MOBIKE Working Group P. Eronen, Ed.
Internet-Draft Nokia
Expires: January 16, 2006 July 15, 2005

# IKEv2 Mobility and Multihoming Protocol (MOBIKE)
**draft-ietf-mobike-protocol-01.txt**



Abstract

This document describes the MOBIKE protocol, a mobility and multihoming extension to IKEv2. MOBIKE allows mobile and/or multihomed hosts to update the (outer) IP addresses associated with IKE and IPsec Security Associations (SAs). The main scenario for MOBIKE is making it possible for a remote access VPN user to move from one address to another while keeping the connection with the VPN gateway active.

Table of Contents

# 1. Introduction

## 1.1 Motivation

IKEv2 is used for performing mutual authentication and establishing and maintaining IPsec security associations (SAs). In the current specifications, the IPsec and IKE SAs are created implicitly between the IP addresses that are used when the IKE_SA is established. These IP addresses are then used as the outer (tunnel header) addresses for tunnel mode IPsec packets. Currently, it is not possible to change these addresses after the IKE_SA has been created.

There are scenarios where these IP addresses might change. One example is mobility: a host changes its point of network attachment, and receives a new IP address. Another example is a multihoming host that would like to change to a different interface if, for instance, the currently used address stops working for some reason.

Although the problem can be solved by creating new IKE and IPsec SAs when the addresses need to be changed, this may not be optimal for several reasons. In some cases, creating a new IKE_SA may require user interaction for authentication (entering a code from a token card, for instance). Creating new SAs often also involves expensive calculations and possibly a large number of roundtrips. Due to these reasons, a mechanism for updating the IP addresses of existing IKE and IPsec SAs is needed. The MOBIKE protocol described in this document provides such a mechanism.

The main scenario for MOBIKE is making it possible for a remote access VPN user to move from one address to another without re-establishing all security associations with the VPN gateway. For instance, a user could start from fixed Ethernet in the office, and then disconnect the laptop and move to office wireless LAN. When leaving the office the laptop could start using GPRS, and switch to a different wireless LAN when the user arrives home. MOBIKE updates only the outer (tunnel header) addresses of IPsec SAs, and the addresses and others traffic selectors used inside the tunnel stay unchanged. Thus, mobility can be (mostly) invisible to applications and their connections using the VPN.

MOBIKE also supports more complex scenarios where the VPN gateway also has several network interfaces: these interfaces could be connected to different networks or ISPs, they may have may be a mix of IPv4 and IPv6 addresses, and the addresses may change over time. Furthermore, both parties could be VPN gateways relaying traffic for other parties.

### 1.2 MOBIKE protocol overview

Since MOBIKE allows both parties to have several addresses, this leads us to an important question: there are up to N*M pairs of IP addresses that could potentially be used. How to decide which of these pairs should be used? The decision has to take into account several factors. First, the parties have may preferences about which interface should be used, due to performance and cost reasons, for instance. Second, the decision is constrained by the fact that some of the pairs may not work at all due to incompatible IP versions, outages somewhere in the network, problems at the local link at either end, and so on.

MOBIKE solves this problem by taking a simple approach: the party that initiated the IKE_SA (the "client" in remote access VPN scenario) is responsible for deciding which address pair is used for the IPsec SAs, and collecting the information it needs to make this decision (such as determining which address pairs work or do not work). The other party (the "gateway" in remote access VPN scenario) simply tells the initiator what addresses it has, but does not update the IPsec SAs until it receives a message from the initiator to do so.

Making the decision at the initiator is consistent with how normal IKEv2 works: the initiator decides which addresses it uses when contacting the responder. It also makes sense especially when the initiator is the mobile node: it is in a better position to decide which of its network interfaces should be used for both upstream and downstream traffic.

The details of exactly how the initiator makes the decision, what information is used in making it, how the information is collected, how preferences affect the decision, and when a decision needs to be changed, are largely beyond the scope of MOBIKE. This does not mean that these details are unimportant: on the contrary, they are likely to be crucial in any real system. However, MOBIKE is concerned with these details only to the extent that they are visible in IKEv2/IPsec messages exchanged between the peers (and thus need to be standardized to ensure interoperability). Issues such as mobility detection and local policies are also not specific to MOBIKE, but apply to existing mobility protocols such as Mobile IPv4 [MIP4] as well.

One important aspect of this information gathering that has to be visible in the messages is determining whether a certain pair of addresses can be used. IKEv2 Dead Peer Detection (DPD) feature can provide information that the currently used pair does or does not work. There are, however, some complications in using it for other

addresses, and thus MOBIKE adds a new IKEv2 message that can be used to "test" whether some particular pair of addresses works or not, without yet committing to changing the addresses currently in use.

MOBIKE also has to deal with situations where the network contains NATs or stateful packet filters (for brevity, the rest of this document talks simply about NATs). When the addresses used for IPsec SAs are changed, MOBIKE can enable or disable IKEv2 NAT Traversal as needed. However, if the party "outside" the NAT changes its IP address, it may no longer be able to send packets to the party "behind" the NAT, since the packets may not (depending on the exact type of NAT) match the NAT mapping state. Here MOBIKE assumes that the initiator is the party "behind" the NAT, and does not fully support the case where the responder's addresses change when NATs are present.

Updating the addresses of IPsec SAs naturally has to take into account several security considerations. MOBIKE includes two features designed to address these considerations. First, a "return routability" check can be used to verify the addresses provided by the peer. This makes it more difficult to flood third parties with large amounts of traffic. Second, a "NAT prevention" feature ensures that IP addresses have not been modified by NATs, IPv4/IPv6 translation agents, or other similar devices. This feature is mainly intended for site-to-site VPNs where the administrators may know beforehand that NATs are not present, and thus any modification to the packet can be considered to be an attack.

### 1.3 Terminology and notations

When messages containing IKEv2 payloads are shown, optional payloads are shown in brackets (for instance, "[FOO]"), and a plus sign indicates that a payload can be repeated one or more times (for instance, "FOO+").

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 [KEYWORDS].

## 2. MOBIKE protocol exchanges

### 2.1 Signaling support for MOBIKE

Implementations that wish to use MOBIKE for a particular IKE_SA MUST include a MOBIKE_SUPPORTED notification in the IKE_SA_INIT request and response messages.

```
   Initiator                     Responder
  -----------                   -----------
   HDR, SAi1, KEi, Ni,
        N(MOBIKE_SUPPORTED),
        [N(NAT_DETECTION_SOURCE_IP)+,
         N(NAT_DETECTION_DESTINATION_IP)]  -->

                            <--  HDR, SAr1, KEr, Nr,
                                      [N(NAT_DETECTION_SOURCE_IP)+,
                                       N(NAT_DETECTION_DESTINATION_IP)]
                                      [CERTREQ],
                                      N(MOBIKE_SUPPORTED)
```

The MOBIKE_SUPPORTED notification payload is described in Section 3.

### 2.2 Additional addresses

Both the initiator and responder MAY include one or more ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS notification payloads in the IKE_AUTH exchange (in case of multiple IKE_AUTH exchanges, in the message containing the SA payload).

```
   Initiator                     Responder
  -----------                   -----------
   HDR, SK { IDi, [CERT], [IDr], AUTH,
             [CP(CFG_REQUEST)]
             SAi2, TSi, TSr,
             [N(ADDITIONAL_*_ADDRESS)+]  -->

                            <--  HDR, SK { IDr, [CERT], AUTH,
                                           [CP(CFG_REPLY)],
                                           SAr2, TSi, TSr,
                                           [N(ADDITIONAL_*_ADDRESS)+] }
```

The recipient stores this information, but no other action is taken at this time.

### 2.3 Changing addresses in IPsec SAs

In MOBIKE, the initiator of the IKE_SA decides what addresses are used in the IPsec SAs. That is, the responder never updates any IPsec SAs without receiving an explicit UPDATE_SA_ADDRESSES request from the initiator. (As described below, the responder can, however, update the IKE_SA in some circumstances.)

The description in this section assumes that the initiator has already decided what the new addresses should be. How this decision is made is beyond the scope of this specification. When this decision has been made, the initiator

- o Updates the IKE_SA and IPsec SAs with the new addresses, and sets the "pending_update" flag in the IKE_SA.

- o If NAT Traversal is not enabled, and the responder supports NAT Traversal (as indicated by NAT detection payloads in the IKE_SA_INIT exchange), and the initiator either suspects or knows that a NAT is likely to be present, enables NAT Traversal.

- o If there are outstanding IKEv2 requests, continues retransmitting them using the addresses in the IKE_SA (the new addresses).

- o When the window size allows, sends an INFORMATIONAL request containing the UPDATE_SA_ADDRESSES notification payload (which does not contain any data), and clears the "pending_update" flag. (See Section 2.6 for description of the COOKIE2 notification.)

```
   Initiator                   Responder
  -----------                 -----------
   HDR, SK { N(UPDATE_SA_ADDRESSES),
             N(COOKIE2),
             [N(NAT_DETECTION_*_IP)],
             [N(NAT_PREVENTION)] } -->
```

- o If a new address change occurs while waiting for the response, starts again from the first step (and ignores responses to this UPDATE_SA_ADDRESSES request).

Note that if the responder has NAT Traversal enabled, it can update the addresses in both the IKE_SA and IPsec SAs as usual (if it implements the "SHOULD" from [IKEv2] Section 2.23).

When processing an INFORMATIONAL request containing the UPDATE_SA_ADDRESSES notification, the responder

o Determines whether it has already received a newer UPDATE_SA_ADDRESSES request than this one (if the responder uses a window size greater than one, it is possible that requests are received out of order). If it has, a response message is sent, but no other action is taken.

o If the NAT_PREVENTION payload is present, processes it as described in Section 2.7.

o Checks that the (source IP address, destination IP address) pair in the IP header is acceptable according to local policy. If it is not, replies with "HDR, SK {N(COOKIE2), N(UNACCEPTABLE_ADDRESSES)}".

o Updates the IP addresses in the IKE_SA with the values from the IP header. (Using the address from the IP header is consistent with normal IKEv2, and allows IKEv2 to work with NATs without needing unilateral self-address fixing [UNSAF].)

o Replies with an INFORMATIONAL response:

```
   Initiator                   Responder
  -----------                 -----------
                          <--  HDR, SK { N(COOKIE2),
                                         [N(NAT_DETECTION_*_IP)] }
```

o If necessary, initiates a return routability check for the new initiator address (see Section 2.6) and waits for the check to finish..

o Updates the IPsec SAs with the new addresses.

o If NAT Traversal is supported and NAT detection payloads were included, enables or disables NAT Traversal.

When the initiator receives the reply, it

o If the response contains the NAT_PREVENTED payload, processes it as described in Section 2.7.

o If the response contains an UNACCEPTABLE_ADDRESSES notification payload, the initiator MAY select another addresses and retry the exchange, keep on using the current addresses, or disconnect.

o If NAT Traversal is supported and NAT detection payloads were included, enables or disables NAT Traversal.

### 2.4 Updating additional addresses

As described in Section 2.2, both the initiator and responder can send a list of additional addresses (in addition to the one used for IKE_SA_INIT/IKE_AUTH exchange) to the initiator in the IKE_AUTH exchange. If this list of addresses changes, a new list can be sent in any INFORMATIONAL exchange request message.

When the responder (of the original IKE_SA) receives an INFORMATIONAL request containing ADDITIONAL_*_ADDRESS payloads, it simply stores the information, but no other action is taken.

```
   Initiator                    Responder
  -----------                  -----------
   HDR, SK { N(ADDITIONAL_*_ADDRESS)+,
             N(COOKIE2) }  -->

                           <--  HDR, SK { N(COOKIE2) }
```

When the initiator receives an INFORMATIONAL request containing ADDITIONAL_*_ADDRESS, it stores the information and also determines whether the currently used addresses need to be changed (for instance, if the currently used address is no longer included in the list); if it does, the initiator proceeds as described in Section 2.3.

```
   Initiator                    Responder
  -----------                  -----------
                           <--  HDR, SK { N(ADDITIONAL_*_ADDRESS)+,
                                          N(COOKIE2) }

   HDR, SK { N(COOKIE2) }  -->
```

If the implementation supports window sizes greater than one, it also has to keep track of the Message ID of the latest update it has received, to avoid the situation where new information is overwritten by older.

There is one additional complication: when the responder wants to send a new additional address list, the currently used addresses may no longer work. In this case, the responder uses the additional address list received from the initiator, the list of its own addresses, and, if necessary, the path testing feature (see Section 2.5) to determine a path that works, updates the addresses in the IKE_SA (but not IPsec SAs), and then sends the INFORMATIONAL request. This is the only time the responder uses the additional address list received from the initiator.

Note that both peers can have their own policies about what addresses are acceptable to use. A minimal "mobile client" could have a policy that says that only the responder's address specified in local configuration is acceptable. This kind of client does not have to send or process ADDITIONAL_*_ADDRESS notification payloads. Similarly, a simple "VPN gateway" that has only a single address, and is not going to change it, does not need to send or understand ADDITIONAL_*_ADDRESS notification payloads.

## 2.5 Path testing

IKEv2 Dead Peer Detection allows the peers to detect if the currently used path has stopped working. However, if either of the peers has several addresses, DPD alone does not indicate which of the other paths might work. The path testing feature allows the parties to determine whether a particular path (pair of addresses) works, without yet committing to changing over to these addresses.

MOBIKE introduces a new IKEv2 exchange type, PATH_TEST, for testing connectivity. This exchange is not part of any IKE_SA, so it is not cryptographically protected. It also does not result in the responder keeping any state.

```
   Initiator                   Responder
  -----------                 -----------
   HDR(0,0), N(COOKIE2),
             [N(NAT_DETECTION_*_IP)]  -->

                          <--  HDR(0,0), N(COOKIE2),
                                         [N(NAT_DETECTION_*_IP)]
```

The reason for introducing a new exchange type, instead of using INFORMATIONAL exchanges, is to simplify implementations by allowing MOBIKE to work with window size 1.

Performing path testing over several different paths is not required if the node has other information that enables it to select which path should be used. Also, responders do not perform path testing unless they update their list of additional addresses (as described in Section 2.4). Implementations MAY do path testing even if the currently used path is working to e.g. detect when a better but previously unavailable path becomes available, or to speed up recovery in fault situations.

Implementations that perform path testing MUST take steps to avoid causing unnecessary congestion. TBD: add some more details here.

### 2.6 Return routability check

Both the initiator and the responder can optionally verify that the other party can actually receive packets at the claimed address. This "return routability check" MAY be done before updating the IPsec SAs, immediately after updating them, or continuously during the connection.

By default, return routability check SHOULD be done before updating the IPsec SAs. In environments where the peer is expected to be well-behaving (many corporate VPNs, for instance), or the address can be verified by some other means (e.g., the address is included in the peer's certificate), the return routability check MAY be skipped or postponed until after the IPsec SAs have been updated.

Any INFORMATIONAL exchange can be used for return routability purposes (with one exception, described below): when a valid response is received, we know the other party can receive packets at the claimed address.

To ensure that the peer cannot generate the correct INFORMATIONAL response without seeing the request, a new payload is added to all INFORMATIONAL messages. The sender of an INFORMATIONAL request MUST include a COOKIE2 notification payload, and the recipient of an INFORMATIONAL request MUST copy the payload as-is to the response. When processing the response, the original sender MUST verify that the value is the same one as sent. If the values do not match, the IKE_SA MUST be closed.

There is one additional issue that must be taken into account. If the INFORMATIONAL request has been sent to several different addresses (i.e., the destination address in the IKE_SA has been updated after the request was first sent), receiving the INFORMATIONAL response does not tell which address is the working one. In this case, a new INFORMATIONAL request needs to be sent to check return routability.

### 2.7 NAT prevention

IKEv2/IPsec implementations that do not support NAT Traversal can, in fact, work across some types of one-to-one "basic" NATs and IPv4/IPv6 translation agents in tunnel mode. This may be considered a problem in some circumstances, since in some sense any modification of the IP addresses can be considered to be an attack.

The "NAT prevention" feature allows both the initiator and responder to have a policy that prevents the use of paths that contain NATs, IPv4/IPv6 translation agents, or other nodes that modify the

addresses in the IP header. This feature is mainly intended for site-to-site VPN cases, where the administrators may know beforehand that NATs are not present, and thus any modification to the packet can be considered to be an attack.

This specification addresses the issue as follows. When an IPsec SA is created, the tunnel header IP addresses (and port if doing UDP encapsulation) are taken from the IKE_SA, not the message IP header. The NAT_PREVENTION payload is used to guarantee that NATs have not modified the address used in IKE_SA. However, all response messages are still sent to the address and port the corresponding request came from.

The initiator MAY include a NAT_PREVENTION payload in an IKE_SA_INIT request. The responder then MUST calculate the expected value based on the values from the IP header, and compare this with the value in the NAT_PREVENTION payload. If they do not match, the responder replies with "HDR(A,0), N(NAT_PREVENTED)" and does not create any state.

If the values do match, the responder initializes (local_address, local_port, peer_address, peer_port) in the to-be-created IKE_SA with values from the IP header. The same applies if neither NAT_PREVENTION nor NAT_DETECTION_*_IP payloads were included, or if the responder does not support NAT Traversal.

If the IKE_SA_INIT request included NAT_DETECTION_*_IP payloads but no NAT_PREVENTION payload, the situation is different since the initiator may at this point change from port 500 to 4500. In this case, the responder initializes (local_address, local_port, peer_address, peer_port) from the first IKE_AUTH request.

IKEv2 requires that if an IPsec endpoint discovers a NAT between it and its correspondent, it MUST send all subsequent traffic to and from port 4500. To simplify things, implementations that support both this specification and NAT Traversal MUST change to port 4500 if the correspondent also supports both, even if no NAT was detected between them.

NAT_PREVENTION payloads can also be included when changing the addresses of IPsec SAs (see Section 2.3). TBD: add better description.

# 3. Payload formats

## 3.1 MOBIKE_SUPPORTED notification payload

The MOBIKE_SUPPORTED notification payload is included in the IKE_SA_INIT messages to indicate that the implementation supports this specification.

The Notify Message Type for MOBIKE_SUPPORTED is TBD-BY-IANA (16396..40959). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

## 3.2 ADDITIONAL_IP4/6_ADDRESS notification payloads

Both initiator and responder can include ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS payloads in the IKE_AUTH exchange and INFORMATIONAL exchange request messages; see Section 2.2 and Section 2.4 for more detailed description.

The Notify Message Types for ADDITIONAL_IP4_ADDRESS and ADDITIONAL_IP6_ADDRESS are TBD-BY-IANA (16396..40959) and TBD-BY-IANA (16396..40959), respectively. The Protocol ID and SPI Size fields are set to zero. The data associated with these Notify types is either a four-octet IPv4 address or a 16-octet IPv6 address.

## 3.3 UPDATE_SA_ADDRESSES notification payload

This payload is included in INFORMATIONAL exchange requests sent by the initiator of the IKE_SA to update addresses of the IKE_SA and IPsec SAs (see Section 2.3).

The Notify Message Type for UPDATE_SA_ADDRESSES is TBD-BY-IANA (16396..40959). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

## 3.4 UNACCEPTABLE_ADDRESSES notification payload

The responder can include this notification payload in an INFORMATIONAL exchange response to indicate that the address change in the corresponding request message (which contained an UPDATE_SA_ADDRESSES notification payload) was not carried out.

The Notify Message Type for UNACCEPTABLE_ADDRESSES is TBD-BY-IANA (40..8191). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

### 3.5 COOKIE2 notification payload

This payload is included in all INFORMATIONAL exchange messages for return routability check purposes (see Section 2.6). It is also used in PATH_TEST messages to match requests and responses (see Section 2.5).

The data associated with this notification MUST be between 8 and 64 octets in length (inclusive), and MUST be chosen in a way that is unpredictable to the recipient. The Notify Message Type for this message is TBD-BY-IANA (16396..40959). The Protocol ID and SPI Size fields are set to zero.

### 3.6 NAT_PREVENTION notification payload

See Section 2.7 for a description of this payload.

The data associated with this notification is the SHA-1 hash [FIPS180-2] of the following data: IKE SPIs (in the order they appear in the header), the IP address and port from which the packet was sent, and the IP address and port to which the packet was sent. The Notify Message Type for this message is TBD-BY-IANA (16396..40959). The Protocol ID and SPI Size fields are set to zero.

### 3.7 NAT_PREVENTED notification payload

See Section 2.7 for a description of this payload.

The Notify Message Type for NAT_PREVENTED is TBD-BY-IANA (40..8191). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

## 4. Security considerations

The main goals of this specification are to not reduce the security offered by usual IKEv2 procedures and to counter mobility related threats in an appropriate manner. In some specific cases MOBIKE is also capable of protecting address changes better than existing NAT Traversal procedures.

The threats arising in scenarios targeted by MOBIKE are:

Traffic redirection and hijacking

Insecure mobility management mechanisms may allow inappropriate redirection of traffic. This may allow inspection of the traffic as well as man-in-the-middle and session hijacking attacks.

The scope of these attacks in the MOBIKE case is limited, as all traffic is protected using IPsec. However, it should be observed that security associations originally created for the protection of a specific flow between specific addresses may be moved through MOBIKE. The level of required protection may be different in a new location of a VPN client, for instance.

Third-party denial-of-service through flooding

Traffic redirection may be performed not just to gain access to the traffic, but also to cause a denial-of-service attack for a third party. For instance, a high-speed TCP session or a multimedia stream may be redirected towards a victim host, causing its communications capabilities to suffer.

The attackers in this threat can be either outsiders or even one of the participants. In usual VPN usage scenarios attacks by participants can be easily dealt with. However, this requires that strong authentication was performed in the initial IKEv2 negotiation. This may not be the case in all scenarios, particularly with opportunistic approaches to security.

Normally such attacks would expire in a short time frame due to the lack of responses (such as transport layer acknowledgements) from the victim. However, as described in [Aura02], malicious participants would typically be able to spoof such acknowledgements and maintain the traffic flow for an extended period of time. For instance, if the attacker opened the TCP stream itself before redirecting it to the victim, the attacker becomes aware of the sequence number space

used in this particular session.

It should also be noted, as shown in [Bombing], that without ingress filtering in the attacker's network such attacks are already possible simply by sending spoofed packets from the attacker to the victim directly. Consequently, it makes little sense to protect against attacks of similar nature in MOBIKE. However, it still makes sense to limit the amplification capabilities provided to attackers, so that they cannot use stream redirection to send 1000 packets to the victim by sending just a few packets themselves.

Note that a variant of the flooding attack exists in IKEv2 NAT Traversal functionality [PseudoNAT]. In this variant, the attacker has to be on the path between the participants, changing the addresses in the packets that pass by. This attack is possible because the addresses in the outer headers are not protected. When the attacker leaves the path, the correct situation is restored after the exchange of the next packets between the participants.

Spoofing indications related to network connectivity

Attackers may also spoof various indications from lower layers and the network in an effort to confuse the peers about which addresses are or are not working. For example, attackers may spoof ICMP error messages in an effort to cause the parties to move their traffic elsewhere or even to disconnect. Attackers may also spoof information related to network attachments, router discovery, and address assignments in an effort to make the parties believe they have Internet connectivity when in reality they do not.

This may cause use of non-preferred addresses or even denial-of-service.

Denial-of-service of the participants through MOBIKE

Inappropriate MOBIKE protocol mechanisms might make it possible for attackers to disconnect the participants, or to move them to non-operational addresses.

MOBIKE addresses these threats using the following countermeasures:

Payload traffic protection

The use of IPsec protection on payload traffic protects the participants against disclosure of the contents of the traffic,

should the traffic end up in an incorrect destination. It is recommended that security policies be configured in a manner that takes into account that a single security association can be used through different paths at different times.

Protection of MOBIKE payloads

The payloads used in MOBIKE are encrypted, integrity protected, and replay protected. This assures that no one except the participants can, for instance, give a control message to change the addresses.

Note, however, that the actual IP address communicated in these messages is in the outer IP header and not protected, just as in IKEv2 NAT Traversal. MOBIKE adds the NAT_PREVENTION payload, however, which can be used to prevent modifications by outsiders. Where this payload is used, communication through NATs and other address translators is impossible, however. This feature is mainly intended for site-to-site VPN cases, where the administrators may know beforehand that NATs are not present, and thus any modification to the packet can be considered to be an attack.

Explicit address change

MOBIKE allows only address changes that are explicitly requested. This provides additional security beyond to what IKEv2 NAT Traversal has, but it should be noted that the benefits of this can only be realized when MOBIKE is used without intervening NATs and NAT Traversal.

When NAT Traversal is supported, the peer's address may be updated automatically to allow changes in NAT mappings. The "continued return routability" feature, implemented by the COOKIE2 payload, allows verification of the new address after the change. This limits the duration of any "third party bombing" attack by off-path (relative to the victim) attackers.

Return routability tests

This specification requires the use of return routability tests (under certain conditions) to ensure that third party flooding attacks are prevented. The tests are authenticated messages that the peer has to respond to in order for the address change to be committed to. The tests contain unpredictable data, and can only be properly responded to by someone who has the keys associated with the IKEv2 security association and who has seen the request packet for the test.

MOBIKE does not provide any protection of its own for indications from other parts of the protocol stack. However, MOBIKE is resistant to incorrect information from these sources in the sense that it provides its own security for both the signaling of addressing information as well as actual payload data transmission. Denial-of-service vulnerabilities remain, however. Some aspects of these vulnerabilities can be mitigated through the use of techniques specific to the other parts of the stack, such as properly dealing with ICMP errors [ICMPAttacks], link layer security, or the use of [SEND] to protect IPv6 Router and Neighbor Discovery.

## 5. IANA considerations

This document does not create any new namespaces to be maintained by IANA, but it requires new values in namespaces that have been defined in the IKEv2 base specification [IKEv2].

This document defines one new IKEv2 exchange, "PATH_TEST", whose value is to be allocated from the "IKEv2 Exchange Types" namespace. This exchange is described in Section 2.5.

This document also defines several new IKEv2 notification payloads whose values are to be allocated from the "IKEv2 Notification Payload Types" namespace. These notification payloads are described in Section 3.

## 6. Acknowledgements

This document is a collaborative effort of the entire MOBIKE WG. We would particularly like to thank Jari Arkko, Francis Dupont, Paul Hoffman, Tero Kivinen, and Hannes Tschofenig. This document also incorporates ideas and text from earlier MOBIKE protocol proposals, including [AddrMgmt], [Kivinen], [MOPO], and [SMOBIKE], and the MOBIKE design document [Design].

# 7. References

## 7.1 Normative references

[FIPS180-2]
National Institute of Standards and Technology, "Specifications for the Secure Hash Standard", Federal Information Processing Standard (FIPS) Publication 180-2, August 2002.

[IKEv2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-17 (work in progress), October 2004.

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

[UDPEncap]
Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

## 7.2 Informative references

[AddrMgmt]
Dupont, F., "Address Management for IKE version 2", draft-dupont-ikev2-addrmgmt-07 (work in progress), May 2005.

[Aura02] Aura, T., Roe, M., and J. Arkko, "Security of Internet Location Management", Proc. 18th Annual Computer Security Applications Conference (ACSAC), December 2002.

[Bombing] Dupont, F., "A note about 3rd party bombing in Mobile IPv6", draft-dupont-mipv6-3bombing-02 (work in progress), June 2005.

[Design] Kivinen, T. and H. Tschofenig, "Design of the MOBIKE protocol", draft-ietf-mobike-design-02 (work in progress), February 2005.

[ICMPAttacks]
Gont, F., "ICMP attacks against TCP", draft-gont-tcpm-icmp-attacks-03 (work in progress), December 2004.

[Kivinen] Kivinen, T., "MOBIKE protocol",

draft-kivinen-mobike-protocol-00 (work in progress), February 2004.

[MIP4] Perkins, C., "IP Mobility Support for IPv4", RFC 3344, August 2002.

[MOPO] Eronen, P., "Mobility Protocol Options for IKEv2 (MOPO-IKE)", draft-eronen-mobike-mopo-02 (work in progress), February 2005.

[PseudoNAT] Dupont, F. and J-J. Bernard, "Transient pseudo-NAT attacks or how NATs are even more evil than you believed", draft-dupont-transient-pseudonat-04 (expired) (work in progress), June 2004.

[SEND] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[SMOBIKE] Eronen, P. and H. Tschofenig, "Simple Mobility and Multihoming Extensions for IKEv2 (SMOBIKE)", draft-eronen-mobike-simple-00 (work in progress), March 2004.

[UNSAF] Daigle, L., "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation", RFC 3424, November 2002.

Author's Address

Pasi Eronen (editor)
Nokia Research Center
P.O. Box 407
FIN-00045 Nokia Group
Finland

Email: pasi.eronen@nokia.com

## 1. Changelog

(This section should be removed by the RFC editor.)

Changes from -00 to -01:

o Editorial fixes and small clarifications (issues 21, 25, 26, 29).

- o Use Protocol ID zero for notifications (issue 30).
- o Separate ADDITIONAL_*_ADDRESS payloads for IPv4 and IPv6 (issue 23).
- o Use the word "path" only in senses that include the route taken (issue 29).

## Intellectual Property Statement



## Disclaimer of Validity



## Copyright Statement



## Acknowledgment

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