Multi6
Internet-Draft
Expires: April 18, 2005

A. Nagarajan
H. Tschofenig
Siemens

October 18, 2004

Comparative Analysis of Multi6 Proposals using a Locator/Identifier Split
draft-nagarajan-multi6-comparison-00.txt

Abstract

An IP address serves as a locator and an identifier in the classical Internet environment. This dual role of the IP address makes mobility and multihoming a challenging task. There have been various proposals to overcome this limitation, particularly from the Multi6 working group.

This memo makes a comparative analysis of these proposals that support a locator/identifier split for multihoming in IPv6 from the

security point of view. The purpose is also to provide a common framework under which future proposals can be compared and chosen for various security requirements.

Table of Contents

1. Introduction

The Multi6 group aims at providing potential solutions for multihoming support in IPv6. Most of these proposals try to define a new convergence layer operating between the classic internet and the transport layers [8]. This eventually splits the dual role of the IP from being both locators and identifiers to only locators. For naming the end hosts, new identifiers are thought of. Such locator/identifier split multihoming solutions bring in new security considerations.

The IETF draft [6] discusses some common threats relating to IPv6 multihoming solutions. In this memo we look into the security issues of some Multi6 proposals that deal with the locator/identifier split and try to compare them on the basis of the threats draft. We look at the proposals from the following point of view: There is a protocol which establishes state at two endpoints. For future state updates (in case of multihoming and mobility) an authorization problem exists as to who is allowed to modify the pre-established state and how. We analyse the properties of the proposed protocols.

2. Strong Identity Multihoming

The SIM proposal [1] assumes that the problem to be solved is site multihoming, with the ability to have the set of site locator prefixes change over time due to site renumbering. It also assumes that public key signatures are not required for normal communication. They are used for verification only at the time of locator prefix changes for a host or when two hosts claim to use the same identifier.

This proposal aims to achieve site multihoming by introducing an M6 shim layer between the classic IP and the transport layer. All the applications and the upper layer protocols use identifiers to name the hosts. These identifiers are hashes of the public key of the hosts. The network layer uses IPv6 to locate the hosts on the internet. The M6 layer is an extension header between the IP and the transport layer that maps the identifiers to the locators and vice versa. From the perspective of the upper layers, the packets seem to travel end to end using identifiers, although they are rewritten with locators on flight.

2.1 Notational Conventions

A is the initiating host and B is the responding host. X is a potentially malicious host.

FQDN (A) is the domain name for A.

Ls(A) is the locator set for A, which consists of L1 (A), L2 (A) until Ln (A).

ID(A) is an application ID for A.ID(A) is a 128 bit number consisting of two fixed bits (e.g., 10) followed by 126 bits of a truncated SHA1 hash of a public key that the host has generated.

CT(A) is a 64 bit "context tag" allocated by A and used when B sends packets to A. The packets contain the low-order 32 bits of the tag, named CT32 (A). The full tag is used for DoS-attack prevention during the PK challenge/response.

2.2 SIM Protocol Overview

The SIM proposal is composed of two parts. The protocol starts with a 3-way hand shake between the communicating parties where a context state is established in the M6 layer. The context state includes information on the peer and local identities, the locator set of the peer and the local host, the verification status of each locator of the peer and the context tags of the initiator and the receiver. The

context establishment as illustrated in Figure 1is composed of the Context Request, Context Response and the Context Confirm messages. The details of this message exchange can be looked into the IETF draft [1].

```
 I with ID(I) at L1(I)
 R with ID(R) at L1(R)

 I-->R, CReq :Nonce(I),ID(I),ID(R),CT(I)

 R-->I, CRes :Nonce(I),Context[ID(I),ID(R),CT(I),CT(R),
              L1(I),L1(R)],TimeStamp,Hash(Context,TimeStamp)

 I-->R, CCon :Context[ID(I),ID(R),CT(I),CT(R),L1(I),L1(R)],
              Hash(Context,TimeStamp)
```

Figure 1: SIM protocol Context Establishment

The second part of the SIM proposal is the challenge request-response protocol shown in Figure 2 that is used when one of the hosts changes its locator. The protocol relies on the fact that the mobile node can prove the knowledge of the 64 bit context tags of both the initiator and the receiver that was used only at the time of state establishment.

```
I with ID(I) at L2(I)
R with ID(R) at L1(R)

I-->R, Data packet from unknown locator L2(I)

R-->I, ChaReq:Nonce(R),ID(I),ID(R),CT32(R),L2(I)

I-->R, ChaRes:[Nonce(R),CT32(R),L2(I),
       Hash(Nonce(R),ID(I),ID(R),CT(I),CT(R)),PK(R)]Sign(R)
```

Figure 2: SIM protocol Readdressing

2.3 SIM Security Considerations

The initiator of the communication first tries to establish a host-pair context with the peer based on information it learns from the DNS.The responder later establishes some initial state with the initiator using the context creation 3-way handshake. Locator updates are later possible using signature verifications. However,

it is very crucial that these locator changes be authenticated to avoid man in the middle attacks.

In order to prevent such attacks during locator updates, the SIM protocol relies on the ability to verify that the entity requesting redirection indeed holds the private key where the hash of the corresponding public key hashes to the ID itself.

1. PREMEDIATED REDIRECTION

From our analysis of this proposal, it is evident that a premediated redirection attack[6] on host B is possible due to the absence of an initiator authentication. When host A initiates a communication with B based on a DNS look up for B, B initiates a 3-way exchange in order to establish a context among themselves. However, B does not make any attempt to reverse look up A in order to verify that A is who he claims to be. They eventually establish a context pair and start exchanging data packets. A malicious host X could easily claim to be A and encourage B to exchange data with it attempting a premediated redirection attack. A reverse lookup with the L(X)->FQDN and then forward lookup for FQDN->ID(X) would prove if X is who he claims to be. SIM fails to do this and remains unknown about the redirection until A genuinely initiates a communication with B.

2. REDIRECTING PACKETS TO ATTACKER

Any protocol that does not know how to authenticate locator changes for a mobile host is open to redirection attacks. The SIM protocol however uses public key signature verifications to verify locator updates and is not susceptible to such attacks. If Host X claims to be A that has just moved and encourages B to send data packets to it, it (and A) will receive a challenge request from B. B waits until it receives the first correct response signed with the private key that corresponds to the public key of A. It would be impossible for anybody other than A (even X) to sign using A's private key. However, host X could attempt to increase the cost of computation on B's side by compelling it to verify more and more signatures. This by itself could be a denial of service attack.

3. REDIRECTING PACKETS TO A BLACK HOLE

This variant of the classic redirection attacks [6] is different because this forces a host to send packets to a nonexistent locator and eventually dropping them off.

Assuming that A and B are already communicating and X sends a locator update to B claiming that it is A and has moved (to a black hole), both A and the black hole locator receive challenge requests from B. It is evident that only A sends an acceptable response forcing B to ignore the locator update from X.

4. REDIRECTING PACKETS TO A THIRD PARTY

Mechanisms that prevent the previous redirection attacks can prevent this one too. Assuming that A and B are already communicating, the SIM proposal makes sure that no other host other than A itself would be able to authenticate to B on the basis of a challenge response that is signed with A's private key.

5. PACKET INJECTION

The SIM protocol introduces a M6 shim layer between the IP and the transport layers. This M6 layer is just an extension header for the data packets as shown below. It would be suffice for an attacker X to learn just the 32 bit context tag to inject false packets in a sequence. Though it would be difficult for nodes out of path between A and B to guess the context tag, it would be sufficiently easy for those between the communicating nodes to learn their context tags, create new forged packets and inject them.

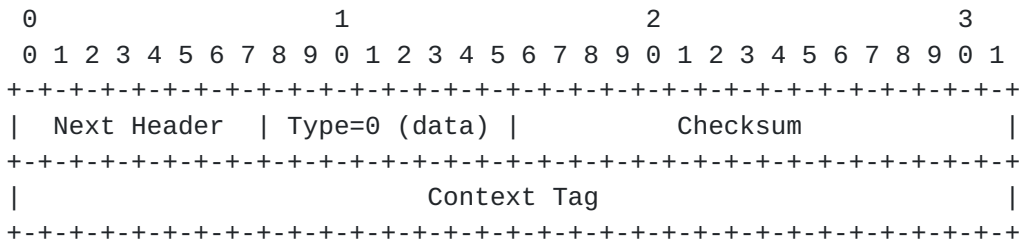


Figure 3: M6 packet header

This situation is no different from other IP packets of today which do not experience any IPsec protection. As a proposed solution, IPsec could be used along with the SIM proposal.

6. RESOURCE EXHAUSTION DENIAL OF SERVICE ATTACKS

During the initial phase of the context establishment, the proposal avoids using heavy computations in order to remain stateless. It does not introduce PK signature verifications until there is a locator change and necessitates one. However, this could in turn present a resource exhaustion denial of service attack. The attacker could spoof one or many challenge response packets to the receiver and force it to do heavy PK signature verifications. This would exhaust the responser's resources posing a denial of service to the original initiator.

3. Multihoming without Identifiers

The NOID proposal [2] assumes that the problem to be solved is site multihoming, with the ability to have the set of site locator prefixes change over time due to site renumbering. It also assumes that the DNS infrastructure can be used for verification of the relationship between locators on both the initiator of communication and the responding peer. Further, doing a reverse look up for each locator in the locator set to its FQDN is assumed not to create significant problem.

NOID proposes an approach for endpoint identifier and locator separation where the endpoint identifier space is drawn from the locator space. Instead of creating a new namespace for endpoint identifiers, the endpoint identifier may be chosen from the set of locators itself. This initial locator could be used for communication among hosts until one of them necessitate a locator update.

3.1 Notational Conventions

A is the initiating host and B is the responding host. X is a potentially malicious host.

FQDN (A) is the domain name for A.

Ls(A) is the locator set for A, which consists of L1 (A), L2 (A) until Ln (A).

AID (A) is an application ID for A. In this proposal, AID (A) is always one member of Ls (A).

3.2 NOID Protocol Overview

The NOID proposal is composed of two parts. The protocol starts with a 3-way hand shake between the communicating parties where a context state is established in the M6 layer. The context state includes information on peer locator which the upper layer protocol uses as the AID, the local locator which the application uses as the AID, the locator set of the peer and the local host, the verification status of each locator of the peer and the FlowIDs of the initiator and the receiver. The second part of the NOID proposal is the locator updates. Assuming that the initiator is mobile and sends a packet to the receiver from a new locator L2(I), the receiver would now have to perform some return routability test to make sure that the L2(I) indeed belongs to the initiator. For this, NOID relies on DNS verifications. The receiver performs a reverse look up for L2(I) to see if L2 belongs to the locator set of I. This way, it confirms

that I is reachable at L2 and makes L2 as the preferred locator.

```
I with AID(I) at L1(I)
R with AID(R) at L1(R)

I-->R, CReq :Nonce(I),AID(I),AID(R),FlowID(I)

R-->I, CRes :Nonce(I),Context[AID(I),AID(R),FlowID(I),FlowID(R),
             L1(I),L1(R)],TimeStamp,Hash(Context,TimeStamp)

I-->R, CCon :Context[AID(I),AID(R),FlowID(I),FlowID(R),L1(I),L1(R),
             TimeStamp,Hash(Context,TimeStamp)
```

Figure 4: NOID protocol Context Establishment

3.3 NOID Security Considerations

Conceptually, the NOID proposal is similar to that of the SIM. It adds a layer to the middle of IP above most IP processing, but below IPsec, fragmentation and reassembly functions [7]. It assigns one of the locators in the locator set of a host as the host identifier and uses global DNS as a mapping system between IDs and Locators.

Since the NOID proposal does not make use of cryptographic identifiers, it does not perform PK signature verifications like the SIM. As an alternative to prevent redirection attacks, it relies on the DNS (for the hosts which support this protocol) being maintained with consistent forward and reverse maps.

1. PREMEDIATED REDIRECTION

   We understand that the NOID proposal is quite secure from the premediation attacks [6] due to timely reverse DNS lookups. When an attacker X using L(X) spoofs like host A and sends a M6 data packet to B to start a communication, B does not authenticate X at this point. It sends the context request message to initiate the 3 way exchange and waits for the context response message from the initiator. However, after B sends the context confirm message to the initiator, it starts asynchronously and incrementally extracting and verifying the locator set of the initiator from the DNS. This verification would fail as L(X) would not look up to FQDN(A).NOID suggests such packets be dropped and an unknown context error be sent.

2. REDIRECTING THE PACKETS

In NOID, redirection attacks on the pretext of locator changes are not possible by an attacker. This is because as soon as a host receives packets from a new locator, it first looks up the context states (already has a list of verified locators) using the flow ID and the locators. If this look up succeeds then the locator is acceptable for incoming packets. However, if the locator has not been verified then it puts the new locator on the top in the list of asynchronous DNS verifications that are needed to be made. Assuming that A and B are already communicating and X (using L(X) and FQDN(A)) is attempting a redirection attack on B, X will never be successful as its L(X) will not be looked up to FQDN (A) in the DNS records.

The strength of the NOID proposal relies on the DNS look up and it is important that any genuine host has authentic DNS records. It is out of the scope of this draft to look at such security threats. The DNS reverse look up method prevents all kinds of redirection attacks including those to the attacker, a third party or a black hole.

3. PACKET INJECTION

The current draft of the NOID proposal does not talk in detail about the format of the data packets. It is however clear that any node in the path between the hosts can look at the source and destination locators and the flow IDs of the packets being exchanged. Therefore, it should not be difficult to inject false packets or manipulate the flow IDs to affect the original data stream.

4. RESOURCE EXHAUSTION DENIAL OF SERVICE ATTACKS

The NOID proposal uses DNS look up to authenticate the peer host and for locator changes verifications. Since DNS look up are so important in NOID, it is also important that the DNS that is looked up into also has only authentic DNS records. Hence, a host could use a signed DNS zone or some secure mechanism for forward or reverse look ups. For instance, when an attacker X pretending to be host A, spoofs one or many packets to host B, host B would put the locators of X on the top of the list for DNS verifications. If it is a signed DNS zone or a secure look up mechanism, it would end up consuming a lot of resources of B.

4. Multihoming using 64-bit Crypto-Based IDs

The CB64 proposal [3] is remarkably similar to the NOID and the SIM proposals. It assumes that the problem to be solved is site multihoming, with the ability to have the set of site locator prefixes change over time due to site renumbering. It also assumes that public key signatures are not required for normal communication. They are used for verification only at the time of locator prefix changes for a host or when two hosts claim to use the same identifier.

According to the NOID proposal, hosts that interact with each other are identified using the AIDs or application IDs. AIDs are also a part of Ls and are of 128 bits. NOID makes use of flow IDs so that mapping to the correct AID at the receiving end can be accomplished.

According to the SIM proposal, hosts that interact with each other use 128 bit cryptographic identifiers that consist of a truncated SHA1 hash of a public key that the host has generated. SIM also makes use of context tags that allow the receiver to find the correct context without relying on the locators in the packet.

CB64 tries to keep the AID and the Flow ID of NOID. However, AIDs in CB64 are a combination of locators and cryptographic identifiers. ID(A) which is a 64 bit hash of the public key is embedded in the lower order bits of AID(A) [which is a part of the Ls (A)]. The upper 64 bits is the subnet locator.

4.1 Notational Conventions

A is the initiating host and B is the responding host. X is a potentially malicious host.

FQDN (A) is the domain name for A.

ID(A) is the 64 bit CBID for A

Ls(A) is the locator set for A, which consists of L1(A), L2(A), until Ln(A). Each Lk(A) contains ID(A) in the low-order bits. The high-order 64 bits of a locator are sometimes referred to as the "subnet locator".(The protocol does not prevent a single host having multiple identities, but that can be viewed multiple entities A, B etc existing on the same host).

AID (A) is an application ID for A. In this proposal, AID (A) is always one member of Ls (A).

4.2 CB64 Protocol Overview

The CB64 proposal is made of two parts. The protocol starts with a 3-way hand shake between the communicating parties where a context state is established in the M6 layer. The context state includes information on peer locator which the upper layer protocol uses as the AID, the local locator which the application uses as the AID, the locator set of the peer with each containing the same 64 bit ID in the lower bits, the locator set of the local host with each locator containing the same 64 bit ID in the lower bits, the verification status of each locator of the peer and the FlowIDs of the initiator and the receiver.

```
I with AID(I) at L1(I)
R with AID(R) at L1(R)

I-->R, CReq :Nonce(I),AID(I),AID(R),FlowID(I)

R-->I, CRes :Nonce(I),Context[AID(I),AID(R),FlowID(I),FlowID(R),
             L1(I),L1(R)],TimeStamp,Hash(Context,TimeStamp)

I-->R, CCon :Context[AID(I),AID(R),FlowID(I),FlowID(R),L1(I),L1(R),
             TimeStamp,Hash(Context,TimeStamp)
```

Figure 5: CB64 protocol Context Establishment

The second part of the CB64 proposal is the challenge request-response protocol that is used when one of the hosts changes its locator. The protocol relies on the fact that the mobile node can prove the knowledge of the FlowIDs of both the initiator and the receiver that was used only at the time of state establishment.

```
I with AID(I) at L2(I)
R with AID(R) at L1(R)

I-->R, Data packet from unknown locator L2(I)

R-->I, ChaReq:Nonce(R),AID(I),AID(R),FlowID(R),L2(I)

I-->R, ChaRes:[Nonce(R),FlowID(R),L2(I),
       Hash(Nonce(R),AID(I),AID(R),FlowID(I),FlowID(R))PK(R)]Sign(R)
```

Figure 6: CB64 protocol Readdressing

4.3 CB64 Security Considerations

CB64 uses 128 bit locators that are embedded with the cryptographic identifiers to locate a host on the internet. Similar to the NOID proposal, it does not use separate identifiers (like the SIM proposal) but assigns one of these locators as the application identifier for a host. However, since application identifiers hold cryptographic identifiers, CB64 performs PK signature verifications similar to SIM for locator updates verifications.

In order to prevent redirection attacks during locator updates, the CB64 protocol relies on the ability to verify that the entity requesting redirection indeed holds the private key where the hash of the corresponding public key hashes to the ID itself.

1. PREMEDIATED REDIRECTION

   From our analysis of this proposal, it is evident that a premediated redirection attack [6] on host B is possible due to the absence of an initiator authentication. This scenario is quite similar to that of the SIM proposal

2. Other security issues

   All the security issues of CB64 are exactly similar to that of the SIM. This is because both the SIM and the CB64 rely on similar PK signature verification mechanism for locator changes authentication. Therefore, it is strongly recommended that security considerations of the SIM proposal be referred.

5. Weak Identifier Multihoming Protocol

The WIMP proposal [5] uses a new logical layer between networking and transport layers that separates the end-point identifier and locator roles from each other. The identifiers are used to name the end-points, while IPv6 addresses are used for routing purposes. Identifiers in WIMP are 128-bit non-routable IDs that are never used in packets on the network. These IDs are locally generated for both local and remote nodes by hashing a nonce (for the initiator's endpoint identity) and the FQDN (for the responder's endpoint identity). The WIMP layer manages the mapping between IDs and locators based on internal state.

Communication between two end-points requires establishment of a WIMP session. Once the session is established, it can be used for multiple simultaneous or sequential connections to the same end-point. During WIMP session establishment, WIMP introduces a separate header into the data packets, between the IP and TCP/UDP headers that contains information about the WIMP session. WIMP does not introduce a separate header into all IPv6 (data) packets. Instead, once a WIMP session is established, the IPv6 FlowID is used to hold an identifier for the WIMP host-pair context associated with a given packet. The FlowIDs serve as a convenient "compression tag" without increasing the packet size.

5.1 Notational Conventions

A is the initiating host and B is the responding host. X is a potentially malicious host.

ID(I) is an identifier for I and ID(R) is an identifier for R.

FQDN(R) is the domain name for R.

Ls(I) is the locator set for I, which consists of L1(I), L2(I) until Ln(I).

Ls(R) is the locator set for R.

F(I) is a FlowID assigned by the initiator and used by the responder. The responder uses F(I) in packets going to the initiator.

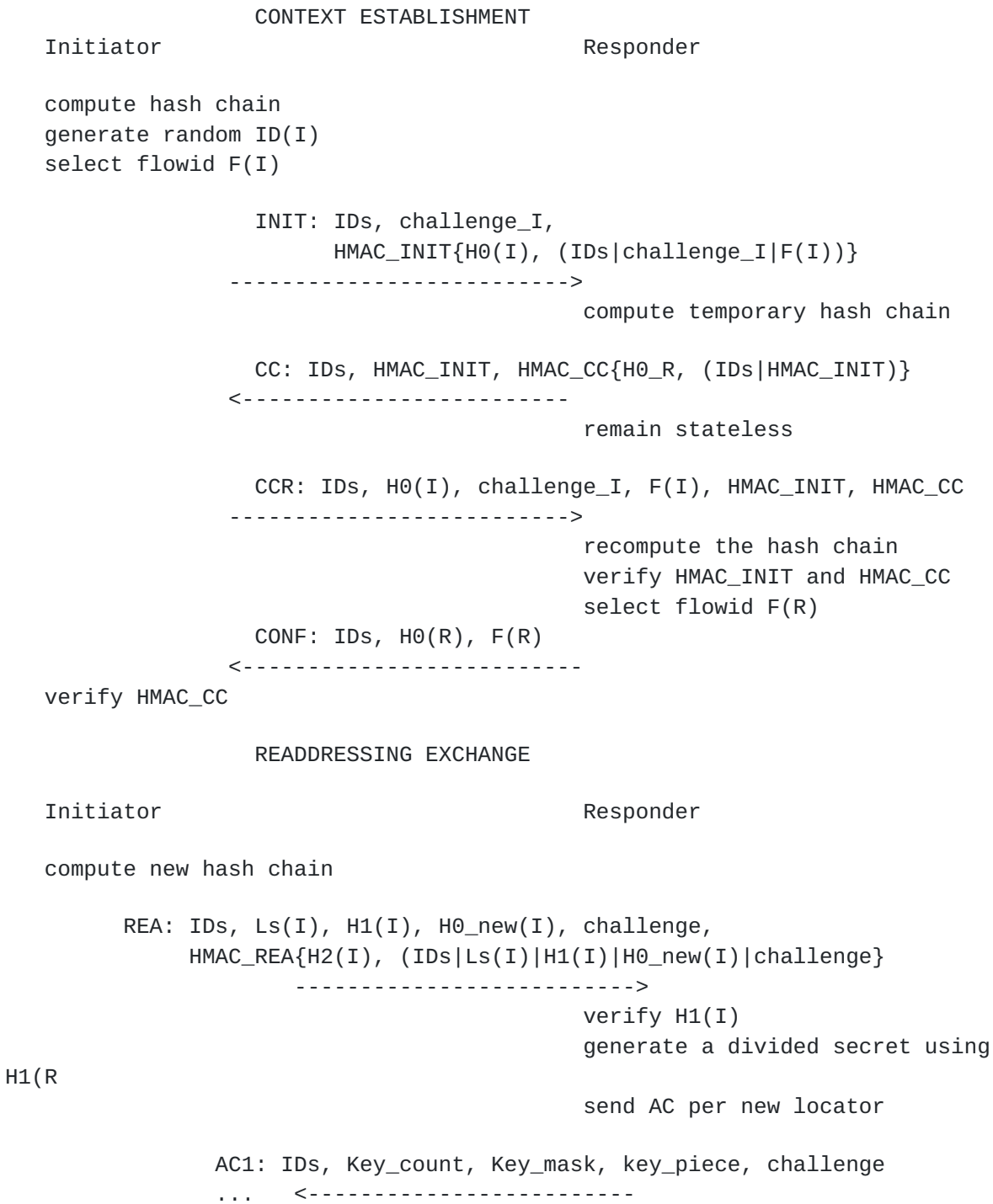
F(R) is a FlowID assigned by the responder and used by the initiator.

Hk(I) is k-th hash value in the initiator's reverse hash chain. The H0(I) is the first revealed value, i.e. the "anchor" of the

reverse hash chain. Note that the parameter k defines the revealing order, not the computation order.

Hk(R) is k-th hash value in a responder's reverse hash chain.

5.2 WIMP Protocol Overview



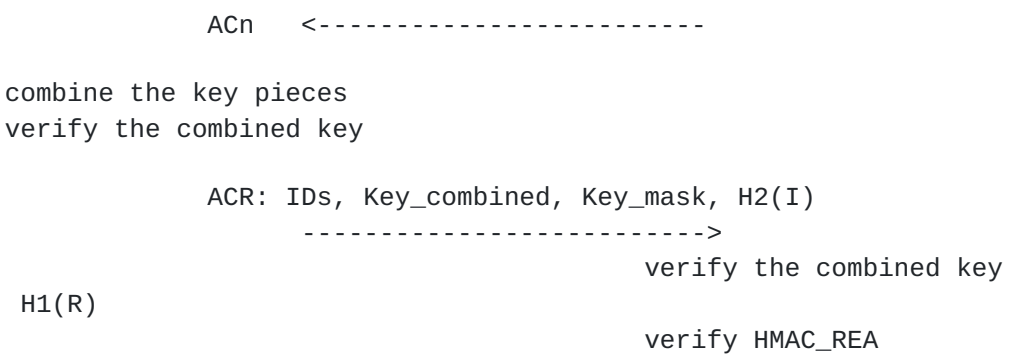


Figure 7: WIMP protocol(context establishment and locator change verification)

5.3 WIMP Security Considerations

In order to prevent redirection attacks WIMP relies on the ability to verify that the entity requesting redirection indeed holds the successor values of a hash chain and is able to combine a divided secret that is sent via parallel paths.

WIMP uses reverse hash chains and context establishment is based on opportunistic authentication. In other words, both the initiator and the responder have to trust each other that the first message comes from an authentic peer. Once the initial messages have been sent out safely, the following messages are secure. It would be impossible for an attacker who had not eaves dropped the initial messages spoof the following messages and data exchange.

1. PREMEDIATED REDIRECTION

   Premediated redirection [6] is possible in WIMP because there is no authentication of the initial messages. The receiving host is forced to trust its peer as whom it claims to be. After the first message, all the other messages are authenticated. A malicious host X could easily spoof as A to initiate a communication with B. Since B needs to trust the first message from the peer, B is fooled to trust X as A. Then X can happily communicate with B pretending to be A. As an alternative to avoid this, IPsec could be used for the initial messages. However, this could have yet another set of protocol overload problems like key exchanges and security association establishments.

2. REDIRECTING THE PACKETS

The WIMP protocol is sufficiently secure against redirection attacks because it verifies that the entity requesting redirection indeed holds the successor values of a hash chain. Assuming that A and B are already communicating and X spoofs as A to pose a redirection attack to B, X sends a readdressing or REA message to B. To avoid a possible redirection attack, B must verify that the initiator indeed is who he claims to be (by verifying if the hash value sent in the REA message is a successor of the previous value) and is reachable at the claimed locations (by sending a challenge message to all the locators). In order to authenticate himself X must be able to guess the successor hash value and to locate at different topological locations, at the same time, to be able to answer to the challenges. Since both of these are far from possible, X would fail to make a successful redirection attack.

3. RESOURCE EXHAUSTION DENIAL OF SERVICE ATTACKS

WIMP uses hash chains as compared to the PK signatures for host authentication and locator change verifications. The trade-off between hash chain based message authentication and signatures is that hash chains are vulnerable for a class of man-in-the-middle attacks ONLY in the initial couple of messages. However, if we accept that the first round-trip of the context establishment exchange is open for such attacks hash chains have other advantages. The hash chain computation is a lightweight operation compared to signature verification and consumes fewer resources at the time of resource exhaustion attacks [5]. It could also be possible that signature verifications be used only for the vulnerable initial messages.

6. Location Independent Network

LIN6 [4] is a protocol supporting multihoming and mobility in IPv6. LIN6 introduces the node id, not the interface id, for each node. Each node can be identified by its node id no matter where the node is connected and no matter how many interfaces the node has. In the IPv6 layer, 64-bit node id called LIN6 ID is used while 128-bit node-id called LIN6 generalized ID is used above the Transport layer. TCP connections and security associations can be preserved even if the node moves to another subnet or the node changes the using interface in a multihoming environment without modifying TCP or IPsec

LIN6 attempts to make transport connections at the TCP layer using some form of a standard ID and then passed on the network layer. When this reaches the network layer, the IPv6 layer overwrites the standard ID with the network ID and then passes it on further.

6.1 Notational Conventions

A is the initiating host and B is the responding host. X is a potentially malicious host.

MA-A is the Mobile Agent of A and MA-B is the Mobile Agent of B.

NS is the name server with AAAA records(network prefix of MA and LIN6 ID of a node).

The LIN6 ID is the ID for a node in the network. This is the EUI 64 standard identifier. The EUI 64 has 64 bits, the upper 24 bits is organizations unique ID (OUI) and is allocated by the IEEE to the organization. The rest of the 40 bits is assigned randomly. Here the organizations ID is the ID of the authors University and is 0x00-0c-21.

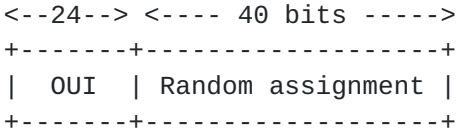


Figure 8

The LIN6 prefix is a predefined constant value that is attached to the head of the LIN6 ID to form the LIN6 generalized ID.

The LIN6 generalized ID is used to identify the hosts ONLY in the transport layer. This ID is NOT used in the IP layer. These LING6 generated ID are used to make SA among the hosts and for IPsec. Therefore from the ULP and the transport layer point of view, all data exchange are identified using the LIN6 generated IDs.

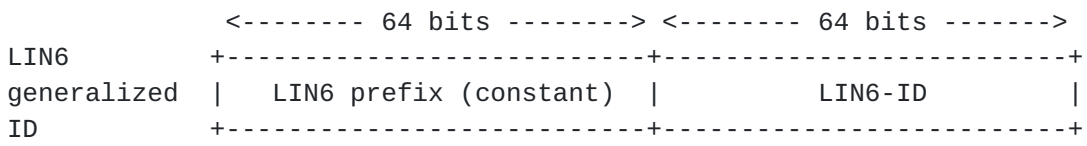


Figure 9

The LIN6 address is used in the network layer and contains the network prefix. The network identifies the network (locator). The LIN6 address is formed by combining the network prefix and the LIN6 ID.

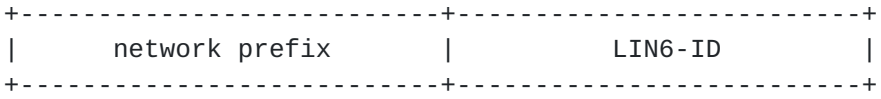


Figure 10

6.2 LIN6 Protocol Overview

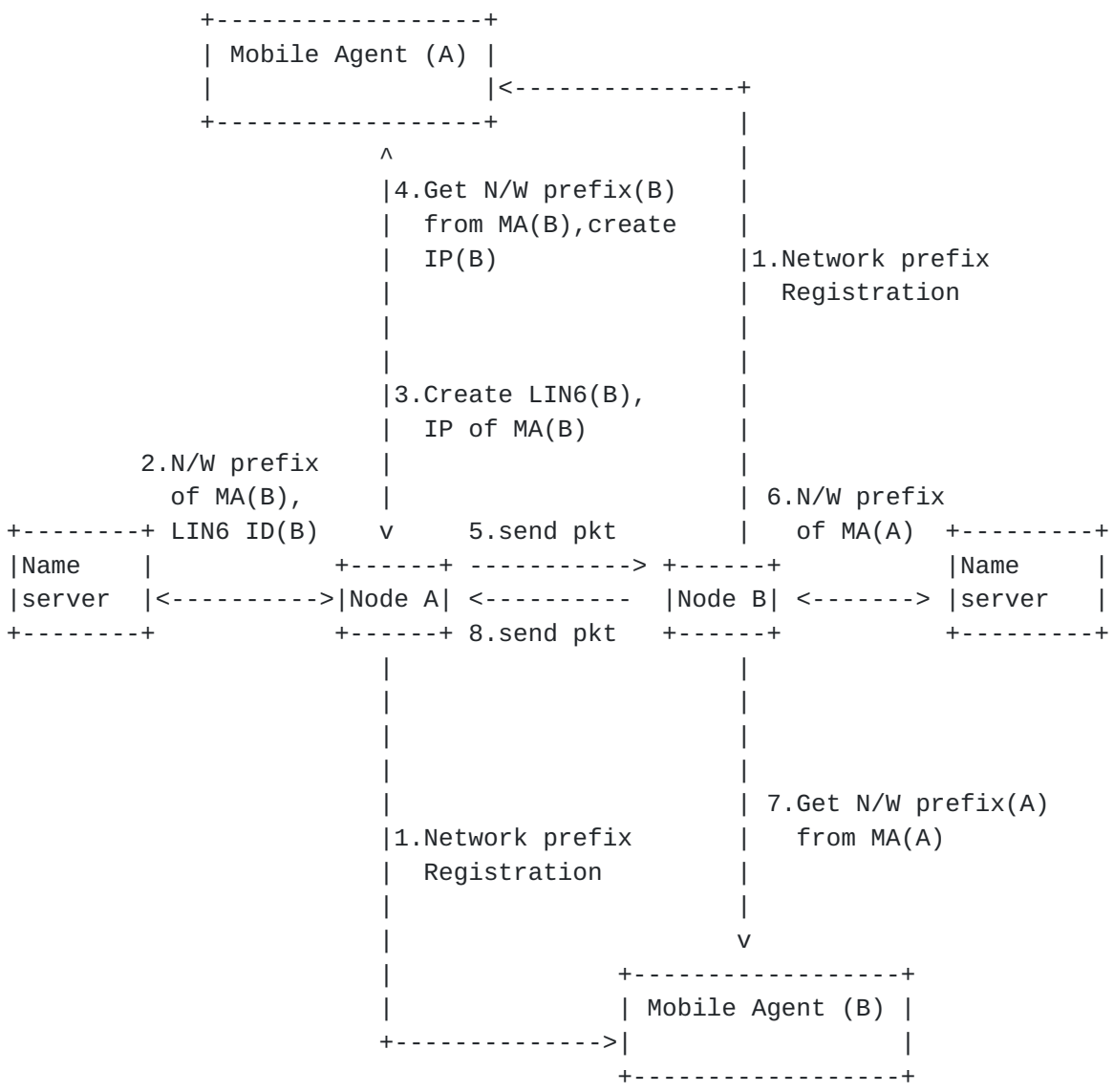


Figure 11

6.3 LIN6 Security Considerations

The security issues of the LIN6 proposal are very similar to that of the Mobile IPv6. This is because LIN6 also makes use of mobile agents and requires binding updates. Assuming that there exists proper authorization of the Binding Updates used by mobile agents (which is still an open issue in the LIN6 proposal), we look at the other security issues.

1. PREMEDIATED REDIRECTION

When node A wishes to communicate with node B, it sends a packet to B (after looking up the AAAA records and the MA-B for B's network prefix). For B to send a packet back to A, I has to obtain the network prefix of host A from MA-A. Host B first obtains the FQDN of host A from the name server by indicating the LIN6 ID of Node-A, and then obtains the MA-A:s network prefix from the name server by indicating the FQDN of Node-A. Finally, it obtains the network prefix of A from the MA-A. Assuming that the DNS records are authentic, it would not be possible of a malicious host X to spoof as A and initiate a communication with B. AAAA records will indeed show up that X is not A.

2. REDIRECTING THE PACKETS

LIN6 proposes three different solutions to support multihoming. Two of these solutions make use of authenticated header mechanism for registering the new IP with the mobile agent. The third proposal makes use of cookies in addition to the authentication headers. All of the proposals make sure that redirection of packets to a third party locator is not possible as long as the mobile agents can be trusted.
Assuming that the binding updates are secure and redirection is not possible at the time of locator updates, there are still possibilities for redirection at the time of DNS look ups. The interface between the correspondent node and the DNS server is very important because security of LIN6 is dependent on the DNS request and response messages. The initial DNS request by a correspondent node returns the address of the Mapping Agent of the initiator. If this message is modified, the correspondent node can be forced to learn the wrong address of the Mapping Agent. A malicious node could easily take advantage of this to perform a redirection a redirection attack. It is also crucial in LIN6 that the DNS holds only authentic AAAA records and are in the secure zone.

3. PACKET INJECTION

Packet injection is possible at the time of data exchange and location updates. The binding update security is still TBD in the LIN6 proposal. Assuming that binding updates always take place using security associations (in the case of the first proposal for mobility in LIN6), packet injection would be difficult.
LIN6 also supports IPsec for data exchange. If IPsec is used in connection establishment and payload protection, then it

would not be possible to inject packets or disrupt the flow of the data stream. However, this could have extra overheads of the IPsec like key generation and exchange.
It should also be noted that packet injection is possible at the time of DNS request and reply. Since this problem is generic to all redirection attacks, it is not a special concern for packet injection problem.

4. RESOURCE EXHAUSTION DENIAL OF SERVICE ATTACKS

Any protocol that uses signature verifications could be subjected to resource exhaustion attacks as these verifications are often expensive for a host. When one of the nodes in a LIN6 protocol is mobile, it has to inform the mobile agent (MA) and/or the corresponding node (CN) about its new location. The corresponding node can start using the new locator ONLY after the new network prefix has been verified for impersonation. Attackers can take advantage of such a situation to send forged location updates to the corresponding node. Its should be noted that LIN6 also makes proposals that are based on SA between the Mobile Node and the Correspondent Node for authenticating binding updates.

7. Host Identity Protocol

The HIP proposal [9]is an ID/Locator separation mechanism intended to solve a much wider problem space than site multi-homing. HIP uses cryptographic identifiers termed Host Identity Tags (HITs) at the application layer, which are mapped to locators (IP Addresses) by a HIP protocol stack layer that interfaces between the transport layer and network layer. The essential characteristic of HIP is it use of opportunistic identity generation, as it uses a cryptographic host identifier as the basis of the persistent identity. The transport session can be agile across locators, or even across IP protocol versions, as the HIT is used to determine session integrity allowing the hosts to determine what packets legitimately form part of the session.

HIP is proposed as a new protocol element, located at layer 3.5 (i.e. above the internetwork IP layer and below the transport layer). The presentation to the transport layer uses 128 bit hash values (the HIT) in place of IP addresses, while the presentation to the internet layer uses conventional IP addresses.

7.1 Notational Conventions

I is the initiating host and R is the responding host. X is a potentially malicious host.

FQDN (R) is a fully qualified domain name of R and this is used to uniquely identify R.

HI (R) is the Host Identifier of R. A host identifier is cryptographic in nature and is the public key of an asymmetric key-pair. Correspondingly, the host itself is the entity that holds the private key of the key pair.

The Host Identity Tag or HIT is a 128 bit hash value of the Host Identifier. Its fixed length makes for easier protocol coding and presents and consistent format.

D-H key is the Diffie-Hellmann key used by a host to create a session key (SK). The resultant session key is used to generate the keying material or the KEYMAT to derive all the other keys for integrity checks and encryption.

7.2 HIP Protocol Overview

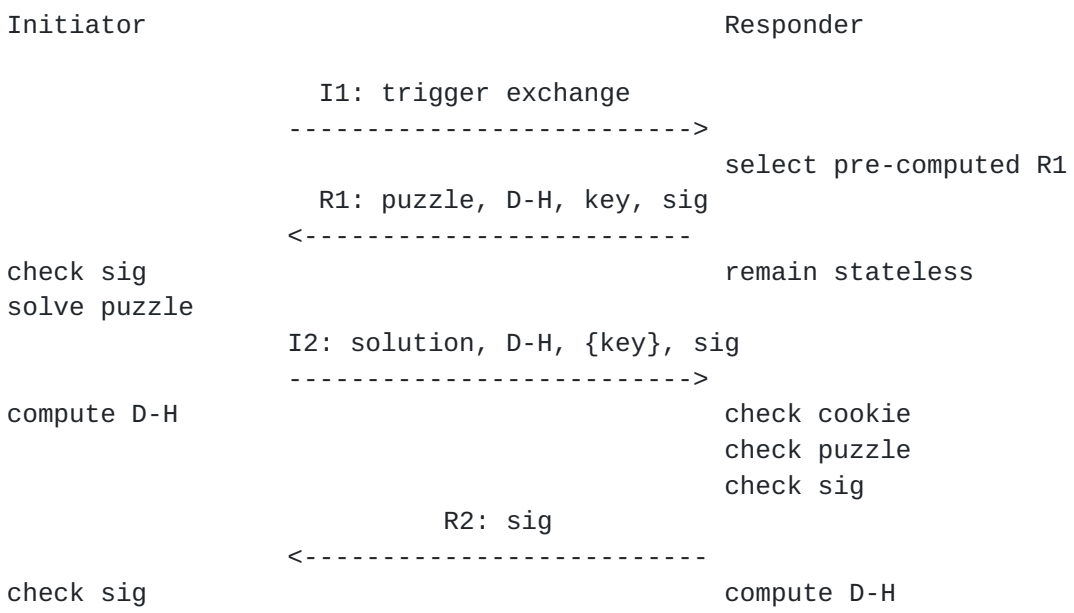


Figure 12: HIP protocol(context establishment)

7.3 HIP Security Considerations

HIP is designed to provide secure authentication of hosts and to provide a fast key exchange for IPsec ESP. HIP also attempts to limit the exposure of the host to various denial-of-service and man-in-the-middle attacks. This is achieved through security associations and PK signature verifications.

In order to prevent redirection attacks during locator updates, the CB64 protocol relies on the ability to verify that the entity requesting redirection indeed holds the private key where the hash of the corresponding public key hashes to the ID itself.

1. PREMEDIATED REDIRECTION

Premediated redirection is not possible in HIP because, HIP tries to authenticate the initiator using PK signatures at the time of base exchange itself. When an attacker X pretending to be an initiator I, sends a spoofed trigger exchange message to the host R, R creates a puzzle and generates a Diffie-Hellmann key for the session and sends it back to the attacker. The attacker needs to solve the puzzle, encrypt his host identity (which is the public authentication key of A) and then sign it (with the private key of X). The responder decrypts the encrypted host identity and uses it to verify the

signature. The private key of the attacker and the public key of A would not make a key pair for signature verification. Authentication fails and the receiver stops the 3-way base exchange. Since the Host Identity of a host is the public authentication key, it is important that these are retrieved from a signed DNS zone, a certificate or some secure methods.

2. REDIRECTING THE PACKETS

HIP proposes some ways to handle locator changes. One of them is to use the readdressing parameter REA and the other is to have Rendezvous Servers for efficient multihoming and mobility.
The former method tries to make a new security association with the hosts every time a mobile node changes its location on the network. Since no other host, even those in the path between the initiator and the receiver cannot know the session keys (exchanged using Diffie-Hellmann key exchange protocol) for the new security associations, it is impossible for a middle man to issue false locator updates to one of the host and redirect the packets.
The later method uses rendezvous servers which maintain a mapping between the Host Identities of HIP nodes for which they provide service and the node's current IP addresses. HIP nodes must notify their Rendezvous Servers about any changes in their IP addresses. It is important that the HIP mobile nodes be authenticated before they update their new IP in their respective Rendezvous Servers. Assuming so, it would be difficult for a malicious host to make false location updates on Rendezvous Servers that do not belong to it. Therefore false location updates are difficult, making all kinds of redirection attacks also difficult.

3. PACKET INJECTION

Since HIP makes use of IPsec ESP in order to secure all the packets, packets that are injected would either be unencrypted or would belong to irrelevant security associations. Such packets could be ignored and this would not affect the regular stream of data flow in HIP.

4. RESOURCE EXHAUSTION DENIAL OF SERVICE ATTACKS

Denial-of-service attacks take advantage of the cost of start of state for a protocol on the Responder compared to the cheapness on the Initiator. HIP makes no attempt to increase the cost of the start of state on the Initiator, but makes an effort to reduce the cost to the Responder. This is done by

having the Responder start the 3-way cookie exchange instead of the initiator by sending a puzzle.
This shifting of the start of state cost to the Initiator in creating the I2 HIP packet, presents yet another DoS attack. The attacker spoofs the I1 HIP packet and sends out the R1 HIP packet. This could tie up the initiator with evaluating the R1 HIP packet and creating the I2 HIP packet. The defense against this attack is to simply ignore any R1 packet where a corresponding I1 or ESP data was not sent.
Message R2 also includes signature verification. However, the responder verifies the signature ONLY if it receives the correct solution for the puzzle it sent out. For an attacker to force the receiver with signature verifications with I2 message, it needs to find the correct solution of the puzzle that was sent out. This would be the price that he would pay for one signature verification attack on the receiver.
Denial of service attack with R2 message is avoided by including a HMAC value in the R2 message. The initiator would verify the signature ONLY if the HMAC value of the message is first verified. It would be impossible for an attacker to compute the correct HMAC as this requires the integrity key that the communicating hosts generated with the session key.
For rendezvous mechanisms for mobility management, the security considerations are yet to be determined. It is assumed that all requests and replies to and from the DNS are secure and that DNS holds only authentic records in a secure zone.

8. Other Security Considerations

Replay attacks - The SIM, NOID and the CB64 proposals make use of nonce and timestamp in the context establishment messages and readdressing messages. This makes replay attacks difficult. This is especially significant for SIM and CB64 proposals that involve PK signature verifications for authentication. In the other proposals like Lin6 and WIMP, replay attacks do not gain anything significant for the attacker. In the HIP proposal, however, replay attacks are avoided by using the cookie mechanism to generate the puzzle. This mechanism makes use of a nonce to calculate the index for the puzzle.

Protecting the context establishment itself - One big draw back of all the proposals except the HIP is that they do not make any attempt to protect the initial context establishment mechanism. For instance, the WIMP proposal assumes that the initial two messages need to be exchanged between authentic nodes. If the initial messages itself are protected, nodes in the path will not be able to create the same hash chain and use them for man in the middle attacks. In the other proposals like SIM, NOID and CB64 protecting the context establishment could protect the Context Tag or Flow ID as they are important for rewriting the locator values back to the identifier values. Otherwise, the CTs and FlowIDs are open to attacks.

9. Comparison of Multi6 Proposals

The following table shows various attacks and relates them to the individual proposals. The symbol 'x' indicates that the attack is applicable to the respective protocol proposal.

	SIM	NOID	CB64	WIMP	LIN6	HIP
Premediated redirection attacks	x		x	x		
Redirecting packets to the attacker					x	
Redirecting packets to a third party					x	
Redirecting packets to a black hole					x	
Packet injection	x	x	x	x		
Resource exhaustion DoS attacks	x	x	x		x	

Figure 13

10. Security Considerations

This document discusses security issues of Multi6 locator/identifier split protocol proposals. As such, it contains a number of security issues.

11. References

11.1 Normative References

11.2 Informative References

[1] Nordmark, E., "Strong Identity Multihoming using 128 bit Identifiers (SIM/CBID128) for use in RFCs to Indicate Requirement Levels", draft-nordmark-multi6-sim-01.txt (work in progress), October 2003.

[2] Nordmark, E., "Multihoming without IP Identifiers", draft-nordmark-multi6-noid-01.txt (work in progress), October 2003.

[3] Nordmark, E., "Multihoming using 64-bit Crypto-based IDs", draft-nordmark-multi6-cb64-00.txt (work in progress), October 2003.

[4] Teraoka, F., Ishiyama, M. and M. Kunishi, "LIN6: A Solution to Multihoming and Mobility in IPv6", draft-teraoka-multi6-lin6-00.txt (work in progress), December 2003.

[5] Ylitalo, J., Torvinen, V. and E. Nordmark, "Weak Identifier Multihoming Protocol (WIMP)", draft-ylitalo-multi6-wimp-00 (work in progress), January 2004.

[6] Nordmark, E. and T. Li, "Threats relating to IPv6 multihoming solutions", draft-nordmark-multi6-threats-01.txt (work in progress), June 2004.

[7] Huston, G., "Architectural Approaches to Multi-Homing for IPv6", draft-huston-multi6-architectures-00.txt (work in progress), May 2004.

[8] Crocker, D., "Choices for Multiaddressing", draft-crocker-multiaddr-analysis-01.txt (work in progress), October 2003.

[9] Moskowitz, R., Nikander, P., Jokela, P. and T. Henderson, "Host Identity Protocol", draft-moskowitz-hip-09.txt (work in progress), February 2004.

[10] Moskowitz, R. and P. Nikander, "Host Identity Protocol Architecture", draft-moskowitz-hip-arch-05 (work in progress), September 2003.

[11] Eggert, L., "Host Identity Protocol (HIP) Rendezvous Mechanisms", draft-eggert-hip-rendezvous-00.txt (work in progress), February 2004.

[12] Nikander, P. and J. Arkko, "End-Host Mobility and Multi-Homing with Host Identity Protocol", draft-nikander-hip-mm-01.txt (work in progress), December 2003.

Authors' Addresses

Aarthi Nagarajan
Siemens
Otto-Hahn-Ring 6
Munich, Bayern 81739
Germany

EMail: aarthi.nagarajan@tuhh.de

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bayern 81739
Germany

EMail: Hannes.Tschofenig@siemens.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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