情報保護學會論文誌
第15卷 第5號, 2005. 10

Redirect 공격과 DoS 공격에 안전한 MIPv6 바인딩 업데이트 프로토콜*

강 현 선,†‡ 박 창 섭
단국대학교

MIPv6 Binding Update Protocol Secure Against both Redirect and DoS Attacks*

Hyun-Sun Kang,†‡ Chang-Seop Park
Dankook University

요 약

본 논문에서는 모바일 노드(mobile node, MN)와 대응노드(correspondent node, CN)사이의 바인딩 업데이트 (binding update, BU) 프로토콜에 대해 기존 프로토콜들에서 알려진 Redirect 공격과 DoS(denial-of service) 공격을 방지하고 효율성을 향상시키기 위해 새로운 BU 프로토콜을 제안한다. 제안 프로토콜에서 홈 에이전트(home agent, HA)는 BU 메시지의 유효성을 확인하는 인증서버(authentication server)의 기능과 MN과 CN를 위한 세션키 분배센터의 기능을 수행한다. 또한 CGA(cryptographically generated address)를 기반으로 한 stateless Diffie-Hellman 키 합의 기법을 소개하고, 제안 프로토콜의 안전성을 기존 프로토콜과 함께 비교하고 분석한다. 제안 프로토콜은 메시지 수, 계산적 오버헤드 측면에서 기존기법보다 효율적이며, Redirect와 DoS 공격으로부터 안전하다.

ABSTRACT

We propose a new binding update(BU) protocol between mobile node(MN) and correspondent node(CN) for the purpose of preventing redirect attacks and DoS attacks observed from the existing BU protocols and enhancing the efficiency of the BU protocol. Home agent plays a role of both authentication server validating BU message and session key distribution center for MN and CN. Also propose the stateless Diffie-Hellman key agreement based on cryptographically generated address (CGA). Security of our proposed protocol is analyzed and compared with other protocols. The proposed protocol is more efficient than previous schemes in terms of the number of message flows and computation overhead and is secure against both redirect and DoS attacks.

Keywords : *MIPv6, Binding update protocol, CGA, Diffie-Hellman key agreement*

Ⅰ. 서 론

Mobile IPv6(MIPv6)[1]는 모바일 노드(MN)의 자유로운 이동성을 지원하기 위한 프로토콜이다. 즉, MN이 인터넷상의 한 지점에서 다른 지점으로의 이동시 전송계층 연결(transport connection)의 중단없이 자유롭게 이동하기 위한 목적으로 제안되었다. MIPv6에서는 MN에 대해 두 가지 유형의 IPv6 주소가 정의된다. 그 중 하나는 MN의 홈 네트워크(home network)에서 정의된 고정된 주소로 사용되는 HoA(home address)이고, 다른 하나는 MN이 외부 네트워크(foreign network)로

접수일 : 2005년 7월 27일 ; 채택일 : 2005년 10월 12일
* 이 연구는 2004학년도 단국대학교 대학연구비의 지원으로 연구되었음.
† 주저자, ‡ 교신저자, sshskang@dankook.ac.kr

이동했을 경우 외부 네트워크에서 동적으로 할당 받게 되는 CoA(care-of address)이다. MN이 외부 네트워크로 이동했을 경우, HoA를 목적지로 하는 패킷을 계속 전달 받기 위해서 MN은 HA(home agent)에게 BU 메시지를 보냄으로써 새로운 CoA를 알리는 홈 등록(home registration)을 수행해야 한다. HA는 성공적인 BU를 통해서 MN에 대한 HoA, CoA 등의 바인딩 정보를 갱신하고, 이후의 HoA로 향하는 패킷을 MN에게 터널링 한다.

만약 대응노드(CN)가 MN에게 패킷을 보내기를 원할 때, CN은 MN의 현재 CoA를 모르기 때문에 우선은 MN의 HoA로 패킷을 전송한다. MN의 HA는 새로운 CoA로 해당 패킷을 전달하고, MN은 CN에게 직접적으로 응답한다. MIPv6에 도입된 route optimization mechanism은 이러한 삼각 라우팅 문제(triangular routing problem)를 해결하기 위해 사용될 수 있다. MN으로부터 BU 메시지를 수신 후, CN은 MN의 HoA, CoA를 바인딩 캐쉬(binding cache)에 유지하고, MN으로 직접 패킷을 보낼 수 있게 된다. 이때, 만약 BU 메시지가 전혀 인증되지 않는다면, 몇몇의 Redirect와 DoS(denial-of service) 공격이 가능하게 된다. CN이 MN과 통신 중이라고 가정하자. 공격자는 MN의 패킷을 자신에게 redirect하기 위해, MN의 HoA와 자신의 주소를 CoA에 담아 가짜의 BU 메시지를 보낸다. 또한, 공격자는 공격대상 호스트나 해당 호스트가 속한 서브넷에 대해 Flooding 공격을 하기 위해 MN의 패킷을 임의의 다른 공격대상 호스트에게 redirect 할 수도 있다. 이 밖에도, 인증되지 않은 BU 메시지에 대해 발생할 수 있는 몇몇의 잠재적인 공격[2-4]이 존재하며, 따라서 BU 메시지는 반드시 보호되어야 한다.

MN과 CN 사이의 BU 프로토콜을 보호하기 위해 제안된 대부분의 인증 메커니즘은 PKI(public key infrastructure)나 KDC(key distribution center)와 같은 security infrastructures를 기반으로 하지 않는다. 따라서 인증 메커니즘에서 중요시되는 부분은 MN과 CN 사이의 SA(security association)를 설정하는 방법이다.

MIPv6[1]에 포함된 RR(return routability) 프로시저와 CGA(cryptographically generated address)[5,6]는 MN과 CN 사이에 교환되는 BU 메시지를 보호하기 위해 사용된다. 하지만, 해당 메커니즘 역시 완전히 안전하거나 효율적이지 못하다. 본 논문에서는 기존의 BU 프로토콜에서 알려진 보안상의 문제를 완화하고 BU 프로토콜의 효율성을 향상시킨 MN과 CN 사이의 새로운 BU 프로토콜을 제안한다. 2장에서는 본 논문과 관련한 기존연구와 문제점을 소개하고, 3장과 4장에서는 각각 새롭게 제안하는 안전하고 효율적인 BU 프로토콜을 제안하고 분석한다. 기존 프로토콜과의 비교분석은 5장에서 소개되며, 마지막으로 6장의 결론으로 본 논문을 마치게 된다.

II. 기존연구와 문제점

MN과 CN의 안전한 BU 프로토콜에서 중요한 부분은 MN이 보낸 HoA, CoA을 CN이 확인하는 메커니즘이며, 해당 메커니즘은 Redirect와 DoS 공격 등 다양한 유형의 공격에 대해 대응할 수 있도록 설계되어야 한다. 이 장에서는, BU 메시지를 보호하기 위해 제안된 기존 기법들을 분석하고, 발견된 취약성과 문제점에 대해 알아본다.

2.1 Return Routability Procedure

RR 프로시저[1]는 MIPv6에 포함된 BU 프로토콜의 안전성을 향상시키기 위한 프로토콜이다. RR 프로시저의 주된 목적은 MN과 CN 사이의 일종의 세션키인 "binding management key (Kbm)"를 설정하는 것이며, 해당 세션키는 차후 BU(binding update)/BA(binding acknowledgement) 메시지를 보호하는데 사용된다. 이를 위해, CN은 두 개의 키 재료(keying materials) kh, kc를 생성하여, MN의 HoA와 CoA를 통해 각각 전송한다. 만약 실제로 HoA와 CoA가 MN이 소유한 주소라면 MN은 두 개의 키 재료를 수신할 수 있고, 이를 기반으로 세션키 $Kbm = h(kh, kc)$를 계산할 수 있게 된다. 위의 $h()$는 일 방향 해쉬함수(one-way hash function)를 나타낸다. 하지만, RR 프로시저에는 보안상의 문제점이 존재한다. RR 프로시저는 HoA와 CoA 사이의 강력한 바인딩을 제공하지 않는다. MN_1과 CN이 HoA_1, CoA_1에 대해 kh_1, kc_1를 교환하였고, MN_2와 CN이 HoA_2, CoA_2에 대해 kh_2와 kc_2를 교환하였다고 가정하자. 키 재료를 전송하는 두 메시지는 평문의 형태로 전송되기 때문에, 키 재료는 쉽게 도청될 수 있고, 공격자는 유효한 세션키 $Kbm' = h(kh_1|kc_2)$를 얻을 수 있

게 된다. 또한, RR 프로시저는 한번의 BU 프로토콜의 실행을 위해 8개의 메시지가 필요하므로 메시지 수 측면에서도 비효율적이다.

2.2 CGA-based BU 프로토콜

CGA는 MN의 IPv6 주소와 공개키를 바인딩하기 위한 개념으로, IPv6 주소의 64비트 인터페이스 식별자(interface identifier, IID)를 생성하는데 사용된다. MN의 공개키 PK_{MN}와 서명용 개인키 SK_{MN}이 주어졌을 때, HoA의 IID는 h(subnet prefix of HoA, PK_{MN}) 계산으로 생성된다. CGA 생성과 관련한 프로세스는 [7]에 상세히 설명되어 있다. CGA-based BU프로토콜[5]에서 MN은 CN에게 BU 메시지 (CoA, CN, HoA, … , PK_{MN}, $Sig(SK_{MN})$)를 보낸다. 앞에서 밑줄 친 두 필드는 소스(source)/목적지(destination) 주소를 나타내고, $Sig(SK_{MN})$는 MN의 서명용 개인키 SK_{MN}를 사용하여 생성한 전자서명을 나타낸다. CN은 공개키 PK_{MN}를 이용하여 서명을 확인한 후, HoA의 IID가 공개키로부터 유도되는지를 검사한다. 해당 프로토콜에서 CN은 계산비용이 큰 서명확인 계산을 수행하기 때문에, 공격자는 CN에게 BU 메시지를 연속적으로 보냄으로써 CN에 대한 DoS 공격을 발생시킬 수 있다.

기본 SUCV 프로토콜[6]은 CGA-based BU 프로토콜로서, CN에 대한 DoS 공격을 막기 위해 클라이언트 퍼즐(Client Puzzle)[12] 개념을 사용한다. 하지만 퍼즐을 풀기위해 소요되는 시간은 BU 프로토콜의 효율성 측면에 바람직하지 못한 영향을 준다. 기본 SUCV 프로토콜은 CN과 MN 사이의 공통키 공유를 위해 Diffie-Hellman(D-H) 키 합의를 사용한다. 해당 프로토콜에서 g는 Z_p의 생성자이고, p는 임의의 큰 소수라 할 때, sucvP2 메시지 내의 D-H 공개값 $g^y \pmod p$는 전혀 보호되지 않으므로, man-in-the-middle(MITM) 공격이 가능하다. 이 문제에 대한 하나의 해결책으로는 CN이 sucvP2 메시지에 CGA와 서명을 사용하는 것이다. 하지만, MN이 CN에게 위조된 sucvP1 메시지를 연속적으로 보냄으로써, sucvP2 메시지의 서명을 위해 연속적인 공개키 계산을 수행해야 하는 CN에 대한 또 다른 DoS 공격을 발생시킨다. 기본적으로, CGA-based BU 프로토콜[5,6]은 MN이 각 BU 메시지마다 전자서명을 생성해야 하기 때문에, 하드웨어에 제약적인 MN에게는 계산적인 부담이 있다.

2.3 Security Proxy based BU 프로토콜

안전한 BU 프로토콜을 위한 [6,8,9]에서 HA는 하드웨어에 제약적인 MN의 공개키 계산의 부담을 덜어주기 위한 목적으로 security proxy 역할을 수행한다. MN이 HA에게 Request 메시지를 보내면 HA는 CN과 authenticated D-H 키 합의 프로토콜을 수행함으로써 세션키를 생성하며, 생성된 세션키는 Response 메시지를 통해 안전하게 MN에게 전송된다. 해당 세션키는 차후 MN과 CN 사이의 안전한 BU/BA 메시지를 위해 사용된다. [8]에서는 authenticated D-H 키 합의를 위해 전세계적인 PKI를 기반으로 하며, 확장된 SUCV[6]와 [8]의 변형기법으로 제안된 [9]에서는 CGA 기반의 전자서명을 사용한다. 하지만, [9]의 BU 프로토콜은 치명적인 보안상의 문제점을 가지고 있다. 즉, HA는 자신이 직접 생성한 공개키/개인키 쌍을 이용하여 CGA를 생성하고, self-signed 공개키 인증서를 생성하며, 해당 인증서는 HA와 CN 사이의 authenticated D-H 키 합의에 사용한다. 그러나, 공개키/개인키 쌍은 어떤 노드든지 생성할 수 있으며, 인증을 위한 인증서는 self-signed 공개키 인증서이므로, MN과 CN에는 다양한 공격이 가능하게 된다.

Ⅲ. 제안 프로토콜

3.1 설계원리

제안 프로토콜은 HA가 이미 MN의 홈 등록을 통해 제공된 최신의 올바른 바인딩 정보를 유지하고 있음을 가정한다. 따라서 MN에 의한 Redirect 또는 DoS 공격은 HA를 통해 추적이 가능하고, 이를 통해 서비스를 중단할 수 있기 때문에, 홈 등록동안에는 MN은 올바르지 않은 CoA를 등록하지 않음을 가정한다. 이와 같은 가정은 [10,pp.158]에서도 언급된 바 있다.

제안 프로토콜에는 CGA 개념이 사용된다. 하지만, BU 메시지의 서명에 CGA를 사용하는 기존기법[5,6]과는 달리 제안 프로토콜에서 CGA는 HA와 CN 사이의 authenticated D-H 키 합의를 목적

으로 사용된다. p를 큰 소수라 하고, g를 그룹 Z_p의 생성자라 할 때, HA는 다음과 같이 MN에 대한 HoA를 생성한다. HA는 임의의 수 y를 선택하고 $g^y \bmod p$를 계산한다. 이후부터, "mod p"는 표기의 단순성을 위해 생략하기로 한다. MN의 HoA의 IID는 h(MN's subnet prefix, g^y)로 계산된다. 초기 서비스 등록 시, HoA는 MN에게 할당되고 D-H 비밀값 y는 MN에게 공개하지 않는다. 같은 방식으로, CN은 한 쌍의 x와 g^x를 생성하고, CN의 IID는 h(CN's subnet prefix, g^x)와 같이 계산된다. 이와 같은 접근으로 서명의 생성, 확인과 같은 공개키 계산을 생략시킬 수 있다.

제안 프로토콜에서 HA는 인증서버와 키 분배센터의 역할을 수행한다. 첫째, MN이 BU 메시지를 CN에게 보내면, CN은 MN에 대한 인증자 (authenticator)가 된다. CN은 MN을 인증하기 위한 어떠한 정보도 가지고 있지 않으므로, HA에게 메시지의 인증을 요청하는데, 이는 HA가 CN에 대한 인증 서버가 됨을 의미한다. 즉, CN은 HA를 통해 BU 메시지의 유효성을 검사한다. 둘째, HA는 MN과 HA의 정적인 SA와 CN과 HA의 동적으로 생성되는 SA를 기반으로, MN과 CN이 사용하게 될 세션키를 분배한다. 제안 프로토콜에서는 HA와 MN 사이에는 미리 설정된 SA가 존재함을 가정으로 하고, 이것은 또한 HA와 MN 사이의 안전한 홈 등록에서도 사용된다.[11] 반면, CN은 HA와 CGA 기반 D-H 키 합의를 사용하여 동적으로 SA를 설정하며, 이를 통해 HA는 MN과 CN 사이의 새로운 세션키 공유를 돕는다. 다음의 설명에서 |는 연결 연산자를 의미하고 ⊕는 XOR 연산자를 의미한다. $H()$는 키를 이용한 해쉬함수를 의미하고 $MAC(K)$은 모든 선행하는 메시지에 대해 키 K를 이용하여 계산한 MAC 값을 의미한다. 간단한 예로 만약 A, B, $MAC(K)$ 형태의 메시지가 있을 때 해당 메시지는 A, B 그리고 선행하는 전체 메시지 (A, B)의 키 K를 이용하여 계산한 MAC 값으로 구성된 메시지를 의미한다. 또한 노드의 개체이름은 해당 개체의 IPv6 주소를 나타낸다.

3.2 CGA 기반 Stateless D-H 키 합의

이번 절에서는 제안 프로토콜의 중요한 개념인 stateless D-H 키 합의 기법을 소개한다. 다음의 시나리오를 D-H 키 합의와 관련하여 생각해 보자.

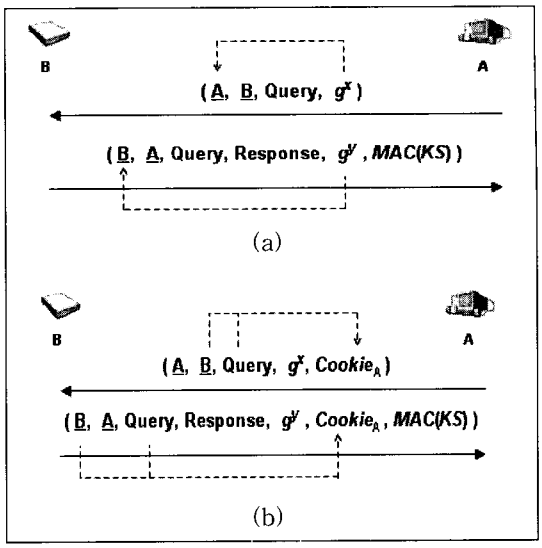

그림 1. stateful, stateless CGA기반 D-H 키합의

그림 1-(a)에서 노드 A는 노드 B에게 Query 메시지를 보낸다. 여기서 A와 B의 IPv6 주소는 각각 D-H 공개값 g^x와 g^y로부터 생성된 CGA이다. 다음으로, 노드 B는 한 쌍의 Query, Response와 $KS=h(g^{xy})$를 이용하여 계산한 $MAC(KS)$으로 구성된 메시지를 노드 A에게 보낸다. 요점의 간단한 설명을 위해, 그림 1-(a)의 Query-Response 프로토콜에서의 재생공격의 가능성은 배제하고, 해당 프로토콜은 A와 B 사이에 단 한번만 수행됨을 가정으로 한다. 노드 A가 자신이 보낸 Query 메시지와 관련한 상태정보를 유지한다고 가정하면 두 노드 사이에는 stateful D-H 키 합의가 수행된다. Query 메시지를 노드 B에게 보낸 후, 노드 A는 소스 주소가 B인 Response 메시지를 수신하기를 기다리게 된다. 만약 공격자가 Response를 수정하고 그에 상응한 $MAC(KS)$을 생성하기를 원한다고 가정하자. 노드 A와 B의 경로상의 공격자 C는 $h()$의 일방향의 속성에 의해 B를 생성한 g^z를 찾는 것이 불가능하며, 소스 주소 B를 바꾸지 못하면 Response 메시지의 g^y를 공격자의 g^z로 대체할 수 없게 된다.

반면, 노드 A가 자신이 보낸 Query 메시지와 관련한 어떠한 상태정보도 유지하지 않는다고 가정하면 stateless D-H 키 합의가 수행된다. 노드 A는 Query 메시지를 누구에게 보냈는지 저장하고 있지 않기 때문에, 다음과 같이 성공적인 MITM 공격이 발생할 수 있다. 공격자 C는 (A, B, Query, g^x)를 (C, B, Query, g^z)로 대체하고 노드 B와

$KS_{BC} = h(g^{yz})$를 공유할 수 있다. 그 결과, 공격자는 노드 B가 보낸 Response 메시지 (B, C, Query, Response, g^y, $MAC(KS_{BC})$)를 다음과 같이 수정할 수 있다.

(C, A, Query, Response′, g^z, $MAC(KS_{CA})$)

즉, 공격자 C는 $KS_{CA} = h(g^{zx})$를 생성한 후 수정된 메시지를 A에게 보내며, 노드 A는 C가 g^z로부터 유도되었고 $MAC(KS_{CA})$이 유효하면 해당 메시지를 수락하게 된다. 이와 같은 stateless D-H 키 합의의 문제점 해결을 위해, 그림 1-(b)와 같이 노드 A에 의해 생성된 $Cookie_A = H(K_A, B \mid Query)$를 도입한다. 앞에서 K_A는 노드 A의 비밀키이다. 해당 쿠키가 B와 Query에 대한 정보를 포함하기 때문에 노드 A는 Response 메시지의 소스 주소를 검사할 수 있고, 이를 통해 위의 MITM 공격이 불가능하게 된다.

3.3 제안 프로토콜

제안 프로토콜에서 HA는 HoA, CoA, K_{HM}, (y, g^y), SN_{HA}, LT_{HA}로 구성된 MN에 대한 바인딩 캐쉬 목록을 유지함을 가정하며, 보안측면에서 MN과 HA를 하나의 개체로 간주한다. K_{HM}은 MN과 HA 사이의 미리 공유된 대칭키를, g^y는 MN의 CGA 계산에 사용된 D-H 공개값을, SN_{HA}는 일련번호를, LT_{HA}는 lifetime을 나타낸다. 다음의 그림 2는 안전한 BU를 위해 MN, CN, HA 사이에 교환되는 일련의 메시지를 나타낸다.

① MN → CN : BU message

MN은 HoA, CoA, BuP로 구성된 메시지를 작성하여 작성한 전체 메시지에 세션키 KS_1을 이용하여 계산한 $MAC(KS_1)$과 함께 CN으로 전송한다. BuP는 SN_{CN}와 LT_{CN}으로 구성된 바인딩 업데이트 파라미터를 의미하며, BuP에 포함된 SN_{CN}은 MN과 CN 사이의 최초의 BU 수행 시 MN에 의해 선택된 임의의 수로 시작한다.

② CN → HA : BU_Auth_Request message

CN은 직접 *BU* message를 인증할 수 없기 때문에, *BU_Auth_Request* message를 작성하여 대신 *BU* message를 인증할 HA에게 전송한다. 이때, HA와 CN 사이에 stateless D-H 키 합의는 수행되며, CN은 MN에게서 수신한 메시지 또는 HA에게 송신한 메시지에 대한 어떠한 상태정보도 유지하지 않는다. 반면, CN은 미리 설정된 n개의 목록으로 구성된 넌스배열 *ArrayN*을 유지하는데, *ArrayN*의 각 목록은 index, nonce, flag로 구성된다.

BU_Auth_Request message 생성 시, CN은 flag가 0으로 설정된 index j를 선택하고, 배열로부터 상응하는 nonce N_j를 이용하여 DoS 공격을 막기 위한 $Cookie_{CN} = H(K_{cn}, HoA|CoA|BuP|N_j)$를 계산하고, flag는 1로 설정한다. 만약 flag가 0으로 설정된 목록이 없다면, 추가적인 *BU* message는 드롭된다. K_{cn}은 CN의 노드키를 나타낸다.

BU_Auth_Response message가 HA로부터 도착되면, CN은 index j와 상응하는 flag를 0으로 설정하고, 대응하는 nonce는 새로운 값으로 갱신한다. 기본적으로, CN은 동시에 처리 가능한 *BU* message의 최대수로 n을 설정하고, 서로 다른 넌스를 할당한다.

BU_Auth_Request message를 수신한 HA는 우선 메시지에 포함된 HoA, CoA가 바인딩 캐쉬 내의 값과 일치하는지 검사한다. 그 다음, HA는 MN과 같은 방법으로 KS_1을 계산하고 $MAC(KS_1)$ 값이 유효한지 검사한다. 또한 g^x로부터 CN의 IID가 유도될 수 있는지 확인한다. 만약 위의 테스트 중 하나라도 실패하게 되면, HA는 메시지를 드롭하고 그렇지 않으면, 메시지에 포함된 g^x와 바인딩 캐쉬에 포함된 MN의 y를 기반으로 D-H 키 합의를 수행하고, $KS_2 = H(g^{xy}, SN_{CN})$를 계산한다.

BU_Auth_Request message 확인 후, HA는 D-H 키 합의와 MN과 CN 사이에 공유해야 할 세션키 정보를 포함한 *BU_Auth_Response* message를 작성하여 CN에게 전송한다.

③ HA → CN : BU_Auth_Response message

HA로부터 *BU_Auth_Response* message를 수신한 CN은 메시지에 포함된 CoA, HoA, BuP가 $Cookie_{CN}$ 생성에 사용된 값과 일치하는지 검사한다. 만약 일치하지 않다면, 해당 메시지를 드롭하고 그렇지 않으면, CN은 g^y를 기반으로 stateless D-H 키 합의를 수행한다. 그리고 MN의 IID가 g^y로부터 유도되는지 여부를 확인한 후, $KS_1 \oplus KS_2$로부터 KS_1을 계산함으로써 MN과 CN 사이의 공

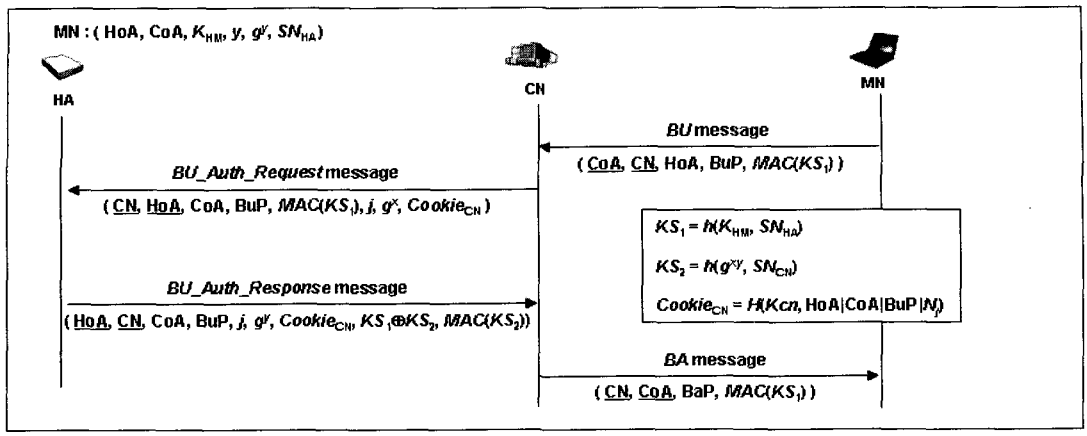

그림 2. 제안 프로토콜

통 세션키 KS_1을 공유한다. 마지막으로, CN은 $MAC(KS_2)$ 값을 확인하고, 만약 인증에 성공하면 메시지 내의 index j와 대응하는 목록의 flag를 0으로 설정하고, nonce를 새로운 값으로 갱신한다. 그렇지 않은 경우, CN은 메시지를 드롭한다.

④ CN → MN : BA message

만약 *BU_Auth_Response* message가 유효하면, CN은 MN에 대한 바인딩 캐쉬를 생성 또는 갱신한 후, MN에게 *BA* message를 전송한다. 해당 메시지는 $MAC(KS_1)$을 기반으로 인증된다. BaP는 BuP와 같거나 CN에 의해 수정된 값을 포함하는 binding acknowledgement parameter를 나타낸다.

Ⅳ. 안전성 분석

이 장에는 안전한 BU 프로토콜을 위한 설계과정에서 발견된 다양한 위협과 공격[2,4]에 대한 제안 프로토콜의 안전성을 분석한다.

4.1 CGA-based Diffie-Hellman 키 합의

평문 형태의 D-H 키 합의 메시지가 MITM 공격에 대해 안전하기 위해서는 반드시 인증되어야 한다. 하지만, 3.2절에서 언급했던 바와 같이 CGA 기반 stateless D-H 키 합의 자체로는 MITM 공격에 대해 안전할 수 없으며, 따라서 제안 프로토콜에서는 MAC과 Cookie가 사용되었다. CN과 HA 경로상의 공격자의 주목적은 *BU_Auth_Response* 메시지를 수정함으로써, CN이 바인딩 캐쉬에 잘못된 HoA′, CoA′, BuP′를 저장하도록 하는 것이다. 우선, 공격자는 D-H값 g^y를 임의로 선택한 g^z로 대체하고, g^z와 일 방향 해쉬함수의 계산으로 HoA′를 생성한다.

BU_Auth_Response′ : (HoA′, CN, CoA′, BuP′, j, g^z, $Cookie_{CN}$′, $R \oplus KS_2$′, $MAC(KS_2')$), $KS_2' = H(g^{xz}, SN_{CN})$

여기서 g^z에 의해 HoA′가 결정되므로 공격자는 특정한 HoA′를 미리 예측하지 못하고, 공격자는 수정한 메시지에 대해 유효한 $MAC(KS_2')$를 계산하기 위해서는 KS_2'를 구해야 한다. 하지만, 공격에 성공하기 위해서는 해결해야 할 두 가지 문제가 있다. 첫째, 공격자는 CN의 비밀 노드키 K_{CN}을 모르기 때문에 유효한 $Cookie_{CN}' = H(K_{CN}, HoA'|CoA'|BuP'|N_j)$를 생성하는 것은 불가능하다. 또한, 공격자는 $KS_1 \oplus KS_2$ 로부터 KS_1을 구별할 수도 없기 때문에 공격자는 $R \oplus KS_2'$는 생성할 수 있지만, $KS_1 \oplus KS_2'$는 생성하지 못한다. 앞에서 R은 공격자에 의해 임의로 선택된 수를 나타낸다. 결국, CN이 KS_1 대신에 R을 구했기 때문에 공격자가 $Cookie_{CN}'$을 계산하더라도 *BA* message는 수락되지 않는다.

4.2 Redirect와 Flooding 공격

보통 BU 프로토콜에 대한 Redirect 공격에는 두 가지 유형이 있다. 첫 번째 유형은(Type 1 Re-

direct Attack), 공격자가 MN으로 향하는 메시지를 다른 노드로 redirect 하기 위한 목적으로, 부당하게 다른 MN의 HoA를 사용하는 것이다. 공격자가 CN과 통신 중인 특정 MN의 HoA를 안다고 가정하자. 만약 공격자가 공격자의 CoA와 MN의 HoA를 포함한 성공적인 BU 메시지를 CN에게 보낼 수 있다면, 공격자는 MN과 CN의 연결을 가로채기 할 수 있다. 같은 방식으로, 네트워크 내의 임의의 호스트에게 redirect 메시지를 이용하여 flooding 공격을 성공시킬 수 있다. 하지만, 제안 프로토콜에서 공격자는 HA를 통해 *BU* message를 인증받을 수 있는 기반이 되는 K_{HM}이나 KS_1을 가지고 있지 않기 때문에 앞에서 설명한 redirect 공격은 불가능하게 된다. 두 키는 오직 특정 HoA를 가진 정당한 MN과 해당 HA만이 알고 있으며, 공격자의 CoA는 HA의 바인딩 캐쉬에 존재하지 않기 때문에, 위조된 *BU* message는 HA에서 처리되지 않는다. 특히, 공격자는 MN의 HoA와 관련한 한 쌍의 올바른 D-H 공개값/비밀값을 모르기 때문에 해당 공격은 불가능 하게 된다.

두 번째 유형은(Type 2 Redirect Attack) 부정한 MN이 자신의 HoA를 이용하여 비정상적인 메시지를 보내는 공격이다. 즉, CN으로부터 대용량의 파일을 다운로드 받기를 시작한 후, 다른 공격대상 호스트에게 redirect를 하기위해 잘못된 *BU* message를 CN에게 보내는 것이다. 기존에 제안된 대부분의 BU 프로토콜[1,5,6,9]은 이러한 공격에 대응하지 못한다. 하지만, 본 논문에서 제안한 프로토콜은 해당 공격에 대해 대응 또는 완화시킬 수 있다. 앞서 HA는 부정한 MN에 대한 최신의 올바른 HoA와 CoA를 유지함을 가정했다. 따라서 MN이 CN에게 HoA와 잘못된 CoA가 포함된 위조된 *BU* message를 보내면, HA는 자신의 바인딩 캐쉬에 저장되어 있는 내용과 일치하지 않으므로 잘못된 정보를 포함한 *BU_Auth_Request* message를 처리하지 않는다. 그 후에, 부정한 MN은 잘못된 *BU* message를 보내자마자 해당 메시지에 상응하는 *BU_Auth_Response* message를 위조하려 할 것이다. MN은 HoA와 관련한 D-H 비밀키 y를 모르기 때문에, MN은 올바른 $MAC(KS_2)$를 계산하지 못하고, *BU_Auth_Response* message를 위조하지 못한다. MN은 홈 등록 동안에는 부정한 행위를 하지 않는다는 초기의 가정에 위배되지만, 부정한 MN이 홈 등록을 통해 HoA와 잘못된 CoA를 등록했다고 가정하자. 잘못된 CoA로 CN에게 *BU* message를 보낼 때, MN은 다른 공격대상 호스트에 대한 flooding 공격을 위해 메시지의 redirect에는 성공할 수 있다. 하지만, HA와 CN의 바인딩 캐쉬에는 부정한 MN의 검출에 활용할 수 있는 두 가지 증거가 남게 된다.

4.3 Resource Exhaustion 공격

인터넷상에서는 공격대상 노드의 메모리와 계산 리소스를 고갈시키는 DoS 공격이 주요 위협이다. 공격자는 공격대상 노드에게 프로토콜이 실행되는 동안 계산 비용이 큰 공개키 계산이나 많은 상태를 생성하는 연속적인 메시지를 보냄으로써 flooding 공격을 할 수 있다. 이번 절에서는 제안 프로토콜이 가짜의 잘못된 메시지를 연속적으로 보냄으로써 발생하는 MIPv6 노드의 flooding 공격에 대응하는 방법을 살펴볼 것이다.

첫째, 공격자는 불필요한 *BU* message를 연속적으로 CN에게 보낸다. CN은 쿠키를 생성을 위한 경량의 키를 이용한 해쉬 계산을 수행하지만, 수신한 각 메시지에 대한 어떠한 상태도 생성하지 않는다. 이 단계에서, CN은 해당 메시지의 필터를 위한 어떠한 효과적인 방법도 가지고 있지 않기 때문에, 불필요한 *BU_Auth_Request* message를 생성하고 HA에게 보낸다. 만약 flag가 0으로 설정된 nonce가 없을 경우, 추가적인 *BU* message는 CN에 의해 드롭된다.

둘째, 연속적인 *BU_Auth_Request* message를 보내는 공격에 방어하기 위해서, 기본적인 제안 프로토콜을 다음과 같이 약간 수정할 수 있다. MN을 위한 바인딩 캐쉬 목록에 *BU_Auth_Request* message 송신자의 ID를 저장하기 위해 추가적으로 "request_identity"라 불리는 작은 크기의 메모리를 할당한다. CN의 정상적인 메시지 수신 후, HA는 메시지의 유효성을 검사하기 위한 계산을 수행하고 성공 시, HA는 CN을 해당 메모리에 저장한다. 공격자가 *BU_Auth_Request* message를 도청하였다가, 이어서 HA에게 동일한 메시지를 연속적으로 보낸다고 가정하자. HA는 송신자가 MN에 대한 바인딩 캐쉬에 존재하는지 여부를 체크한다. 앞서 소개한 공격의 경우, 바인딩 캐쉬에 송신자가 이미 존재하기 때문에, 이러한 메시지는 D-H 계산을 수행하지 않고 드롭된다. 또한 메모리 공간

의 절약을 위해 일정 lifetime이 지나면 해당 필드(request_identity)는 삭제될 수 있다.

셋째, 도청된 *BU_Auth_Response* message의 연속적인 전송은 nonce index *j*를 이용하여 검출될 수 있다. 앞서 언급한 대로, CN은 *nonce*의 배열인 *ArrayN*을 유지하므로 정상적인 처리에서, 메시지 처리가 성공적이라면 메시지 내의 index와 상응하는 flag가 1로 설정된다. 그러나, 만약 동일한 index 값을 포함한 몇몇의 메시지가 수신된다면, flag가 0으로 설정되어 있거나 대응하는 index 값이 변경되었기 때문에, 처리되지 않고 드롭된다.

4.4 가짜 HoAs를 이용한 공격

일 방향 해쉬함수의 속성으로 인해 특정 HoA와 상응하는 한 쌍의 D-H 공개키/비밀키를 찾는 것은 불가능하지만, 공격자는 임의의 한 쌍의 D-H 공개키/비밀키를 생성하는 것은 가능하다. 이를 기반으로 가짜의 HoA를 위조할 수 있고, 또한 위조된 *BU* message를 CN에게 보낼 수 있다. 이러한 약점 때문에, CN이 위조된 *BU* message를 연속적으로 수신하고 처리한다면, CN의 바인딩 캐쉬는 불필요한 바인딩 정보로 채워질 것이다. 이러한 문제점은 제안 프로토콜뿐 아니라 다른 CGA 기반 BU 프로토콜[5,6,7,9]에도 역시 적용될 수 있다. 하지만, 이번 절에서는 해당 공격이 제안 프로토콜에서는 실효성이 없음을 보인다.

공격자가 D-H 파라미터 집합 (z_1, g^{z1}), (z_2, g^{z2}), …, (z_m, g^{zm})을 생성하고, 그에 대응하는 가짜의 IPv6 HoAs (HoA_1, HoA_2, …, HoA_m)와 임의의 CoAs (CoA_1, CoA_2, …, CoA_m)를 생성했다고 가정하자. 공격자는 이제 CN에게 가짜의 HoA와 CoA을 기반으로 한 몇몇의 가짜 *BU* message (BU_1, BU_2, …, BU_m)를 생성하여 보낼 것이다. BU_i (i = 1, 2, …, m)을 수신하면, CN은 위조된 *BU* message의 HoA_i와 관련한 HA에게 그에 상응하는 $BU\_Auth\_Request_i$ message를 보내지만, HoA_i가 가짜이기 때문에 해당 메시지는 HA에 의해 처리되지 않는다. 따라서 공격자는 공격을 성공시키기 위해서 CN에게 위조된 $BU\_Auth\_Response_i$ message를 보내야 하며, 해당 메시지의 작성을 위해서는 CN에 의해 생성된 cookie가 필요하므로 $BU\_Auth\_Request_i$ 메시지를 도청해야만 한다. 즉, 공격자가 CN의 바인딩 캐쉬에 불필요한 정보 1개를 채우기 위한 목적으로 공격자는 계산적 오버헤드가 큰 D-H 계산을 수행해야 하며, $BU\_Auth\_Request_i$ message를 도청해야 한다. 따라서 가짜의 HoA에 기반 한 위조된 BU 메시지의 연속적인 송신으로 CN의 바인딩 캐쉬에 불필요한 정보로 채우기 위한 DoS 공격은 제안 프로토콜에서는 효과적이지 못한 공격이다.

4.5 불필요한 BU 프로토콜 발생 공격

만약 MN이 HA를 통해 새로운 CN의 IP 패킷을 수신 받는다면, MN은 자동적으로 CN에게 *BU* message를 보낸다. 이러한 속성을 이용하여, 공격자는 다수의 모바일 노드와 어떤 일정한 대응노드 사이의 불필요한 BU 프로토콜을 발생시킬 수 있다.[2-4] 공격자가 다수의 모바일 노드들의 HoA, CoA와 홈 에이전트를 알고 있다고 가정하자. 그렇다면, 일정한 CN을 대상으로 한 많은 거짓 패킷들이 생성되고, 각각의 모바일 노드들에게 보내진다. 그 후, CN은 상응하는 MN 혹은 HA와 몇몇의 불필요한 BU 프로토콜을 수행하고, 바인딩 캐쉬에 많은 불필요한 바인딩 정보를 생성한다. 제안 프로토콜은, 이러한 유형의 공격을 미리 설정된 MN과 HA 사이의 SA를 사용하여 쉽게 막을 수 있다. CN으로부터 수신한 패킷을 MN의 CoA로 터널링할 때, HA는 소스 인증의 목적으로 전체 패킷을 보호하는 $MAC(K_{HM})$을 포함한다.

V. 비교 분석

MIPv6 BU 프로토콜의 계산적, 메시지의 오버헤드는 지연에 민감한 응용환경에서는 심각한 영향을 줄 수 있다. 따라서 BU 프로토콜에서 MIPv6 개체들 사이에 교환되는 메시지 수를 최소화하고, 계산적 오버헤드를 최소화하는 것이 바람직하다. 다음의 표 1은 기존 프로토콜과 제안 프로토콜의 메시지 수과 각 개체의 오버헤드 비교와 안전성에 대해 비교해 놓은 것이다. 안전성 비교에서는 Ⅳ장의 안전성 분석에서 언급한 공격 시나리오를 대상으로 하였다. 메시지 수 측면에서는 제안 프로토콜과 기본 SUCV 프로토콜이 가장 효율적이다. 계산적 오버헤드의 비교를 위해 MN, CN, HA에 의해 수행되는 D-H 계산과 전자서명 생성과 확인 횟수를 비교하였다. 제안 프로토콜이 계산 복잡도 측면에서 기존

표 1. 프로토콜 비교
[1] Return Routability [6-1] Basic SUCV
[6-2] Extended SUCV [8] PKI-based Proxy
[9] CGA-based Proxy
DH=D-H 계산 DS=전자서명 생성/확인

	[1]	[6-1]	[6-2]	[8]	[9]	Ours
메시지 수	8	4	7	9	9	4
MN의 계산	None	DH(1) DS(2)	None	None	None	None
CN의 계산	None	DH(1) DS(2)	DH(1) DS(2)	DH(1) DS(1)	DH(1) DS(1)	DH(1) DS(0)
HA의 계산	None	None	DH(1) DS(2)	DH(1) DS(1)	DH(1) DS(1)	DH(1) DS(0)
Security Infra.	None	None	None	PKI	None	None
Redirect Attack (Type 1)	X	○	○	○	X	○
Redirect Attack (Type 2)	X	X	X	△	X	△
Resource Exhaustion Attack	○	X	X	○	X	○
Attack using fictional HoA	X	X	X	○	X	○

Secure(○), Insecure(X), Mitigated(△)

의 프로토콜에 비해 더욱 효율적임을 알 수 있다.

기존 프로토콜은 위의 공격으로부터 완전한 안전성을 보장하지 않은 반면, Ⅳ장에서 살펴본 것과 같이 제안 프로토콜은 Redirect와 DoS 공격으로부터 안전성을 제공한다. 즉, 제안 프로토콜은 다양한 공격으로부터 안전성을 제공하며, 위의 표에서는 나타내지 않았지만 [2,3,4]에서 언급한 다른 공격들에 대해서도 안전성을 제공한다.

Ⅵ. 결 론

본 논문에서는 MN과 CN 사이의 안전하고 효율적인 새로운 BU 프로토콜을 제안하였다. 제안 프로토콜에서 HA는 MN에 대한 최신의 올바른 바인딩 정보를 유지하고 있음을 가정으로 하고, 새로운 BU 메시지의 유효성을 검사하는 보안 프록시 기능과 MN과 CN에 대한 세션키 분배센터의 기능도 수행한다. 본 논문에서는 또한 공개키 계산없이 MIPv6 개체들 사이의 새로운 SA 설정을 위한 CGA 기반 stateless Diffie-Hellman 키 합의라는 새로운 기법을 제안하였다. 본 논문의 안전성과 비교분석에서 본 바와 같이 제안 프로토콜은 메시지 수, 계산적 오버헤드 측면에서 효율적이며, 다양한 공격으로부터 안전성을 제공하는 MIPv6 BU 프로토콜이다.

참 고 문 헌

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

[2] Aura, T., Roe, M. and Arkko, J., "Security of Internet Location Management," *In Proc. The 18th Annual Computer Security Applications Conference*, Las Vegas, Dec. 2002.

[3] Aura, T., "Mobile IP Security," *Security Protocols: The 10th Int'l Workshop, Cambridge*, U.K., Apr. 17-19, 2002, LNCS 2845, Springer Verlag, 2003.

[4] Nikander, P., Arkko, J. and Aura, T., Montenegro, G., Nordmark, E., "Mobile IP version 6 Route Optimization Security Design Background," *draft-ietf-mip6-ro-sec-02*, Oct. 2004.

[5] O'Shea, G. and Roe, M., "Child-proof Authentication for MIPv6 (CAM)," *ACM Computer Communications Review*, 31 (2), July 2001.

[6] Montenegro, G., Castelluccia, C., "Statistically Unique and Cryptographically Verifiable Identifiers and Addresses,"*In Proc. ISOC Symposium on Network and Distributed System Security (NDSS 2002)*, San Diego, Feb. 2002.

[7] Aura, T., "Cryptographically Generated Addresses,"*RFC 3972*, March 2005.

[8] Deng, R., Zhou, J., and Bao, F., "Defending against Redirect Attacks in Mobile IP," *In Proc. The 9thACM conference on Computer and communications security*, Washington D.C., Nov 18-22, 2002.

[9] You, I. and Cho, K., "A Security Proxy Based Protocol for Authenticating the Mobile IPv6 Binding Updates," *Computational Science and Its Applications ICCSA 2004:International Conference*, Assisi, Italy, May 14-17, 2004, LNCS 3043, Springer Verlag, 2004.

[10] Soliman, S., "*Mobile IPv6 : Mobility in a Wireless Internet,*" Addison-Wesley, 2004.

[11] Arkko, J., Devarapalli, V. and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents," *RFC 3776*, June 2004.

[12] Aura, T., Nikander, P., and Leiwo, J., "DoS-resistant Authentication with Clients Puzzles," *Security Protocols: The 8th Int'l Workshop*, Cambridge, U.K., Apr. 25-27 2000, LNCS 2133, Springer Verlag, 2001.

〈著者紹介〉

강 현 선 (Hyun-sun Kang)
2002년 2월: 단국대학교 전자계산학과 졸업
2004년 2월: 단국대학교 전자계산학 석사
2004년 3월~현재: 단국대학교 전자계산학 박사과정
〈관심분야〉 암호이론, 보안 프로토콜, IPv6

박 창 섭 (Chang-seop Park)
1983년: 연세대학교 경제학과 졸업
1983년: 한국 IBM 근무
1990년: 미국 Lehigh Univ. 전자계산학 박사
1990년~현재: 단국대학교 전자컴퓨터학부 교수
〈관심분야〉 부호이론, 암호학