안전한 모바일 IPv6 바인딩 갱신을 위한 개선된 프로토콜

유 일 선[†] · 원 유 석[††] · 조 경 산[†††]

요 약

모바일 IPv6 환경에서 검증되지 않은 바인딩 갱신은 MN과 CN에게 다양한 보안위협을 유발할 수 있다. 따라서 모바일 IPv6에서 바인딩 갱신 보호는 매우 중요하며 이를 위해 안전한 바인딩 갱신 프로토콜들이 제안되었다. 본 논문에서는 모바일 IPv6 환경에서 안전한 바인딩 갱신을 위한 보안 프로토콜을 제안한다. 제안 프로토콜은 Deng과 Zhou, Bao가 제안한 프로토콜에 Aura의 2중 해쉬 기반의 CGA 기법을 적용함으로써 제 3의 신뢰할 수 있는 CA(Certificate Authority)의 필요성을 제거하였다 [2,9]. 이러한 2중 해쉬 기반의 CGA 기법을 통해 제안 프로토콜은 다른 CGA 기반의 프로토콜보다 강력한 보안성을 지원하며 동시에 기존 연구에서 제안한 프로토콜보다 적은 비용으로 HA(Home Agent)의 공개키 검증을 할 수 있다. 기존 프로토콜 및 CAM-DH, SUCV와의 비교는 제안 프로토콜이 보안성 및 성능과 관리면에서 우수함을 보인다.

An Improved Protocol for the Secure Mobile IPv6 Binding Updates

Il-Sun You[†] · Youseuk Won[††] · Kyungsan Cho[†††]

ABSTRACT

In MIPv6, unauthenticated binding updates expose the involved MN and CN to various security attacks. Thus, protecting the binding update process becomes of paramount importance in the MIPv6, and several secure binding update protocols have been proposed. In this paper, we propose a novel protocol for the secure binding updates in MIPv6, which can resolve the drawbacks of the Deng-Zhou-Bao's protocol [2], by adopting Aura's CGA scheme with two hashes [9]. Aura's scheme enables our protocol to achieve stronger security than other CGA-based protocols without a trusted CA, resulting in less cost of verifying the HA's public key than the Deng-Zhou-Bao's protocol. Through the comparison of our protocol with other protocols such as the Deng-Zhou-Bao's protocol, CAM-DH and SUCV, we show that our protocol can provide better performance and manageability in addition to stronger security than other approaches.

키워드 : 모바일 IPv6(Mobile IPv6), 바인딩 갱신(Binding Update), Cryptographically Generated Address(CGA), Return Routability(RR)

1. 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 but also all other nodes of the Internet to various security threats [1]. The essential requirement to address the security threats is for the CN 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 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 [2-8]. Recently, the Return Routability (RR) protocol has been accepted as the basic technique for securing the BUs. 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(Child-proof Authentication for MIPv6)

※ 이 연구는 2004학년도 단국대학교 대학연구비의 지원으로 연구되었음.
† 정 회 원 : 단국대학교 대학원 전산통계학과
†† 준 회 원 : 단국대학교 대학원
††† 종신회원 : 단국대학교 정보컴퓨터학부 교수
논문접수 : 2003년 11월 3일, 심사완료 : 2004년 6월 11일

protocol, CAM-DH protocol, SUCV(Statistic Uniqueness and Cryptographic Verifiability) protocol and ABKs (Address Based Keys) protocol have been proposed based on public key [2-6, 8]. 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 the novel methods such as Cryptographically Generated Address (CGA) and identity-based cryptosystems. There are two important design considerations in the public key based protocols [2]. The first is the performance since public key cryptographic operations are computationally intensive. It is desirable to minimize the expensive cryptographic operations in mobile devices with constraint computational power, such as PDAs and cellular phones. Among the above public key based approaches, CAM-DH and SUCV provide the option to off-load the expensive cryptographic operation of the MN to its HA. But CAM-DH does not fully remove the expensive cryptographic operations from the MN and SUCV results in the HA's additional cost of managing the MN's private key. The second is the mechanism who generates and possesses the private/public key pair, and how the public key is securely bound with its owner. In the above public key based approaches, the MN generates and possesses the private/public key pair and binds the generated public key with its home address (HoA). However, such a bind is not desirable due to several reasons mentioned in [2]. Alternatively, subnet prefixes for home links, much more traceable and manageable, can be used.

Deng, Zhou and Bao proposed a public key based protocol [2]. Unlike other protocols, their protocol uses the public key certificates (PKC), issued for home links, containing home link subnet prefixes as subject names instead of the public keys bound with the MNs's HoAs. Therefore, their protocol with such PKCs can be much more traceable, manageable and scalable than the above public key based approaches. Moreover, it uses the home agents (HA) as trusted security proxies to off-load the public key cryptographic operations of the MNs to the HAs under the MIPv6's assumption that communication between the MNs and their HAs is protected with pre-established security association. In spite of the above strength, their protocol has a critical limitation [11]. That is, it needs trusted Certification Authorities (CA) to issue the PKCs containing home link subnet prefixes as subject names for home links. Also, the verification of the PKCs is burden to the CNs.

In this paper, we propose a security proxy based protocol for authenticating the BUs, which combines the Deng-Zhou-Bao's protocol with Aura's two hash-based CGA scheme [9] to avoid the use of trusted CAs. That is, in our protocol, the HAs use the addresses derived from the their public keys via the CGA method instead of the PKCs issued by the trusted CAs. Like Deng-Zhou-Bao's protocol, our protocol uses the HAs as the trusted security proxies to minimize the expensive cryptographic operations in the MNs.

The rest of the paper is organized as follows. Section 2 reviews the Deng-Zhou-Bao's protocol. In section 3, we describe the two hash-based CGA scheme and propose a security proxy based protocol for securing the BUs. Section 4 analyzes the proposed protocol. Finally, section 5 draws some conclusions.

2. Review of Deng-Zhou-Bao's protocol

Deng, Zhou and Bao designed their protocol to possess the following features [2]. First, it performs one-way authenticated key-exchange between the MN and the CN where the MN authenticates itself to the CN and the exchanged session key is used to secure the BU messages from the MN to the CN. Second, it employs public key cryptosystems and is secure against powerful adversary who is able to launch both passive and active attacks. Third, it is easy to manage and scalable. Instead of issuing PKCs containing the MNs's HoAs as subject names for the MNs, their scheme issues PKCs containing home link subnet prefixes as subject names for home links. Fourth, no public key cryptographic operations are performed at the MNs. The HAs function as trusted security proxies for the MNs in the protocol. They testify the legitimacy of the MNs's HoAs, facilitate authentication of the MNs to the CNs, and establish shared secret session keys for them.

2.1 Notation

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

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

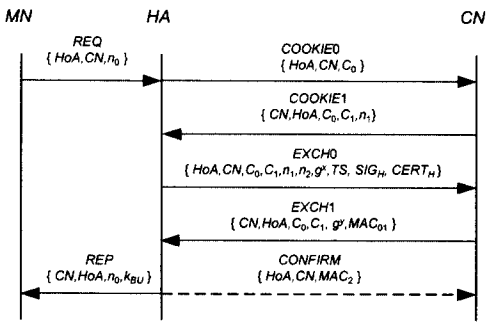
P_X/S_X : a public and private key pair of X.
$S_X(m)$: node X's digital signature on a message m.
$m|n$: concatenation of two messages m and n.

2.2 System Setup

A home link is associated with a public/private key pair P_H and S_H in a digital signature scheme. The private key S_H is kept by a HA in the home link. The home link obtains a PKC, $Cert_H = \{HL, P_H, VI, SIG_{CA}\}$ from a CA, where HL is the home link subnet prefix, VI is the valid duration of the certificate, and SIG_{CA} is CA's signature on HL, P_H and VI. It is assumed that CN's can obtain CA's public key via various means. The protocol also uses the Diffie-Hellman key exchange algorithm to arrive at a mutual secret value between parties of the protocol. Let p and g be the public Diffie-Hellman parameters, where p is a large prime and g is a generator of the multiplicative group Zp^*. To keep notations compact, $g^x \ mod \ p$ is written simply as g^x. It is assumed that the values of p *and* g are agreed upon before hand by all the parties concerned.

2.3 Protocol Operation

The protocol messages exchanged among a MN, its HA and its CN are shown in (Figure 1).



(Figure 1) Deng-Zhou-Bao's protocol

In the protocol, the existence of and operations performed by the *HA* are transparent to both the *MN* and the *CN*. As far as the *MN* is concerned, it sends message *REQ* to and receives *REP* from its *CN*. Similarly, from the *CN*'s point of view, it receives *COOKIE*0, *EXCH*0 and *CONFIRM* from and sends *COOKIE*1 and *EXCH*1 to the *MN*. The use of cookies during the key exchange is a weak form of protection against an intruder who generates a series of request packets, each with a different spoofed source IP address and sends them to a protocol party. For each request, the protocol party will first validate cookies before performing computationally expensive public key cryptographic operations. If the authentication process is successful, the *CN* creates a cache entry for the *MN*' *HoA* and the session key K_{BU}, which will be used for authenticating binding update messages from MN. After that, the *MN* proceeds to send CN BU messages protected using K_{BU} as in the RR protocol.

2.4 Weakness of the Deng-Zhou-Bao's protocol

Because of issuing PKCs containing home link subnet prefixes as subject names for home links, the Deng-Zhou-Bao's protocol is able to be much more manageable and scalable than other public key based approaches. Furthermore, with the PKCs, it can achieve a strong one-way authentication of the MN/HoA to the CN and allow the CN to securely share a secret session key with the MN.

Nevertheless, there is a critical limitation that the protocol should employ trusted CAs to issue the PKCs for home links. It is not feasible solution where no global CA is available. To protect against the man-in-the-middle attack, the CN should validate $Cert_H$ certificate path and revocation status in addition to the signature on $Cert_H$. Obviously, such validation is heavy burden to the CN.

Thus, the Deng-Zhou-Bao's protocol needs to be enhanced to avoid the use of trusted CAs and PKCs issued by them.

3. The Proposed Protocol

In this section, we propose a security proxy based protocol for authenticating the BUs, which combines the Deng-Zhou-Bao's protocol with Aura's two hash-based CGA method to avoid the use of trusted CAs. In our protocol, the HA uses the address derived from its public key via the CGA method instead of the PKC issued by a trusted CA.

3.1 The CGA

CGA is IPv6 address where the interface identifier is generated by hashing the address owner's public key. However, as computers become faster, the 64 bits of the

interface identifier will not be sufficient to prevent attackers from searching for hash collisions.

Recently, Aura proposed a new CGA scheme where two hash values are computed instead of one [9]. 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. The above definition can be stated in terms of the following three bit masks (Mask1, Mask2, Mask3) as shown in (Figure 2).

```
Sec    = Address & 7
Mask1  = 0x00000000000000000000000000000000 if Sec = 0,
         0xfff00000000000000000000000000000 if Sec = 1,
         0xffffff000000000000000000000000 if Sec = 2,
         0xffffffff0000000000000000000000 if Sec = 3
         0xfffffffffff00000000000000000000 if Sec = 4,
         0xffffffffffffff000000000000000000 if Sec = 5,
         0xffffffffffffffffff0000000000000 if Sec = 6, and
         0xffffffffffffffffffffff00000000 if Sec = 7
Mask2  = 0x00000000000000000300000000000000
Mask3  = 0x0000000000000000ffffffffffffffff8

   (Hash1 & Mask3) || Mask2  == Address & Mask3
                Hash2 & Mask1 == 0

where '&' means bit-and operation and '||' means bit-or operation
```

(Figure 2) The definition of a CGA using bit masks

3.2 System Setup

In our protocol, a home link is associated with a public/private key pair P_H and S_H 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_H.

Each CGA can be associated with a self-signed X.509 v3 certificate.

(Figure 3) shows the self-signed X.509 v3 certificate structure, its extension and two 128-bit hash values (Hash1 and Hash2) [9, 10]. As an alternative to the certificate, an optimized parameter format can be used. The optimized format is simply the concatenation of the DER-encoded subjectPublicKeyInfo and CGAParameters data value.

```
Certificate ::= SEQUENCE {
    tbsCertificate       TBSCertificate,
    signatureAlgorithm   AlgorithmIdentifier,
    signature            BIT STRING } -- signature must be verified

TBSCertificate ::= SEQUENCE {
    version         [0]  Version DEFAULT v1,
    serialNumber         CertificateSerialNumber,
    signature            AlgorithmIdentifier,
    issuer               Name,  -- value : home link subnet prefix
    validity             Validity, -- validity must be checked
    subject              Name,  -- value : home link subnet prefix
    subjectPublicKeyInfo SubjectPublicKeyInfo,
                              -- value : address owner's public key

    issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                         -- If present, version shall be v2 or v3
    subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                         -- If present, version shall be v2 or v3
    extensions      [3]  Extensions OPTIONAL
                         -- If present, version shall be v3 -- }

Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension

Extension ::= SEQUENCE {
    extnID     OBJECT IDENTIFIER,        -- value : cgaExtnID =
                                            {1 3 6 1 4 1 311 TBD}
    Critical   BOOLEAN DEFAULT FALSE, --value : false
    extnValue  OCTET STRING }            --value : encoded
                                            CGAParameters

CGAParameters ::= SEQUENCE {
                  modifier       OCTET STRING (SIZE 12),
                  routingPrefix  OCTET STRING (SIZE 8),
                  collisionCount INTEGER (0..2) }

Hash1 = MD5(DER_encode(SubjectPublicKeyInfo) |CGAParameters
        datavalues)
Hash2 = MD5(DER_encode(SubjectPublicKeyInfo) |modifier data
        values)
```

(Figure 3) A Self-Signed X.509 v3 certificate structure for the CGA

The process of obtaining a new CGA is as follows.

① Generate a public/private key pair P_H and S_H for a home link.

② Generate a new CGA via the algorithm presented in (Figure 4).

③ Create and sign a self-signed X.509 v3 certificate, which contains an extension where the extnID has the value cgaExtnID, critical has the value false or true, and the extnValue contains the encoded CGAParameters data value. As an alternative to the certificate, an optimized parameter format can be created.

Like the Deng-Zhou-Bao's protocol, it is assumed that the public Diffie-Hellman parameters p and g are agreed upon before hand by all the parties.

```
/*-------------------------------------------------
// type
SubjectPublicKeyInfo  : an ASN.1 structure of type
                        SubjecctPublicKeyInfo
CGAParameters         : an ASN.1 structure of type
                        CGAParameters
//input
HL : the home link subnet prefix (a 64-bit Routhing Prefix)
PH : a HA's public key
Sec : security parameter Sec, which is an unsigned 3-bit integer
-------------------------------------------------*/

IPv6Addr *generateCGA(IPv6AddrPrefix HL, PublicKey PH,
unsigned int Sec)
{
    SubjectPublicKeyInfo *pkInfo = NULL ;
    CGAParameters *cgaParams = NULL ;
    unsigned char *derPKInfo = NULL, *derCgaParams = NULL ;
    unsigned char *Hash1 = NULL, *Hash2 = NULL ;
    IPv6Addr *newCGA = NULL ;

        // 1. DER-encode a HA's public key as an ASN.1
           structure of the type SubjectPublicKeyInfo

    pkInfo = new SubjectPublicKeyInfo(PH ); CHK_ERR(pkInfo) ;
    derPKInfo = pkInfo → DER_encode() ; CHK_ERR(derPKInfo) ;

        // 2. Create an ASN.1 structure of type CGAParameters
    cgaParams = new CGAParameters() ; CHK_ERR(cgaParams) ;
    cgaParams → modifier = 0 ;
    cgaParams → routingPrefix = HL ;
    cgaParams → collisionCount = 0 ;
    derCgaParams = cgaParams → DER_encode() ;
                   CHK_ERR(derCgaParams) ;

        // 3. Compute Hash2
    while(1) {
        Hash2 = MD5(DER_Concatenate(derPKInfo, DerValue
        (cgaParams → modifier))) ; CHK_ERR(Hash2) ;
                // compare the 12*Sec leftmost bits of Hash2
                   with zero
        if (Is_Leftmost_bits_Zero(Hash2, 12*Sec)) break ;
        cgaParams → modifier++ ;
        if(cgaParams → modifier == Max_Modifier))
                        goto error_handler ;
    }
            // 4. Generate a new CGA
    while(1) {
        Hash1 = MD5(DER_Concatenate(derPKInfo,
        derCgaParams)) ; CHK_ERR(Hash1) ;

                // a new CGA = HL | rightmost 64 bits of Hash1
        newCGA = Make_IPv6_Address(HL, Rightmost_64_bits_of
        (Hash1)) ;
        CHK_ERR(newCGA) ;
        Set_Group_Bit(newCGA, 1) ;
        Set_Universal_Bit (newCGA, 1) ;
        Set_Sec_Bits(newCGA,Sec) ;
        if (!Is_There_Address_Collision(newCGA)) break ;

        cgaParams → collisionCount++ ;
        if(cgaParams → collisionCount > 2) goto error_handler ;
        Free_DER_Value(derCgaParams) ;
        derCgaParams = cgaParams->DER_encode() ;
    }

        ……      // 5. deinitialize values
    return newCGA

error_handler :                 // 6. Handle errors
        ……

    return NULL ;
}
```

(Figure 4) CGA Generation Algorithm

3.3 Protocol Operation

In our protocol, the *HA*s function as security proxies for the *MN*s. They testify the legitimacy of the MN's HoA, facilitate authentication of the MNs to the CNs, and establish shared secret session keys for them. For the appliance of the CGA, our protocol modifies the Deng-Zhou-Bao's one by replacing the $Cert_H$ of *EXCH0* with the self-signed X.509 v3 certificate or the optimized parameter format. Thus, when the *CN* receives the modified *EXCH0*, it should verify the *HA*'s *CGA* using the self-signed X.509 v3 certificate or the optimized parameter format instead of $Cert_H$. The algorithm for verifying the *HA*'s *CGA* is shown in (Figure 5).

```
/*-------------------------------------------------
        // constant
cgaExtnID : { 1 3 6 1 4 1 311 TBD }

        // input
CGA : a HA's address
pkInfo : a HA's public key information
cgaParams : CGAParameters

-------------------------------------------------*/

BOOL VerifyCGA(IPv6Addr* CGA,
               SubjectPublicKeyInfo* pkInfo,
               CGAParameters* cgaParams)
{
           unsigned char *derPKInfo = NULL,
                         *derCgaParams = NULL ;
           unsigned char *Hash1 = NULL, *Hash2 = NULL ;

           unsigned int Sec;

                   // 1. Check input values
           CHK_ERR(CGA) ; CHK_ERR(pkInfo) ;
           CHK_ERR(cgaParams) ;

                   // 2. Compare the group and universal bits
                   //                 in the address to one
           if (!Is_Set_Group_Bit (CGA) ||
             !Is_Set_Universal_Bit (CGA))
                    goto error_handler ;

                   // 3. Get Sec value and
                   // check that the collisionCount value is 0, 1 or 2
           Sec = Sec_Value_of(CGA) ;
           if (cgaParams → collisionCount > 2) goto error_handler ;

                   // 4. Verify the subnet prefixe of the CGA
           if(!Is_Equal_Two_Subnet_Prefixes(
                  Subnet_Prefix_of(CGA),
                  cgaParams → routingPrefix)) goto error_handler ;

                   // 5. Verity the interface identifier of the CGA
           derPKInfo = pkInfo → DER_encode() ;
                       CHK_ERR(derPKInfo) ;
           derCgaParams = cgaParams → DER_encode() ;
                          CHK_ERR(derCgaParams) ;
           Hash1 = MD5(DER_Concatenate(derPKInfo,
                  derCgaParams)) ; CHK_ERR(Hash1) ;

           if(!Verify_Interface_Identifier(
                  Interface_Identifier_of(CGA),
                  Rightmost_64_bits_of(Hash1))) goto error_handler ;
```

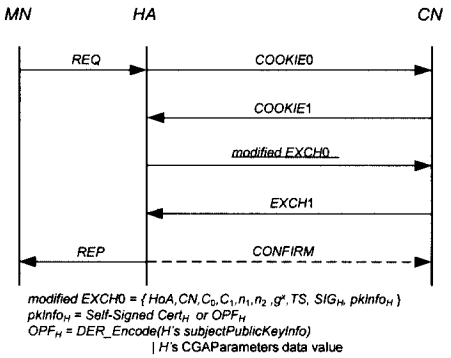
```
        // 6. Verity the Sec of the CGA
    Hash2 = MD5(DER_Concatenate(derPKInfo, DerValue
        (cga  Params →modifier))) ;
    CHK_ERR(Hash2) ;
        // compare the 12*Sec leftmost bits of Hash2
           with zero
    if(!Is_Leftmost_bits_Zero(Hash2, 12*Sec)) goto
      error_handler ;

    ……    // 5. deinitialize values
    return TRUE ;

error_handler :          // 6. Handle errors
            ……
    return FALSE ;
}
```

(Figure 5) CGA Verification Algorithm

The algorithm takes three inputs : the HA's *CGA*, the HA's SubjectPublicKeyInfo data value *subjectPublicKeyInfo* and the HA's CGAParameters data value *cgaParams*. *subjectPublicKeyInfo* and *cgaParams* are retrieved from $pkInfo_h$, which may be the self-signed X.509 v3 certificate or the optimized parameter format. In a case of using the self-signed X.509 v3 certificate, the CN should validate the signature on the certificate besides the HA's CGA. Our protocol is outlined in (Figure 6).



(Figure 6) The Proposed Protocol

4. Analysis

Our protocol is analyzed in terms of security, performance and manageability. Then, it is compared with other protocols such as the Deng-Zhou-Bao's protocol, CAM-DH and SUCV.

4.1 Security

As computers become faster, the 64 bits of the interface identifier will not be sufficient to prevent attackers from searching for hash collisions. Our protocol uses the two hash- based CGA scheme to prevent such brute-force attacks. The scheme 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, our protocol is more secure than other CGA based approaches such as CAM-DH and SUCV, which require the cost of brute-force attacks, $O(2^{62})$.

4.2 Performance

We evaluate the performance of our protocol in terms of the cost of verifying the HA's public key P_H and the public key cryptographic operations that the MN should perform.

<Notation>

$Cost_{Cert\text{-}Signature}$	: the cost of verifying the signature on the certificate
$Cost_{Revocation\text{-}Status}$	: the cost of checking the revocation status of the certificate
$Cost_{Cert\text{-}Path}$	: the cost of verifying the certificate path
$Cost_{2Hash\text{-}CGA}$	: the cost of verifying two hash-based CGA
$Cost_{1Hash\text{-}CGA}$	: the cost of verifying one hash-based CGA
$Cost_{DZB}$	: the cost that the Deng-Zhou-Bao's approach verifies P_H included in $Cert_H$
$Cost_{Self\ Signed\text{-}Cert}$	: the cost that our approach verifies P_H included in a *HA*'s self-signed certificate
$Cost_{OPF}$	: the cost that our approach verifies P_H included in a *HA*'s optimized parameter format
$Cost_{CAM\text{-}DH}$	: the cost that the CAM-DH verifies P_{MN}
$Cost_{SUCV}$	: the cost that the SUCV verifies P_{MN}

<Cost>

$Cost_{DZB}$	=	$Cost_{Cert\text{-}Signature} + Cost_{Revocation\text{-}Status} + Cost_{Cert\ Path}$
$Cost_{Self\ Signed\text{-}Cert}$	=	$Cost_{2Hash\text{-}CGA} + Cost_{Cert\text{-}Signature}$
$Cost_{OPF}$	=	$Cost_{2Hash\text{-}CGA}$
$Cost_{CAM\text{-}DH}$	=	$Cost_{1Hash\text{-}CGA}$
$Cost_{SUCV}$	=	$Cost_{1Hash\text{-}CGA}$
$Cost_{Revocation\text{-}Status}$	+	$Cost_{Cert\text{-}Path} > Cost_{Cert\text{-}Signature} > Cost_{2Hash\text{-}CGA} > Cost_{1Hash\text{-}CGA}$
$Cost_{SUCV}$	=	$Cost_{CAM\text{-}DH} < Cost_{OPF} < Cost_{Slef\text{-}Signed\text{-}Cert} < Cost_{DZB}$

(Figure 7) The Cost of Verifying the HAs (or the MNs) Public Key

(Figure 7) shows the cost of verifying the HA's (or the MN's) public key. Our protocol needs $Cost_{Self\text{-}Signed\text{-}Cert}$ or $Cost_{OPF}$ to verify the HA's public key P_H, which are less than $Cost_{DZB}$. Especially, because $Cost_{2Hash\text{-}CGA} \approx Cost_{1Hash\text{-}CGA}$, our protocol with the optimized parameter format has almost the same cost as the cost of one hash-based ap-

proaches such as $Cost_{CAM-DH}$ and $Cost_{SUCV}$. From the viewpoint of the MN, the MN is allowed to perform no public key cryptographic operations. That is, the security proxy HA performs the expensive cryptographic operations on behalf of the MN. CAM-DH and SUCV provide the option to off-load the expensive cryptographic operation of the MN to its HA. But CAM-DH does not fully remove the expensive cryptographic operations from the MN and SUCV needs for the HA to manage the MN's private key.

4.3 Manageability

Because our protocol needs no trusted CA and allows the HA, instead of the MN, to use the address derived from its public key, it is more manageable and scalable than other protocols.

The comparison of our protocol with other protocols such as the Deng-Zhou-Bao's protocol, CAM-DH and SUCV is summarized in <Table 1>.

As shown in <Table 1>, our protocol can provide good performance and manageability in addition to stronger security than one hash-based CGA protocols.

<Table 1> The comparison of the proposed protocol with other protocols

	Ours	Deng-Zhou-Bao	CAM-DH	SUCV
1	×	○	×	×
2	two hash-based CGA	PKC	one hash-based CGA	one hash-based CGA
3	HA	HA	MN	MN
4	$O(2^{59+12*Sec})$	$O(2^{128})$ or $O(2^{160})$	$O(2^{62})$	$O(2^{62})$
5	$Cost_{Self\ Signed-Cert}$ or $Cost_{OPF}$	$Cost_{DZB}$	$Cost_{CAM\ DH}$	$Cost_{SUCV}$
6	High	High	Low	Low
7	×	×	○	×
8	Diffie-Hellman	Diffie-Hellman	Diffie-Hellman	Diffie-Hellman
9	cookie	cookie	return routability	puzzle

1. Trusted CA
2. Mechanism binding the public key with its owner.
3. Node who generates and manages the private key/public key pair
4. Cost of brute force attacks
5. Cost of verifying the public key
6. Manageability and Scalability
7. Public key cryptographic operations the MN should perform
8. Method that generates and distributes a session key
9. Method that prevents denial of service attacks

5. Conclusion

In this paper, we propose a security proxy based protocol for authenticating the BUs, which combines the Deng-Zhou-Bao's protocol with Aura's two hash-based CGA scheme to avoid the use of trusted CAs. Because the two hash-based CGA scheme increases the cost of brute-force attacks by a factor of 2^{12*Sec} (ie. from 2^{59} to $2^{59+12*Sec}$), our protocol can achieve stronger security than other CGA-based protocols. Moreover, its cost of verifying the HA's public key is less than the one of the Deng-Zhou-Bao's protocol, and with the optimized parameter format, the cost of our protocol is almost the same as that of one hash-based approaches. Also, the security proxy HA allows for the MN to perform no public key cryptographic operations. Because our protocol needs no trusted CA and allows the HA, instead of the MN, to uses the address derived from its public key via the CGA method, it is more manageable and scalable than other protocols.

The comparison of our protocol with other protocols such as the Deng-Zhou-Bao's protocol, CAM-DH and SUCV shows that our protocol can provide good performance and manageability in addition to stronger security than one hash-based CGA protocols.

References

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

[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>, Work in progress, Feb., 2002.

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

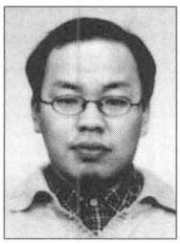
[6] G. Montenegro, C. Castelluccia, "SUCV Identifiers and Addresses," IETF, <draft-montenegro-sucv-02.txt>, Work in progress, Nov., 2001.

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

[8] YouSeuk Won, Kyungsan Cho, "Comparison and Analysis of Protocols for the Secure Binding Updates in MIPv6," the KIPS transactions : part C, Vol.10-C, No.6, Oct., 2003.

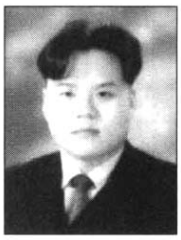
[9] T. Aura, "Cryptographically Generated Addresses (CGA),"

IETF, <draft-aura-cga-00.txt>, Work in progress, Feb., 2003.
[10] R. Housley, W. Ford, T. Polk and D. Solo, "Internet X.509 public key infrastructure certificate and CRL profile," RFC 2459, Jan., 1999.
[11] Ii-Sun You and Kyungsan Cho, "A Security Proxy Based Protocol for Authenticating the Mobile IPv6 Binding Updates," ICCSA 2004, Springer-Verlag LNCS 3043, 2004.



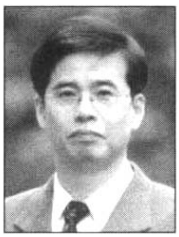
유 일 선

e-mail : qjemfahr@hanmail.net
1995년 단국대학교 전산통계학과(이학사)
1997년 단국대학교 일반대학원 전산통계학과(이학석사)
2002년 단국대학교 일반대학원 전산통계학과(이학박사)
1997년~2000년 (주)한조엔지니어링 연구원
2000년~2004년 (주)인터넷시큐리티 선임연구원
2004년~현재 (주)씬멀티미디어 책임연구원
관심분야 : 침입탐지, 네트워크보안, 사용자 인증 및 접근통제



원 유 석

e-mail : server11@dankook.ac.kr
2000년 단국대학교 전산통계학과 학사
2002년 단국대학교 대학원 전산통계학과 이학석사
2002년~현재 단국대학교 대학원 박사과정
관심분야 : 네트워크 및 이동 통신 보안, 시뮬레이션, 에이전트



조 경 산

e-mail : kscho@dankook.ac.kr
1979년 서울대학교 전자공학과(학사)
1981년 한국과학원 전기 및 전자공학과 (공학석사)
1988년 텍사스 대학교(오스틴) 전기전산공학과(Ph.D.)
1988년~1990년 삼성전자 컴퓨터부문 책임연구원
1990년~현재 단국대학교 정보컴퓨터학부 교수
관심분야 : 컴퓨터 시스템, 네트워크 보안, 성능 분석