MIPSHOP Working Group
Internet Draft
Expires: January 7, 2009

Jaeduck Choi
Doohwan Kim
Souhwan Jung
Soongsil University
July 7, 2008

A Handover Authentication based on AAA server in FMIPv6
draft-choi-mipshop-handover-authentication-aaa-00.txt

Status of this Memo

Abstract

This document describes a handover authentication protocol based on the AAA server in FMIPv6. The proposed scheme employs the Diffie-Hellman (DH) algorithm to enhance security aspects, and modifies the DH key exchange to reduce computational cost at the Mobile Node (MN) by delegating exponential operation to the AAA server. The MN and Access Router (AR) establish the handover key HK through the AAA server. The main advantage of this document is more secure and suitable to a light-weight mobile terminal.

Table of Contents

1. Introduction

In the mobile IP networks [2],[3], the handover authentication should be provided to protect signaling messages against security vulnerabilities such as the Denial of Service (DoS) attack or the intercept attack by packet redirection. Also, it should require less computing power to be suitable for a light-weight mobile terminal.

This document describes a handover authentication based on a light-weight DH key exchange with the AAA server. The MN and the AR establish the handover key HK through the AAA while the MN belongs to the AR domain. The proposed protocol also supports a robust security feature such as Perfect Forward Secrecy (PFS) and Perfect Backward Secrecy (PBS). Also, it requires less computation at the MN by delegating the DH half-key generation to the AAA server.

This document defines two messages HAReq and HAResp, and new options to carry out handover authentication.

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 RFC-2119 [1].

New terminologies are defined in this document.

Handover Authentication Request (HAReq):
HAReq is a message to deliver parameters for the handover authentication to the AAA.

Handover Authentication Response (HAResp):
HAResp is a message to deliver parameters for the handover authentication to the MN, and notify the MN of the success or failure of the handover authentication.

HK_i :
Handover key between the MN and AR_i for securing FBU message

AK :
Authentication key derived from the master key after an initial full EAP authentication

3. Protocol Operations

3.1. Overview

In the proposed scheme, we assume that the MN and AAA server perform an initial full EAP authentication [4][5] during a bootstrapping resulting that a master key is established between them. And then, the MN and AAA server derive the Authentication Key (AK) from the master key, and the MN and AR share a Handover Key (HK) that is also derived from the master key. In the future, the AAA server authenticates the MN using the AK when the MN requests the handover authentication to the AAA server. Also, we assume that a secure channel exists between the AR and AAA server using the TLS or IPSec to protect handover authentication messages.

The basic idea of our scheme is using the DH key exchange to enhance security aspects, and is modifying the DH algorithm to reduce computational cost at the MN by delegating exponential operation to the AAA server. Figure 1 shows the sequential steps of a proposed protocol in FMIPv6. The MN and the AR_1 share the handover key HK_1 after the bootstrapping authentication. When the MN moves from the AR_1 to the AR_2, the MN protects the FBU message using the HK_1. The MN in the AR_2 domain may exchange the HK_2 with the AAA server to protect the FBU message for the next handover. This key exchange procedure is achieved among the MN, AR_2, and AAA server.

The AAA server authenticates the MN using the Message Authentication Code (MAC) with the AK. If the validation is successful, the MN and AR_2 generate a new handover key HK_2 using the DH key exchange. The HK_2 is used for handover authentication when the MN moves from the AR_2 to the next AR (AR_3). The MN repeats the same procedure whenever it handovers.

Step 1
- Bootstrapping authentication
Step 2
- FBU procedure protected by HK_1 when MN moves from AR_1 to AR_2
Step 3
- Handover from AR_1 to AR_2
Step 4
- Authentication procedure between the MN and AAA server using AK
- Key exchange (HK_2) based on DH key algorithm
Step 5
- FBU procedure protected by HK_2 when MN moves from AR_2 to AR_3
Step 6
- Handover from AR_2 to AR_3

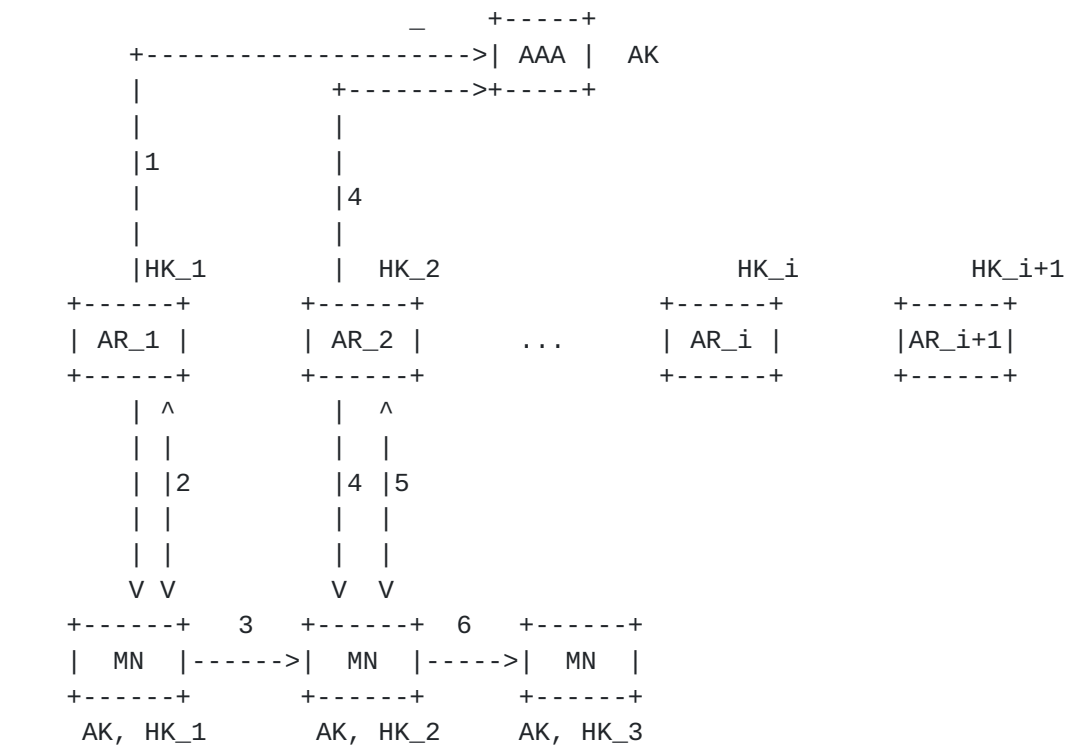

Figure 1 Protocol Overview.

3.2. Protocol Details

Figure 2 shows the protocol of the handover authentication. In Figure 2, the MN SHOULD generate and store a random number r and value g^r after a bootstrapping authentication. Two values are reused to reduce computational overhead at the MN during their lifetime.

The MN moves from the AR (AR_i-1) to the current AR (AR_i) at the time T_1. The MN and AR MUST exchange a handover key HK_i to protect the FBU message for the next handover when the MN belongs to the AR (AR_i) domain as shown in the following figure.

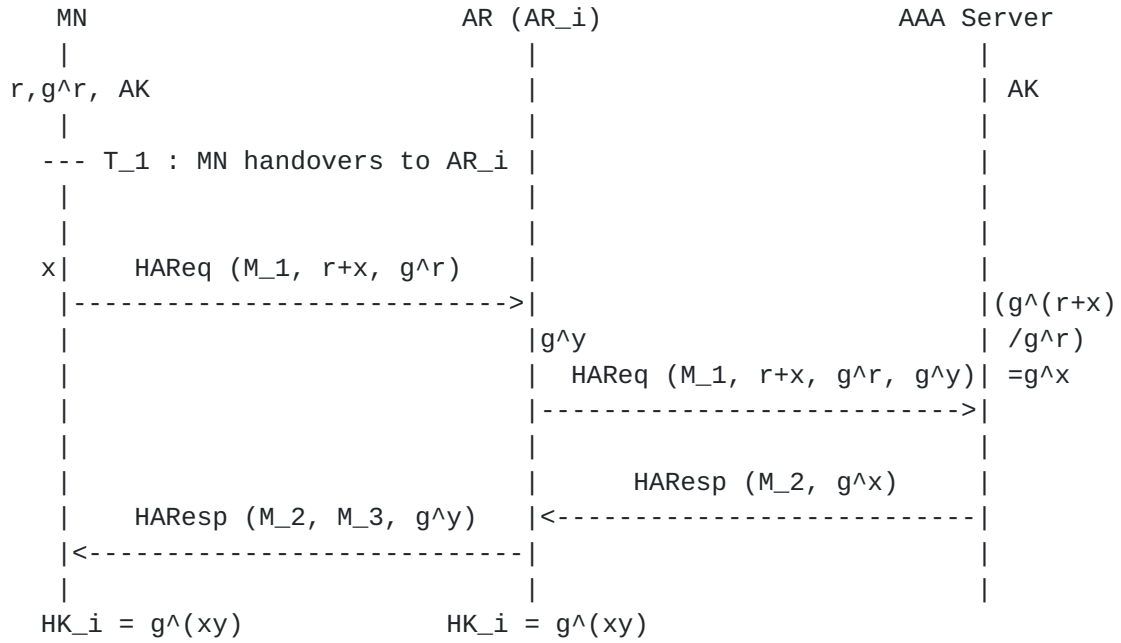

Figure 2 Procedure of Handover Authentication.

The MN generates a new random value x and computes the message M_1. And then, the MN sends the HAReq message to the AR. The HAReq message contains following values.

M_1=H(AK, ID_MN||ID_AR||ID_AAA||r+x||g^r)
HAReq [M_1, ID_MN, ID_AR, ID_AAA, r+x, g^r]

Upon receiving the messages, the AR generates a random number y and computes its DH public key g^y. The AR forwards the receiving

messages with the g^y to the AAA. The HAReq message contains following values.

HAReq [M_1, ID_MN, ID_AR, ID_AAA, r+x, g^r, g^y]

The AAA server verifies the message M_1. If the validation is successful, the AAA server computes a value g^(r+x), and then extracts the value g^x, the MN's DH public key, by computing g^(r+x)/g^r. The AAA server computes the M_2, and then replies with the HAResp message to the AR. The AAA server also notifies the success of authentication to the AR. The HAResp message contains following values.

M_2=H(AK, ID_MN||ID_AR||ID_AAA||g^y)
HAResp ["Success", M_2, ID_MN, ID_AR, ID_AAA, g^x]

If the AR receives the messages including the failure notification from the AAA server, the AR notifies the MN of the failure of the handover authentication. Otherwise, the AR computes the new handover authentication key HK_i=(g^x)^y= g^(xy) using the private key y and the received value g^x from the AAA server. The AR computes M_3, and sends the message HAResp to the MN. Note that the AR SHOULD cache the HK_i and ID_MN for securing FBU procedure when the MN will move from the AR (AR_i) to the next AR (AR_i+1). The HAResp message contains following values.

M_3=H(g^(xy), M_2||ID_MN||ID_AR||ID_AAA)
HAResp ["Success", M_2, M_3, ID_MN, ID_AR, ID_AAA, g^y]

The MN verifies the M_2 using the AK. If a success, the MN computes the new handover authentication key HK_i=(g^y)^x=g^(yx) using the private key x and the public key g^y, and then verifies the M_3. If the MN fails to verify the M_2 or M_3, the authentication fails. In the future, the MN MUST perform securing FBU using the HK_i when it moves from the AR (AR_i) to the next AR (AR_i+1).

- Securing FBU procedure: FBU, H(HK_i, FBU)

3.2.1. MN Behavior

The MN MUST share the AK with the AAA server after initial bootstrapping authentication. Also, the MN SHOULD store a random value r and value g^r during their lifetime.

The MN MUST use the HK_i for protecting the FBU message when the MN handovers to the next AR (AR_i+1). After moving to the AR_i+1, the MN MUST initiate a HAReq message to exchange the HK_i+1 with the AR_i+1.

If the MN receives the HAResp message including the success notification from the AAA server, the MN MUST generate the HK_i and cache it for next handover authentication. In the future, the MN MUST use HK_i for securing FBU between the MN and AR_i when the MN moves from the AR (AR_i) to the next AR (AR_i+1).

The MN that belongs to the AR (AR_i+1) domain MUST store the HK_i until the MN and AR_i+1 exchange the HK_i+1. Also, the MN SHOULD cache HK_i for its life time. If the MN comes back to the AR_i domain, the MN SHOULD reuse its HK_i.

3.2.2. AR Behavior

Upon receiving the message HAReq from the MN, the AR MUST generate a random number y and a value g^y. Also, the AR MUST forward the received message HAReq with the value g^y to the AAA server, and create cache table including the ID_MN, ID_AR, ID_AAA, y, and g^y.

The AR (AR_i) MUST compute the new handover authentication key HK_i using its DH private key y in cache table and the received value g^x from the AAA server, when the AR receives the message HAResp including the success notification. Also, the AR MUST compute the message M_3 to confirm the handover key HK_i with the MN. The AR MUST send the message HAResp including the life time of the HK_i to the MN. If the AR receives the failure of authentication from the AAA server, the AR MUST delete the MN's cache table except the DH parameters y and g^y that are not used for generating handover key. The AR SHOULD reuse the stored value y and g^y without computing new DH public keys.

The AR MUST verify a FBU message using the HK_i when the MN moves from the AR (AR_i) to the next AR (AR_i+1).

3.2.3. AAA Server Behavior

The channels between the AAA server and ARs SHOULD be established by the IPsec or TLS. Upon receiving a HAReq message from the AR (AR_i), the AAA server MUST verify the value M_1 in the HAReq message using the AK shared with the MN. If the AAA server fails to verify the M_1, the AAA server MUST send a HAResp message including the failure of authentication to the AR, and ignore the received HAReq message.

Otherwise, the AAA server MUST compute a g^x by computing g^(r+x)/g^r. And then the AAA server MUST generate the message M_2. The AAA server MUST send the HAResp message with the result of M_1 verification to the AR.

4. Message Formats

This document defines two messages HAReq and HAResp, and new options to carry out handover authentication.

4.1. Handover Authentication Request (HAReq)

The HAReq MUST be sent from the MN to the AAA server through the AR for the handover authentication. Receiving the HAReq message from the MN, the AR MUST forward it to the AAA server. The HAReq message SHOULD use Options to deliver extra variables for authentication (to be assigned by IANA).

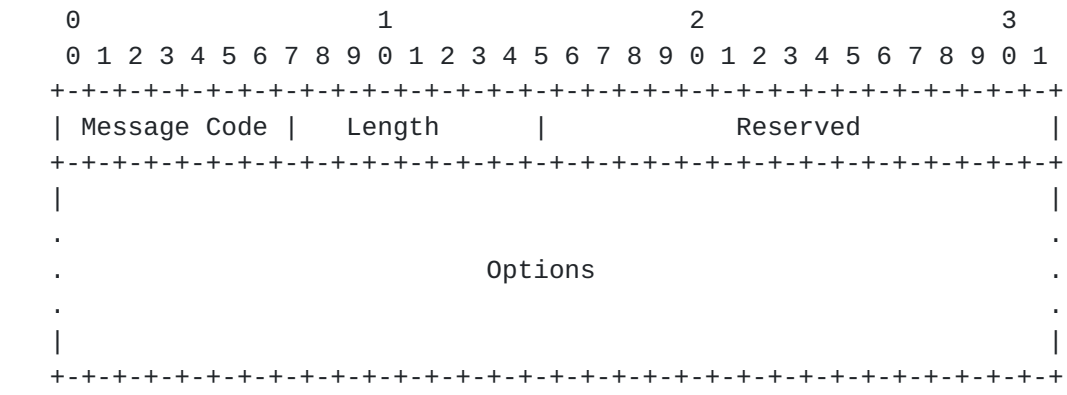

Figure 3 Handover Authentication Request (HAReq) Message.

HAReq Fields

Message Code

8-bit field indicates the handover authentication protocol, the value of which is taken from the IANA. A value of '1' (to be assigned by IANA) indicates the HAReq message.

Length

8-bit field is the length of the HAReq message.

Reserved

16-bit field reserved for future use. This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Options

Options field includes extra variables for handover authentication.

4.2. Handover Authentication Response (HAResp)

The HAResp MUST be sent to the MN in responding to a HAReq message.

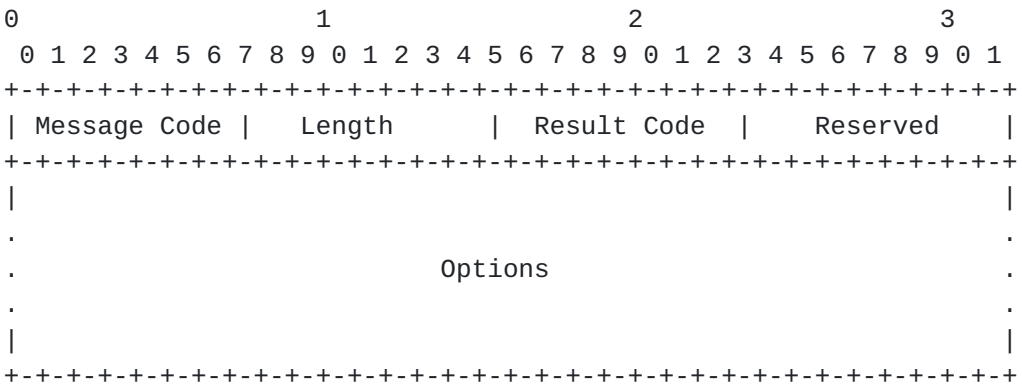

Figure 4 Handover Authentication Response (HAResp) Message.

HAResp Fields

Message Code

8-bit field indicates the handover authentication protocol, the value of which is taken from the IANA. A value of '2' (to be assigned by IANA) indicates the HAResp message.

Length

8-bit field is the length of the HAResp message.

Result Code

8-bit field indicates the result of authentication. A value of '200' (to be assigned by IANA) indicates the "Success", and '400' (to be assigned by IANA) indicates the "Fail"

Reserved

16-bit field reserved for future use. This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Options

Options field includes extra variables for handover authentication.

4.3. Options

The HAReq and HAResp messages SHOULD accommodate various options as follows.

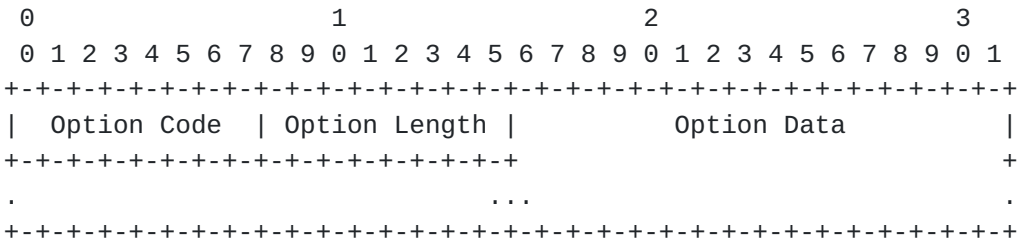

Figure 5 Option Message.

Option Fields

Option Code

8-bit field indicates extra variables as follows.

M_1 (Code number: to be assigned by IANA)
M_2 (Code number: to be assigned by IANA)
M_3 (Code number: to be assigned by IANA)
ID_MN (Code number: to be assigned by IANA)
ID_AR (Code number: to be assigned by IANA)
ID_AAA (Code number: to be assigned by IANA)

Random_1 (Code number: to be assigned by IANA)
 : This value is r+x.
Random_2 (Code number: to be assigned by IANA)
 : This value is g^r
DH_MN (Code number: to be assigned by IANA)
 : This value is g^x.
DH_AR (Code number: to be assigned by IANA)
 : This value is g^y.
HK_LifeTime (Code number: to be assigned by IANA)

Option Length

8-bit field is the length of the Options field.

5. Security Considerations

The proposed scheme assumes that there are secure channels among the AAA server and ARs. Therefore, communications between the AAA server and AR are secure against the MITM (Man-in-the-Middle) attack. Although there is no secure channel between the MN and the AR, the MN secures the messages using the MAC with the AK shared with the AAA server. This can also protect the MN and the AR against MITM attack.

The DoS attack for exhausting resource has become a major security threat. The DoS attack considered on our scheme is the CPU exhaustion attack such as exponent operation when an attacker sends a number of fake requests to the AR. In the proposed scheme, by reusing unused DH public keys, ARs protect themselves against malicious attackers who will try to exhaust their computing power. The AR requires two exponent operations per handover procedure: a DH public value g^y and handover key (g^x)^y. Upon receiving the fake requests, the AR will generate DH public keys and forward the fake requests with the DH public keys to the AAA server. However, the AAA server may fail to verify the fake requests due to unknown AK, and then it notifies the failure of authentication to the AR. For the resistance against DoS attack, if the AR receives the failure of authentication from the AAA server, the AR should keep the generated DH public key (g^y) to be reused for the next request. The proposed protocol can also protect the nodes against replay attack by using a random number and caching the MN's ID and HK_i.

The proposed scheme provides the PFS and PBS. The PFS and PBS mean that even if a handover key HK_i is compromised by some reasons, it never reveals all the previous and next handover keys. We use the DH key agreement protocol to provide the PFS and PBS. If the HK_i is exposed by some attacks, an attacker gives no clues to guess the previous and next handover keys. The reason is that the MN and AR generate the handover key HK_i using new DH private key x_i and y_i whenever the MN performs the handover authentication.

Our protocol is robust to the ping-pong problem. If the MN quickly changes its position between the ARs, there may be the ping-pong problem. When the MN frequently moves between the AR_i and AR_i+1, the handover key HK_i and HK_i+1 should be changed according to its movement area. The proposed scheme securely caches the MN's HK at the MN and the AR. Hence, the MN can securely reuse the HK without disclosing it and performing redundant handover authentication procedure.

6. IANA Considerations

The IANA will allocate the numbers to the HAReq, HAResp, and options.

7. References

7.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[2] Koodli, R., "Mobile IPv6 Fast Handovers", RFC 5268, June 2008.
[3] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
[4] Aboba, B., "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
[5] Aboba, B., "Extensible Authentication Protocol (EAP) Key Management Framework", draft-ietf-eap-keying-22 (work in progress), October 2006.
[6] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 33588, September 2003.
[7] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

Authors' Addresses

Jaeduck Choi
Soongsil University
1-1, Sangdo-dong, Dongjak-gu
Seoul 156-743
KOREA

Email: cjduck@cns.ssu.ac.kr

Doohwan Kim
Soongsil University
1-1, Sangdo-dong, Dongjak-gu
Seoul 156-743
KOREA

Email: shapja@cns.ssu.ac.kr

Souhwan Jung
Soongsil University
1-1, Sangdo-dong, Dongjak-gu
Seoul 156-743
KOREA

Email: souhwanj@ssu.ac.kr

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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