논문 2005-42CI-5-3

주소기반의 키를 사용하는 모바일 IPv6 바인딩 갱신 프로토콜 개선

(An Improvement of Mobile IPv6 Binding Update Protocol Using Address Based Keys)

유 일 선*, 최 승 교**

(Ilsun You and Sung Kyo Choi)

요 약

최근에 주소기반의 키를 적용하는 모바일 IPv6 바인딩 갱신 프로토콜(BU-ABK)이 제안되었다. BU-ABK는 주소기반의 키를 통해 광역의 보안 인프라 없이 강력한 인증과 안전한 키교환을 지원하지만 공개키 연산을 위한 암호화 매개변수의 위조를 탐지할 수 없기 때문에 중간자 공격과 서비스 거부공격에 취약한 문제점을 갖는다. 본 논문에서는 이러한 BU-ABK의 취약점을 개선하고 제한된 전력을 갖는 이동장치를 위한 최적화 옵션을 제안한다. 또한, BU-ABK와의 비교를 통해 제안 프로토콜이 이동노드 상의 과중된 오버헤드를 초래하지 않고 강력한 보안을 제공함을 보인다.

Abstract

Recently, a mobile IPv6 binding update protocol using Address Based Keys (BU-ABK) was proposed. This protocol applies Address Based Keys (ABK), generated through identity-based cryptosystem, to enable strong authentication and secure key exchange without any global security infrastructure. However, because it cannot detect that public cryptographic parameters for ABKs are altered or forged, it is vulnerable to man-in-the-middle attacks and denial of service attacks. Furthermore, it has heavy burden of managing the public cryptographic parameters. In this paper, we show the weaknesses of BU-ABK and then propose an enhanced BU-ABK (EBU-ABK). Furthermore, we provide an optimization for mobile devices with constraint computational power. The comparison of EBU-ABK with BU-ABK shows that the enhanced protocol achieves strong security while not resulting in heavy computation overhead on a mobile node.

Keywords: Mobile IP Version 6(MIPv6), Binding Updates, ABKs, RR, CGA

I. Introduction

The route optimization operation in Mobile IP Version 6 (MIPv6) environment allows direct routing from any correspondent node (CN) to any mobile node (MN)[2]. But the route optimization requires that the MN constantly informs its CNs about its new care-of-address (CoA) by sending them binding update (BU) messages. Without a security solution, the route optimization functionality exposes the involved MNs and CNs to various security threats[1]. To address the security threats, the CN is required to authenticate the MN sending the BU message. Only after successfully authenticating the MN, the CN has to update its binding cache entries. Unfortunately, it is so difficult to achieve strong authentication between two previously unknown nodes (MN and CN) where no global security infrastructure is available. Thus, the need has arisen for a security solution to enable sufficient

* 정회원, 한국성서대학교 정보과학부
(Department of Information Science, Korean Bible University)
** 정회원, 삼척대학교 컴퓨터공학과
(Department of Computer Engineering, National Samcheok University)
접수일자: 2005년3월14일, 수정완료일: 2005년9월6일

authentication between the CN and the MN, excluding the use of traditional secret- or Public Key Infrastructure (PKI) based authentication infrastructures.

Several researches have been conducted to solve this security issue[2-7]. Recently, the Return Routability (RR) protocol has been accepted as the basic technique for the secure BU. Nevertheless, the RR protocol has some potential drawbacks, both in terms of its security properties and also performance[2]. Unlike the RR protocol, the protocols such as CAM, CAM-DH and SUCV have been proposed based on public key[2-6]. The public key based protocols attempted to associate the MN's address with its public key to avoid the use of additional security infrastructure such as PKI, by using Cryptographically Generated Address (CGA) method. CGA is IPv6 address where the interface identifier is generated by hashing the address owner's public key[3,8]. The address owner can use the corresponding private key to assert address ownership and to sign messages sent from the address without any additional security infrastructure. However, in spite of such strength, the CGA-based protocols are vulnerable to brute-force attacks searching for hash collisions, since they use only the 62 bits of the interface identifier as the hash value for the address owner's public key.

On the other hands, recently, a public key based binding update protocol using Address Based Keys (BU-ABK) was proposed[5]. In this protocol, a MN's public key is a hash of its HoA and an expiration time, and its private key is generated by its HA through identity-based cryptosystem. Therefore, once a CN has obtained parameters for a HA, it does not need to request an authenticated public key of each MN in the domain of the HA. The protocol, unlike the CGA-based protocols, is thus free from the limited size of the interface identifier. Although the protocol overcomes the shortcoming of the CGA method, we find that it is vulnerable to man-in-the-middle attacks and denial of service attacks. Furthermore, it has heavy burden of managing the public cryptographic parameters. Because such problems mainly result from there being no way to detect that public cryptographic parameters for ABKs are altered or forged, BU-ABK needs a mechanism to securely distribute the parameters without any security infrastructure. To provide the mechanism, we adopt Aura's two hash-based CGA scheme as a solution[8].

In this paper, we show the weaknesses of BU-ABK and then provide an enhanced BU-ABK (EBU-ABK). Furthermore, we provide an optimization for MNs with constraint computational power, which is an important design consideration for public key based binding update protocols[2].

The rest of the paper is organized as follows. Section II reviews and analyzes BU-ABK. Section III shows the weaknesses of BU-ABK. In section IV, we describe the two hash-based CGA scheme and propose EBU-ABK. Section VI analyzes EBU-ABK. Finally, section 6 draws some conclusions.

II. Analysis of BU-ABK

BU-ABK is a public key based protocol to secure MIPv6 binding updates. Especially, it usesidentity-based cryptosystem to support strong authentication and secure key exchange between MNs and CNs without any global security infrastructure. In this section, we briefly describe identity-based cryptosystems and then analyze BU-ABK.

1. Notation

h() : a cryptographic secure one-way hash function

prf(k, m) : a keyed hash function. It accepts a secret key *k* and a message *m*, and generates a pseudo random output

m|n : concatenation of two messages *m* and *n*

IPrK: the identity-based the private key for the MN

IPuK: the identity-based public key for the MN

param: the public cryptographic parameters which are necessary for the identity-based

cryptographic function

ENCRYPT(m,IPuK,param): The identity-based encrypt function. It accepts a message *m*, *IPuK* and *param*, and generates an encrypted message.

DECRYPT(m,IPrK,param): The identity-based decrypt function. It accepts an encrypted message *m*, *IPrK* and *param*, and generates a decrypted message.

2. Applying Identity-Based Cryptosystem

BU-ABK uses identity-based cryptosystems to construct a public/private key associated with the MN's HoA. In the protocol, the publicly known identifier, the MN's public key, is a hash of the MN's HoA and an expiration time, and the HA serves as an Identity-based Private Key Generator (IPKG) for its all MNs. Prior to generating the MN's private key, the HA should have a secret master key known only to itself and public cryptographic parameters, which are necessary for the identity-based cryptographic algorithm. The HA uses the secret master key to generate the private key and returns it with the expiration time. The parameters are used to perform cryptographic operations by two nodes involved in securing or encrypting a message. The secret master key can expire or become compromised, and in that case the publicly known parameters would have to be updated. It is assumed that all valid public/private key pairs associated with the HA will have the same expiration time.

3. Protocol Operation

BU-ABK is composed of three phases: configuration phase, authentication phase and BU phase. Fig. 1 outlines configuration phase. In this phase, the MN is configured to have an identity-based public/private key pair that is associated with its 128-bit HoA, along with the public cryptographic parameters. Secure distribution of a private key and cryptographic parameters is available through the pre-established IPSec security association between the MN and its HA in the MIPv6[2,7].

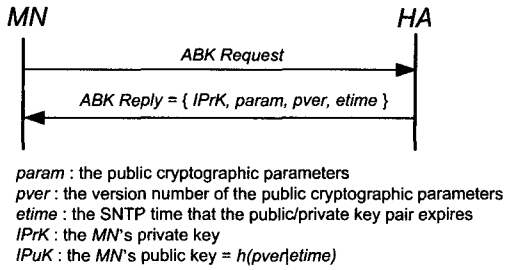


그림 1. 키 초기화 단계
Fig. 1. Configuration Phase.

Fig. 1 outlines configuration phase. In this phase, the MN is configured to have an identity based public/private key pair that is associated with its 128-bit HoA. This phase for securely distributing the private key and cryptographic parameters to the MN consists of the two messages: ABK Request and ABK Reply. While the ABK Request message requests private key and parameters, the ABK Reply message returns private key and parameters.

Since communication between the MN and its HA is protected with pre-establish IPsec security association in the MIPv6, secure distribution of a private key and cryptographic parameters is available[2,7].

In authentication phase, the CN obtains the cryptographic parameters from the HA, and then shares a secret key *Km* with the MN. *Km* is used to generate a session key to encrypt each the MN's BU message to the CN. Since *Km* is encrypted by the MN's public key and sent to the MN, it can be

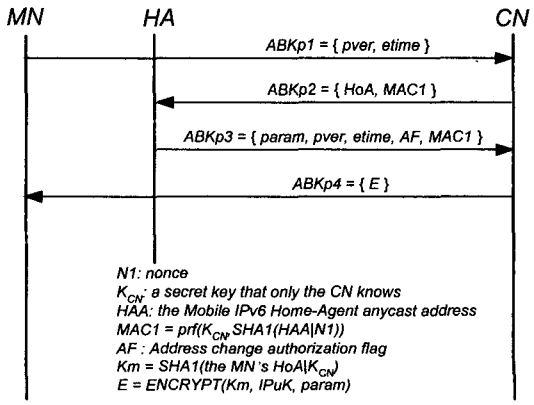
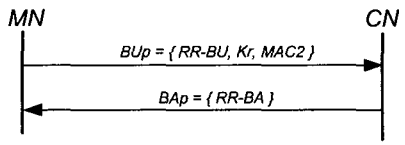


그림 2. 인증단계
Fig. 2. Authentication Phase.



RR-BU: standard binding update message used in RR protocol
Kr : random value generated by the MN
K = SHA1(Km|Kr)
MAC2 = prf(K,SHA1(RR-BU|Kr))
RR-BA : standard binding acknowledge message used in RR protocol

그림 3. 바인딩 갱신 단계
Fig. 3. Binding Update Phase.

obtained by only the MN. Therefore, if the MN's BU message is validated through *Km*, the CN can trust the MN besides its BU message.

As shown in Fig. 2, this phase consists of the four messages: ABKp1- ABKp4. If the CN already has the HA's parameters, it will skip ABKp2 and ABKp3 and send ABKp4. The nonce *N1* for ABKp2 should be refreshed periodically, but the same nonce is used for all home agents with which the CN corresponds during the same time period. The CN should also keep a cache of recently used nonces. In ABKp3, if AF is not set, the MN should be using a globally unique interface identifier, and the CN should check that the interface identifiers of the MN's HoA and the MN's CoA are the same. If AF is set, some other method of authorizing MN's CoA to change the routing should be used.

Fig. 3 outlines BU phase. In this phase, a new BU between the MN and its CN is established according to the standard MIPv6 procedure[7]. For securing BU, the MN uses the secret key Km shared in the authentication phase to generate a new session key K, and then compute MAC2 with K. Before verifying MAC2, the CN validates the MN's new CoA according to AF. If AF is not set, the CN checks that the interface identifiers of the MN's HoA and the MN's CoA are the same. Otherwise, it uses some other method. Then, it verifies MAC2 with Km. If the verification is positive, the BU is successfully established.

III. Weaknesses of BU-ABK

1. Vulnerability to Man-in-the Middle Attacks

In BU-ABK, since ABKp1, ABKp2 and ABKp3 are not signed or encrypted, it is possible for an intruder to change them without being detected[5]. Though both RR and CGA (with an initial round of RR) have the same vulnerability as claimed in [5], BU-ABK is more critical than other approaches since the CN can be incorrectly configured for the HA in addition to the MN[4-5,7]. That is, public cryptographic parameters can be replaced with faked or malicious one. Such misconfiguration enables an intruder to easily masquerade as any MN in the domain of the HA. Also, it can cause the CN to deny all MN in the domain of the HA. Therefore, BU-ABK needs to solve this problem. In this section, two types of attacks, which exploit the vulnerability, are analyzed.

A. Faked Parameters Attacks

For each ABKp1, the CN checks its parameters table to see if it has the parameters for the relevant HA. If it does not, the CN will send ABKp2 to the appropriate HA to request parameters. Upon receiving ABKp3, the CN checks param and computes *MAC1*. If param is not null and the received *MAC1* and the computed one are same, the CN caches *param*, *pver*, *etime* and AF.

However, since ABKp3 is not signed by the HA, t

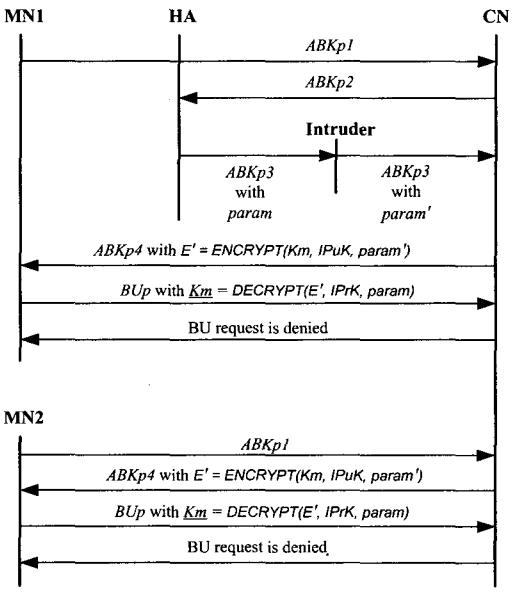


그림 4. 위조된 암호화 매개변수 공격
Fig. 4. Faked Parameters Attacks.

he CN cannot be sure that *param*, *pver* and *etime* are not altered. Therefore, while not being detected by the CN, an intruder can replace *param* of ABKp3 with faked parameters *param'*. Later, the CN with *param'* will deny the MN's BU request as shown in Fig. 4, even though the MN correctly generates BUp.

B. Malicious Parameters Attacks

By altering ABKp1 and ABKp3, an intruder can easily trick the CN into caching the public cryptographic parameters corresponding to the compromised master key or his own one. Such man-in-the middle attacks can be mounted as shown in Fig. 5.

First, an intruder on the HA-CN link sends a spoofed ABKp1 including a new version number *pver'* and *etime'* to the CN. When the CN receives the message, it finds that it does not have the parameters with *pver'*, and then sends ABKp2 to the HA which is appropriate for the received ABKp1. The message ABKp2 is intercepted by the intruder, who makes and sends a spoofed ABKp3 which contains the public cryptographic parameters *param'* corresponding to the compromised master key or his own one. When receiving the spoofed ABKp3, the CN checks *param'* and *MAC1*, and caches *param'*, *pver'*, *etime'* and AF. ABKp4 sent by the CN will be discarded.

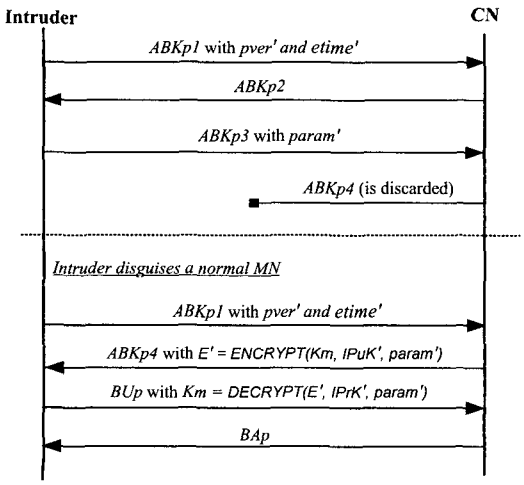


그림 5. 악의적인 암호화 매개변수 공격
Fig. 5. Malicious Parameters Attacks.

Now, the intruder can establish a fake BU by using his master key and *param'*. For that, he sends a new ABKp1 including *pver'* to the CN, while disguising himself as a normal MN. Since the CN has the parameters with *pver'*, it skips ABKp2 and ABKp3 and sends ABKp4 to the intruder. Upon receiving ABKp4 from the CN, the intruder decrypts $E' = ENCRYPT(Km, IPuK', param')$ with the faked MN's private key *IPrK'* to retrieves *Km*. While *IPuK'* can be generated through hashing the faked MN's HoA and *etime'*, *IPrK'* can be generated through the intruder's master key. As a result, the retrieved Km enables the intruder to establish a fake BU.

The attacks that try to configure the CN with faked parameters are distinguished with these attacks because they replace param of ABKp3 with just faked parameters.

2. Vulnerability to Denial of Service Attack

Denial of service (DoS) attacks can be classified into two categories: resource-exhaustion attacks and non resource-exhaustion attacks. Unluckily, BU-ABK is vulnerable to both types of attacks. Since non resource-exhaustion attacks are mentioned in Faked Parameters Attacks of the previous section, we focus resource-exhaustion attacks.

Upon a receipt of ABKp1, the CN, which caches the HA's parameters, skips ABKp2 and ABKp3, computes *E* and sends ABKp4 to the MN. However, such operation causes the CN to be vulnerable to resource-exhaustion attacks, since the CN performs the asymmetric cryptographic operation *ENCRYPT (Km, IPuK, param)* for ABKp4 without verifying ABKp1. Thus, an intruder can try to attack the CN by sending a storm of ABKp1 messages.

3. Burden of Managing the Public Cryptographic Parameters

In BU-ABK, it is difficult to enable CNs to timely update the public cryptographic parameters distributed to themselves. Even in the case that a HA's master key is compromised, the CN does not update the old parameters corresponding to the

compromised master key, until it receives ABKp1 including a new *pver*. Such vulnerability allows an intruder to use the compromised master key and the old parameters to retrieve *Km*, which results in a fake BU. For that, the only thing that the intruder should do is to send just ABKp1 including the old *pver* to the CN. In addition to the public cryptographic parameters, the HA's master secret key should be securely managed because the leakage of the key may result in various attacks.

Thus, the burden of managing the HA's secret key and public cryptographic parameters may be a large obstacle to applying BU-ABK for real systems.

Ⅳ. EBU-ABK

Since BU-ABK does not protect public cryptographic parameters for ABKs, it cannot prevent man-in-the middle attacks and non resource-exhaustion attacks. Therefore, it is desirable to protect the parameters. For that, the use of X.509 v3 certificate[9] is recommended in [5]. However, such a solution is not appropriate in MIPv6 environment where no global security infrastructure is available. Also, because public cryptographic parameters, managed by the HA, tend to be a target of attackers, it is not desirable to apply the CGA method, which uses only the 62 bits of the interface identifier as the hash value for the address owner's public key. For the reason mentioned above, Aura's two hash based CGA, which increases the cost of brute-force attacks by a factor of 2^{12*Sec} (from 2^{59} to $2^{59+12*Sec}$), is adopted as a solution[8]. In this section, we propose a protocol that combines BU-ABK with Aura's two hash-based CGA scheme. Furthermore, we provide an optimization for MNs with constraint computational power.

1. Applying the Two Hash Based CGA Scheme

Recently, Aura proposed a new CGA scheme where two hash values are computed instead of one[8]. The first hash value (Hash1) is used to produce the interface identifier (i.e. rightmost 64 bits) of the address. The purpose of the second hash (Hash2) is to artificially increase that computational complexity of generating new addresses and, consequently, the cost of brute-force attacks.

In the proposed CGA scheme, a CGA format is defined as an IPv6 address where the 12*Sec leftmost bits of the second hash value Hash2 are zero, and the rightmost 64 bits of the first hash value Hash1 equal the interface identifier of the address. The three rightmost bits of the address, which encode the security parameter Sec to determine the level of security, and the universal and group bits are ignored in the comparison. The latter two bits must both be one.

In EBU-ABK, a home link is associated with a public/private key pair P_{HA} and S_{HA} in a digital signature scheme. A HA in the home link keeps the public/private key pair, and derives a CGA from the public key P_{HA}. Each CGA is associated with an optimized parameter format including the HA's public key information and CGA parameters[8].

The process of obtaining a new CGA is as follows.

- Generate a public/private key pair P_{HA} and S_{HA} for a home link.
- Generate a new CGA via the algorithm presented in [8].
- Create an optimized parameter format. The format is simply the concatenation of the DER-encoded *subjectPublicKeyInfo* and CGAParameters data value *cgaParams*, which are defined in [8-9].

2. Protocol Operation

To employ the two hash-based CGA scheme, EBU-ABK enhances authentication phase of BU-ABK. Configuration phase and BU phase are exactly as BU-ABK.

Fig. 6 outlines EBU-ABK where the HA testifies the legitimacy of the MN's HoA, and computes digital signature to certify the validity of public cryptographic parameters.

EBU-ABK improves the drawbacks of BU-ABK as follows.

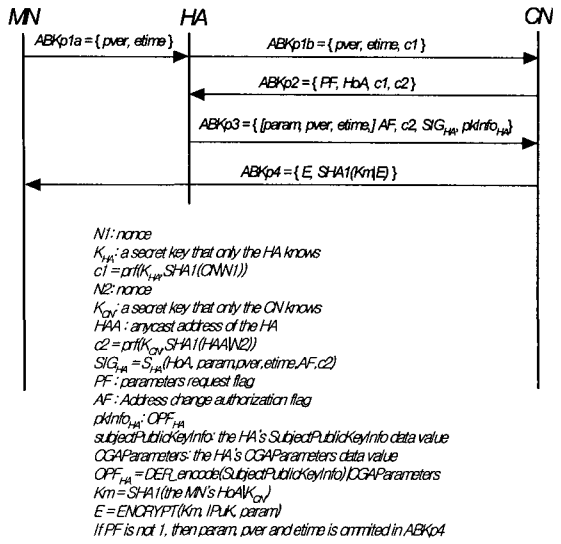


그림 6. 개선된 BU-ABK
Fig. 6. Enhanced BU-ABK.

First, to protect public cryptographic parameters for ABKs, EBU-ABK uses digital signature with Aura's two hash based CGA. For that, the HA keeps the public/private key pair P_{HA}/S_{HA} and uses CGA, derived from its public key P_{HA}, as its own MIPv6 address. The CN should validate the public key P_{HA} with the HA's CGA before verifying the signature SIG_{HA}.

Second, to prevent resource-exhaustion attacks on the HA and the CN, cookies such as *c1* and *c2* are created and added to the protocol messages. Only if they are valid, the HA and the CN should perform asymmetric cryptographic operations.

A. ABKp1a = { *pver, etime* }

In ABKp1a, the MN tries to contact the CN, giving the version and the expiration time of public cryptographic parameters. ABKp1a is sent to the MN's home link via the IPSec protected secure tunnel.

B. ABKp1b = { *pver, etime, c1*}

Upon arriving atthe home link, ABKp1a is intercepted by the HA using IPv6 Neighbor Discovery[7,10]. The HA computes a cookie c1 and sends ABKp1b to the CN. The HA should periodically refresh N1, instead of creating a new one for each ABKp1b. That is, the same nonce is used for all CNs during the same time period. The HAshould also keep a cache of recently used nonces. Thus, the HA can be stateless and protected from being flooded with nonces.

C. ABKp2 = { *PF, HoA, c1, c2* }

When receiving ABKp1b, the CN searches the public cryptographic parameters having pver from its cache. If it already has the parameters, it sets the parameters request flag *PF* to 0. Otherwise, it sets the flag to 1. It then computes a cookie *c2* and sends ABKp2 to the HA. *N2* is created and managed in the same way as *N1* by the CN. Thus, the CN, like the HA, can be stateless and protected from being flooded with nonces.

D. ABKp3 = { *[param, pver, etime,], AF, c2, SIG_{HA}, $pkInfo_{HA}$* }

When receiving ABKp2, the HA validates the cookie *c1* to prevent resource-exhaustion attacks. If *c1* is valid, it computes SIG_{HA} with its private key S_{HA}, and then sends ABKp4 to the CN. When *PF* is 0, ABKp3 can omit *param*, *pver* and *etime*.

E. ABKp4 = { *E, SHA1(Km|E)* }

On receipt of ABKp3, the CN firstly checks the cookie *c2*, and then verifies SIG_{HA} with the HA's public key P_{HA}. Since the CN performs asymmetric cryptographic operations in the case that *c2* is valid, it resists against resource-exhaustion attacks. Before verifying SIG_{HA}, the CN should verify the HA's CGA. The algorithm for verifying the HA's CGA is defined in [8]. If the HA's CGA is valid, the CN verifies SIG_{HA} using $pkInfo_{HA}$. When the verification is positive, the CN can be confident that the MN's HoA and the public cryptographic parameters are valid. If *PF* is 0 and the parameters are included in the message, the CN caches or updates the ones. Thus, the CN can timely update the HA's parameters, while preventing the man-in-the middle attacks and non resource-exhaustion attacks. After that, the CN computes the MN's public key *IPuK* =

h(the MN's HoA|*etime)* and *E* = *ENCRYPT(Km, IPuK, param)*, and sends ABKp5 to the MN. When the MN receives ABKp5, it computes the secret key *Km* = *DECRYPT(E, IPrK, param)*, which is used in BU phase.

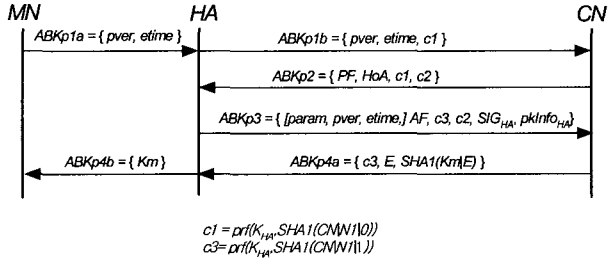


그림 7. 저전력의 이동노드를 위한 최적화
Fig. 7. Optimization for Low-Power Mobile Nodes.

3. Optimization

Because asymmetric cryptographic operations are computationally intensive, it is desirable to provide an optimization for MNs with constraint computational power, such as PDAs and cellular phones. For that, EBU-ABK allows the HA to perform the expensive operations on behalf of the MN. Fig. 7 shows EBU-ABK with this optimization

To prevent resource-exhaustion attacks, the HA's cookie *c3* is added to ABKp3 and ABKp4a. The HA intercepts ABKp4a and verifies the cookie *c3* in it. If *c3* is valid, the HA decrypts *E* with the MN's private key and then forwards *Km* to the MN. Thus, the HA off-loads the expensive cryptographic operations of its MNs to itself. In EBU-ABK, such optimization is available, since the HA can regenerate the MN's private key through identity-based cryptosystems and communication between the MN and the HA is protected with pre-establish security association in the MIPv6.

V. Analysis of EBU-ABK

In this section, our protocol is analyzed in terms of security and performance. Then, it is compared with BU-ABK.

1. Security

A. Redirection Attacks

Redirection attacks are malicious acts which redirect traffic from CNs to locations chosen by intruders[2]. These attacks can be classified into two categories, session hijacking and malicious mobile node flooding.

Since BU-ABK is not vulnerable to these attacks[5], EBU-ABK, which improves BU-ABK, can resists against them.

B. Cost of Brute-Force Attacks for Cracking CGA

The two hash-based CGA scheme, adopted in EBU-ABK, includes the routing prefix of the address in the input for the first hash value Hash1 and uses the second hash value Hash2 to increase the cost of brute-force attacks. During address generation, the input for Hash2 is modified by varying the value of modifier until the leftmost 12*Sec bits of Hash2 are zero. This increases the cost of address generation approximately by a factor of 2^{12*Sec}. It also increases the cost of brute-force attacks by the same factor (ie. From 2^{59} to $2^{59+12*Sec}$). Thus, EBU-ABK is more secure than other CGA based approaches such as CAM-DH and SUCV, which require the cost of brute-force attacks, $O(2^{62})$.

C. Man-In-The-Middle Attacks

(1) Man-In-The-Middle Attacks in ABKp3

To mount man-in-the-middle attacks, intruders can fabricate ABKp3 messages, which include public cryptographic parameters, SIG_{HA} and $pkInfo_{HA}$. For that, intruders must have a private/public key pair that can generate SIG_{HA}. Such a private/public key pair is able to be obtained by performing the brute-force attack to compromise the two hash-based CGA. Thus, man-in-the-middle attacks in ABKp3 need at least the cost of the brute-force attack for cracking the two hash-based CGA, $O(2^{59}$ to $2^{59+12*Sec})$, which enables EBU-ABK to prevent them.

(2) Man-In-The-Middle Attacks in ABKp4

Intruders can modify ABKp4 messages to make

the subsequent BU requests denied. For successful attacks, they should replace E with E' and then compute $SHA1(Km|E')$. Since $SHA1(Km|E')$ cannot be made without Km, they must perform the brute-force attack to find Km. Such a brute-force attack needs the cost, $O(2^{128})$, which is sufficient to prevent man-in-the-middle attacks in ABKp4.

D. Resource-Exhaustion Attacks

As mentioned in Ⅳ, EBU-ABK verifies the cookies *c1*, *c2* and *c3* before performing asymmetric cryptographic operations. Thus, intruders should have the valid cookies to flood CN or HA with asymmetric cryptographic operations. Intruders can perform the brute-force attack to find the valid cookies. Such an attack needs at least the cost, $O(2^{160})$, which is sufficient to resist against resource-exhaustion.

표 1. EBU-ABK와 BU-ABK의 비교
Table 1. The Comparison of EBU-ABK with BU-ABK.

	EBU-ABK with the optimization for low-power MN	BU-ABK
1	Two hash-based CGA	X
2	*Csign* + *Cverify* + *Ccga*	0
3	*Ca-enc* + *Ca-dec*	*Ca-enc* + *Ca-dec*
4	3**Ccookie*	*Ccookie*
5	○	○
6	○	X
7	○	X
8	High	Low
9	0	1

1. Mechanism verifying public cryptographic parameters
2. Cost of signing and verifying public cryptographic parameters
3. Cost of encrypting and decrypting the shared secret *Km*
4. Cost of preventing resource-exhaustion attacks
5. Ability to preventing redirection attacks
6. Ability to preventing man-in-the-middle attacks
7. Ability to preventing resource-exhaustion attacks
8. Ability to timely update public cryptographic parameters
9. Asymmetric cryptographic operations in the MN

* notation
Csign: Cost for signing public cryptographic parameters
Cverify: Cost for verifying public cryptographic parameters
Ccga: Cost for verifying a CGA (= two hash operations)
Ca-enc: Cost for an asymmetric encryption
Ca-dec: Cost for an asymmetric decryption
Ccookie: Cost for generating and verifying a cookie

attacks.

2. Performance

In comparison to BU-ABK, EBU-ABK has an additional cos for validating the public cryptographic parameters and preventing resource-exhaustion attacks, which may result in performance degrade. Also, EBU-ABK, unlike BU-ABK, has to exchange ABKp2 and ABKp3 in each authentication phase. However, since the authentication phase of EBU-ABK provides high-level security, a new shared secret key Km established in this phase can be used in BU phase for a relatively long period of time. That is, once Km is established, the MN and the CN can perform binding updates without exchanging a secret key for a long lifetime. It is desirable for the lifetime of Km to be much longer than at least 420 seconds that are the maximum limitation of the RR protocol[2,7]. Such long lifetime of Km allows the additional cost not to result in critical performance degrade.

When the optimization for low-power MNs is used, the MN in the authentication phase just receives Km decrypted by its HA without performing any cryptographic operations, whereas the MN in the authentication phase of BU-ABK should decrypt E with its private key.

Table 1 summarizes the comparison of EBU-ABK with BU-ABK.

Ⅵ. Conclusion

In this paper, we show the weaknesses of BU-ABK and propose a protocol that enhances BU-ABK with Aura's two hash-based CGA scheme. Although BU-ABK, unlike the CGA method, resists against brute-force attacks searching for hash collisions, it is vulnerable to man-in-the-middle attacks and denial of service attacks. Furthermore, it has heavy burden of managing the public cryptographic parameters. The weaknesses mainly result from there being no way to detect that public cryptographic parameters for ABKs are altered or

forged. BU-ABK thus needs an additional mechanism to securely distribute the parameters. To provide the mechanism, we adopt Aura's two hash-based CGA scheme as a solution. BU-ABK is improved in a way that the public cryptographic parameters is digitally signed with the HA's private key. With the HA's CGA, the CN can validate the public key P_{HA} before verifying the signature. Because the two hash-based CGA scheme increases the cost of brute-force attacks by a factor of 2^{12*Sec} (from 2^{59} to $2^{59+12*Sec}$), EBU-ABK can achieve stronger security than other CGA-based protocols. Also, we use cookies to prevent denial of service attacks. With such cryptographic methods, EBU-ABK can resist against the above attacks, while lightening the burden of managing the parameters. Furthermore, EBU-ABK provides an optimization for mobile nodes with constraint computational power.

The comparison of EBU-ABK with BU-ABK shows that it achieves strong security while not resulting in heavy computation overhead on a mobile node.

References

[1] J. Arkko, "Security Framework for Mobile IPv6 Route Optimization," *IETF, <draft-arkko-mipv6ro-secframework-00.txt>*, Nov. 2001. Work in progress.

[2] R. Deng, J. Zhou, and F. Bao, "Defending Against Redirect attacks in Mobile IP," *Proceedings of the 9th ACM Conference on Computer and Communications Security*, Nov. 2002.

[3] G. O'Shea and M. Roe, "Child-proof authentication for MIPv6 (CAM)," *ACM Computer Communications Review*, Vol. 31, No. 2, April 2001.

[4] M. Roe, T. Aura, G. O'Shea, and J. Arkko, "Authentication of Mobile IPv6 Binding Updates and Acknowledgments," IETF, <draft-roe-mobileip-updateauth-02.txt>, Feb. 2002. Work in progress.

[5] S. Okazaki, A. Desai, C. Gentry and et.el., "Securing MIPv6 Binding Updates Using Address Based Keys (ABKs)," IETF, <draft-okazaki-mobileip-abk-01.txt>, Oct. 2002. Work in progress

[6] G. Montenegro, C. Castelluccia, "SUCV Identifiers and Addresses," IETF, <draft-montenegro-sucv-02.txt>, Nov. 2001. Work in progress.

[7] D. Johnson, C. Perkins and J. Arkko, "Mobility Support in IPv6," IETF, <draft-ietf-mobileip-ipv6-24.txt>, Jun. 2003. Work in progress.

[8] T. Aura, "Cryptographically Generated Addresses (CGA)," IETF, <draft-aura-cga-01.txt>, Aug. 2003. Work in progress.

[9] R. Housley, W. Ford, T. Polk, and D. Solo, "Internet X.509 public key infrastructure certificate and CRL profile," RFC 2459, Jan, 1999.

[10] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, Dec. 1998.

[11] Pekka Nikander, Tuomas Aura, Jari Arkko and Gabriel Montenegro, "Mobile IP version 6 (MIPv6) Route Optimization Security Design," *Proceedings of IEEE Vehicular Technology Conference Fall 2003*, Orlando, FL USA, October 2003. IEEE Press.

저 자 소 개

유 일 선(정회원)

1995년 단국대학교 전산통계학과 학사 졸업

1997년 단국대학교 전산통계학과 석사 졸업

2002년 단국대학교 전산통계학과 박사 졸업

2005년~현재 한국성서대학교 정보과학부 전임강사

<주관심분야 : 인터넷 보안, 접근통제, MIPv6>

최 승 교(정회원)

1982년 단국대학교 전기공학과 학사 졸업

1992년 단국대학교 전산통계학과 석사 졸업

2001년 단국대학교 전산통계학과 박사 졸업

1994년~현재 삼척대학교 컴퓨터공학과 교수

<주관심분야 : 컴퓨터구조, 성능평가, 인터넷보안 컴퓨터 통신>