情報保護學會論文誌
第15券 第3號, 2005. 6

AAA 오버헤드를 최소화한 효율적인 MIPv4 등록 프로토콜*

강 현 선,†‡ 박 창 섭
단국대학교

An Efficient MIPv4 Registration Protocol With Minimal Overheads Of AAA*

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

요 약

MIPv4는 MN의 이동성을 지원하기 위한 프로토콜이며, MN의 이동에 대해 등록 프로토콜을 수행함으로써 MN의 바인딩 정보를 관리하며 MN에게 지속적인 통신을 제공하게 된다. 이러한 등록 프로토콜은 무선 환경에서 다수의 MN에 의해서 수행되기 때문에 인증이 반드시 필요하며, 인증을 위해 키 분배센터 역할의 AAA를 도입하는 것이 일반적인 접근방식이다. 본 논문에서는 기본적인 AAA방식에 도메인 키 개념을 도입하여 AAA의 접속을 최소화하는 효율적인 등록 프로토콜을 제안한다. 또한 제안 프로토콜은 다양한 유형의 재생공격에 대응 가능하며, MN에게 네트워크 서비스를 제공함에 따라서 발생되는 과금문제를 해결하기 위한 부인방지 서비스도 제공한다.

ABSTRACT

MIPv4 supports node mobility, manages MN's binding list and provides seamless communication through registration protocol. Since the registration protocol usually operating in the wireless environment involves authenticating MNs, it is a general approach to introduce the AAA infrastructure as key distribution center for the purpose of authentication. In this paper, we propose an efficient registration protocol with lightweight AAA based on domain key. Proposed protocol also withstands various replay attacks, and provides non-repudiation service for the accounts of the usage of the network service.

Keywords : *MIPv4, Registration protocol, AAA*

Ⅰ. 서 론

MIPv4(Mobile IPv4)[2]는 IPv4 네트워크에서의 모바일 노드 MN(Mobile Node)에게 현재 진행 중인 전송계층 연결(transport connection)을 중단함이 없이 이동성을 제공하기 위한 목적으로 제안되었다. MIP에서는 MN에 대해서 두 가지 유형의 IP 주소가 정의된다. 그 중 하나는 MN의 홈 도메인 (home domain)에서 정의된 고정된 주소로 사용되는 HOA(Home-Of Address)이고, 다른 하나는 MN이 외부 도메인 (foreign domain)을 방문했을 경우 외부 에이전트 FA(Foreign Agent)에 의해서 동적으로 할당 받게 되는 COA(Care-Of Address)이다. MN은 현재의 COA를 자신의 홈 에이전트 HA (Home Agent)에게 등록 프로토콜(Registration protocol)을 통해 등록함으로써 이

접수일 : 2005년 1월 3일 ; 채택일 : 2005년 6월 7일
* 본 연구는 정보통신부 대학 IT연구센터 육성·지원사업의 연구결과로 수행 되었습니다.
† 주저자, ‡ 교신저자, sshskang@dankook.ac.kr

동 중에도 패킷 전달 (packet forwarding) 서비스를 받게 된다.

기본적인 MIPv4[2-4]의 등록 프로토콜은 무결성 보장 및 재생공격을 방지하기 위한 MAC (Message Authentication Code)기반의 인증기법이 포함이 되어 있지만, MAC 생성에 소요되는 공유키를 포함하는 SA(Security Association)는 사전에 설정/분배되어 진다고 가정을 하고 있다. 이와 관련하여 두 가지 유형의 SA 설정과 관련한 메커니즘이 제안된다. Mobile IPv4에 참여하는 임의의 두 개체 간에 SA를 공유하기 위해서는 키 분배센터의 역할을 하는 AAA (Authentication Authorization Accounting) 서버를 이용하는 방식[1,8]이외에도, PKI 기반의 공개키 암호 시스템[9]을 적용할 수도 있다. 하지만, 현재의 상황에서 전 세계적인 규모의 PKI구축을 가정하기에는 무리가 있으며 또한 하드웨어 제약적인 MN이 계산 복잡도가 높은 공개키 관련 작업을 수행하는 데에도 한계가 있다. 결론적으로, 대칭키 암호에 기반을 둔 AAA 프로토콜을 통해서 SA의 설정/분배작업을 수행하는 것이 보다 현실적인 접근방법이 된다. 최근 AAA를 기반으로 하는 기법[11]에서도 볼 수 있는 것처럼 AAA를 기반으로 하는 안전하고 빠른 기법을 설계하는데 초점을 맞추고 있다. 본 논문에서는 AAA의 도입에 따른 지연을 감소시킬 수 있는 등록 프로토콜을 제안하며, 또한 등록 프로토콜을 통해 MN에게 네트워크 서비스를 제공함에 따라서 발생되는 과금문제를 해결하기 위한 부인방지 메커니즘도 제안한다. 특히, 이와 관련된 전반적인 키 관리체계에 대해서도 논의한다. 다음의 소개될 Ⅱ장에서는 안전한 등록 프로토콜의 관련연구를 설명하며, Ⅲ장과 Ⅳ장에서는 본 논문에서 제안하는 프로토콜의 설계원리와 프로토콜을 소개한다. Ⅴ장에서는 제안 프로토콜의 안전성, 성능평가를 서술하며 마지막으로 Ⅵ장에서 결론을 맺는다.

Ⅱ. MIPv4의 안전한 등록 프로토콜 관련연구

2.1 기본적인 MIPv4 등록 프로토콜

다음 그림 1은 FA에게 제공받은 COA를 HA에게 등록하기 위한 MIPv4 등록 프로토콜을 나타낸다.

모바일 노드 MN이 이동함에 따라서, MN은 FA의 에이전트 광고 AA(Agent Advertisement) 메시지를 통해 자신의 위치 이동을 감지하고 FA로부터 새로운 COA를 제공 받는다. FA로부터 제공받은 COA를 HA에게 등록하기 위한 MIPv4의 기본적인 등록 프로토콜의 동작과정은 다음과 같다.

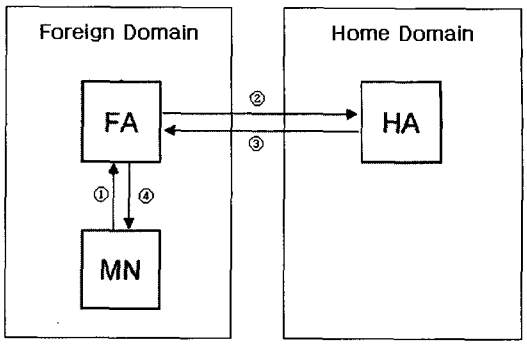

그림 1. MIPv4의 기본적인 등록 프로토콜

① MN은 새로운 COA를 포함한 등록요청 메시지를 FA에게 전송
② FA는 수신한 등록요청 메시지를 HA에게 전달
③ HA는 MN의 새로운 COA를 등록 등록응답 메시지를 FA에게 전송
④ FA는 수신한 등록응답 메시지를 MN에게 전달

2.2 안전한 MIPv4 등록 프로토콜

기본 MIPv4의 등록 프로토콜에서는 등록요청 메시지에 대한 무결성 보장 및 재생공격 방지를 위한 2가지의 방안을 기술하고 있다. 첫째, 무결성 보장은 MN과 FA 그리고 FA와 HA간에 전달되는 메시지들에 대해서 대칭키 기반의 MAC 값을 담은 인증 확장필드의 추가를 통해서 제공된다. 둘째, MIPv4에서의 재생공격은 공격자가 특정 MN의 정상적인 등록요청 메시지를 도청하였다가 일정한 시간이 지난 후 다시 재생시킴으로써 해당 MN에 대한 서비스 거부공격을 초래하거나 또는 자신이 불법적인 네트워크 접속 서비스를 받기 위해 사용된다. 이에 대응하기 위하여 등록요청/응답 메시지에는 타임스탬프 또는 난수를 담은 ID(Identification) 필드가 존재하여 등록요청 메시지에 대한 freshness를 보장한다. 난수를 사용하는 경우는 다음 등록요청에서 사용하게 될 난수를 HA가 생성하여 등록응답 메시지를 통해 MN에게 보내게 된다. 하지만, 이 방법은 등록요청 메시지에 대한 freshness 검사가 HA에 의해서만 행해지기 때문에, 만약 공격

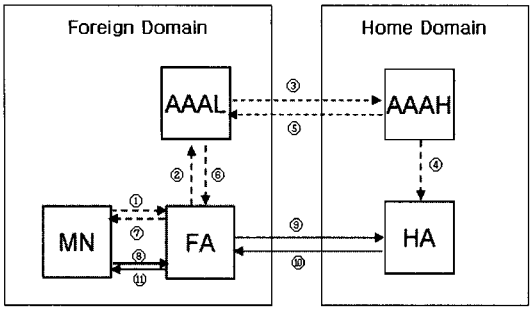

그림 2. AAA 기반 등록 프로토콜

자의 목적이 단지 외부 도메인 상에서의 네트워크 접속이라면 FA에 대한 다른 유형의 재생공격[10]이 가능하게 된다. 따라서, MIPv4 등록 프로토콜에 적용 가능한 시도-응답(challenge-response)프로토콜이 제안된다.[5,6] MN은 FA가 방송하는 난수에 FA 와 사전에 공유하고 있는 대칭키로 계산된 MAC을 보냄으로써 등록 요청 메시지에 대한 freshness 속성을 제공하게 된다. 이 방식은 재생공격에는 대응적이지만 FA가 생성하여 방송하는 수많은 시도 값을 유지, 관리해야 하는 복잡성의 문제를 안고 있다.

2.3 AAA 기반 등록 프로토콜

지금까지 논의한 MIPv4의 무결성 보장과 재생공격 방지를 위한 메커니즘들에는 당사자들 간에 사전에 대칭키가 공유되어 있다는 가정을 하고 있다. 하지만, MN, HA, FA들 간의 공유키 분배/설정은 선결되어야 할 과제인데, 인터넷 상에 존재하는 모든 MN들과 에이전트들에 대해서 각각 임의의 한 쌍에 대한 공유키를 설정한다는 것은 거의 불가능한 일이 된다. 따라서, 이에 대한 2가지의 접근방식이 있다. 첫째는 PKI기반의 공개키 서명 방식[9]이다. 이 방식은MAC 방식 대신에 등록요청/응답 메시지에 대한 서명을 함으로써 무결성 보장뿐만 아니라, MN의 등록요청에 대한 부인방지 기능도 부수적으로 제공된다. 하지만, 이 방식에서도 전 세계적인 규모의 PKI 구축을 가정하기에는 무리가 있으며, 공개키 서명에 따른 계산적인 오버헤드와 메시지 오버헤드 등에 대한 문제가 있다. 두 번째 접근방식은 AAA 서버를 MIPv4에 도입하는 것[1,8]이다. MIPv4에서의 AAA는 MN에 대한 인증, 권한검증, 과금, 개체간 인증의 기능을 수행하며, 노드 간에 사용할 공유키를 생성하여 등록과정 중에 동적으로 분배하는 키 분배 센터 역할도 수행한다.

그림 2는 제안 프로토콜과 비교에 사용할 AAA 기반의 등록 프로토콜을 나타낸다. 이 모델에서는 AAAH는 단지 개체간의 인증에 사용할 공유키를 생성하여 분배하는 역할을 수행한다. 홈 도메인 상의 AAA 서버인 AAAH와 외부 도메인 상의 AAAL 간에는 사전에 SA가 설정되어 있고, AAAL와 FA 그리고 AAAH와 HA, 그리고 MN과 AAAH간에도 역시 SA가 사전 공유되어 있음을 가정한다.

- ①번 ~ ⑦번

AAAH로부터 생성된 공유키를 분배받는 과정으로 ①에서 MN은 FA에게 해당 세션에서 사용할 키를 요청하면 FA와 AAAL은 ②, ③ 메시지를 통해 AAAH에게 MN의 인증과 공유키의 분배를 요청한 다. AAAH는 각 개체들 간에 사용하게 될 공유키를 생성하여 ④, ⑤, ⑥, ⑦ 메시지를 통해 분배한다.

- ⑧번 ~ ⑪번

공유된 키를 기반으로 한 MIPv4 등록과정으로 그림 1의 MIPv4 기본 등록 프로토콜의 동작과 같은 원리로 동작한다.

그림 2의 모든 전송은 앞서 가정한 개체 간 SA를 사용하여 전달하게 된다. 즉, MN의 이동에 따른 등록과정 수행에 있어서 AAA를 도입한 경우는 FA와 HA간의 정규 메시지 교환 이외에도 AAAH와 AAAL간의 추가적인 메시지 교환이 이뤄지게 되므로 등록 프로토콜 수행 시간이 길어지게 된다. 또한, 동일한 홈 도메인에 속해있지만 서로 다른 HA에 등록되어 있는 MN의 이동에 대해 각 MN의 이동마다 AAAL와 AAAH의 접속을 통한 등록과정이 개별적으로 이루어지기 때문에 AAA에 과다한 오버헤드가 발생하게 된다.

앞서 살펴본 바와 같이 등록과정에서 다양한 이유로 발생하는 오버헤드는 MN의 이동에 따른 등록 프로토콜의 처리를 지연시키고 등록이 지연되는 동안 패킷손실이 발생할 수 있게 된다. 따라서 등록 프로토콜에서는 안전성과 효율성을 제고하기 위한 다양한 요구사항이 존재하며, 본 논문에서는 AAA 방식을 기반으로 AAA의 오버헤드를 최소화하고 인증, 부인방지 서비스를 제공하는 재생공격에 대응적인 새로운 기법을 제안하고자 한다.

Ⅲ. 설계원리

이번 장에서는 본 논문에서 제안하는 프로토콜 설계에 적용된 주요 핵심적인 사항들을 발췌하여 소개하고, 다음 장에서 이 원리에 기반을 둔 제안 프로토콜을 기술한다. 제안하는 등록 프로토콜이 작동되는 기본적인 네트워크 및 보안 환경은 다음과 같다. 첫째, 홈 도메인과 외부 도메인에는 다수의 HA 및 FA들이 각각 존재한다. 둘째, 상이한 2개의 도메인에 각각 존재하는 AAA 서버 간에는 SA가 이미 설정되어 있다. 만약 다수의 상이한 도메인이 존재할 경우에는 계층적 형태의 브로커(broker) AAA 서버가 존재할 수도 있지만, 본 논문에서는 논의의 간소화를 위해서 2개의 도메인을 대상으로 등록 프로토콜을 기술한다. 셋째, AAA 서버는 해당 도메인 내에서의 인증기관의 역할을 담당하며 또한 자신의 공개키 인증서는 도메인 상의 모든 에이전트에게 사전 분배한다. 넷째, 한 도메인 내의 각각의 에이전트들과 AAA 서버와의 개별적인 SA가 존재하며, 또한 AAA 서버와 에이전트들은 일반적으로 동일한 기관에 의해서 관리, 운영되기 때문에 이들 간에는 신뢰관계가 이미 구축 되어져 있다고 가정하며 이들 간에 공유되는 도메인 키(domain key)가 존재한다. 다섯째, MN은 AAAH와 개별적인 SA를 사전에 공유한다.

3.1 도메인 기반 키 관리와 AAA 오버헤드의 최소화

일반적인 AAA 기반 MIPv4에서는 Ⅱ장에서 설명한 바와 같이 MN이 FA를 통해 등록을 요청할 때마다, MN-FA, HA-FA 간에 공유될 공유키를 설정하기 위한 AAAL와 AAAH 사이의 메시지 교환이 이루어지게 된다. 본 논문에서 제안하는 등록 프로토콜의 기본적인 개념은 동일한 홈 도메인에 속해 있는 다수의 MN들이 특정 외부 도메인의 서로 다른 FA들을 통해서 등록을 요청할 경우에, 결국 동일한 2개의 AAA 서버가 관여되는 것이기 때문에 AAAL와 AAAH간의 메시지 교환을 1번으로 단축시키자는 것이다. 이를 위해서 우리는 다음과 같은 키 관리체계를 제안한다. 각 도메인에 속하는 AAA 서버와 다수의 에이전트들은 일반적으로 동일한 기관에 의해서 관리, 운영되기 때문에 이들 간에는 신뢰관계가 이미 구축 되어 있다고 가정하여, 정적인 그룹키(group key)의 역할을 하는 도메인 키가 사전에 공유됨을 가정한다. 이 도메인 키의 공유는 AAA 서버와 에이전트 간에 사전에 설정해 놓은 SA를 기반으로 분배되며 MIPv4 개체간의 세션키 유도를 위해 사용된다. $H=\{HA_1, HA_2, \cdots, HA_l\}$와 $F=\{FA_1, FA_2, \cdots, FA_m\}$를 각각 홈 도메인과 외부 도메인에 속해 있는 HA와 FA들의 집합이라 하고, K_{Home}을 $H \cup \{AAAH\}$의 공통된 도메인 키라 하자. 또한 MN_{ij}를 홈 도메인의 HA_i를 에이전트로 사용하는 j번째의 MN이라 하자.

임의의 $i \in [1,l]$과 $k \in [1,m]$에 대해 홈 도메인의 $HA_i \in H$와 외부 도메인의 $FA_k \in H$ 사이에 설정되는 공유키는 $K_{HA_FA} = H^+(K_{Home}, AAAL)$로 AAAH에 의해서 계산되어지며, AAAL을 경유해 FA_k에게 전달된다. $H^+()$는 Pseudo Random Generator이며 HA_i는 K_{Home}을 알고 있기 때문에 공유키 K_{HA_FA}를 계산할 수 있게 된다.

3.2 AAAL의 인증캐쉬의 사용

앞 절에서 언급했듯이 동일한 홈 도메인과 특정한 외부 도메인 사이의 MN의 이동에서는 도메인 기반의 키 관리를 통해 AAAH와 AAAL간의 메시지 교환을 생략할 수 있으며, 대신 AAAL의 인증캐쉬에 저장되어 있는 공유키를 사용하면 된다. 기존의 FA에서 각각의 MN과 HA에 대한 SA를 저장, 관리하는 방식[1,8]과 비교하였을 때 AAAL에서 각각의 MN과 HA에 대한 SA를 저장, 관리하는 것은 여러 가지 이점이 있다. 우선 FA의 SA 관리의 부담을 줄일 수 있으며 MIPv4 등록과정에서 AAAH의 관여를 최소화할 수 있는 방안이다. 또한 외부 도메인에서의 MN에 대한 과금관련 정보의 관리를 고려할 때 AAAL의 SA 관리는 매우 효율적인 방안이다.

3.3 해쉬체인을 이용한 부인방지 서비스

본 논문에서는 해쉬체인[7]을 이용해서 모바일 노드가 등록요청을 통해 서비스를 받은 사실을 차후에 부인하지 못하게 하는 기능을 제공한다. 일반적인 해쉬체인(hash chain)은 seed 값이 주어졌을 때, 이를 기반으로 반복적인 일방향 해쉬함수 $h()$를 적용하여 생성되는 값들의 체인을 의미하며, 이러한 해쉬체인은 역 방향으로의 계산이 불가능하다는 특

성을 가진다. MN은 자신이 임의로 선정한 seed, h_w를 기반으로 w개의 해쉬체인을 생성하고, 루트 값 h_1은 서비스 초기화 과정에서 AAAH로부터 단 한번 서명을 받게 된다. 이 해쉬체인을 매번 등록요청 시마다 사용함으로써 AAA는 MN의 등록요청에 따른 서비스 제공의 확인 자료로 사용된다. 현재 MIPv4 표준 문서에서 부인방지 서비스에 대해서는 언급하고 있지 않다. 하지만 모바일 노드에게 네트워크 서비스를 제공함에 따라서 발생되는 과금문제를 고려할 때 부인방지 서비스는 매우 중요하고 반드시 필요한 기능이다. 일반적으로 부인방지 서비스를 제공하기 위해서는 공개키 서명을 사용한다. 하지만 매 등록/요청 메시지마다 전자서명을 사용하는 방식은 계산적 오버헤드와 패킷 오버헤드는 물론 인증서 관리라는 문제점이 있기 때문에 본 논문에서는 계산량과 오버헤드에서 훨씬 개선된 부인방지 서비스를 제공한다.

3.4 재생공격에 대한 대응책

FA를 대상으로 한 재생공격[10]에 대처하기 위해서 FA와 MN간에 적용되는 시도-응답 프로토콜[5,6]은 2장에서 설명한 바와 같이 시도-응답 관련하여 등록요청 메시지에 확장 필드가 추가되며 FA는 자신이 생성한 수많은 시도 값에 대한 유지, 관리를 해야 하는 부담을 안고 있다. 본 논문에서는 시도-응답 프로토콜을 사용하지 않고 재생공격에 대응적인 대안을 제시한다. 기본적인 원리는 MN이 FA에 보내는 등록요청 메시지와 HA가 FA에 전송하는 등록응답 메시지에 대한 무결성 보장 및 재생공격 방지를 위해서 등록 세션키의 개념을 도입하는 것이다. 즉, 등록요청이 시행될 때마다 MN-FA, HA-FA 간에 새로운 세션키를 사용하고, 특히 HA-FA 간의 세션키 설정은 FA가 주도케 하는 것이다. V장의 안전성 분석에서는 이와 관련된 사항들에 대해서 논의한다.

Ⅳ. 제안 프로토콜

4.1 용어정의

등록요청 메시지에는 ID, Lifetime, HOA, HA, COA 필드가 포함된다. ID는 단순 재생공격 방지와 등록요청/응답 메시지 동기화를 위한 필드이며 HA는 MN의 홈 에이전트의 주소를 나타낸다. 등록요청 메시지를 수신한 HA는 바인딩 갱신 (BU : binding update) 과정을 수행하며 수행 결과에 대해 등록응답 메시지를 전송하게 된다. 바인딩 갱신이란 자신이 관리하는 MN에 대한 HOA, COA, Lifetime 등으로 구성된 이동성 바인딩 리스트(mobility binding list)를 HA가 Lifetime에 명시된 기간만큼 유지 관리하는 것을 의미한다. 앞으로의 논의에 있어서 *MN*, *FA*, *HA*, *AAAH*, *AAAL*는 각각 해당 개체의 IP 주소를 의미하며, 추가적으로 다음과 같은 용어를 정의한다.

$RREQ$	: 등록요청 메시지
$RREP$	: 등록응답 메시지
K_{Home}	: 홈 도메인 내의 공통된 도메인 키
K_{MN}	: MN의 노드 키(node key)
K_{HA_MN}	: HA와 MN 간의 세션키
K_{HA_FA}	: HA와 FA 간의 공유키
K_{HA_FA}'	: HA와 FA 간의 세션키
Ks_i	: i 번째 등록시 사용되는 세션키
h_i	: i 번째 등록시 사용되는 해쉬체인 값
$E(k,m)$	: 대칭키 k로 메시지 m을 암호화
$Sig()$	: 개인키로 서명한 값
$h()$	: 일방향 해쉬함수 (해쉬체인 생성)
$H^*()$	: $2n$ 비트를 출력하는 Pseudo Random Generator
$H^+()$	: n 비트를 출력하는 Pseudo Random Generator
$Auth(k)$	: 앞에 존재하는 전체 메시지를 대칭키 k를 기반으로 계산한 MAC
$LEFT[k]$	: k의 최상위 비트에서 중간지점까지 해당하는 값
$RIGHT[k]$	: k의 중간지점에서 최하위 비트까지 해당하는 값

4.2 서비스 초기화 과정

MN은 MIPv4 등록과정과는 별도로 AAAH와의 서비스 초기화 과정을 수행한다. AAAH와 MN 간에는 각각 개별적인 SA가 존재하며, 해당 SA를 기반으로 AAAH로부터 $K_{MN} = H^+(K_{Home}, MN, HA)$

즉, 노드 키와 홈 도메인의 특정 HA를 안전하게 할당 받는다. 3.3절에서 설명한 것처럼 MN은 자신이 임의로 선정한 seed, h_w를 기반으로 w개의 해쉬체인 $\{h_{j_1} = h(h_j) \mid j = w, w-1, \cdots, 2\}$을 생성하고, AAAH는 h_1을 자신의 서명용 개인키로 서명한 $Sig(h_1)$을 MN에게 제공한다. 이와 같은 단 한 번의 해쉬체인에 대한 서명으로 차후 모바일 노드의 네트워크 사용에 대한 과금문제를 해결하기 위한 부인방지 기능을 제공하게 된다. 즉, MN은 서비스 초기화 과정에서 $\{K_{MN}, HA, (Sig(h_1), h_2, \cdots, h_w)\}$을 획득하게 된다. 위에서 생성한 해쉬체인은 모든 해쉬체인 값이 사용된 후에는 다시 초기화 과정을 통해 해쉬체인을 재 생성해야 하며, 본 논문의 성능평가는 한 번의 서비스 초기화 과정을 기준으로 한다.

4.3 등록과정

MN은 i 번째의 등록요청 메시지를 작성하기 위해서, 서비스 초기화 과정에서 제공받은 노드 키 K_{MN}을 기반으로 2개의 세션키를 다음과 같이 생성한다.

$$Ks_i = LEFT[H^*(K_{MN}, h_i)]$$
$$K_{HA\_MN} = RIGHT[H^*(K_{MN}, h_i)]$$

Ks_i은 현재 i 번째의 등록과정에서 FA와 공유하게 되는 인증용 세션키인데, FA는 이 세션키를 HA로부터 전달 받게 된다. K_{HA_MN}은 MN이 HA와 공유하게 되는 인증용 세션키이다. 다음의 그림 3은 제안하는 등록 프로토콜에서 대한 메시지의 흐름을 나타낸 것으로 이를 기반으로 제안 프로토콜을 설명한다. ①~⑤번까지의 점선으로 표시된 부분이 AAAH로부터 생성된 공유키를 분배받는 과정이고, ⑥~⑧번까지의 실선으로 표시된 부분이 공유된 키를 기반으로 MIPv4 등록과정의 수행을 나타낸다.

① MN→FA :
$RREQ, Auth(K_{HA\_MN}), AAAH, Auth(Ks_i)$

MN은 기본적인 $RREQ$ 메시지 필드에 h_i를 포함한 등록요청 메시지를 작성한다.

작성된 $RREQ$메시지를 K_{HA_MN}로 계산한 MAC 값에 해당하는 전체 메시지에 대해 Ks_i로 계산한 MAC 값을 FA에게 전송한다. 여기서 h_i는

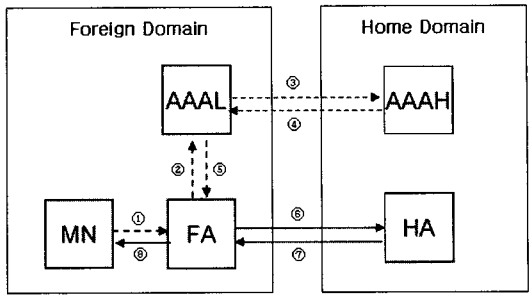

그림 3. 제안 프로토콜

MN의 i 번째 등록요청에 사용되어 질 해쉬체인 값이며, 나중에 서비스 부인방지를 위한 증거 자료로 사용된다. FA의 입장에서는 현재 Ks_i를 모르기 때문에 $Auth(Ks_i)$에 대한 인증을 당장 수행할 수는 없고, ⑦번 메시지를 수신한 후에 가능하게 된다.

② FA→AAAL : $MN, HA, AAAH$

FA는 AAAL에게 HA를 홈 에이전트로 이용하고 있는 MN의 등록요청을 처리하는 데에 사용될 인증용 세션키를 요청한다. AAAL는 이전에 MN의 홈 도메인의 AAAH로부터 전송받은 세션키가 인증캐쉬 내에 존재하는지를 검사한다. 만약 존재한다면 다음의 ③번, ④번 과정을 생략하고 ⑤번 과정을 수행하게 되며, 만약 존재하지 않는다면 다음의 ③번 과정을 수행하게 된다.

③ AAAL→AAAH : MN, HA

AAAL는 다시 AAAH에게 MN의 등록요청에 필요한 인증용 세션키의 생성을 요청한다.

④ AAAH→AAAL : K_{HA_FA}

AAAH는 자신의 도메인 키 K_{Home}를 기반으로 자신의 홈 도메인 소속의 임의의 HA와 AAAL가 속해있는 외부 도메인의 FA간에 공유될 세션키 $K_{HA_FA} = H^+(K_{Home}, AAAL)$을 계산한 후, AAAH와 AAAL가 사전에 공유하고 있는 SA를 기반으로 안전하게 AAAL에게 전달되고, AAAL의 인증캐쉬에 저장된다. 저장된 키는 차후 동일한 홈 도메인 내의 다른 MN이 어느 HA에 등록되어 있는 지에는 상관없이, 동일 외부 도메인의 임의의 FA를 통해서 등록요청을 하는 경우에 재사용된다. 이것은 동일한 도메인 내의 AAAH와 모든 HA가 도메인 키를 공

유하기에 가능하며 차후 AAA와의 접속을 줄일 수 있는 방안이기도 하다.

⑤ AAAL→FA : K_{HA_FA}

AAAL는 인증캐쉬에 저장되어 있거나 ④에서 전달 받은 K_{HA_FA}를 사전에 공유하고 있는 키를 기반으로 FA에게 안전하게 전달한다.

⑥ FA→HA : $RREQ, Auth(K_{HA\_MN}),$
$AAAL, r, Auth(K_{HA\_FA}')$

FA는 임의의 난수 r을 선택하여 HA와 FA와의 인증용 세션키 $K_{HA_FA}' = H^+(K_{HA_FA}, r)$를 계산하고, $\{RREQ, Auth(K_{HA_MN}), AAAL, r\}$에 대한 MAC 값 $Auth(K_{HA_FA}')$을 첨부하여 HA에게 전송한다. HA는 사전에 AAAH와 공유한 도메인 키 K_{Home}과 전달받은 $AAAL$을 통해 K_{HA_FA}를 계산하고 r을 이용하여 $K_{HA_FA}' = H^+(K_{HA_FA}, r)$을 계산한다. 계산된 키를 기반으로 해당 MAC 값 $Auth(K_{HA_FA}')$을 확인하고, K_{Home}과 $RREQ$의 h_i를 통해 K_{HA_MN}를 계산하고 이를 기반으로 해당 MAC값을 확인하여 MN을 인증한다. 이 부분에서 FA가 HA와 사용하게 될 세션키를 매 등록마다 임의의 r을 사용하여 새롭게 설정하여 사용함으로써 재생공격에 대응할 수 있게 되며, r은 세션키 계산을 위한 값으로 세션키 계산 후 따로 유지, 관리할 필요는 없다. 또한 MN의 첫 번째 등록을 위해 사용하는 해쉬체인 값은 AAAH의 개인키로 서명된 값이므로 HA는 AAAH의 공개키로 확인하고 그 값을 저장해 놓는다. 차후 이를 이용하여 MN의 과금 문제에 대한 부인을 방지할 수 있다. 이와 같이 제안 프로토콜은 단 한 번의 공개키 기반 서명을 통해 부인방지 서비스를 제공할 수 있게 된다.

⑦ HA→FA : $RREP, Auth(K_{HA\_MN}),$
$E(K_{HA\_FA}', h_i \| Ks_i), Auth(K_{HA\_FA}')$

등록요청 메시지에 대한 정상적인 처리가 완료되면 등록응답 메시지 $RREP$를 작성하고, 이를 K_{HA_MN}로 계산한 MAC값, h_i와 Ks_i를 K_{HA_FA}'로 암호화한 값과 전체 메시지에 대해 K_{HA_FA}'로 계산한 MAC값을 FA에게 전달한다. FA는 K_{HA_FA}'을 이용하여 Ks_i를 복호화함으로써 Ks_i를 MN과 공유하게 되고, MN으로부터 전달받은 ①번 메시지에 대한 인증을 수행할 수 있다. 또한 해당 MAC값을 확인함으로써 HA를 인증한다.

⑧ FA→MN :
$RREP, Auth(K_{HA\_MN}), Auth(Ks_i)$

FA는 HA로부터 전달받은 $RREP$와 그 MAC값과, 전체 메시지에 대해 Ks_i로 계산한 MAC값을 MN에게 전달한다. MN은 해당 MAC값을 확인함으로써 FA와 HA를 인증하게 되고, 이 과정을 통해 MN, FA 간에는 공통된 세션키 Ks_i를 공유하게 된다.

V. 분석 및 성능평가

5.1 안전성 분석

앞 장의 프로토콜의 설계원리와 등록과정에서 본 것과 같이 본 논문에서 제안한 프로토콜은 AAA와 도메인 키를 기반으로 하며, AAA의 도입에 따른 지연을 최소화하며 시도-응답 프로토콜을 사용하지 않고 재생공격에 대응적인 기법을 제안하였다. 이번 장에서는 제안 프로토콜의 안전성 분석을 위해 전반적인 개체 간의 키 생성/분배에 대해 살펴보고, 프로토콜의 안전성을 분석한다. 제안 프로토콜의 안전성을 위한 기본 원리는 MN이 FA에게 보내는 등록요청 메시지와 HA가 FA에게 전송하는 등록응답 메시지에 대한 무결성 보장 및 재생공격 방지를 위해 등록 세션키의 개념을 도입하는 것이다. 즉, 모든 개체 간에는 매 등록마다 새로운 세션키를 사용함으로써 개체 간에 교환되는 메시지에 freshness 속성을 제공한다.

제안 프로토콜에서 HA-MN 간의 세션키는 MN의 노드키를 기반으로 MN과 HA 각 개체에서 생성되며, MN은 $K_{MN} = H^+(K_{Home}, MN, HA)$의 계산에 의해 노드키를 생성하며, 이를 기반으로 HA와의 세션키는 K_{HA_MN}이 계산되어 진다. MN에 대한 노드키는 해당 MN에 대한 고유한 장치키 개념으로 사용된다. 따라서 HA-MN 간의 공유키 생성에 K_{MN}의 노드키를 사용함으로써 해당 세션키가 특정 MN에 대한 키임을 확인할 수 있게 된다. 또한 이로써 임의의 MN이 공격을 목적으로 자신이 생성한 HA-MN 간의 세션키를 생성하더라도, 해당 세션키

는 정당한 노드키를 기반으로 생성되지 않았기 때문에 해당 공격에 대응할 수 있게 된다.

FA-MN 간의 세션키는 HA-MN 간의 세션키와 유사한 방식으로 생성된다. 즉, Ks_i는 MN에 의해서 $Ks_i = LEFT[H^*(K_{MN}, h_i)]$로 생성되며 생성된 키는 HA에게 전달되고, HA에 의해 HA-FA 간의 세션키로 암호화된 형태 $E(K_{HA_FA}', h_i \| Ks_i)$로 FA에게 전달된다. FA는 HA-FA 간의 세션키를 사용하여 FA-MN 간의 세션키 Ks_i를 공유할 수 있게 된다. 여기서 HA가 세션키와 함께 해쉬체인 값을 함께 암호화해서 보내는데, 이는 HA의 등록응답 메시지가 MN의 등록요청에 대한 정당한 응답 메시지인가에 대한 확인을 위한 것이다. 또한 이 과정을 통해 FA-MN 간의 세션키에 대한 키 확인을 하게 된다.

앞 장에서 우리는 시도-응답 프로토콜로 대응할 수 있는 재생공격의 유형에 대해 간략히 언급했었다. 공격자가 FA를 대상으로 하는 재생공격으로, 해당 등록 프로토콜이 시도-응답 프로토콜이 아닌 난수를 통한 기본적인 재생공격에만 대응적이라고 가정할 때 발생할 수 있는 공격방식이다. 공격방식을 단계별로 살펴보면, 공격자는 정당한 MN에 의해 정상적으로 처리된 등록요청/응답 메시지 한 세트를 도청해 놓는다. 일정 시간이 흐른 후, 공격자는 도청된 등록요청 메시지를 FA에게 보내고, FA는 해당 메시지를 HA에게 보낸다. HA는 해당 메시지의 ID가 이전 등록응답에서 지정한 것과 상이함을 확인하고 에러 메시지를 FA에게 보낸다. 이때, HA의 에러 메시지가 FA에 도착되기 전에 공격자는 중간에서 FA에게 도청된 등록응답 메시지를 보내고 FA에서는 정당한 등록으로 처리하게 되는 것이다. 이로써 공격자는 불법적으로 외부 네트워크의 자원을 사용할 수 있게 된다. 이러한 공격에 대응하기 위한 제안으로는 시도-응답 프로토콜이 있지만, 이 방식에서는 확장 필드가 추가되며 FA는 자신이 생성한 수많은 시도 값에 대한 유지, 관리해야 하는 문제점이 있다. 본 논문에서는 시도-응답 프로토콜을 사용하지 않고 재생공격에 대응적인 대안을 제시한다. 기본 원리는 HA-FA 간에 매 등록마다 세션키를 사용하고, 해당 세션키를 FA가 주도하여 설정하도록 하는 것이다. 즉, HA-FA 간의 공유키를 K_{HA_FA}라 할 때, 해당 공유키를 기반으로 FA는 매 등록과정 마다 자신이 생성한 임의의 수 r을 기반으로 새로운 세션키 $K_{HA_FA}' = H^+(K_{HA_FA}, r)$를 생성하여 사용한다. 만약 시도-응답 프로토콜을 사용하지 않고 제안 프로토콜을 사용하고, 앞서 설명한 공격자의 재생공격이 발생한다고 가정하면 FA-MN 또는 HA-FA 간의 새롭게 설정된 세션키를 통해 재생공격 된 메시지는 인증에 실패하게 된다. 즉, FA에서는 재생된 메시지를 검출할 수 있고, 해당 메시지에 대해서는 에러처리를 하게 된다. 따라서 제안 프로토콜은 시도응답 프로토콜을 사용하지 않고도 재생공격에 대응할 수 있는 효율적인 프로토콜이라 할 수 있다.

5.2 성능평가

이번 절에서는 제안 프로토콜의 성능평가를 위해 AAA를 도입한 기본적인 MIPv4 방식[1,8]과 제안 프로토콜과 비교하게 된다. 비교를 위한 두 방식에서 AAAH는 키 분배 센터의 역할을 수행하며 HA와 FA에 의해서 등록과정이 수행된다. 프로토콜 비교의 간소화를 위해 편의상 2개의 도메인을 기준으로 하며, 홈 도메인과 외부 도메인에 속해 있는 HA와 FA들의 집합을 $H = \{HA_1, HA_2, \cdots, HA_l\}$과 $F = \{FA_1, FA_2, \cdots, FA_m\}$라 하고, K_{Home}을 공통의 홈 도메인 키라고 하자. 또한, MN_{ij}를 홈 도메인의 HA_i를 에이전트로 사용하는 j번째의 MN이라 하자. 이와 같은 환경에서 MN의 이동에 따른 등록과정에 필요한 전체 노드간의 교환 메시지 수를 비교하고, AAAH의 오버헤드 비교를 위해 해당 메시지에서 가장 큰 지연을 차지하는 AAAL과 AAAH 간에 교환되는 메시지 수를 비교한다. 두 프로토콜 간 비교는 홈 도메인의 MN_{11}이 외부 도메인의 FA_1로 이동했다가 $FA_2, \cdots, FA_n$까지 해당 도메인 내부에서 이동한 경우와 홈 도메인 내의 MN_{11}이 FA_1로 이동하고 연이어 홈 도메인 내의 HA에 관계없이 MN_{11}과 동일한 도메인 내의 $n-1$개의 임의의 MN_{ij}가 외부 도메인 내의 임의의 $FA_k (k \in [1, m])$로 이동한 경우에 대해 각각 비교하였다.

5.2.1 메시지 수 비교

첫 번째 경우는 홈 도메인의 MN_{11}이 외부 도메인의 FA_1로 이동했다가 $FA_2, \cdots, FA_m$까지 해당 도메인 내부에서 이동한 경우이다. 기존 프로토콜[1][8]은 각 개체간의 공유키 분배를 위해서 기본적으

로 7개의 메시지가 필요하며 공유된 키를 기반으로 MIPv4 등록 프로토콜을 위해 4개의 메시지가 필요하다. 이 경우 기존 프로토콜에서는 각 FA로의 이동에 따라 각각 키 분배와 등록과정을 수행해야 하므로 한 번의 이동에 대해 11번의 메시지가 필요하게 된다. 반면 제안 프로토콜은 각 개체간의 공유키 분배를 위해서 기본적으로 5개의 메시지가 필요하며 등록을 위해 3개의 메시지가 필요하고, 이후의 해당 도메인 간에 공유된 키를 기반으로 MIPv4 등록 프로토콜을 위해 6개의 메시지가 필요하다. 기존 기법보다 제안 프로토콜에서 키 분배를 위한 메시지 수가 감소한 것은 AAAH는 FA와 HA 간의 공유키만을 생성/분배하며 HA와 MN은 각각 직접 유도를 통해 키를 생성하므로 따로 전달할 필요가 없기 때문이다. 또한 제안 프로토콜에서는 도메인 키를 기반으로 하기 때문에 이와 같은 경우 초기등록 이후 AAAH가 관여하지 않기 때문에 AAAH의 부하를 줄일 수 있고 메시지 수 또한 현저하게 감소하게 된다. 이 경우 제안 프로토콜에서는 각 FA로의 이동에 따라 단 한 번의 키 분배와 등록과정을 위한 8개의 메시지와 $m-1$번의 이후의 등록과정을 위한 6번의 메시지가 필요하게 된다. 두 프로토콜의 메시지 수를 일반화 해보면 다음과 같으며, m이 증가할수록 제안 프로토콜은 더욱 효과적인 프로토콜이 된다.

기존 프로토콜 : $(7+4)m = 11m$
제안 프로토콜 : $8+6(m-1) = 6m+2$

두 번째 경우는 홈 도메인 내의 MN_{11}이 FA_1로 이동하고 연이어 홈 도메인 내의 HA에 관계없이 MN_{11}과 동일한 도메인 내의 $n-1$개의 임의의 MN_{ij}가 외부 도메인 내의 임의의 FA_k로 이동한 경우이다. 이 경우 기존 프로토콜은 각각의 MN은 AAAH 를 통한 키 분배 과정과 등록과정을 n번 수행해야 한다. 반면 제안 프로토콜은 도메인 키 개념을 기반으로 하기 때문에 한번의 키 분배와 등록과정을 위한 8개의 메시지와 $n-1$번의 이후의 등록과정이 필요하게 된다. 예를 들어, MN_{12}가 FA_3을 통해서 등록을 할 경우에 HA_1과 FA_3간에 설정되는 공유키 K_{HA_FA}는 홈 도메인의 도메인 키 K_{Home}을 기반으로 AAAH에 의해 계산되어 AAAL에게 제공되고 인증 캐쉬(Authentication cache)에 저장된다. 차후에 MN_{45}가 FA_6을 통해서 등록을 할 경우에는 HA_4와 FA_6간에 설정되는 공유키 K_{HA_FA}는 역시 동일하기 때문에 AAAH와 AAAL간의 메시지 교환은 생략된다. 또한 기존방식[1,8]에서는 AAAH에서 각 개체 사이에 공유할 키를 모두 생성해 주는 것과는 달리 본 논문에서는 HA와 MN은 자신이 가지고 있는 키 재료를 사용하여 사용할 키를 유도해 낼 수 있다. 이와 같은 MN과 HA에서의 자동 키 설정 기능은 메시지 수를 현저하게 감소시킬 수 있게 된다. 메시지 수를 일반화 해보면 다음과 같으며, n이 증가할수록 제안 프로토콜은 더욱 효과적인 프로토콜이 된다.

기존 프로토콜 : $(7+4)n = 11n$
제안 프로토콜 : $8+6(n-1) = 6n+2$

AAA를 기반으로 등록과정을 수행할 때 가장 큰 지연을 차지하는 부분은 AAAL과 AAAH의 메시지 교환부분이다. 따라서 이번에는 등록에 필요한 전체 메시지 수 중 AAAH의 오버헤드 비교를 위해 AAAL과 AAAH 간에 교환되는 메시지 수를 비교한다. 다음의 표 1에서 첫 번째 경우를 case i, 두 번째 경우를 case ii로 표시한다.

표 1. AAA 간의 메시지 수 비교

구 분	case i	case ii
기존기법	$2m$	$2n$
제안기법	2	2

위에서 살펴보면, 기존 프로토콜은 $2m$번 $2n$번의 AAAL과 AAAH와의 메시지 교환이 필요하고 제안 프로토콜은 각각 2번의 메시지 교환이 필요하다. 이는 선형 복잡도 $O(n)$에서 상수 복잡도 $O(1)$로 개선한 매우 효율적인 프로토콜이라 할 수 있다.

Ⅵ. 결 론

MIPv4 등록 프로토콜에서는 안전성과 효율성을 제고하기 위한 다양한 요구사항이 존재한다. 본 논문에서는 AAA 방식을 기반으로 인증을 위한 각 개체에서의 키 관리 문제점을 해결하였으며, 도메인 키와 인증캐쉬 개념을 도입함으로써 AAA의 오버헤드를 최소화하였다. AAA의 오버헤드 최소화는 네

트워크를 이동하는 메시지 수를 현저하게 줄일 수 있으며 이는 네트워크 성능 향상에도 커다란 기여를 할 수 있다. 또한 제안 프로토콜은 인증뿐 아니라 네트워크 서비스를 제공함에 따라서 발생되는 과금 문제를 해결하기 위한 부인방지 서비스도 제공하며 다양한 유형의 재생공격에도 대응적이다. 새로운 프로토콜의 설계에는 안전성과 더불어 중요한 부분이 효율성 측면이다. 본 논문에서 제안한 프로토콜은 성능평가에서 볼 수 있듯이 안전성과 함께 효율성을 만족하는 MIPv4 등록 프로토콜이다.

참 고 문 헌

[1] C. Perkins, Ed., "AAA Registration Keys for Mobile IP," Internet Draft, *⟨draft-ietf-mobileip-aaa-key-12.txt⟩*, May 2003.

[2] C. Perkins, Ed., "IP Mobility Support," *RFC2002*, October 1996.

[3] C. Perkins, Ed.,"IP Mobility Support for IPv4," *RFC3344*, August 2002.

[4] C. Perkins, Ed.,"IP Mobility Support for IPv4, revised," Internet Draft,*⟨draft-ietf-mip4-rfc3344bis-00.txt⟩*,June 2004.

[5] C. Perkins,"Mobile IPv4 Challenge /Res-ponse Extensions,"*RFC3012*, November 2000.

[6] C. Perkins, "Mobile IPv4 Challenge /Res-ponse Extensions (revised)," Internet Draft,*⟨draft-ietf-mip4-rfc3012bis-02.txt⟩*, June 2004.

[7] L. Lamport, "Password authentication with insecure communication. Commun "", *Mag. of ACM*, 24 (11), pp. 770-772. 1981.

[8] S. Glass, T. Hiller, S. Jacobs, C. Perkins, "Mobile IP Authentication, Authorization and Accounting Require-ment," *RFC2977*, October 2000.

[9] S. Jacobs, "Mobile IP Public Key Based Authentication," Internet Draft, *⟨draft-jacobs-mobileip-pki-auth-00.txt⟩*, August 1998.

[10] Sufatrio, Kwok Yan Lam, "Mobile IP Registration Protocol: A Security Attack and New Secure Minimal Public-Key Based Authentication," *ISPAN'99*, June 1999

[11] 김현곤, "AAA 기반 Mobile IP 환경에서 안전 하고 빠른 핸드오프 기법 설계", *정보보호학회 논문지*, 14호1권, 2004. 2.

⟨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