherwise is, in order to pass the authentication check, the attacker would have to be able to reverse a one-way function and produce T0 from T1.

6.5. Tunnel Protection

There are three ways that the tunnel between the home agent and the mobile node can be protected against attack.

- The home agent can encrypt all packets destined for the mobile node, e.g., with ESP. This is the RECOMMENDED method.

- The home agent can disturb some of the data fields of the BKE message in some way that has been arranged in advance with the mobile node, beyond the scope of this specification.

- The home agent can insert some additional options, not defined in this specification.

7. Protocol Processing

In this section, processing details for protocol messages are specified.

7.1. Initiation of the protocol

A Mobile Node may initiate the protocol at any convenient moment. To do so, it includes a Binding Warning Destination Option into any packet containing the Home Address Destination Option.

The Binding Warning contains two 128-bit pieces of data: a nonce

N1, i.e., a newly generated random number, and a cryptographic token
T1, calculated over the nonce and the addresses involved. The key

K_(MN,HA), shared between the MN and the HA, is also used in the token generation, allowing the HA to check T1 if it wishes (see section 6.2.1). The token T1 is generated in such a way that it also allows the Correspondent Node to check whether the Binding Warning and the Binding Key Establishment messages have been sent by the same party. This is done by sending a value (here called T0) in the latter message that is could only have been produced with knowledge possessed by the party that created T1. If the latter was MN, and if MN does not share this information, then the correspondent node can be assured that both the Binding Warning and the Binding Key Establishment messages were both sent by the same mobile node.

Instead of being a random number, it is also possible to use a counter as the nonce N1. In that case, the Mobile Node MUST increment the counter every time it sends a Binding Warning to any node. Using a counter instead of a nonce allows the Home Agent to check T1 against replay attacks. By special agreement between the MN and the HA, more complex arrangements are possible; for instance, the uppermost 64 bits of N1 could be randomly generated and the lower 64 bits a counter.

[PNR: Should we be more specific about N1, and define that it consists of a 32-bit counter and 96-bit random number?]

[CEP: Yes, we should]

The mobile node computes T0, the pre-image of the token T1, using the algorithm specified in section 8.1. This pre-image will be sent to the Corresponding Node as a part of the Binding Key Establishment, and therefore the Mobile Node may want to save it. An alternative to saving is to recompute it when needed. After creating T0, the mobile node creates token T1 using the algorithm specified in section 8.2.

Once the MN has generated N1 and computed T1, it is ready to create the Binding Warning. Along with the Binding Warning option, the packet MUST also contain the Home Address option. Thus, the packet will have the following headers, in the following order.

```
     IP header
             destination address := Correspondent Node address (CNA)
             source address := Care-of-Address (CoA)
     Routing Header, if present
     Destination Options header, containing
             Home Address option
             Binding Warning option
     Fragmentation Header, if present
     IPsec headers, if present
     Upper layer data, if present
```

As shown, the packet MAY also contain higher-level protocol payload. For example, the upper layer data could be a TCP SYN packet, The Mobile Node MUST save the <N1,CoA,CNA> triplet so that it can more effectively match the Binding Key Request to be received. It MAY also want to save T0 and/or T1. In any case, it MUST store enough data so that T0 can be later retrieved, either by recomputing it or by remembering it.

If, after transmitting the Binding Warning, the mobile node does not receive a Binding Key Request, it MAY retransmit the Binding Warning up to BW_RETRIES times. The first retransmission MUST be delayed for BW_REXMIT_DELAY seconds, and every subsequent retransmission must be delayed for twice the amount of additional time as the previous retransmission was delayed.

7.2. Processing a received Binding Warning

When a Correspondent Node receives a packet containing a Binding Warning, it SHOULD process the Binding Warning and send a Binding Request based on that. However, it SHOULD limit creation of state until it receives the Binding Key Establishment packet back from the Mobile Node.

The reason for not creating state is to protect the CN from memory exhausting Denial-of-Service attacks. The present protocol MAY be used to defer state creation for upper layer protocols as well.

The incoming Binding Warning requires very little processing. The nonce N1 is basically just a random number, and since the correspondent node does not have the key K_(MN,HA), the token T1 appears random number as well.

7.3. Generating data for the Binding Key Request

To avoid creating state, the Correspondent Node encodes the relevant information into a cryptographic token T2. The same token will be included in the Binding Key Establishment, allowing the Correspondent Node to check that the Binding Key Establishment actually contains the same pieces of information that the Binding Warning did.

In addition to creating the the token T2, the CN also creates a structured nonce N2. From the protocol point of view, the nonce N2 is a random number, eventually used to create the keying material needed for the BSA. However, due to the requirement of not creating state at this point, the CN cannot simply generate a new random number and use it, since that would require that the CN remembered the generated number. Thus, instead, the CN cryptographically combines its key K_CN and the received token T1 to create N2. This allows the CN to later reconstruct N2. N2 inherits randomness from both K_CN and T1.

The CN first generates N2, using the received 128-bit token T1, according to the algorithm in section 8.3. Then T2 is generated using the nonce N2, according to the algorithm in section 8.4.

The implementation MAY also use some other method for generating the token T2, or include more information in the generation (like TCP port numbers), depending on the state that it wants to protect and forget.

7.3.1. Encoding TCP connection

As an OPTIONAL implementation issue, we describe how the Corresponding Node MAY encode initial TCP state into the token T2, allowing it to forget TCP state after sending a TCP SYN-ACK along the Binding Key Request.

Basically, the TCP protocol control block (PCB) created as a result of receiving a TCP SYN would contain the following information

- 16-bit local port number
- 16-bit remote port number
- 32-bit local sequence number
- 32-bit remote sequence number

Since these will be received in a reconstructible way in the forthcoming first ACK packet, the implementation MAY opt to encode this information into the token T2, and cease creating an explicit protocol control block. The eventual ACK packet will allow this information to be reconstructed, and reception of T2 allows the host to check that the reconstructed state matches with what is expected.

To encode the state into T2, [XXX: the details need to be written.]

7.4. Sending the Binding Key Request

Once the CN has the nonces N1 and N2 and the tokens T1 and T2 available, N1 and T1 from the received Binding Warning and N2 and T2 as generated above, it is ready send the Binding Key Request. The Binding Key Request MUST be sent to the Home Address of the Mobile Node, i.e., the address indicated in the Home Address destination option of the received Binding Warning packet. The transmission MUST bypass any possibly existing Binding Cache entry for that specific

Home Address, and there MUST NOT be any routing headers containing any of the mobile node's care-of addresses in the resulting packet.

Allowing non-mobility related routing headers, such as ones configured through manual means, should be considered very carefully. There might be legitimate reasons for doing so. Likewise, explicit tunneling requires careful study, and there are a number of good reasons to permit it.

The Binding Key Request packet has the following headers, in the following order:

```
     IP header
             destination address := mobile node's Home Address (HoA)
             source address := Correspondent Node address (CNA)
     Routing Header, if present (no mobile node CoA)
     Destination Options header, containing
             Home Address option
             Binding Key Request option
     Fragmentation Header, if present
     IPsec headers, if present
     Upper layer data, if present
```

As an example, the upper layer data could be a TCP SYN-ACK packet, i.e. the second packet in TCP three-way handshake.

7.5. Forgetting State

Once the CN has sent the Binding Key Request packet to the Home Address, it SHOULD forget all state created. It can simply discard all Binding related data since T0 and T2 are given as part of the Binding Key Establishment option; together with the key K_CN, this will allow it to reconstruct all necessary state variables. Furthermore, if the CN has coded the upper layer state in the T2, it MAY also opt to discard upper layer state such as a TCP protocol control block.

7.6. Processing Binding Key Request at the Home Agent

As a minimal requirement, Home Agent MUST forward the Binding Request to the Mobile Node through the established tunnel, unless it follows any of the more specific strategies outlined below.

The construction of the token T1 makes it possible to the Home Agent to verify that the Mobile Node has once generated T1 for the very purpose of communicating from the named CoA with the named

Corresponding Node. Furthermore, if there is a counter component in
N1, the Home Agent may also check against replay attacks.

In addition to OPTIONALLY checking T1, the Home Agent MAY also modify N1, and correspondingly T1, as per mutual agreement between the Home Agent and the Mobile Node. However, such practices go beyond the current scope of this specification. See section 6.5.

7.7. Processing Binding Key Request by the Mobile Node

The Mobile Node will receive the Binding Key Request as tunneled by the Home Agent. Whenever the Mobile Node is away from home, it MUST NOT accept Binding Key Requests that are sent directly to it instead of being tunneled by the Home Agent. Additionally, if the MN has an ESP protected tunnel with the HA, it MUST silently discard Binding Requests unless they have been protected with _that_particular ESP security association.

As a part of accepting the Binding Key Request, the Mobile Node MUST check that the outer header has the CoA as the destination address and the Home Address as the source address. If ESP is used, this check is typically performed as a part of the ESP policy processing. Additionally, it MUST check that the inner header has the Home Address (HoA) as the destination address and the Correspondent Node address (CNA) as the source address. The check over the inner header destination address MAY also be performed as a part of the ESP policy processing. However, checking of the source address of the inner header typically has to be performed separately. The Mobile Node MUST silently drop the packet if it fails the tunneling checks.

Once the Binding Key Request has passed the tunneling checks, further checks are made. First, if the MN saved a <N1,CoA,CNA> when making calculations for sending the Binding Warning (see section 7.1), it SHOULD check that the received nonce N1' matches the saved nonce N1, taking into account the OPTIONAL modification(s) by the HA, if used (see 6.5). Next, the Mobile Node SHOULD check that the received token T1' matches with the saved/recomputed token T1. In the basic case (no special arrangements between the MN and the HA), to perform the check the MN simply either uses its saved T1 value or recomputes T0 and T1, and compares the recomputed T1 value with the received T1'. If the MN knows that the HA may have changed the T1, the method for verifying T1 depends on the mutual agreement between the MN and the HA, and goes beyond the scope of this document.

Once the MN has accepted the Binding Key Request, it proceeds to establish the MN's end of a security association and prepare data for the Binding Key Establishment

7.8. Establishing the BSA

To establish the Mobile Node's end of the BSA that is used to authenticate the future Binding Updates, the MN generates a new random number N_BK. Then it proceeds to create BK from this new random number and the nonce N2 received in the Binding Key Request (see section 8.5).

```
BK := HASH_BK( N_BK || N2 )
```

The MN sets up an outgoing BSA using the BK as the keying material. The policy of the BSA MUST be set as follows.

1. The BSA MUST be only applied if the outgoing packet contains a Binding Update. It MUST NOT be applied if there are no BUs.

2. The destination address of the packet MUST be the address of the Correspondent Node.

Correspondingly, the MN sets up an incoming BSA using the BK as the keying material. The policy of this security association MUST be set as follows.

1. The BSA only protects Binding Acknowledgements. Unless there is additional IPsec protection, the packet is NOT considered integrity protected for any other purpose but informing the MN that the CN has received and processed the BU. The BSA MUST NOT be used for authenticating other packet data, or any fields in packets not containing a Binding Acknowledgement.

   DISCUSSION: Should we require that the IPv6 header has some particular source or destination IP addresses? The BSA should already be considered to identify the sender of the Binding Update.

7.9. Sending Binding Key Establishment (and Binding Update)

In addition to N_BK, the Binding Key Establishment option also needs T0, the pre-image of the token T1. Thus, the MN must provide this, either by recomputing it from N1 and the addresses, or by saving <N1,CoA,CNA,T0> after making the calculation during processing of the Binding Warning, as described in section 6.1.

To finish its part of the protocol run, the MN sends a packet that MUST contain at least the Binding Key Establishment (BKE) option and the Home Address option, located in the Destination Options header. The BKE option MUST immediately follow the Home Address option.

Typically the same packet would contain also the Binding Update that the MN initially wanted to send. If so, that Binding Update MUST be protected with the new BSA. The packet would also typically contain upper layer data such as a the first real data packet within a TCP connection.

Thus, the packet containing the BKE would appear as follows:

```
     IP header
             source address := mobile node's care-of Address (CoA)
             destination address := Correspondent Node address (CNA)
     Routing Header, if present
     Destination Options header, containing
             Home Address option
             Binding Key Establishment option
     Fragmentation Header, if present
     IPsec headers, if present
     Destination Options Header, containing
             Binding Update
     Upper layer data, if present
```

This packet structure leads to natural processing of the packet at the receiver end, allowing it to first verify the Binding Key Establishment and create the BSA before processing the Binding Update Header.

7.10. Receiving Binding Key Establishment

To the CN it may, effectively, appear to receive a Binding Key Establishment (BKE) option at any time, since it is likely to forget about previous Binding Warning messages.

When the CN starts to process the BKE option, it MUST check that there has been a Home Address Destination Option earlier in the destination options header.

To authenticate the BKE option, the CN collects the MN's current Care-of-Address from the IP header, the MN's home address (HoA) from the Home Address destination option, and its own IP address (CNA) from the IP header. It first calculates the token T1, according to the algorithm in section 8.2, using the received token T0 from the BKE option. It then recomputes the token T2 according to the algorithm in section 8.4, and compares the recomputed T2 with the received T2', as found in the received BKE option. If they are equal, the check succeeds.

Otherwise, if the comparison fails, the CN repeats the attempted verification using the next previous K_CN value, and recomputes the token T2. The exact number of previous saved previous keys and repeated checking attempts is a local policy issue. However, it is RECOMMENDED that at most three previous K_CN values are tried. This limits the amount of resources that a resource-exhausting denial-of-service attack may consume.

As an additional measure against resource-exhausting denial-of-service attacks, the correspondent node MAY limit the rate in which it checks Binding Key Establishment messages. It MAY also dynamically modify the number of additional checking steps that it is ready to perform in the case the the first check fails.

7.11. Establishing the CN Security Associations

Once the Binding Key Exchange has been verified, the CN proceeds to establish its BSA. To do so, it first recomputes the nonce N2, using the algorithm in section 8.3, and then combines the nonce N2 with the random number N_BK received in the BKE option to generate the same keying material (BK) that the MN generated.

The CN sets up an incoming BSA using BK as the keying material. The policy of this security association MUST be set as follows.

1. The BSA only protects the Binding Update. Unless there is additional IPsec protection, the data within the packet is NOT integrity protected for any other purpose except entering a new care-of address into the CN's binding cache.

2. The destination address of the packet SHOULD be the CN's address.

The CN sets up an outgoing BSA using the BK as the keying material. The policy of the security association MUST be set as follows.

1. The BSA MUST be only applied if the outgoing packet contains a Binding Acknowledgement. It MUST NOT be applied otherwise.

2. The destination address of the packet MUST be a home address of the Mobile Node.

7.12. Processing the rest of the packet

The rest of the packet MAY contain an AH header to protect the other headers and the payload. In that case, the rest of the packet is processed normally according to the rules in RFC 2406 [3].

7.13. Operations when the Mobile Node is at home

If the Mobile Node wants to run the described protocol while it is still at home, it MAY do so. In that case, the Care-of-Address and the Home Address will be the same. However, the packets containing the Binding Warning and Binding Key Establishment MUST still contain the Home Address destination option. Thus, the CN implementation does not need to care about whether the MN is at home or away from home while running the protocol -- from its point of view, the protocol will still run the same.

[XXX: Check SHA-1 and HMAC-SHA-1 input and output sizes.] [XXX: Check source and dest address order in IPv6 header, and update their inclusion order in hashes so that you can copy both with a single copy instead of requiring two.]

7.14. Processing For Correspondent Node Initiation

In addition to the Mobile Node, the Correspondent Node MAY also want to initiate the protocol. However, in that case some of the threats are different. Especially, since the CN is the initiator and the MN is the responder, it is important to protect the MN from resource exhausting denial-of-service attacks. Thus, from the security point of view, the MN initiated and the CN initiated versions of the protocol are different cryptographic protocols. However, the CN initiated version has been designed in such a way that almost all of the implementation is shared between these two protocol variants.

7.14.1. CN Initiation by sending a Binding Key Request

When the CN wants to initiate the protocol, it assigns zero for the Nonce N1, assigns zero for the Token T1, and generates the Nonce N2 and Token T2 as defined in Section 7.3. The values for Nonce N1 and Token T1 MUST be zero as they signal the MN that this is a CN initiated version of the protocol.

Note that since the key K_CN and the Home Address are used in the computation, the generated N2 is hard to predict even when T1 is fixed to zero. On the other hand, since the CN will be creating state and therefore will remember N2, it MAY just simply use a random number generator to create N2. The same applies for T2.

Once the CN has prepared values for nonces N1 and N2 and tokens T1 and T2, it proceeds to send a Binding Key Request as usual. The Binding Request MUST be sent to the Home Address of the Mobile Node.

When the CN is initiating the protocol (see appendix A, it MUST NOT clear its internal state. Instead it MUST remember that it has initiated the protocol with the MN.

7.14.2. Handling a CN initiated Binding Key Request by a HA

A Home Agent cannot do much for a CN initiated Binding Key Request. The optional check of token T1 would fail, but the fact that N1 and

   T1 are zero indicate that the protocol was initiated by the CN. In particular, the Home Agent SHOULD NOT replace the value T1 with another value that would pass the integrity test at the MN.

[XXX: There may be problems with this approach and the various ways we have thought the Home Agent could possibly participate to the protocol. More work is needed here.]

7.14.3. Handling a CN initiated Binding Key Request by a MN

When a MN receives a CN initiated Binding Key Request, zero N1 indicates it that it has not itself initiated the protocol this time.

Now, the MN has basically two different paths to follow. If it determines that it already has traffic going on with the CN, it MAY proceed the fast way and send a BKE packet immediately (see below). For example, the implementation MAY scan the active protocol control blocks, and finding a PCB with the CN as the remote address can be considered as positive indication that there is already traffic going on with the CN.

On the other hand, if the MN determines that it does not recognize the CN, it SHOULD defer creating state and continue in a stateless fashion. Again, there are two options here. First, the Mobile Node MAY decide to ignore the Binding Key Request completely. Second, it MAY send a Binding Warning in a stateless fashion.

If the MN decides to create state and send a Binding Key Establishment message, it cannot compute T0. On the other hand, it knows that the CN has state. Thus, to allow the CN to recognize the case, it simply sets T0 to zero. T2 is copied over from the Binding Request as usual, and N_BK is a random number as usual.

If the MN decides to send a Binding Warning in a stateless fashion, it proceeds as defined in Section 7.1. However, in this case the Mobile Node SHOULD NOT save the T1 or any other information about the fact that it has sent the Binding Warning. That is, when the MN receives a Binding Key Request from the CN the second time, it notices that the Token T1 is a valid one but that there does not exist any state, and continues to establish the state (i.e. BSA) and send the BKE.

In either case, the resulting BSA should have fairly short timeout. The timeout can be extended once the MN receives a properly authenticated Binding Acknowledgement from the CN.

7.14.4. Further operations at the CN end

As the CN has state, it will be expecting a Binding Warning or Binding Key Establishment. It will recognize the received Binding

Warning simply by the addresses. In that case proceeds normally other than that it SHOULD NOT forget its state after sending the (second) Binding Request, simplifying checks later on when the BKE is received.

On the other hand, if the CN receives a Binding Key Establishment, it will recognize the BKE as a reply to its Binding Key Request since T0 is zero, and again use the addresses to find its saved state. After that, it simply checks the received T2' against the remembered T2, and proceeds normally.

7.15. Simultaneous operation by two Mobile Nodes

In this section, we present a nonexistent discussion about various issues when two Mobile Nodes, both acting as a Correspondent Node to each other, want to run this protocol at the same time.

8. Cryptographic Algorithms

The various algorithms for creating tokens and keys are collected in this section. This is done for three reasons:

- To make sure the same algorithm is used even when specified at different places in the document

- To allow for easy modifications if consensus develops within the working group

- For easy cross-reference throughout the document

8.1. Algorithm for Computing Token T0

The input for the algorithm HASH_T0 is created by concatenating the following data:

1. N1, the nonce generated by the mobile node for this purpose.

2. CoA, the mobile node's 128-bit care-of-address

3. CNA, the 128-bit Correspondent Node address

4. HoA, the mobile node's 128-bit home-address

This concatenation (N1 || CNA || CoA || HoA) results in a 512-bit bitstring.

Calculate a hashed MAC, as defined in [4], over the 512-bit bitstring, using the key K_(MN,HA). The RECOMMENDED algorithm is

HMAC-SHA-1 [5], but since the exact algorithm is an issue private to the MN and HA, other algorithms can be used according to mutual agreement.

Obtain the result by taking the rightmost 128 bits of the resulting HMAC code. This is the pre-image of the token T1, called T0. This pre-image will be sent to the Corresponding Node as a part of the Binding Key Establishment, and therefore the Mobile Node may want to save it. An alternative to saving is to recompute it when needed.

8.2. Algorithm for computing token T1

The input data for the algorithm HASH_T1 is created by padding the 128-bit T0 to the left with 32 zero bits to form a bitstring with 160 bits.

Apply basic SHA-1 [5] over the bitstring.

Obtain T1 (the result) by taking the rightmost 128 bits of the the SHA-1 result.

8.3. Algorithm for computing nonce N2

The input data for algorithm HASH_N2 is obtained by concatenating a 128-bit (16 byte) zero value and the received 128-bit token T1, resulting in a bitstring with 256 bits.

Calculate a hashed MAC, as defined in [4], over the bitstring, using tke key K_CN. The RECOMMENDED algorithm is HMAC-SHA-1 [5], but since the exact algorithm is an issue private to the correspondent node, the implementation MAY elect to use some other algorithm.

Obtain the result by taking the rightmost 128 bits of the resulting HMAC code. This is the nonce N2.

8.4. Algorithm for computing token T2

The input data for algorithm HASH_T2 is obtained by concatenation of the following values:

1. T1, the received 128-bit token generated by the mobile node in the Binding Warning destination option

2. CoA, the mobile node's 128-bit care-of-address, and

3. CNA, the 128-bit Correspondent Node address from the IP header

destination address field

4. HoA, the 128-bit home-address from the Home Address destination option

This concatenation (T1 || CoA || CNA || HoA) results in a 512-bit bitstring.

Calculate a hashed MAC, as defined in [4], over the 512-bit bitstring, using the key K_CN. The RECOMMENDED algorithm is HMAC-SHA-1 [5], but since the exact algorithm is an issue private to the CN, the implementation MAY select to use some other algorithm.

Obtain the result by taking the rightmost 128 bits of the resulting HMAC code. This is the token T2.

8.5. Algorithm for computing key material BK

The data for algorithm HASH_T2 is obtained by concatenation of the following values:

1. N_BK, the random number generated for this purpose by the mobile node and included in the Binding Key Establishment message

2. N2, the 128-bit nonce calculated by the correspondent node and delivered to the mobile node in the Binding Key Request message.

This concatenation (N_BK || N2) results in a 256-bit input bitstring.

```
KeyMat := MD5( N_BK || N2 )
```

Obtain the result by hashing (using MD5 [9]) the 256-bit input. This is the desired key material BK.

9. Protocol Data Unit Formats

9.1. Binding Warning Message

A mobile node may use the Binding Warning destination option to enable a correspondent node to initiate the process of establishing a Binding Key for use when authenticating its Binding Update messages.

Nonce N1	A 128-bit opaque value.
Token T1	A 128-bit hashed value, computed over the addresses involved.

See section 7.1 for processing details on the selection of the Nonce N1, and the calculation of the Cryptographic Token T1.

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                      |  Option Type  | Option Length |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      =                      Nonce N1 (128 bits)                      =
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      =                 Cryptographic Token T1 (128 bits)             =
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: The Binding Warning Destination Option format

9.2. Binding Key Request Message

A correspondent node may use the Binding Key Request destination option to transmit key material to a mobile node.

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                      |  Option Type  | Option Length |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      =                      Nonce N1 (128 bits)                      =
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      =                 Cryptographic Token T1 (128 bits)             =
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      =                   Structured Nonce N2 (128 bits)              =
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      =                 Cryptographic Token T2 (128 bits)             =
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: The Binding Key Request Destination Option format

Nonce N1	A 128-bit opaque value, copied over from corresponding value in the Binding Warning sent from the mobile node.
Token T1	A 128-bit hashed value, copied over from corresponding value in the Binding Warning sent from the mobile node.
Nonce N2	A 128-bit hashed value created using a new nonce and the information from the mobile node.
Token T2	A 128-bit authentication token which includes all information whose integrity must be protected, as well as T1.

See sections 7.3 and 6.2 for details on the the calculation of the Structured Nonce N2, and the Cryptographic Token T2. Including T1 into the authentication token T2 helps to block some active attacks.

9.3. Binding Key Establishment Destination Option

A mobile node may use the Binding Key Establishment destination option to transmit key material to a correspondent node.

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                       |  Option Type  | Option Length |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                        Lifetime (32 bits)                     |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       =                       Nonce N1 (128 bits)                     =
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       =                Cryptographic Token T0 (128 bits)              =
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       =                Cryptographic Token T2 (128 bits)              =
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       =                  Random Number N_BK (128 bits)                =
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: The Binding Key Establishment
Destination Option format

Lifetime	A 32-bit value specifying the lifetime of the BSA in seconds.
Nonce N1	A 128-bit opaque value, copied over over from the information in the Binding Key Request.
Token T0	A 128-bit hashed value, previously computed and possibly saved when sending the Binding Warning.
Token T2	A 128-bit authentication token which is copied over from the information in the Binding Key Request.
Random N_BK	A 128-bit random number, used as the second half of the keying material to be created.

See sections 6.2 and 8 for details on the the calculation of the Cryptographic Tokens T0 and T2. The mobile node SHOULD check that the nonce N1 corresponds to the nonce which it sent in the Binding Warning message to the correspondent node.

Acknowledgements

We would like to thank the members of the Mobile IP and IPng Working Groups for their comments and suggestions on this work. We would particularly like to thank (in alphabetical order) N Asokan (Nokia) Francis Dupont (ENST Bretagne) Michael Thomas (Cisco)

References

[1] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. Request for Comments (Best Current Practice) 2119, Internet Engineering Task Force, March 1997.

[2] D. Johnson and C. Perkins. Mobility Support in IPv6 (work in progress). draft-ietf-mobileip-ipv6-13.txt, October 2000.

[3] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). Request for Comments (Proposed Standard) 2406, Internet Engineering Task Force, November 1998.

[4] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. Request for Comments (Informational) 2104, Internet Engineering Task Force, February 1997.

[5] C. Madson and R. Glenn. The Use of HMAC-SHA-1-96 within ESP and AH. Request for Comments (Proposed Standard) 2404, Internet Engineering Task Force, November 1998.

[6] G. Montenegro and C. Castelluccia. Statistically Unique and Cryptographically Verifiable Identifiers (work in progress). draft-montenegro-sucv-00.txt, April 2001.

[7] T. Narten and R. Draves. Privacy Extensions for Stateless Address Autoconfiguration in IPv6, January 2001.

[8] P. Nikander. An Address Ownership Problem in IPv6 (work in progress). draft-nikander-ipng-address-ownership-00.txt, February 2001.

[9] R. Rivest. The MD5 Message-Digest Algorithm. Request for Comments (Informational) 1321, Internet Engineering Task Force, April 1992.

A. Correspondent Node Initiated Operation

From the security analysis point of view, the CN initiated variant of the protocol is a separate protocol from the MN initiated one. Thus, it must be analyzed separately. Furthermore, since we are sharing much of the implementation between these two protocols, it is also necessary to analyze the protocols together in order to discover potential pitfalls. In this section, we analyze the CN initiated protocol variant in isolation.

In the CN initiated case, we have two possibilities. The first possibility is that the MN already knows the CN, and therefore the initial knowledge sets can be expressed as follows.

```
MN: { CoA, HoA, K_(MN,HA), CNA }
HA: { CoA, HoA, K_(MN,HA) }
CN: { CNA, HoA }
```

In this case the main threat seems to be replay attacks. That is, resource exhausting DoS is fairly hard since both the MN and the CN already have their peer's address, and no actual new state must be preserved. On the other hand, unless proper freshness of the BSA keying material can be assured, various replay attacks seem to be fairly easy.

The shortened protocol can be described as follows. (Zero elements have been removed.)

```
CN->HA: < CNA, HoA, N2, T2 >
HA->MN: TUNNEL < CNA, HoA, N2, T2 >
MN->CN: < CoA, CNA, HoA, T2, N_BK >
```

Here the token T2 assures freshness of the BKE message. On the other hand, the MN has no way of determining if the first Binding Key Request is fresh or not. XXX: More analysis needed.

The case when the MN does not know the CN is more difficult. The initial knowledge sets can be described as follows.

```
MN: { CoA, HoA, K_(MN,HA) }
HA: { CoA, HoA, K_(MN,HA) }
CN: { CNA, HoA }
```

Thus, the MN learns the CN address CNA only by receiving the Binding Key Request. If it were to simply accept it and proceed with establishing state, this would open up a trivial denial-of-service attack. Furthermore, as the MN may be a small device with limited storage capacity, such an attack would be even more serious.

The first and safest option for the MN is simply ignore such a Binding Request. A second option is to proceed much in the same way as the CN works when it receives an unsolicited Binding Warning, i.e., in a way that does not necessitate creating state. The basic protocol can be expressed as follows.

```
     First Binding Update:
             CN->HA: < CNA, HoA, N2, T2 >
             HA->MN: TUNNEL < CNA, HoA, N2, T2 >
     Binding Warning:
             MN->CN: < CoA, CNA, HoA, N1, T1 >
     Binding Key Request:
             CN->HA: < CNA, HoA, N1, T1, N2', T2' >
             HA->MN: TUNNEL < CNA, HoA, N1, T1, N2', T2' >
     Binding Key Exchange:
             MN->CN: < CoA, CNA, HoA, T0, T2', N_BK >
```

As said, the crucial issue here is to ensure that the MN does not need to create state when it receives the first Binding Key Request. Following the principles that we used in the MN initiated variant of the protocol, it would be best to compute the the hash value T1 so that it covers N2 and/or T2 in addition to the addresses and N1 that it usually covers, to require that the CN uses the same N2 and T2 when sending the second Binding Key Request, and let the MN to check this coverage upon receiving the second Binding Key Request.

DISCUSSION. Including T2 to T1 and requiring that T2' == T2 is not included in the spec below, but perhaps it should be. Further protocol analysis is required in any case. In order to ease initial implementations, we have simply specified that the protocol flows as usual. We need more experience about the actual implementation.

[XXX: Add full protocol analysis here.]

B. An open question about this draft

We are currently considering whether we should add authenticated Diffie-Hellman (D-H) support as an option. One suggested way to accomplish this is to unify the first message (Binding Warning) of this protocol with the first message of some other protocol supporting D-H, e.g. [6]. Another way would be to further complicate this protocol and explicitly allow D-H based key exchange, combined with strong authentication, in messages 2 and 3 (Binding Key Request and Binding Key Establishment).

In any case, we do not currently see much value of adding

non-authenticated D-H support to this protocol, since such an
extension would be equally vulnerable to active attacks, and the

added protection against passive attacks does not pay off the added complexity.

Chair's Address

The Working Group can be contacted via its current chairs:

```
       Basavaraj Patil                      Phil Roberts
       Nortel Networks Inc.                 Megisto Corp.
                                            Suite 120
       2201 Lakeside Blvd.                  20251 Century Blvd
       Richardson, TX. 75082-4399           Germantown MD 20874
       USA                                  USA
       Phone:  +1 972-684-1489              Phone:  +1 847-202-9314
       Email:  bpatil@nortelnetworks.com    Email:  PRoberts@MEGISTO.com
```

Authors' Addresses

Questions about this memo can also be directed to the authors:

```
     Pekka Nikander                   CharlesCE.oPerkinsmmunications Systems
Lab
     Ericsson Research NomadicLab
     P.O.BOX 58                       Nokia3Research1Center3 Fairchild Drive
     FIN-02131 ESPOO
     Finland                          MountainUView,SCaliforniaA94043
     Phone:  +358-40-721-4424
     Pekka.Nikander@nomadiclab.com    Phone:Em+1-650a625-2986il:
charliep@iprg.nokia.com
     Fax:  +358-9-299-5055            Fax:  +1 650 625-2502
```