MIPv6을 위한 티켓 기반의 바인딩 갱신 프로토콜[1)]

구중두[1] 김상진[2] 오희국[1]
한양대학교 컴퓨터공학과[1], 한국기술교육대학교 인터넷미디어공학부[2]
{jdkoo, hkoh}@cse.hanyang.ac.kr
sangjin@kut.ac.kr

Ticket based Binding Update Protocol for MIPv6

Jungdoo Koo[1], Sangjin Kim[2], Heekuck Oh[1]
Hanyang University, Department of Computer Science and Engineering[1],
Korea University of Technology and Education, School of Internet-Media Engineering[2]

요 약

기존 MIPv6을 위한 바인딩 갱신(binding update) 프로토콜은 이동 노드가 새로운 외부 링크로 이동할 때마다 동일한 프로토콜을 수행하여 바인딩을 갱신한다. 그러나 이동을 하지 않아도 바인딩 갱신의 수명 때문에 같은 네트워크 링크에 있더라도 바인딩 갱신을 계속 수행하여야 하므로 매번 동일한 과정을 반복하여 바인딩을 갱신하는 것은 비효율적이다. 이 논문에서는 CGA (Cryptographically Generated Address) 방식의 MIPv6을 위한 새로운 티켓 기반 바인딩 갱신 프로토콜을 제안한다. 제안한 프로토콜에서 처음으로 바인딩을 갱신할 때 대응노드는 이동 노드에게 티켓을 발급하여 준다. 이후 갱신에서는 홈 에이전트의 도움 없이 이동 노드는 대응 노드와 직접 바인딩 갱신 프로토콜을 수행할 있다. 따라서 기존 프로토콜과 달리 두 번째 갱신부터는 매우 효율적으로 바인딩을 갱신할 수 있다. 이 논문은 몇 가지 공격 시나리오를 통해 제안한 프로토콜의 안전성을 분석하며, 기존 프로토콜과 비교하여 효율성을 입증한다.

1. 서 론

모바일 IPv6[1]에서 모든 노드는 자신의 현재 위치와 상관없이 접근할 수 있는 정적인 홈 주소(home address)와 홈 링크가 아닌 외부 링크에 접속되어 있을 경우 그 링크로 패킷들이 전달될 수 있도록 동적인 의탁 주소(CoA, care of address)를 사용한다. 이 주소는 그 노드를 관리하고 있는 홈 에이전트에 통보해야 한다. 그러나 중요한 문제는 기존에 통신하고 있던 대응 노드들과도 바인딩을 해야 한다는 것이다. 이 과정이 없으면 항상 데이터가 홈 에이전트를 통해 전달되게 되어 비효율적인 경로를 통해 전달되는 문제점이 있다. 이 문제점 때문에 MIPv6에서는 라우트 최적화(route optimization)라는 통신 방식을 도입하여 대응 노드가 이동 노드의 홈 주소가 아닌 의탁 주소로 직접 메시지를 보낼 수 있도록 하였다.

공격자들이 바인딩이 갱신되는 과정을 공격하여 특정 노드의 의탁 주소를 다른 주소로 변경하여 갱신되도록 하면 세션을 가로챌 수 있고, 특정 노드에 대한 서비스 거부 공격을 시도할 수 있다. MIPv6에서 이동 노드와 그 노드의 홈 에이전트 간에는 IPSec을 통해 안전하게 바인딩 업데이트가 가능하지만 이동 노드와 임의의 대응 노드 간에는 IPSec을 사용할 수 없는 경우가 많아 안전하게 바인딩을 갱신하는 새로운 메커니즘이 요구되었다.

기존 바인딩 갱신 메커니즘은 한결 같이 매번 동일한 프로토콜을 통해 바인딩을 갱신한다. 즉, 지난 바인딩 갱신을 통해 얻은 정보나 값을 활용하여 그 이후부터는 보다 효율적으로 갱신하는 방법을 사용하지 않고 있다. 이런 접근 방식은 크게 두 가지 문제점이 있다. 첫째, 빈번하게 새 링크로 이동하는 노드의 경우에는 매번 동일한 과정을 반복하는 것은 비효율적이다. 둘째, 바인딩 갱신은 수명이 있으며, 이 수명은 끝나면 이동하지 않은 경우에도 바인딩을 다시 갱신해야 한다. 따라서 이 논문에서는 처음으로 바인딩을 갱신할 때 대응 노드가 모바일 노드에게 티켓을 발급하여 주도록 하여, 향후 갱신이 필요할 경우에는 보다 효율적으로 갱신할 수 있는 프로토콜을 제안한다. 즉, 두 번째 갱신부터 홈 에이전트의 도움 없이 직접 대응 노드와 바인딩 업데이트를 수행할 수 있다. 이렇게 하면 업데이트의 효율성뿐만 아니라 두 가지 추가적인 장점을 지니고 있다. 첫째, 티켓은 대응 노드가 자신만이 알고 있는 대칭키로 발급하며, 티켓에 이동 노드를 인증하기 위한 세션키가 포함되어 있으므로 대응 노드는 상태(티켓)를 유지할 필요가 없다. 둘째, 홈 에이전트의 도움 없이 업데이트를 수행할 수 있으므로 홈 에이전트가 동작하지 않더라도 업데이트를 수행하고 통신을 계속할 수 있다.

이 논문에서 제안하는 프로토콜에서 홈 주소와 위탁 주소는 모두 CGA 방식[2]을 사용한다. CGA 방식에서 주소는 서브넷 프리픽스 정보와 공개키를 이용하여 생성된다. 이런 주소를 사용하면 대응되는 개인키를 이용하여 안전하게 노드들을 인증할 수 있다. 물론 누구나 새로운 가짜 주소를 만들 수 있지만 기존 주소를 도용할 수는 없다. 이 때문에 이 논문에서는 기존 제안들과 달리 홈 에이전트가 CGA를 생성하며, 오직 홈 에이전트만이 대응되는 개인키를 알고 있다고 가정한다. 즉, 각 노드는 대응되는 개인키를 모른다.

본 논문의 나머지 구성은 아래와 같다. 2장에서는 기존에 제안된 논문들에 대해서 살펴보고 3장에서는 이 논문에서 제안하고 있는 바인딩 갱신 프로토콜을 상세히 서술한다. 4장에서는 제안한 프로토콜의 안전성 및 효율성에 분석하며, 끝으로 5장세 결론 및 향후 연구 방향에 대해서 제시한다.

2. 관련연구

기존에 제안된 프로토콜은 크게 CGA에 기반한 프로토콜과 그렇지 않은 프로토콜로 나누어서 살펴보도록 한다.

CGA 기반의 CAM[3] 프로토콜은 한 번의 메시지만으로 바

1) 본 연구는 정보통신부 및 정보통신 연구진흥원의 대학IT연구센터(홈네트워크 연구센터) 육성•지원사업의 연구결과로 수행되었음.

인딩 갱신을 수행한다는 장점을 가지고 있다. 그러나 이동 노드가 PDA나 핸드폰과 같이 계산 능력과 전원 수명에 제한을 갖는 노드일 경우에는 CAM 프로토콜은 적합하지 않을 수 있다. 이것은 이동노드에서 계산량이 많은 전자서명을 수행하는 것은 부담되기 때문이다. 또한 대응노드에서 서명 확인을 통해 노드를 인증하기 때문에 도스 공격 또는 리소스 고갈 공격 위험에 노출될 수 있다.

CBID (Crypto-Based Identifiers)[4] 프로토콜도 CGA 기반이지만 CAM과는 다르게 이동노드에서 전자서명을 하는 대신에 연산능력이 뛰어난 홈 에이전트에서 처리한다. 이 프로토콜은 또한 도스 공격을 막기 위해 클라이언트 퍼즐 개념을 이용하고 있다.

ECBU (Extended Certificate-Based Binding Update)[5] 프로토콜은 인증센터에서 발행한 인증서를 가지고 안전하게 바인딩 업데이트를 수행하는 프로토콜이다. 이 프로토콜의 바인딩을 갱신하기 위해서 필요한 메시지 수가 많다는 단점을 가지고 있다.

3. 프로토콜

제안하는 프로토콜은 CGA 기반의 티켓 방식을 사용하는 프로토콜이다.

3.1 프로토콜의 가정

* 이동 노드와 대응 노드는 서로의 CGA에 대해 확신한다.
* 이동 노드와 홈 에이전트는 공유된 대칭키를 가지고 있다.
* 대응 노드는 이동 노드가 아닌 고정 노드이다.

3.2 표기법

* MN/MH/CN : 이동노드/이동노드의 홈 에이전트/대응노드.
* HoA/CoA: 이동노드의 홈 주소/의탁주소.
* MH_{addr}/CN_{addr}: 홈 에이전트의 주소/대응노드의 주소.
* BU/BA: 바인딩 업데이트/바인딩 업데이트에 대한 응답.
* $T_e/N_e/L$: 노드 e의 타임 스탬프/난수/라이프 타임.
* MAC(K,M): 암호키 K를 이용한 메시지 M에 대한 MAC 값
* K_{MH-CN}/K_{MN-CN}: MH와 CN사이의 비밀키/MN과 CN사이의 세션키.
* x_{MN}/g^x_{MN}: MN의 Diffie Hellman 개인키/공개키 쌍.
* y_{CN}/g^y_{CN}: CN의 Diffie Hellman 개인키/공개키 쌍.
* Sig(): 전자 서명
* TckMN-CN: MN과 CN사이의 티켓. 이동노드에서 핸드오프 발생 시에 CN과의 인증을 위해서 사용.
* A||B: 메시지 구성요소 A와 B의 비트 결합.
* prf(k,m): Pseudo 랜덤 함수(k:키, m:메시지)

3.3 제안하는 프로토콜

MN과 CN사이의 최초 BU 프로토콜은 그림 1과 같이 수행된다. 초기 BU에서 MN은 티켓 발행 및 계산 부담으로 인해서 HA를 통해서 CN과 BU를 수행한다. 이 단계에서 HA와 CN은 두 노드 사이의 비밀키를 생성하고 CN은 MN을 위한 티켓을 발행한다. 티켓은 MN이 최초 BU 이후에 핸드오프가 일어났을 경우 CN에게 전송함으로써 인증 및 두 노드 사이의 비밀키를 공유하는데 사용된다.

3.3.1 최초 바인딩 업데이트 프로토콜

MN은 MH에게 CoA를 등록한다. 또한 MN은 MH를 통한 CN과의 BU 및 티켓 발행을 요청한다. MN은 MH와 공유하고 있는 비밀키를 가지고 MAC을 통해서 메시지를 인증한다.

$$CoA, MH_{addr}, CN_{addr}, HoA, n_{MN}, T_{MN}, L_{BU}, MAC(K_{MN-MH}, CoA||CN_{addr}||HoA||n_{MN}||T_{MN}) \quad \text{---------- } M_1$$

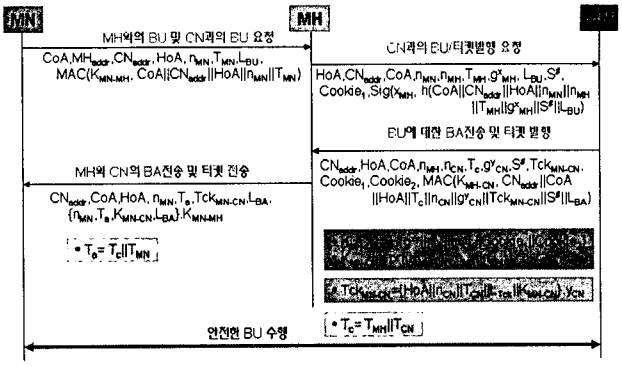

그림 1 MN과 MH 및 CN과의 초기 BU 프로토콜

MN이 보내는 n_{MN}는 MN과 CN사이에 사용할 세션키 생성에 사용된다. L_{BU}는 바인딩 업데이트의 라이프 타임이다.

$$HoA, CN_{addr}, CoA, n_{MN}, n_{MH}, T_{MH}, g^x_{MH}, L_{BU}, S^{\#}, Cookie_1, Sig(x_{MH}, h(CoA||CN_{addr}||HoA||n_{MN}||n_{MH}||T_{MH}||g^x_{MH}||S^{\#}||L_{BU})) \quad \text{---------- } M_2$$

M_2 메시지는 MN이 전송한 메시지를 수신한 MH가 티켓 발행 요청 및 MN과의 BU를 요청하는 메시지이다. T_{MH}는 HA의 타임스탬프로써 이 메시지를 수신하는 CN에서 일차적인 필터링 역할을 하기 위해서 사용되는 파라미터이다. $Cookie_1$ 역시 T_{MH}와 같이 두 노드 사이에 존재할 수 있는 도스 공격이나 Redirect 공격에 대응하기 위해서 보내지는 일차적 필터링 역할을 수행하는 파라미터이다. 또한 MH는 메시지에 전자서명을 해서 보냄으로써 CN에서 MH를 인증한다. $S^{\#}$는 Replay attack을 방지하기 위해 사용되는 파라미터이다.

$$CN_{addr}, HoA, CoA, n_{MH}, n_{CN}, T_c, g^y_{CN}, S^{\#}, Tck_{MN-CN}, Cookie_1, Cookie_2, MAC(K_{MH-CN}, CN_{addr}||CoA||HoA||T_c||n_{CN}||g^y_{CN}||Tck_{MN-CN}||S^{\#}||L_{BA})$$

$$\bullet K_{MH-CN}=prf(x_{MH}||n_{MH}||n_{CN}||Cookie_1||Cookie_2)$$

$$\bullet K_{MN-CN}=prf(K_{MH-CN}, n_{MN}||n_{CN})$$

$$\bullet Tck_{MN-CN}=[HoA||n_{CN}||T_{CN}||L_{Tck}||K_{MN-CN}]\ y_{CN} \quad \text{---------- } M_3$$

$$\bullet T_c= T_{MH}||T_{CN}$$

M2메시지를 수신한 CN은 일차적으로 MH에서 수신한 타임스탬프 T_{MH}와 $Cookie_1$을 확인한다. 먼저, g^x를 통해서 MN의 HoA의 인터페이스 식별자를 생성할 수 있는지 확인한다. 서명을 확인함으로써 MH를 인증한다. CN은 $Cookie_1$에 대한 응답으로 $Cookie_2$를 생성하고 MH와 CN사이에 사용할 비밀키 K_{MH-CH}를 생성한다. 또한, 티켓 안에 들어갈 MN과 CN이 사용할 세션키 K_{MN-CN}을 생성한다. 마지막으로 MN에게 발행해 줄 티켓 Tck_{MN-CN}을 생성한다. L_{BA}는 BA에 대한 라이프 타임이다.

$$CN_{addr}, CoA, HoA, n_{MN}, T_e, Tck_{MN-CN}, L_{BA}, \{n_{MN}, T_e, K_{MN-CN}, L_{BA}\}, K_{MN-MH}$$

$$\bullet T_e= T_c||T_{MN} \quad \text{---------- } M_4$$

M3 메시지를 수신한 HA는 Cookie와 CN과 자신의 타임스탬프를 결합한 Tc를 일차적으로 확인한다. 올바른 사용자로부터 온 메시지라는 것이 확인되면 수신한 파라미터들을 이용해서 CN과 HA사이에 사용할 비밀키 및 MN과 CN사이에 사용할 세션키를 생성한다. 또한, MN에게 CN으로부터 수신한 티켓을 전송한다. 또한 MH는 BU에 대한 BA를 MN에게 전송한다. MN은

자신이 보낸 난스와 각 노들의 타임스탬프를 결합한 T_a 및 L_{BA}를 확인하고 티켓을 얻는다.

3.3.2 차후의 바인딩 업데이트 프로토콜

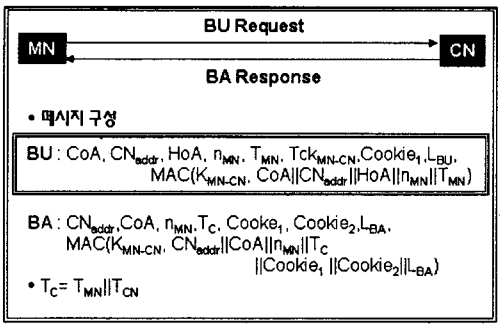

그림 2 MN과 CN사이의 BU 프로토콜

다음은 MN이 최초 바인딩 갱신을 수행 한 후에 핸드오프가 일어났을 경우에는 그림 2와같이 수행된다. 먼저 MN은 BU 메시지에 CN으로부터 받은 티켓과 쿠키를 추가한다. 또한 메시지 인증을 위해 MAC값을 추가한다. BU 메시지를 수신한 CN은 먼저 타임스탬프와 쿠키를 통해 정당한 메시지인지를 확인한다. 또한, 티켓을 복호화 한 후에 두 노드 사이에 사용할 세션키를 확인한다.

CN은 MN에게 MN의 쿠키와 자신이 생성한 쿠키 및 메시지 인증을 위해 MAC 값을 전송한다. 이 메시지를 수신한 MN은 먼저 두 노드가 생성한 쿠키 값과 타임스탬프의 결합인 T_c 및 BA의 라이프타임을 확인한 후에 MAC값을 확인한다.

4. 프로토콜 분석

4.1 프로토콜의 안전성 분석

프로토콜의 안전성은 DoS (Denial of Service)공격과 Redirect 공격, MITM (Man-in-the-middle-attack), Replay 공격 및 기타 공격 시나리오에 안전하다는 것을 증명한다.

* **DoS 공격**: 공격자는 불필요한 또는 위조된 바인딩 업데이트 메시지를 CN에게 플러딩 할 수 있다고 가정하자. 그럴 경우에 공격은 성공적으로 이루어진다. 그러나 우리의 프로토콜에 CN은 먼저 MN으로부터 온 쿠키와 타임스탬프 및 시퀀스 넘버를 확인한 후에 올바르지 않은 메시지일 경우에는 바로 메시지를 드롭한다. 또한 쿠키정보 역시 캐쉬에 저장하는 것이 아니기 때문에 메모리 오버플로우 공격에도 안전할 수 있다.
* **Redirect 공격**: MN과 CN사이의 통신에서, 공격자가 Redirect 공격의 일종인 Session Hijacking 공격을 할 수 있다고 가정하자. 그러나 우리의 프로토콜은 CGA 기반의 티켓 방식을 사용하기 때문에 공격자는 MN의 HoA 인터페이스 식별자를 생성하지 못한다. HoA 식별자와 CoA를 공격자가 얻었다고 해도 공격자는 티켓을 위조할 수 없으므로 Redirect 공격의 일종인 Session Hijacking 공격을 성공시킬 수 없다.
* **중간자 공격 및 재생 공격**: 공격자는 MN과 CN사이에서 중간자 공격 및 재생 공격을 할 수 있다고 가정하자. 그러나 앞서 살펴 본 공격들과 같이 공격자는 티켓을 위조할 수 없기 때문에 중간자 공격은 어렵다. 또한 재생 공격 역시 메시지에 포함된 타임스탬프 및 시퀀스 넘버로 인해서 어렵다.

4.2 프로토콜의 효율성 분석

MN에서 핸드오프가 일어난 후에 BU 프로토콜의 메시지 수와 각 노드에서의 계산량을 비교함으로써 프로토콜의 효율성을 분석한다.

* DH/DS: Diffie-Hellman 계산/전자서명
* None: DH 및 DS를 계산하지 않음
* BCBID/ECBID: 기본 CBID/확장 CBID

표 1 프로토콜의 효율성 분석

	RR [1]	ECBU [5]	CBID [3]		우리의 프로토콜	
			BCBID	ECBID	최초 BU	이후의 BU
메시지수	8	8	4	7	4	2
MN의 계산	None	None	DH(1) DS(2)	None	None	None
CN의 계산	None	DH(1) DS(2)	DH(1) DS(2)	DH(1) DS(2)	DH(1)	None
HA의 계산	None	DH(1) DS(2)	None	DH(1) DS(2)	DH(1) DS(1)	None

5. 결론

이 논문에서는 마이크로나 피코 셀 환경과 같이 핸드오프가 빈번하게 일어나는 환경에 적합한 새로운 바인딩 업데이트 프로토콜을 제안하였다. 제안된 프로토콜은 최초 바인딩 갱신이 일어날 때 티켓을 교환하여 향후 바인딩 갱신이 필요할 때 홈 에이전트의 도움 없이 매우 효율적으로 갱신할 수 있다. 이 논문에서는 대응노드를 고정노드라고 가정하였지만 대응노드도 이동노드가 될 수 있다. 따라서 향후에는 이런 환경에 적합한 바인딩 갱신 프로토콜에 대한 연구가 필요하다.

참고문헌

[1] D. Johnson, C. Perkins, J. Arkko, "Mobility Support in IPv6", IETF RFC 3775, June 2004.
[2] T. Aura, "Cryptographically Generated Addresses (CGA)", IETF RFC 3972, March 2005.
[3] G. Montenegro, C. Castelluccia, "Crypto-Based Identifiers (CBID): Concepts and Application", ACM Trans. on Information and System Security, Vol. 7, No. 1, pp. 97-127, Feb. 2004.
[4] G. O'Shea, M. Roe, "Child-proof Authentication for MIPv6 (CAM)", ACM Computer Communication Review, Vol 31 Issue 2, pp. 4-8, Apr. 2001.
[5] Y. Qiu, J. Zhou, F. Bao, "Protecting All Traffic Channels in Mobile IPv6 Network", 2004 Wireless Communication and Networking Conf., Vol. 1, pp. 160-165, Mar. 2004.