모바일 IPv6환경에서 SEND 프로토콜을 위한 대리 서명

류재현, 홍충선
경희대학교 컴퓨터공학과
e-mail: jhryu@khu.ac.kr ,cshong@khu.ac.kr

Proxy Signature For SEND Protocol in Mobile IPv6 Environment

Jae-Hyun Ryu , Choong-Seon Hong
**Dept of Computer Engineering, Kyung-Hee University

요 약

IPv6 환경에서 SEcure Neighbor Discovery(SEND) 프로토콜은 Neighbor Discovery 프로토콜의 위협 요소를 제거하고 단말 노드의 안전한 통신을 위한 환경을 제공함으로써 보다 안정적으로 인터넷 서비스를 제공할 수 있게 되었다. 하지만 SEND 프로토콜을 Mobile IPv6와 연동하는 과정에서 모바일 노드가 다른 네트워크로 이동하였을 경우 SEND 프로토콜 동작 과정에서의 취약점을 발견하고 기존의 방식과 차별화된 RSA기반의 위임 정보에 기반한 대리 서명 기법을 사용하여 문제를 해결하였다

1. 서론

인터넷 사용자의 증가로 인하여 부족한 주소 공간의 문제를 해결하기 위한 IPv6 프로토콜의 사용이 얼마 남지 않았다 앞으로 사용될 IPv6 프로토콜은 인접 노드와 통신을 하기 위해 인접 노드 탐색 프로토콜(Neighbor Discovery Protocol)[1]을 사용하여 디폴트 라우터 설정, Prefix 정보 획득, 중복 주소 검사, IP 주소와 링크 로컬 주소의 매핑 등 여러 가지 동작을 한다.

단말 노드가 이런 기본적인 설정을 제공하는 인접 노드 탐색 프로토콜에 보안 기능이 제공되지 않는다면 공격자는 보안상 약점을 이용하여 네트워크의 일부 또는 전체를 마비시키고 홈 네트워킹의 경우 개인정보 유출을 발생 시킬 수도 있다.

이러한 공격에 대비하기 위하여 IETF에서는 SEND(SEcure Neighbor Discovery) 워킹 그룹을 구성하여 IPv6 ND 프로토콜에 보안 기능을 지원하는 SEND 프로토콜의 표준화를 완료 하였다.[2]

2. 관련 연구

2.1 SEND 프로토콜과 CGA

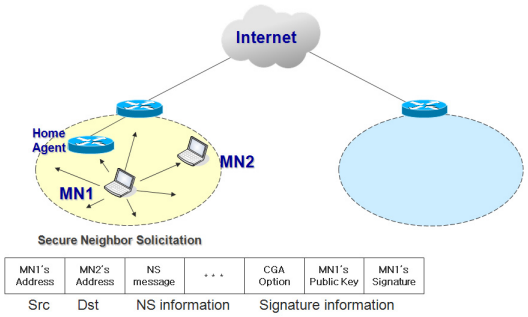
SEND 프로토콜은 메시지를 보낸 노드가 정당한 노드이며 정당한 방법으로 메시지를 생성하였고 메시지의 전송 과정에서 변조된 사실이 없다는 것을 증명하기 위해 CGA(Cryptographically Generated Address)[3] 방식을 사용 한다. 이 방식은 별도의 키 교환 방법이나 CA(Certification Authority) 또는 보안 인프라 구조 없이 송신자가 비밀키와 공개키를 생성하여 공개키는 주소에 포함시켜 CGA 주소를 생성하고 비밀키로 서명을 생성 할 수 있다.

단말 노드가 Neighbor Solicitation(NS) / Neighbor Advertisement(NA) 메시지에 대한 메시지 무결성 입증을 위해 자신이 생성한 비밀키로 메시지에 대한 서명을 생성하고 공개키를 메시지에 포함시켜 메시지를 전송한다. 이 메시지를 받은 노드는 메시지에 포함 되어 있는 공개키로 메시지의 무결성을 확인을 한다. 이 과정에서 메시지에 포함되어 있는 공개키로 공격자뿐만 아니라 어떠한 노드도 메시지의 내용을 볼 수 있다. 하지만 이 메시지에 대한 서명은 그 비밀키를 생성한 노드만이 할 수 있다.

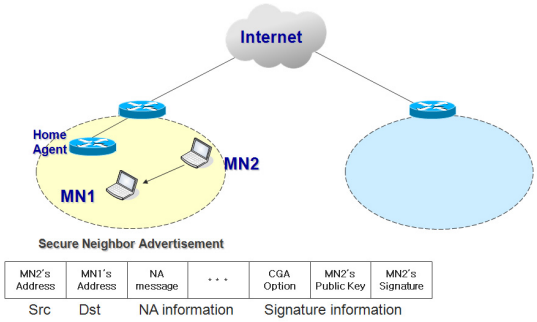
본 논문의 구성은 다음과 같다. 2장에서는 Mobile IPv6 환경에서 동작하는 SEND 프로토콜의 문제점과 대리서명에 관하여 살펴보고 3장에서는 본 논문에서 제안 하는 대리 서명 기법을 기술한다.

1) This paper was supported by ITRC and MIC

2.2 Mobile IPv6 환경에서의 SEND 프로토콜

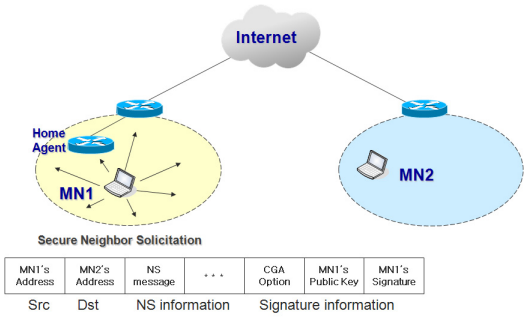


(그림 1) MN1이 MN2에게 Secure NS를 전송



(그림 2) MN2가 MN1에게 Secure NA를 전송

(그림 1)과 (그림 2)는 Secure NS/NA 메시지를 주고 받을 때 정상적인 동작 과정을 나타낸 그림이다.



(그림 3) MN1이 MN2에게 Secure NS를 전송

(그림 3)은 Mobile IPv6 환경에서의 SEND 프로토콜의 동작 과정에서의 취약점을 나타낸 그림이다. MN2는 자신의 홈 네트워크에서 다른 네트워크로 이동한 후 HA에게 Binding Update를 보내어 자신의 CoA(Care of Address) 주소를 등록 한다. 이 상황에서 MN1이 MN2의 홈 네트워크로 진입한 후 주소 생성 과정에서 MN2의 Home Address와 같은 주소를 생성하였고 중복 검사를 위해 Secure NS 메시지를 전송 하는 상황이다

MN1은 MN2에게 중복 주소 검사를 위해 Secure NS 메시지를 보낸 후 MN1은 자신에게 오는 Secure NA 메시지를 보고 중복 주소가 존재 하는지의 여부를 판단 한다. MN1은 Secure NA를 받지 못하였으므로 중복된 주소가 없다고 판단하고 HA에 자신이 생성한 주소를 등록하는 메시지를 보낸다. 이때 HA는 MN1이 등록 하려는 주소는 이미 다른 노드가 사용하고 있기 때문에 등록을 거부한다.

이처럼 Secure NS 메시지를 보내고 Secure NA 메시지를 받아야 함에도 불구하고 MN2의 위치가 외부 네트워크에 있다면 Secure NA 메시지를 받지 못하는 문제점이 발생한다.

2.3 대리 서명(Proxy Signature)

대리 서명 기법은 원 서명자(Original Signer)가 자신의 서명을 대리 서명자(Proxy Signer)가 대신 할 수 있도록 위임하는 서명 기법으로 대리 서명자는 원 서명자를 대신하여 서명을 하는 방법이다.

M. Mambo, K. Usuda와 E. Okamoto는 대리 서명 방식을 위임 형태에 따라 완전 위임, 부분 위임, 보증 위임으로 분류 하였다.[4]

완전 위임은 원 서명자가 자신의 비밀키를 대리 서명자에게 주는 방법으로 자신만이 가지고 있는 비밀키를 전달함으로써 안전성에 큰 위협을 받게 된다.

부분 위임은 원 서명자가 자신의 비밀키를 이용하여 대리 서명자의 비밀키를 생성하는 방법으로 원서명자와 대리 서명자의 구분이 가능하며 서로 상대방의 키를 추정할 수 없어야 한다.

보증 위임은 원서명자가 자신과 대리 서명자의 정보와 관련된 권한에 대해서 서명하고 검증자는 이 정보를 기초로 하여 원서명자로부터 선택된 대리 서명자임을 확인 할 수 있다.

3. 모바일 IPv6에서의 SEND프로토콜

Mobile IPv6환경에서의 SEND 프로토콜은 2.2절에서와 같이 문제점이 발견 되었다. MN2에 대한 정보는 MN2와 HA가 알고 있고 MN2는 Secure NS 메시지를 받지 못하는 상황에서 MN2는 HA에게 대리 서명을 위한 키를 전송함으로써 HA는 대리 서명을 할 수 있다.

MN2가 핸드오버 할 때 Secure NS 메시지를 받을 수 없는 상황이 발생하므로 이때 MN2는 HA에게 AAA(Authentication, Authorization, Accounting) 메시지를 보낼 때 마다 대리 서명을 위한 정보와 위임정보를 함께 전송하여 대리 서명을 요청한다. (AAA 프로토콜은 모

바일 노드가 외부 네트워크에 있을 때 주기적으로 홈네트워크의 인증을 받는 프로토콜이다.) MN2가 HA에게 대리 서명키의 사용을 제한하기 위하여 위임 정보에는 대리 서명이 가능한 메시지의 종류, 대리 서명이 가능한 시간, 대리 서명자등의 정보를 포함한다.[3]

대리 서명을 위한 정보 생성과 전달 및 검증, 대리 서명의 생성 과 검증 방법은 다음과 같다.

표 1 대리 서명 파라미터

S_{OS-s}	원 서명자의 서명값1	S_{PS-s}	대리 서명자의 서명값1
R_{OS-s}	원 서명자의 서명값2	R_{PS-s}	대리 서명자의 서명값2
K_{OS-PUB}	원 서명자의 공개키	K_{PS-PUB}	대리 서명자의 공개키
K_{OS-PRV}	원 서명자의 비밀키	K_{PS-PRV}	대리 서명자의 비밀키
$r1$ $r2$	랜덤 값	$r11$, $r12$	랜덤값
N_{OS}	원서명자의 moduler값	N_{PS}	대리 서명자의 moduler값
h()	해쉬 함수	ID_{PS}	대리서명자의 ID
|	OR 연산	ID_V	검증자의 ID
M_W	위임 정보	ID_{OS}	원서명자의 ID

■ 원 서명자의 대리 서명을 위한 정보 생성 (MN2)

임의의 수 r_1과r_2를 선택 (단 r_1+r_2 =D_{OS-PRV})

$$S_{OS-s} = (K_{OS-PRV} + r1)^{K_{OS-PRV}} (\mathrm{mod}\, N_{OS})$$

$$R_{OS-s} = h(M_W | ID_{OS})^{r2} (\mathrm{mod} N_{OS})$$

■ 원 서명자(MN2)는 대리 서명자(HA)에게

R_{OS-s}와 S_{OS-s}을 전송한다.

■ 대리 서명을 위한 정보의 검증 (HA)

$$[h(M_W | ID_{OS})^{(S_{OS-s})^{K_{OS-PUB}} (\mathrm{mod}\, N_{OS})} \cdot R_{OS-s}]^{K_{OS-PUB}}$$

$$\equiv h(M_W | ID_{OS})^2 (\mathrm{mod} N_{OS})$$

■ 대리 서명 생성 (HA)

임의의 숫자 $r11$과 $r12$를 선택 한다.

(단, $(S_{OS-s})^{K_{OS-PUB}} (\mathrm{mod} N_{OS}) = r11 + r12$)

$$S_{PS} = (r11 + h(M | ID_{PS} | ID_V))^{K_{PS-PRV}} (\mathrm{mod} N_{PS})$$

$$R_{PS} = R_{OS-s} \cdot h(M_W | ID_{OS})^{r12} (\mathrm{mod} N_{PS})$$

■ 대리 서명자는 대리 서명을 한 후 최종 검증자에게 R_{PS-s}와 S_{PS-s} 를 보낸다.

■ 최종 검증자의 검증(MN1)

$$[h(M_W | ID_{OS})^{(S_{PS-s})^{K_{PS-PUB}} (\mathrm{mod} N_{PS})} \cdot R_{PS-s}]^{K_{PS-PUB}}$$

$$\equiv h(M_W | ID_{OS})^{2 + h(M | ID_{PS} | ID_V)} (\mathrm{mod} N_{OS})$$

위에서 언급한 대리 서명을 위한 수식은 본 논문에서 제안한 RSA 기반의 위임정보에 기반한 대리 서명 방법이다. 기존의 대리 서명 방법은 CA로 부터 받은 대리 서명키를 사용하거나, 서명 정보(R , S) 이외에 추가적인 정보를 사용한 준간 결과값의 전송을 필요로 한다 본 논문에서 제안하는 방법은 CA를 통해 대리 서명키를 받을 필요가 없으며 추가적인 정보는 사용 하지만 검증을 위한 연산 과정에서 그 정보가 상쇄 되어 추가적인 정보가 필요하지 않다 이 연산에서 필요한 추가적인 정보는 CGA 확장 옵션에 포함하여 전송한다.[6]

3. 결론

본 논문에서는 Mobile IPv6 환경에서의 SEND 프로토콜의 적용에 관한 문제점을 언급 하고 이에 대한 해결 방안으로 위임 정보를 포함한 RSA 기반의 대리 서명 방식을 제안 하였다. 메시지를 받지 못하는 노드가(MN2) 메시지를 받을 수 있는 노드에게(HA) 제한된 대리 서명의 권한을 위임하고 대리 서명 권한을 가진 노드는 원 서명자에 대한 Secure NS 메시지에 대해 Secure NA 메시지를 전송하여 네트워크에서 일어날 수 있는 문제를 해결 하였다.

참고 문헌

[1] P. Nikander, Ed,“IPv6 Neighbor Discovery (ND) Trust Models and Threats”,RFC 3756 , May 2004

[2] J.Arkko, Ed. “SEcure Neighbor Discovery(SEND)” RFC 3971, Mar 2005

[3] Tuomas Aura, “Cryptographically Generated Addresses” RFC 3972, Mar 2005

[4] Mambo M, Usuda K , Okmamoto E, “Proxy signatures: delegation of the power to sign message” IEICE Transaction Functional E79-A(9), 1996

[5] Haifeng Qian, Zhenfu Cao, “A Novel ID-Based Partial Delegation with Warrant proxy signature Scheme”, LNCS 3759, 2005

[6] M.Bagnulo, “Cryptographically Generated Addresses (CGA) Extension Field Format”, RFC 4581, Oct 2006