# CGA 기반의 HMIPv6 보안 프로토콜 개선

유일선* · 김흥준** · 이진영***

## Improving the CGA-based HMIPv6 Security Protocol

Ilsun You* · Heungjun Kim** · Jinyoung Lee***

요 약

2006년에 Haddad와 Krishnan, Soliman은 HMIPv6의 지역적 바인딩 갱신을 보호하기 위한 표준안으로서 Cryptographically Generated Address 기반의 보안 프로토콜을 제안하였다. 비록 이 프로토콜이 공개키 기반의 강력한 메시지 인증과 바인딩 갱신키 교환을 제공하고 HMIPv6와 유기적인 연동이 가능하지만 Router Solicitation 메시지를 악용한 서비스 거부 공격과 후속 바인딩 갱신 과정에서 악의적인 사용자에 의한 방향전환 공격에 취약한 문제점을 갖는다. 본 논문에서는 이러한 문제점을 개선한다. 또한, 개선된 프로토콜을 보안성과 성능을 중심으로 면밀히 분석하고, 이 두 가지 척도를 종합적으로 고려할 때 기존 프로토콜에 비해 우수하다는 것을 보인다.

ABSTRACT

In 2006, Haddad, Krishnan and Soliman proposed a Cryptographically Generated Address based protocol as a standard for protecting HMIPv6. Though this protocol can provide both the strong message authentication and binding update key negotiation based on the public-key cryptography, it is still vulnerable to several attacks such as denial of service attacks and redirection attacks. This paper improves the problems caused by the protocol. The improved protocol is analyzed in terms of security and performance, and then is shown to be better than the previous one considering the two factors together.

키워드

HMIPv6 Security, CGA, SEND, Localized Mobility Management

## Ⅰ. 서 론

차세대 네트워크에서 이동성 관리를 위한 유력한 표준으로 Mobile IP Version 6 (MIPv6)가 주목받고 있다[1]. MIPv6는 구성 노드(Node)들이 위치와 이동여부에 상관없이 지속적인 통신을 할 수 있도록 지원하며 이를 위해 각각의 이동노드(Mobile Node)에게 두 개의 주소 즉 *HoA*(Home Address)와 *CoA*(Care of Address)를 부여한다. (이 두 주소의 연계를 '바인딩(binding)'이라 함). HoA는 이동노드의 홈에이전트(Home Agent)에서 설정된 영구적인 주소로 이동노드의 식별과 지속적인 접근성을 위해 사용되고 CoA는 이동노드가 외부 네트워크에서 부여 받은 주소로 라우팅을 위해 사용된다. 이동노드는 새로운 네트워크로 이동할 때 마다 홈에이전트와

* 한국성서대학교 정보과학부 조교수
** 진주산업대학교 컴퓨터공학부 부교수(교신저자)
*** 강남대학교 교양학부 부교수

접수일자 2008. 11. 05

대응노드(Correspondi -ng Node)에게 자신의 위치(즉 새롭게 부여받은 CoA)를 알려야 하는데 이를 위해 바인딩 갱신(Binding Update) 절차를 수행해야 한다. 그러나 바인딩 갱신 절차는 이동노드의 핸드오버(Handover) 성능에 과도한 오버헤드를 유발하였고 그 결과 바인딩 갱신의 최적화를 위한 다양한 연구와 표준안이 제안되었다[2-6].

특히, Hierarchical MIPv6(HMIPv6)는 Mobility Anchor Point (*MAP*)라는 새로운 노드를 추가함으로써 바인딩 갱신의 지역화에 의한 성능 개선을 꾀하였다[2]. 이를 위해 HMIPv6의 네트워크는 여러 개의 도메인들로 계층화되며 각 도메인에는 한 개 이상의 *MAP*이 존재한다. *MAP*은 이동노드의 도메인 내 이동을 외부(이동노드의 홈에이전트와 대응노드들)로부터 숨기고 바인딩 갱신 절차를 지역화함으로써 오버헤드를 최소화한다. HMIPv6에서 이동노드의 CoA는 *RCoA*(Regional CoA)와 *LCoA* (On-link CoA)로 세분화 된다. RCoA는 MAP의 Prefix로부터 생성되며 도메인 내에서 이동노드를 식별해 주는 HoA와 같은 역할을 한다. LCoA는 도메인 내에서 이동노드의 위치를 알려주는 역할을 하며 이동노드의 위치가 바뀔 때마다 갱신된다. 홈에이전트와 대응노드는 이동노드의 RCoA를 CoA로 인식하고 이동노드와의 통신을 위해 RCoA를 사용한다. 즉 RCoA가 변경되지 않는 한(이동노드가 도메인을 벗어나지 않는 한) 이동노드가 이동하더라도 바인딩 갱신이 발생하지 않는다. (이 바인딩 갱신은 아래에 언급될 지역적 바인딩 갱신과 다르다) MAP은 RCoA와 LCoA 사이의 패킷 중계를 담당하며 이를 위해 이동노드의 지역적 바인딩 정보 즉 RCoA와 LCoA를 관리한다. 이동노드는 도메인 내에서 새로운 네트워크로 이동할 때마다 MAP과 지역적 바인딩 갱신 절차를 수행함으로써 자신의 새로운 위치(LCoA)를 알린다. 이러한 지역적 바인딩 갱신은 MIPv6의 일반 바인딩 갱신처럼 안전하게 보호되지 않는다면 세션강탈 공격(Session stealing attack), 악의적인 사용자에 의한 방향전환 공격(Redirect attack by a malicious mobile node), 서비스 거부 공격(Denial of Service Attack) 등과 같은 다양한 공격을 초래할 수 있다[4-6].

2006년에 Haddad와 Krishnan, Soliman은 HMIPv6의 지역적 바인딩 갱신을 보호하기 위한 표준안으로서 Cryptographically Generated Add -ress(CGA) 기반의 보안 프로토콜을 제안하였다[7][8] (이하 HKS 프로토콜). HKS 프로토콜은 CGA 기반의 공개키 암호화 기법을 활용하여 강력한 키 교환을 제공하고 기존의 Router Solicitation (RtSol) 메시지와 Router Advertisem -ent (RtAdv) 메시지를 적극 활용함으로써 HMIP -v6와 유기적인 연동을 지원한다. 그러나 이 프로토콜은 RtSol 메시지를 악용한 서비스 거부 공격과 후속 바인딩 갱신에서 악의적인 사용자에 의한 방향전환 공격에 취약한 문제점을 갖는다. 또한, 이동노드가 RtSol 메시지를 보내기 전에 이미 HMIPv6 도메인에 진입한 사실을 알아야 하는 제약이 있는데 이는 이동노드의 핸드오버 성능과 프로토콜 보급에 장애가 될 수 있다.

본 논문에서는 앞서 언급된 HKS 프로토콜의 문제점을 분석한 후에 이에 대한 개선안을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 HKS 프로토콜을 기술하고 문제점을 분석한다. 3장에서는 개선 프로토콜을 제안하고 4장에서는 개선 프로토콜을 보안성과 성능의 측면에서 분석한 후에 HKS 프로토콜과 비교한다. 5장에서는 향후 연구 제시와 함께 결론을 맺는다.

## Ⅱ. HKS 프로토콜

본 장에서는 HKS 프로토콜을 기술한 후, 프로토콜의 취약점을 분석한다.

### 2.1 기호

- *Msg(SA, DA)* : SA에서 DA로 전송되는 메시지
- *MN* : 이동노드, *AR* : Access Router
- *MAP* : Mobility Anchor Point, CN : 대응노드
- *Prefix(X)* : X가 속한 네트워크의 64비트 프리픽스
- *IID(X)* : X의 64비트 Interface Identifier
- *LBU* : 지역적 바인딩 갱신 메시지
- *BA* : 바인딩 응답 메시지
- $PU_X$ : X의 공개키, $PR_X$ : X의 개인키
- *ts* : 타임스탬프
- *E(X, m)* : 메시지 m이 암호화 키 X에 의해 암호화 됨
- *S(X, m)* : 메시지 m이 개인키 X에 전자서명 됨
- | : 결합연산자
- *SHA1(m)* : 메시지 m의 SHA1 해쉬값
- *HMAC_SHA1(k, m)* : k 키를 사용하여 계산된 메시지 m의 HMAC 값 (SHA1 사용됨)

- *First(n,m)* : 메시지 m의 첫 번째 n비트

**2.2** 가정

- 이동노드의 *LCoA*는 SEcure Neighbor Discov
- ery(SEND) 프로토콜[9]을 적용할 수 있도록 CGA 기술에 의해 생성된다.
- 이동노드가 *RtSol* 메시지를 전송하기 전에 HMIPv6 도메인 내에 진입했음을 알 수 있다.
- *MAP* 트리 내에 있는 *AR*들(*MAP* 포함) 사이에는 신뢰관계가 설정되어 있다.

**2.3** 프로토콜 동작

HKS 프로토콜은 크게 초기 과정과 후기 과정으로 나뉜다. 초기 과정은 이동노드가 HMIPv6 도메인에 진입한 후 처음으로 수행하는 지역적 바인딩 갱신 과정으로 이때 후속 바인딩 갱신 과정에서 사용될 바인딩 갱신키 *Kbm*을 교환한다. 후속 과정은 초기 과정이후에 이동노드가 도메인내에서 이동할 때마다 수행하는 지역적 바인딩 갱신과정으로 초기 과정에서 설정된 *Kbm*에 의해 보호된다. HKS 프로토콜의 초기 과정은 그림 1과 같이 4단계로 구성된다.

(1단계) *MN*이 HMIPv6 도메인에 진입한 것을 알게 되면 *MAP*과 *AR*의 Prefix를 요청하는 *RtSol* 메시지를 전자서명 하여 방문 네트워크의 *AR*에게 전송한다. 이때 *RtSol*은 *MN*의 공개키로부터 계산된 64비트의 IIDMN을 포함한다.

*(2단계) AR*가 *RtSol*을 수신하면 먼저 $IID_{MN}$과 CGA 기법을 통해 *MN*의 공개키를 검증하고 메시지에 포함된 전자서명을 검증한다. 전자서명이 유효하면 *AR*은 *LBU*의 보호를 위한 *Ks* 키를 생성하고 그림 1에 있는 수식에 따라 *MN*의 *LCoA*와 *RCoA*를 계산한다. 이후에 *AR*은 *MN*에게는 *RtSol* 메시지를 *MAP*에게는 *PBU* 메시지를 전송한다. 여기서 *RtSol* 메시지는 *AR*과 *MAP*의 Prefix 정보 이외에 *MN*의 공개키로 암호화된 *Ks*를 포함하기 때문에 *MN*은 이 메시지로부터 *LCoA*와 *RCoA*를 계산할 수 있고 아울러 *LBU*를 보호할 수 있다. 또한, *MAP*은 *PBU* 메시지를 통해 *Ks*와 *MN*의 지역적 바인딩 정보 즉 *LCoA*와 *RCoA*를 알 수 있다.

*(3-4단계) MN*이 *RtSol* 메시지를 수신하고 두 개의 *CoA* 즉 *LCoA*와 *RCoA*를 성공적으로 생성하게 되면 곧바로 *MAP*과 *LBU* 메시지와 *BA* 메시지를 교환함으로써 지역적 바인딩 갱신 과정을 수행한다. *MN*과 *MAP*은 *LBU*와 *BA* 메시지의 보호를 위해 *AR*가 보내 준 *Ks*를 이용한다. 그러나 현재 방문 중인 네트워크의 *AR*가 *Ks*를 알기 때문에 이후의 발생할 바인딩 갱신을 위해 계속해서 *Ks*를 적용하는 것은 바람직하지 않다. 따라서 후속 바인딩 갱신의 보호를 위해 *Diffie-Hellman*의 키교환을 적용하여 바인딩 갱신 키 *Kbm*을 교환한다. 즉 *MN*은 *Diffie-Hellman* 개인키 *X*를 생성한 후, 공개키 $g^X$를 계산하여 *LBU* 메시지에 첨부한다. 이와 유사하게 *MAP*은 *Diffie-Hellman* 개인키 *Y*를 생성한 후, 공개키 $g^Y$를 계산하여 *BA* 메시지에 첨부한다. 두 노드는 각각 $SHA1(g^{XY})$를 계산하여 *Kbm*을 공유한다.

초기 과정이후에 이동노드가 도메인내에서 핸드오

$$
\begin{aligned}
&(1)\ MN \rightarrow AR: RtSol\ including\ \{IID_{MN},\ CGAP_{MN},\ S(PR_{MN},\ RtSol)\}\\
&(2a)\ AR \rightarrow MN: RtAdv\ including\ \{E(PU_{MN},\ Ks)\}\\
&(2b)\ AR \rightarrow MAP: PBU\ including\ \{LCoA\ ,RCoA\ ,Ks)\}\\
&\quad -\ LCoA = \mathrm{Prefix}(AR) + IID_{MN}\\
&\quad -\ RCoA = \mathrm{Prefix}(MAP) + First(64,\ SHA1(Ks|LCoA))\\
&(3)\ MN \rightarrow MAP: LBU(LCoA,\ MAP)\\
&\quad -\ LBU\ includes\ \{Seq,\ RCoA,\ g^X,\ HMAC(Ks,\ LBU)\}\\
&(4)\ MAP \rightarrow MN: BA(MAP,\ LCoA)\\
&\quad -\ BA\ includes\ \{Seq,\ g^Y,\ HMAC(Ks,\ BA)\}\\
&\quad -\ Kbm = SHA1(g^{XY})
\end{aligned}
$$

그림 1. HKS 프로토콜
Fig. 1 HKS protocol

버를 하게 되면 이동노드는 바인딩 갱신키 *Kbm*을 사용하여 공개키 암호화 연산없이 효율적으로 바인딩 갱신 과정을 보호한다.

### 2.4 취약점 분석

HKS 프로토콜은 CGA 기법을 활용한 공개키 기반의 강력한 보안을 지원한다. 또한, 기존의 SEND 프로토콜을 적극 활용하기 때문에 HMIPv6 프로토콜과 유기적인 연동이 가능하다는 장점이 있다. 그러나 이 프로토콜은 다음과 같은 문제점을 내포한다.

(1) (1)단계의 *RtSol* 메시지는 nonce나 time stamp를 포함하지 않기 때문에 이 메시지를 악용한 서비스 거부 공격이 가능하다. 예를 들어 공격자는 일정기간 동안 목적 *AR*에게 전송된 유효한 *RtSol* 메시지를 수집하여 이들을 특정 시간에 한꺼번에 재전송하는 서비스 거부 공격을 시도할 수 있다. 특히, 유효한 *RtSol* 메시지는 *AR*로 하여금 *MAP*에게 *PBU*를 전송하도록 유도하기 때문에 공격자는 *AR*와 *MAP*을 동시에 공격하는 효과를 기대할 수 있다.

(2) *MAP*과 *Kbm*을 공유한 유효한 *MN*은 공격대상 네트워크로 이동하지 않고도 이동했다고 위장하는 *LBU* 메시지를 *MAP*에게 전송하여 패킷의 방향전환 공격을 할 수 있다. 비록 *MN*의 입장에서는 자신의 공개키로부터 파생된 *LCoA*를 선택해야 하기 때문에 특정 노드의 주소를 *LCoA*로 선택해서 공격할 수 없지만 특정 네트워크를 대상으로 공격할 수 있다.

(3) *MN*은 새로운 HMIPv6 도메인에 진입할 경우에만 RtSol 메시지를 전자서명해서 보내야 하는데 2.2에서 언급한 것처럼 이것은 *MN*이 미리 HMIPv6 도메인에 진입한 사실을 알아야 하는 조건이 있다. 이 조건은 핸드오버 성능의 저하를 초래하고 동시에 프로토콜 보급에 걸림돌이 될 수 있다.

## Ⅲ. 제안 프로토콜

본 장에서는 앞서 언급된 HKS 프로토콜의 문제점을 개선하는 프로토콜을 제안한다. 제안 프로토콜은 HKS 프로토콜과 유사하게 초기 바인딩 갱신 과정과 후속 바인딩 갱신 과정으로 분리된다.

초기 바인딩 갱신 과정은 그림 2와 같이 3단계로 구성된다:

(1단계) MN이 AR로부터 RtAdv 메시지를 받은 후, 새로운 HMIPv6 도메인에 진입함을 인식하게 되면 메시지에 첨부된 Prefix 정보와 CGA 기법을 통해 자신의 LCoA와 RCoA를 생성한다.

(2-3단계) (1) 단계 이후 MN은 MAP과 함께 LBU 메시지와 BA 메시지를 교환함으로써 지역적 바인딩 갱신을 수행한다. 여기서 LBU 메시지는 MN의 전자서명 S(PRMN, LBU)에 의해 보호되며 MAP에 대한 서비스 거부공격을 예방하기 위해 타임스탬프 ts를 포함한다. AR은 LBU 메시지를 MAP에게 전달하기 전에 이 메시지가 자신의 네트워크로부터 전송되었음을 증명하는 HMAC (Kam, LBU) 값을 첨부한다. 이 HMAC 값은 AR이 MAP과 신뢰관계를 수립할 때 설정한 비밀키 Kam을 통해 생성되기 때문에 MN에 의해서 위조될 수 없고 결과적으로 악의적인 MN에 의한 방향전환 공격을 예방할 수 있

$$
\begin{aligned}
&(1)\ AR \rightarrow MN: RtAdv \\
&(2)\ MN \rightarrow MAP: LBU(LCoA,\ MAP),\ HMAC(Kam,\ LBU) \\
&\quad - LCoA = \mathrm{Prefix}(AR) + IID_{MN} \\
&\quad - RCoA = \mathrm{Prefix}(MAP) + IID_{MN} \\
&\quad - LBU\ includes\ \{Seq,\ ts,\ RCoA,\ CGAP_{MN},\ S(PR_{MN},\ LBU)\} \\
&(3)\ MAP \rightarrow MN: BA(MAP,\ LCoA) \\
&\quad - EKbm = E(PU_{MN},\ Kbm) \\
&\quad - BA\ includes\ \{Seq,\ ts,\ EKbm, CGAP_{MAP},\ S(PR_{MAP},\ BA)\}
\end{aligned}
$$

그림 2. 개선 프로토콜 - 초기 지역적 바인딩 갱신 과정
Fig. 2 Improved protocol for the first local binding update

$(1)\ AR \rightarrow MN: RtAdv$
$(2)\ MN \rightarrow MAP: LBU(LCoA,\ MAP),\ HMAC(Kam,\ LBU)$
$-\ LCoA = \mathrm{Prefix}(AR) + IID_{MN}$
$-\ RCoA = \mathrm{Prefix}(MAP) + IID_{MN}$
$-\ LBU\ includes\ \{Seq,\ RCoA,\ HMAC(Kbm,\ LBU)\}$
$(3)\ MAP \rightarrow MN: BA(MAP,\ LCoA)$
$-\ BA\ includes\ \{Seq,\ HMAC(Kbm, BA)\}$

그림 3. 개선 프로토콜 - 후속 지역적 바인딩 갱신
Fig. 3 Improved protocol for successive local binding updates

다. LBU 메시지가 도착하면 MAP은 먼저 AR이 첨부한 HMAC 값을 검증하고 LBU 메시지에 포함된 ts가 현재 시간을 기준으로 적절한 범위 내에 있는지 확인한다. 또한, CGA 기법을 적용하여 MN의 공개키를 인증한다. 이러한 과정에서 오류가 발견되지 않는다면 MAP은 LBU의 전자서명을 검증함으로써 메시지 인증과정을 마무리 한다. LBU 메시지를 신뢰할 수 있다면 MAP은 MN의 RCoA와 LCoA를 바인딩 캐쉬에 저장하고 후속 바인딩 과정을 위한 비밀키 Kbm을 생성한다. 그리고 Kbm을 MN의 공개키로 암호화 하고 그 결과 값과 LBU의 ts를 첨부한 BA 메시지를 생성한다. BA 메시지는 MAP의 개인키로 전자서명 되어 MN에게 전달된다. MN이 BA 메시지를 수신하면 우선 ts가 유효한지 확인하고 MAP의 전자서명을 검증한다. 이후에 자신의 개인키를 적용하여 후속 바인딩 갱신키 Kbm을 추출한다.

첫 번째 지역적 바인딩 갱신이후에 MN은 그림 3과 같이 HMAC(Kbm, LBU) 값을 적용하여 LBU 메시지를 보호한다. 이와 유사하게 BA 메시지도 HMAC(Kbm, BA) 값에 의해 보호되며 암호화된 Kbm을 포함하지 않는다. 이처럼 후속 지역적 바인딩 갱신 과정은 첫 번째와 달리 공유 바인딩 갱신키 Kbm을 활용하여 효율적으로 보호된다.

## Ⅳ. 분석

본 장에서는 제안 프로토콜의 보안성을 분석하고 제안 프로토콜의 연산 오버헤드를 중심으로 성능을 분석한다.

### 4.1 보안성 분석

- 단말간 보안성 제공: 제안 프로토콜에서 MN은 AR에 의존하지 않고 직접 MAP과 바인딩 갱신 및 키 교환을 하기 때문에 단말간 보안성이 보장된다. 이러한 특성은 AR이 붕괴될 경우 그로 인한 영향을 최소화 할 수 있는 장점이 있다.
- 상호인증: 제안 프로토콜에서는 BA가 MAP이 생성하는 전자서명이나 HMAC 값을 통해 보호되기 때문에 MN은 메시지 인증과 함께 MAP을 인증 할 수 있다. 이러한 특성은 MAP 위장 공격을 예방할 수 있다.
- 서비스 거부 공격: 제안 프로토콜은 RtSol 및 RtAdv 메시지를 위해 별도의 공개키 연산을 요구하지 않기 때문에 이 메시지들을 악용한 서비스 거부 공격에 취약하지 않다. 또한, 후속 바인딩 갱신키 교환시 서버의 공개키 연산이 요구되는 LBU 메시지의 경우 타임스탬프 ts를 포함하고 AR에 의한 HMAC값이 함께 전송되기 때문에 서비스 거부 공격을 방어할 수 있다. 즉 MAP은 공개키 연산 이전에 이 두 값의 유효성을 미리 검증하여 서비스 거부 공격에 대비한다.
- 세션강탈 공격: 이 공격을 위해서는 공격자가 다른 이동노드의 트래픽을 자신에게로 전환하도록 유도하는 거짓 LBU 메시지를 MAP에게 전송해야 한다. 그러나 MN의 개인키 PRMN 혹은 바인딩 갱신키 Kbm을 모른다면 LBU 메시지를 위조할 수 없기 때문에 제안 프로토콜은 세션강탈 공격에 취약하지 않다.
- 악의적인 노드에 의한 방향전환 공격: 제안 프로토콜에서 각각의 MN은 LCoA의 IID를 자신의 공개키로부터 파생된 CGA로 생성하기 때문에 LCoA를 공격대상 노드의 주소로 자유롭게 지정할 수 없다. 그러나 LCoA

의 Prefix는 임의로 지정될 수 있기 때문에 특정 네트워크를 목적으로 하는 방향전환 공격을 방어하기 위해서 LBU 메시지가 LCoA의 네트워크에서 전송되었다는 것이 증명되어야 한다. 이를 위해 제안 프로토콜에서는 AR가 LBU 메시지를 전달하기 전에 MAP과 사전에 공유한 비밀키 Kam을 활용하여 LBU메시지의 HMAC 값을 함께 첨부한다. MAP은 HMAC 값을 통해 LBU 메시지가 LCoA의 네트워크로부터 전송되었다는 것을 확신할 수 있고 결과적으로 악의적인 노드에 의한 방향전환 공격을 무력화 할 수 있다.

### 4.2 성능 분석

본 절에서는 연산 오버헤드를 중심으로 제안 프로토콜의 성능을 분석한다. 표 1과 2는 각각 HSK와 제안 프로토콜의 단계별 연산 오버헤드를 상세하게 기술하고 있다.

표 1. HKS 프로토콜의 연산 오버헤드
Table 1. Computation overhead of HKS protocol

| 단계 | | 메시지 | 연산량 | | |
|---|---|---|---|---|---|
| | | | MN | AR | MAP |
| 초기과정 | (1) | RtSol | S | 2H+V | 0 |
| | (2) | RtAdv PBU | D | E | 0 |
| | (3) | LBU | HM | 0 | HM |
| | (4) | BA | HM+DH | 0 | HM+DH |
| | 총계 | | 2HM+S+D+DH | 2H+V+E | 2HM+DH |
| 후속과정 | (1) | RtSol | 0 | 0 | 0 |
| | (2) | RtAdv | 0 | 0 | 0 |
| | (3) | LBU | HM | 0 | HM |
| | (4) | BA | HM | 0 | HM |
| | 총계 | | 2HM | 0 | 2HM |

* H: hash 연산, HM: HMAC연산, S: 전자서명, V: 서명검증, E: 공개키 암호연산, D: 공개키 복호연산, DH: Diffie-Hellman 키 교환

표 2에서 알 수 있듯이 제안 프로토콜은 초기 과정에서 바인딩 갱신키 교환을 위해 MN과 MAP에게 연산 오버헤드가 큰 공개키 암호화 연산을 요구하지만 후속 과정에서는 설정된 키를 바탕으로 효율적인 바인딩 갱신을 수행한다. HKS 프로토콜과 제안 프로토콜을 비교하면 초기 과정의 경우 AR과 MAP의 연산 오버헤드에서 두 프로토콜 사이에 극명한 차이가 있다. 이러한 이유는 HSK 프로토콜과 대조적으로 제안 프로토콜이 키 교환 과정에서 AR의 개입을 배재하였기 때문이다.

표 2. 제안 프로토콜의 연산 오버헤드
Table 2. Computation overhead of proposed protocol

| 단계 | | 메시지 | 연산량 | | |
|---|---|---|---|---|---|
| | | | MN | AR | MAP |
| 초기과정 | (1) | RtAdv | 0 | 0 | 0 |
| | (2) | LBU | S | HM | 2H+HM+V |
| | (3) | BA | 2H+V+D | 0 | S+E |
| | 총계 | | 2H+S+V+D | HM | 2H+HM+S+V+E |
| | HSK와 차이 | | 2H+V-2HM-DH | S-DH | |
| 후속과정 | (1) | RtAdv | 0 | 0 | 0 |
| | (2) | LBU | HM | HM | 2HM |
| | (3) | BA | HM | 0 | HM |
| | 총계 | | 2HM | HM | 3HM |
| | HSK와 차이 | | 0 | 2HM | |

표 3은 두 프로토콜을 보안성과 성능 측면에서 종합적으로 비교하였다. 표 3에 의하면 제안 프로토콜은 HKS에 비해 보안성을 강화하면서 성능에서 부가적 오버헤드를 유발하였다. 그러나 부가적 오버헤드는 제안 프로토콜의 전체 성능에 경미한 영향을 끼칠 것으로 판단되기 때문에 보안성 강화는 의미 있다고 볼 수 있다.

표 3. 두 프로토콜의 비교
Table 3. Comparison of two protocols

| 항목 | | 제안 프로토콜 | HKS 프로토콜 |
|---|---|---|---|
| 보안성 | ① | O | X |
| | ② | O | X |
| | ③ | O | X |
| | ④ | O | O |
| | ⑤ | O | X |
| 성능 | 초기 | 2H+S+V-2HM-2DH | |
| | 후기 | 2HM | |

* ①단말간 보안성 ②상호인증 ③서비스 거부 공격 ④세션강탈 공격 ⑤악의적인 노드에 의한 방향전환 공격

이처럼 보안성과 성능을 동시에 고려할 때 제안 프로토콜이 HKS 프로토콜보다 우수하다는 것을 알 수 있다. 또한, 성능과 보안성 이외에도 제안 프로토콜은 HKS 프로토콜과 달리 RtSol과 RtAdv 메시지를 확장하지 않았기 때문에 MN이 RtSol 메시지를 보내기 전에 HMIPv6 도메인에 진입하였다는 사실을 알아야만 하는 제약이 없다.

## V. 결 론

본 논문에서는 HMIPv6 프로토콜을 보호하기 위해 제안된 HSK 보안 프로토콜을 개선하였다. 특히, 개선된 프로토콜은 HSK 보안 프로토콜과 달리 초기 과정에서 LBU와 BA 메시지를 통해서 후속 바인딩 갱신키를 교환하고 이들 메시지를 보호하기 위해 CGA 기반의 전자서명을 적용한다. 이러한 특징은 RtSol 메시지를 악용한 서비스 공격을 무력화 시킬 수 있고, 이동노드가 RtAdv 메시지를 받기 전에 미리 HMIPv6 도메인에 진입한 사실을 알아야 하는 제약을 해제하였다. 또한, 제안 프로토콜은 지역적 바인딩 갱신 과정에서 AR의 협력을 통해 악의적인 노드에 의한 방향전환 공격을 방어할 수 있다. 개선 프로토콜은 보안성과 성능을 중심으로 면밀히 분석되었고, 이 두 가지 척도를 종합적으로 고려할 때 개선 프로토콜이 HKS 프로토콜에 비해 우수하다는 것을 알 수 있었다.

향후 연구로는 실제 HMIPv6 환경에서 제안 프로토콜의 구현과 적용을 통해 이동노드의 핸드오버 과정에서 발생하는 구체적인 성능 오버헤드를 분석하는 것이 요구된다.

## 참고문헌


[1] D. Johnson, C. Perkins and J. Arkko, Mobility Support in IPv6," RFC 3775, June 2004

[2] H. Soliman, C. Castelluccia, K. El Malki and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," RFC 4140, August 2005

[3] R. Koodli, "Fast Handovers for Mobile IPv6," RFC 4068, July 2005

[4] J. Arkko, C. Vogt and W. Haddad," Enhanced Route Optimization for Mobile IPv6," RFC 4866, May 2007

[5] 유일선, 김흥준, "MIPv6 최적화 프로토콜 시리즈의 후속 단계 개선 연구," 한국해양정보통신학회 논문지, 2007년 11월

[6] 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

[7] W. Haddad, S. Krishnan and H. Soliman, " Using Cryptographically Generated Address -es (CGA) to secure HMIPv6, " IETF Draft, August 2006 (work in progress)

[8] T. Aura, "Cryptographically Generated Addresses (CGA)," RFC 3972, March 2005

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


## 저자소개


유 일 선(Ilsun You)

1995년 단국대학교 전산통계학과 학사 졸업
1997년 단국대학교 전산통계학과 석사 졸업
2002년 단국대학교 전산통계학과 박사 졸업
2005년~현재 한국성서대학교 정보과학부 조교수
※관심분야 : MIPv6, 인터넷 보안, 접근통제

김흥준(HeungJun Kim)

1989년 단국대학교 전자계산학과 졸업(학사)
1993년 단국대학교 대학원 전산통계학과 (석사)
1999년 단국대학교 대학원 전산통계학과 (박사)
1999년~현재 진주산업대학교 컴퓨터공학부 부교수
※관심분야 : 컴퓨터구성, 모바일 네트워킹, P2P

이진영(JinYoung Lee)

1995년 단국대학교 대학원 전산통계학과 졸업(석사)
1998년 단국대학교 대학원 전산통계학과 수료(박사)
1999년~현재 강남대학교 교양학부 부교수
※관심분야: 이미지프로세스, 생체인식, 침입탐지 시스템