情報保護學會論文誌
第12券 第2號, 2002. 4

적응적 서비스 모드에 기반한 이동보안멀티캐스트 구조 및 프로토콜에 관한 연구*

안 재 영**, 구 자 범**, 이 재 일***, 박 세 현**

A Study of Secure Mobile Multicast Architecture and Protocol based on Adaptive Service Mode

Jae Young Ahn**, Ja Beom Gu**, Jae Il Lee***, Se Hyun Park**

요 약

본 논문은 이동환경에서 다수의 이동 단말을 대상으로 안전하고 효율적인 멀티캐스트를 실현하는 이동보안멀티캐스트 구조 및 프로토콜을 제안하고자 한다. 제안된 이동보안멀티캐스트 구조 및 프로토콜은 이동 단말의 움직임에 따라 적응적으로 이동보안멀티캐스트 서비스를 제공해 줌으로써, 전송지연을 낮추고 키갱신 횟수를 줄임과 동시에 이동 단말의 이동으로 인한 서비스 끊김 및 데이터 손실을 방지하여 이동보안멀티캐스트 서비스의 QoS를 높였다.

ABSTRACT

In this paper, we propose an architecture and a protocol for Secure Mobile Multicast(SMM) offering efficient and secure multicast services to many mobile nodes. In this framework, we use Indirect and Direct Service Mode adaptively, according to the movement of mobile nodes around the overlapped service area, to provide reliably secure multicast with low latency, minimum key update, and minimum data loss.

keyword : *Secure mobile multicast*

Ⅰ. 서 론

현재 인터넷의 주요 동향 중 하나는 무선환경에서의 이동성에 다양한 응용 프로토콜을 결합하여 양질의 서비스를 제공하는데 있다. 한편, 특정 다수를 대상으로 한 실시간 멀티미디어통신의 필요성이 나날이 대두됨에 따라서, 현재 인터넷의 네트워크 자원과 구조적 한계를 극복하기 위한 여러 제안과 시도가 진행되고 있다. 대표적으로 네트워크 자원을 절약하여 효율적으로 특정 다수의 수신자들에게 데이터를 전송하는 멀티캐스트(multicast)[16]은 차세대 무선 인터넷의 중요한 역할을 할 것으로 기대된다. 또한, 인터넷상의 특정 다수 사용자들을 대상으로 하는 유료 시청(pay per view) 및 실시간 증권정보 서비스를 위한 응용 프로토콜들은 그 특성상 멀티캐스트를 이용한 구현과 동시에 보안이 반드시 필요하다. 이러한 보안멀티캐스트(secure multicast)[2]를 무선 인터넷에서 구현할 경우, 시스템과 네트워크 자원을 잘

* 이 논문은 2001학년도 중앙대학교 학술연구비, 한국과학재단(R01-2001-00303), 한국정보보호진흥원 및 한국전자통신연구원의 지원에 의한 결과임
** 중앙대학교 전자전기공학부 인터넷세계 보안연구실(shpark@cau.ac.kr)
*** 한국정보보호진흥원(KISA)

관리함과 동시에 이동성을 충분히 고려하여야 한다.

보안멀티캐스트는 그룹키(group-key)에 의한 멀티캐스트 데이터의 암호화를 통하여 이루어지고 있다. 보안멀티캐스트에서 가장 중요한 점은 그룹 멤버의 가입과 탈퇴시의 키관리 문제이다. 이는 크게 두 가지 측면에서 고찰해 볼 수 있다. 첫째는 멀티캐스트 데이터에 대한 접근 권한에 관한 문제이다. 즉, 멀티캐스트 그룹에 가입된 멤버들만 멀티캐스트 그룹키를 소유함으로써, 멀티캐스트 데이터를 복호화 하여 볼 수 있다. 두 번째는 키갱신의 효율성이다. 멀티캐스트 데이터에 대한 접근권한의 문제를 해결하기 위해서는 그룹 멤버의 가입과 탈퇴 시에는 필연적으로 그룹키의 갱신이 뒤따른다. 양질의 QoS를 제공하는 안전한 보안멀티캐스트 서비스를 제공하기 위해서는 키갱신은 신속하고 효율적으로 이루어져야 한다. 이를 위하여 현재 많은 연구들이 진행되고 있다[3~5]. 하지만 이동 환경에서는 멤버들의 이동으로 인하여 키관리 문제는 보다 복잡한 양상을 띤다. 이동 환경에서 멀티캐스트를 구현하려는 연구[6~8]는 활발하게 진행되고 있지만, 키관리에 많은 어려움을 갖고있는 이동 환경에서의 보안멀티캐스트에 대한 연구는 아주 미미한 실정이다.

하지만 최근 이동 통신 환경에서는 보안멀티캐스트를 요구하는 다양한 응용 분야의 필요성이 제기되고 있다. 이에 본 논문은 잦은 움직임을 갖는 이동 단말에게 멀티미디어 유료 시청이나 실시간 증권정보 등과 같은 전송지연과 데이터 손실에 민감한 멀티캐스트 서비스를 제공하기에 적합한 이동보안멀티캐스트(Secure Mobile Multicast : 이하 SMM) 구조 및 프로토콜을 제시하고자 한다. 제안된 SMM 구조는 이동 단말의 움직임에 따라 적응적으로 SMM 서비스를 제공해 줌으로써, 전송지연을 낮춤과 동시에 이동 단말의 이동으로 인한 서비스 끊김 및 데이터 손실을 방지하여 SMM 서비스의 QoS를 높였다. 또한 빈번한 가입·탈퇴와 함께 이동 단말의 이동으로 인해 키갱신이 요구될 때, 이를 일정기간 모아서 처리하는 Batch Process에 의해 처리함으로써 키갱신의 빈도를 줄였다. 아울러, 이동 단말에 대한 데이터 베이스를 분산하여 관리함으로써 중앙 서버의 부담을 덜어주었다.

본 논문은 다음과 같이 구성되었다. 2장에서는 본 논문과 관련된 연구에 대해 설명한다. 3장에서는 제안된 SMM 구조에 관한 기본적 원형을 알아보고, 4장에서는 구체적인 SMM 프로토콜을 제시한다. 5장에서는 시뮬레이션 및 결과를 살펴보고, 6장에서 본 논문의 결론을 맺는다.

Ⅱ. 배경지식

2.1 보안멀티캐스트

멀티캐스트에 보안을 적용하는 보안멀티캐스트에 관한 연구가 활발히 진행되고 있다[2~5]. 보안멀티캐스트에서는 그룹키를 통한 데이터의 암호화로 데이터를 보호하고 있다. 보안멀티캐스트에서는 그룹에 가입한 멤버들만이 멀티캐스트 데이터에 접근하게 해야 한다. 즉, 한 멤버가 새로 보안멀티캐스트 그룹에 가입할 때, 그 멤버는 그전의 멀티캐스트 데이터에 접근할 수 없어야 하며, 이와는 반대로 한 멤버가 그룹을 탈퇴했을 경우, 그 멤버는 그 후의 멀티캐스트 데이터에 접근할 수 없게 하여야 한다. 따라서 멤버의 가입과 탈퇴 시에는 그룹키를 갱신하는 것이 필연적이라고 할 수 있다. 하지만 그룹키의 갱신에는 시스템 및 네트워크 측면에서 많은 오버헤드가 유발될 수 있고, 또한 보안적 측면에서도 많은 문제점들을 야기할 수 있다. 따라서 보안멀티캐스트에서는 그룹키를 얼마나 효율적으로 관리하는가가 매우 중요한 부분을 차지하고 있다.

기본적으로 생각해 볼 수 있는 키관리 방법에는 그룹 콘트롤러(group controller)에 의존하는 중앙 집중적인 키관리 방법을 들 수 있다. 이 방법의 가장 단순한 구조는 그룹 콘트롤러가 그룹의 멤버들에게 안전한 유니캐스트(unicast) 채널을 통해 그룹키를 나눠주는 형태이다. 하지만 이는 그룹의 크기에 비례해 암호화 비용 및 네트워크의 트래픽을 증가시키므로 많은 오버헤드를 갖고 있다. 이를 해결하기 위해서 [3]에서는 계층적 키 트리(key tree)를 이용하여 이 문제를 해결하였다. 그 결과로 한번의 키갱신이 있을 때, 암호화 비용을 대수적(logarithm)으로 줄여주는 한편, 한번의 브로드캐스트(broadcast)로써 키갱신을 수행함으로써 네트워크 자원을 절약하였다.

하지만, 하나의 그룹키를 모든 멤버들이 공유하는 중앙 집중적인 키관리 방법은 여전히 많은 문제점을 안고 있다. 첫째는 '1 affects n'[4] 이란 말로 잘 설명될 수 있다. 전술한 바와 같이 한 멤버의 가입과 탈퇴 시에는 그룹은 키갱신을 수행하여야 한다. 따라서 멤버의 가입과 탈퇴와는 무관한 멤버들이 잦은

키갱신을 수행하는 비효율성을 낳는다. 또한, 이러한 잦은 키갱신은 멤버들에게 이 키갱신 메시지를 놓칠 수 있는 확률을 증가시킨다. 둘째로, 키갱신의 비일관성에 관한 문제이다. 과다한 트래픽이나 네트워크 장비의 이상으로 인하여 하나의 송신자로부터의 데이터 패킷은 넓은 지역에 분산되어 있는 다수의 수신자에게 같은 시간에 전송될 수 없다. 이러한 특성 때문에 키갱신 메시지를 전달하는 경우에 어떤 멤버들은 다른 멤버들보다 먼저 이 메시지를 수신하거나 어떤 멤버들은 이 키갱신 메시지를 놓칠 수 있게 된다. 따라서 키갱신 메시지를 늦게 받은 멤버들은 일정기간 SMM 서비스를 받지 못하거나, 악의를 품은 그룹의 전 멤버가 예전의 그룹키로 위조된 메시지를 다른 멤버들에게 보내는 문제 등이 발생할 수 있다. 이와 같은 문제를 해결하기 위하여 분산구조 형태로 키를 관리하는 방법이 대두하게 되었다. Iolus[4]는 분산구조의 대표적인 예로 볼 수 있다.

2.2 이동멀티캐스트

이동 환경에서 멀티캐스트를 구현하려는 노력은 여러 각도로 이루어지고 있다. 그 중 하나가 Mobile IP상에서 멀티캐스트를 구현하는 것이다[9]. Mobile IP에서는 두개의 IP를 사용한다. 하나는 Home Address로서 Home Network에 위치할 때 사용한다. 또, 다른 IP는 Care-of-Address로서 이동 단말이 Foreign Network에 위치할 때 끊김 없는 데이터 전송을 받기 위해 사용한다. Home Network에 위치하여 이동 단말에게 서비스 해주는 Home Agent(HA)와 Foreign Network에 위치하여 이동 단말에게 서비스 해주는 Foreign Agent(FA)는 긴밀한 상호작용을 통해 이동 단말의 이동성을 뒷받침 해준다.

이러한 Mobile IP에서 멀티캐스트를 구현하려는 두 가지 방법에 대해 [7]에서는 다음과 같이 기술하고 있다.

첫째, Remote Subscription이라고 불리는 방법으로, 이는 이동 단말이 Foreign Network로 이동했을 경우 이동 단말은 자신이 위치한 곳(Foreign Network)에서 멀티캐스트 서비스를 받는 방법이다. 이 방법의 가장 큰 장점은 멀티캐스트 데이터의 전송경로가 이상적이라는 것이다. 따라서, 전송 지연이 작게 된다. 하지만 이 경우 source mobility가 보장되지 않기 때문에 이동 단말들이 모두 수신자임을 가정한다. 더욱이, 모든 네트워크에는 멀티캐스트를 지원하는 라우터들이 갖추어져 있어야 하기 때문에 시스템 자원 측면에서 오버헤드가 있다.

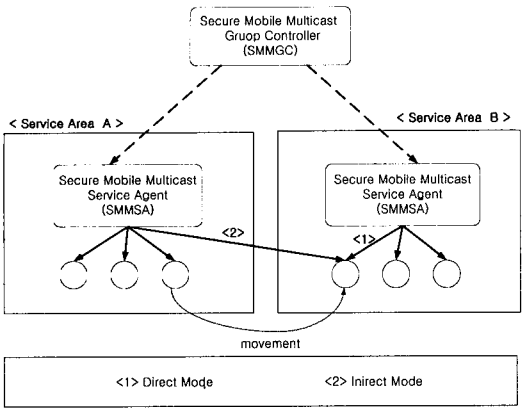
둘째, Bi-directional tunneling이라고 불리 우는 방법으로 HA가 이동 단말이 현재 위치한 위치로 터널링(tunneling)을 통한 유니캐스트로 멀티캐스트를 지원하는 방법이다. 이 경우 source mobility와 recipient mobility를 둘 다 지원한다. 그러나 이 방법의 가장 큰 단점은 멀티캐스트 경로가 매우 길어질 수 있다는 점이다. 또한, HA는 자신이 멀티캐스트 서비스를 해주고 있는 이동 단말이 같은 Foreign Network에 위치하고 있더라도, 각각 하나씩의 중복된 멀티캐스트 데이터를 보낼 수 있다. 따라서, 네트워크 자원이 낭비된다.

[7]에서는 이 두 가지 모드를 거리에 따라서 조합해 사용함으로써 전송지연을 최소화하는 방안을 제안하였는데, 멀티미디어 유료 시청이나 실시간 증권 정보의 경우처럼 이동 단말이 수신자 입장에서 보안멀티캐스트 서비스를 받는 경우 전송 지연뿐만 아니라 보안 멀티캐스트 서비스 제공시 이동중인 단말 서비스 영역간을 이동할 때 발생하는 데이터 손실을 최소화 할 수 있어야 한다. 따라서, 본 논문에서는 SMM 구조로 이러한 문제점을 해결하고자 한다.

Ⅲ. 제안하는 SMM 프로토콜

3.1 SMM 기본 구조

본 논문은 잦은 움직임을 갖는 이동 단말에게 전송지연과 데이터 손실에 민감한 멀티캐스트 서비스를 제공하기에 적합한 이동보안멀티캐스트 구조 및 프로토콜을 제시한다. 제안된 SMM 구조는 서비스 영역의 경계에서 이동 단말의 움직임에 따라 적응적으로 SMM 서비스를 제공해 줌으로써, 전송지연을 낮춤과 동시에 이동 단말의 이동으로 인한 서비스 끊김 및 데이터 손실을 방지하여 SMM 서비스의 QoS를 높였다. 이를 위하여 분산구조를 형성하고 Direct Mode 및 Indirect Mode를 적응적으로 적용하였다. 한편 빈번한 가입·탈퇴와 함께 이동 단말의 이동으로 인해 키갱신이 요구될 때 이를 일정기간 모아서 처리하는 Batch Process를 제시 적용함으로써 키갱신의 빈도를 줄였다. 이러한 SMM의 기본 구조는 [그림 1]과 같으며, 제안된 SMM 구조의 구성은 다음과 같다.



〔그림 1〕 SMM의 기본 구조

SMM은 넓은 지역에서의 확장성 있는 서비스를 위하여 Iolus[4]와 같은 분산구조를 갖는다. 즉, 일정한 범위로 서비스 영역을 나누고 각 서비스 영역별로 서비스 영역키를 갖는다. 이 서비스 영역키는 해당 서비스 영역에 등록되어 있는 이동 단말에게 멀티캐스트 데이터를 암호화하여 서비스 해 줄 때 사용된다. 각 서비스 영역에는 서비스 영역키를 관리하고 이동단말에게 SMM 서비스를 제공하는 SMMSA (Secure Mobile Multicast Service Agent)가 위치한다. 또한 중앙에는 멀티캐스트 서비스를 총괄하고 SMM 서비스의 소스(source)인 SMMGC (Secure Mobile Multicast Group Controller)가 위치한다. SMMGC로부터 SMM 그룹에 가입한 각 멤버들에게 SMM 패킷이 전달되기 위해서 SMMGC와 SMMSA들과 보안멀티캐스트 트리가 형성된다. SMMGC에서 각 SMMSA까지 패킷이 안전하게 전달되기 위해서 SMMGC와 SMMSA들은 하나의 비밀키(secrete key)를 공유하게 되고, SMM 패킷은 이 비밀키로 암호화되어 SMMGC와 SMMSA들 사이에 형성된 멀티캐스트 트리를 통해 멀티캐스트 되어 전달된다. 각 SMMSA는 전달받은 멀티캐스트 패킷을 복호화 한 후에 다시 각각의 서비스 영역키로 암호화되어 이동 단말에게 전달된다. 한편 각 SMMSA 사이에 관리 패킷(control packet)을 주고받기 위해서, 각 SMMSA들은 각각 둘 사이에 공유하는 비밀키를 가지고 있다.

서비스영역 내에서는 서비스 영역키로 SMM 패킷을 암호화하여 이동 단말에게 서비스 해주는데, 이동 단말의 가입이나 탈퇴, 이동 등에 의하여 그룹키를 갱신하게 될 때에는 [3]에서 제시된 계층적 키트리를 이용하여 키갱신을 수행하게 된다. 한편, 각 이동 단말에 대한 정보는 각 이동 단말의 Home SMMSA에 의해 관리하는 분산구조를 취하였다.

본 논문에서는 Direct Mode와 Indirect Mode의 두 가지 SMM 서비스 모드를 적응적으로 적용하여 이동 단말에게 낮은 전송지연을 보장해주고, 데이터 손실을 방지하고 끊김 없는 SMM 서비스를 제공할 수 있는 프로토콜을 제시한다. 본 논문에서 제시하는 프로토콜은 전송지연 측면에서 유리한 Direct Mode를 기반으로 하고, 부가적으로 이동 단말이 새로운 서비스 영역으로 이동할 경우를 지속적 서비스 보장을 위해 Indirect Mode를 제한적으로 사용한다. Direct Mode는 〔그림 1〕의 〈1〉처럼 자신이 위치한 서비스 영역으로부터 SMM 서비스를 받는 방법이고, Indirect Mode는 〔그림 1〕의 〈2〉처럼 Tunneling[13]을 통하여 간접적으로 서비스 받는 방법이다. 이에 대한 자세한 내용은 4장에서 다루도록 한다.

3.2 SMM의 구성

이 절에서는 본 논문에서 제시한 SMM 구조를 이루고 있는 각각의 구성 및 용어에 대해 정의 내리고 설명한다.

- 〈*Group Join*〉: 이동 단말이 SMM 서비스를 받기 위해, SMM 그룹 자체에 처음 가입하는 과정이다. 이때 이동 단말은 자신의 ID와 SMM 서비스에 쓰일 Password, 자신의 공개키를 담고 있는 인증서 및 기타 정보들을 담은 *Group Join Request* 메시지를 Home SMMSA에게 제출하게 된다.
- 〈*Service Join*〉: 〈*Group Join*〉된 후 이동 단말이 현재의 SMM 서비스를 받기 위해 서비스에 참여하는 과정이다. 이때, 이동 단말은 자신의 ID와 Password 등을 포함한 *Service Join Request* 메시지를 자신이 서비스 받고자 하는 SMMSA에게 보낸다.
- 〈*Group Leave*〉: 이동 단말이 SMM 그룹을 영구히 탈퇴하는 것을 말한다. 이동 단말은 자신의 Home SMMSA에게 *Group Leave Request* 메시지를 보내게 된다. 이때 Home SMMSA에 저장되어 있는 이동 단말에 대한 데이터는 삭제된다.
- 〈*Service Leave*〉: SMM 서비스를 받던 이동 단

말이 현재의 SMM 서비스를 중단 받으려고 탈퇴하는 과정이다. 이동 단말은 현재 자신을 서비스 해주고 있는 SMMSA에게 *Service Leave Request* 메시지를 보내게 된다. 이때 Home SMMSA에 저장되어 있던 이동 단말에 대한 데이터는 삭제되지 않는다.

- 〈*Registration*〉: 이동 단말이 SMM 서비스를 받기 위해 SMMSA에 등록하는 과정이다. 이동 단말은 등록을 원하는 SMMSA에 *Registration Request* 메시지를 전송한다. 〈Service Join〉된 이동 단말이 새로운 서비스 영역으로 이동해서 Direct Mode 서비스를 받기 위해서는 이동한 SMMSA에 등록해야 하므로, 이때 *Registration Request* 메시지를 해당 SMMSA에게 보낸다. 한편, 〈*Service Join*〉은 〈*Registration*〉 과정을 내포하고 있어서 이동 단말이 새로운 서비스 영역으로 이동할 때 해당 SMMSA에 등록하는데 사용한다.
- 〈*Deregistration*〉: 이동 단말이 등록을 해지하는 과정으로, 〈*Service Leave*〉의 경우와 Indirect Mode로 서비스를 받던 이동 단말이 *Indirect Stop Request* 메시지를 보낼 때 발생한다.
- Home SMMSA: 이동 단말이 〈*Group Join*〉할 경우 이를 담당했던 SMMSA이다. Home SMMSA는 이동 단말이 〈*Group Leave*〉하기 전까지, 이동 단말에 대한 정보를 저장하고 있다. Home SMMSA는 이동 단말의 〈*Group Join*〉을 담당할 뿐 아니라 이동 단말과 관련된 *Group Leave Verify Request, Service Join Verify Request, Service Leave Verify Request, Registration Verify Request* 등의 메시지를 검증하여 이동 단말에 대한 서비스 유효성에 대한 결과를 해당 SMMSA에게 알려준다. 한편 이동 단말에 대한 정보를 Home SMMSA가 나누어서 관리하게 되므로, 완전한 분산체제를 성립할 수 있다.
- SMMSP(SMM Service Provider): 이동 단말 입장에서 현재 SMM 서비스를 제공하는 SMMSA를 말한다. SMMSP는 Direct Mode와 Indirect Mode로 자신에게 등록된 이동 단말에게 SMM 패킷을 전달해준다.
- Join Latency: 이동 단말이 〈*Service Join*〉 과정에 생기는 지연으로, 이동 단말이 *Service Join Request*를 한 후 해당 SMMSA로부터 서비스 영역키를 받는 데까지 걸리는 시간이다.
- Leave Latency: 이동 단말이 〈*Service Leave*〉 과정에 생기는 지연으로, 이동 단말이 *Service Leave Request*를 한 후 해당 서비스 영역의 키가 갱신되는 데까지 걸리는 시간이다.
- Registration Latency: 이동 단말이 〈*Registration*〉 하는데 걸리는 지연으로, 이동 단말이 *Registration Request* 메시지를 보내고 해당 SMMSA로부터 서비스 영역키를 받는 데까지 걸리는 시간이다.

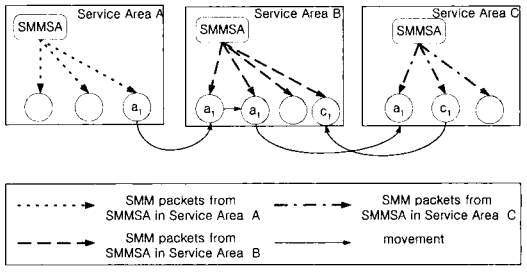
3.3 Direct Mode

SMM은 보안 QoS를 고려하여 두 가지 방식의 서비스 모드를 혼합하여 사용한다. 그 중 기본적으로 사용되는 모드가 Direct Mode이다. Direct Mode ([그림 2])는 Mobile IP의 Remote Subscription[9] 방법과 같이 현재 자신이 위치한 서비스 영역으로부터 직접 SMM 서비스를 받는 방식이다. 따라서 이동 단말이 SMM 패킷을 받는 경로가 이상적이다.

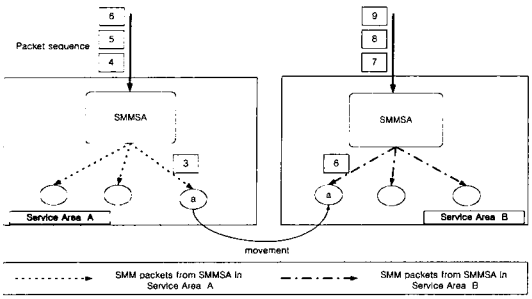
[그림 2]에서 볼 수 있듯이 초기에 서비스 영역 A에 위치하여 서비스 영역 A로부터 SMM 서비스를 받던 이동 단말 a_1은 서비스 영역 B로 이동하면 서비스 영역 B의 SMMSA에 등록하여 서비스 영역 B로부터 SMM 서비스를 받는다. 또한, a_1이 서비스 영역 C로 이동하면, 위와 마찬가지 과정으로 서비스 영역 C로부터 서비스를 받는다. 즉, 이동 단말은 항상 자신이 위치한 서비스 영역에 등록하여 서비스를 받는 방법이 Direct Mode이다.

하지만 이동 단말이 한 서비스 영역에서 새로운 서비스 영역으로 이동했을 때 보안 서비스 등으로 인한 오버헤드로 다음과 같은 두 가지 문제가 발생할 수 있다.

첫째, 이동 단말이 새로운 서비스 영역에 등록되어 서비스 받기까지, 즉 Registration Latency 동안 SMM 서비스 공백기간이 생기게 된다. 이 Registration Latency 동안은 이동 단말에게 SMM 서비스가 끊



[그림 2] Direct Mode



〔그림 3〕 SMM 패킷 시퀀스의 비일관성

기게 되고, 결국 데이터의 손실이 생기게 된다. 2장에서 설명했듯이 너무 잦은 키갱신은 멤버들이 키갱신 메시지를 놓칠 확률을 증가시키고, 또한 키노출 확률도 증가시킨다. 이러한 문제를 해결하기 위해서 본 논문에서는 3.6 절에서 제안하는 Batch Process에 의한 키갱신을 수행한다. 그러나 이러한 과정은 멤버들의 Join latency, Leave latency, Registration Latency를 증가시키는 요인이 될 수도 있다. 따라서, Direct Mode만으로 SMM 서비스를 제공한다면, 키갱신으로 증가되는 Registration Latency로 양질의 끊김 없는 서비스가 어렵다.

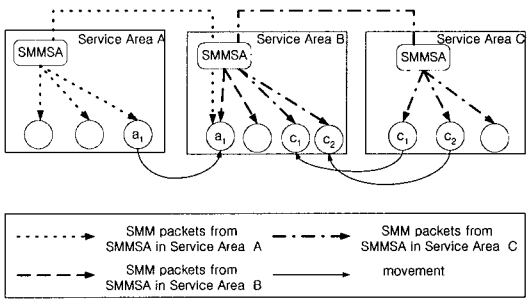
둘째, 전송지연에 따른 각 서비스 영역마다의 SMM 패킷 시퀀스(sequence)의 비일관성 문제가 발생할 수 있다. 〔그림 3〕과 같이 이동 단말 a_1이 서비스 영역 A로부터 서비스 영역 B로 이동했을 경우, SMMGC로부터 각 서비스 영역에 전달되는 패킷은 전송지연으로 인하여 전달시간의 차이가 발생한다. 따라서, 서비스 영역 A가 4, 5, 6번째 SMM 패킷을 받는 동안 서비스 영역 B는 7, 8, 9번째 SMM 패킷을 받는 경우가 발생될 수 있다. 이때 서비스 영역 A에서 3번째 SMM 패킷을 받은 이동 단말은, Registration Latency가 아주 작다고 하더라도 패킷 시퀀스의 불일치로 서비스 영역 B에서는 6번째 패킷부터 받는 경우가 발생할 수 있다. 따라서, 이 경우 4, 5번째 SMM 패킷을 잃게 된다. 각각 20%의 비율로 0, 5, 40, 70, 100Km/h의 속도로 이동하는 이동 단말이 500byte의 패킷 사이즈로 CBR(Constant Bit Rate) 128Kbps의 데이터를 받는 경우를 시뮬레이션 한 결과, 패킷 시퀀스의 비일관성으로 야기되는 패킷의 손실률은 약 3.8 %정도를 보였다(5장 참고).

이러한 두 가지 주요한 이유로, 이동 단말에게 SMM 서비스를 제공함에 있어 Direct Mode를 보완해줄 방안이 요구되며, 본 논문에서는 다음에 설명되는 Indirect Mode를 적응적으로 적용하여 문제를 해결하고자 한다.

3.4 Indirect Mode

SMM의 또 다른 모드 중 하나는 Indirect Mode이다. Indirect Mode(〔그림 4〕)는 자신에게 등록되어 있는 이동 단말 중 자신의 서비스 영역에 위치하지 않은 이동 단말에게 간접적으로 tunneling[13]을 통해 SMM 서비스를 제공하는 모드이다. Mobile IP의 Bidirectional tunneling은 이동 단말이 어디에 위치하든지 항상 HA로부터 tunneling을 통해서 패킷을 받지만, 이 경우에 라우팅 경로가 매우 길어질 수 있다. 따라서, 본 논문에서는 Indirect Mode를 앞 절에서 살펴본 SMM 서비스를 위한 Direct Mode의 문제점을 보완하기 위해 제한적으로만 사용하도록 제안한다.

〔그림 4〕는 서비스 영역 A에 위치했던 이동 단말 a_1이 서비스 영역 B로 이동하여 서비스 받는 과정과, 서비스 영역 C에 위치했던 이동 단말 c_1, c_2가 서비스 영역 B로 이동하여 서비스 받는 과정을 보여준다. 이동 단말은 각 서비스 영역의 SMMSA가 주기적으로 브로드캐스트하는 *SMMSA Advertisement* 패킷을 받고, 이를 기준으로 자신의 위치를 파악한다. 이동 단말 a_1이 서비스 영역 A와 B의 경계 부근에 다가가면, a_1은 A와 B로부터 모두 *SMMSA Advertisement* 패킷을 받음으로써, A와 B의 경계에 다다른 것을 파악한다. 따라서, 이때 a_1은 A로부터 B쪽으로의 Indirect Mode Request 메시지를 보낸다. 이때부터 a_1은 A로부터 Indirect Mode로 서비스 받다가 더 이상 A의 *SMMSA Advertisement* 패킷을 못 받으면, 자신이 완전히 B로 이동했음을 인지하고 B에게 *Registration Request*를 전송한다. a_1은 계속 Indirect Mode로 서비스 받다가 B에

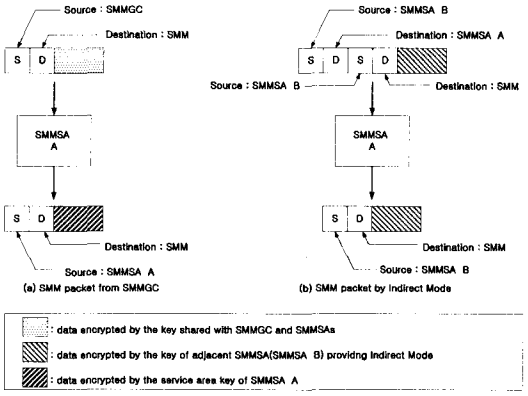


〔그림 4〕 Indirect Mode

등록과정을 마치면 Direct Mode로 서비스 받게 된다. 이때 각 서비스 영역의 패킷 시퀀스의 불일치로 인하여 데이터 손실이 발생할 수 있으므로, 본 논문에서 제시한 SMM 서비스를 위한 Indirect Mode는 다음과 같은 두 가지 방법을 사용한다. 첫째로, 만약 Direct Mode로 받은 패킷의 시퀀스 번호가 Indirect Mode로 받은 마지막 패킷의 시퀀스보다 클 경우에는, Direct Mode로부터 받은 첫 번째 패킷의 시퀀스 번호와 동일한 중복된 패킷을 받을 때까지 Indirect Mode로 서비스를 계속 받는다. 즉, 일정기간 Indirect Mode, Direct Mode의 두 모드로 함께 서비스 받는다. Indirect Mode로 받은 패킷의 시퀀스 번호와 Direct Mode의 첫 번째 패킷의 시퀀스 번호가 같아지는 순간 이동 단말은 Indirect Mode로 서비스를 제공해주던 SMMSA에게 *Indirect Mode Stop Request* 메시지를 보내게 된다. *Indirect Mode Stop Request* 메시지를 받은 SMMSA는 해당 이동 단말을 자신의 Service List에서 삭제하고 〈*Deregistration*〉 시킨다. 동시에 이동 단말은 Direct Mode로 완전히 전환하게 된다. 둘째, 만약 Direct Mode로 받은 패킷의 시퀀스 번호가 Indirect Mode로 받은 마지막 패킷의 시퀀스보다 작거나 같을 경우는 곧바로 *Indirect Mode Stop Request* 메시지를 보내어 Direct Mode로 전환한다. 〔그림 4〕에서 c_1과 c_2는 서비스 영역 B에 등록되기 전 Indirect Mode로만 서비스 받고 있는 경우를 보여준다.

3.5 SMMSA에서의 SMM 패킷 처리

Indirect Mode는 이동 단말에게 끊김 없는 SMM 서비스를 제공하지만, 이때 각 이동 단말마다 Indirect Mode를 유니캐스트로 제공하는 것은 매우 비효율적이다[6]. 즉, Indirect Service List에 있는 이동 단말 중 같은 서비스 영역에 위치한 이동 단말 각각에게 중복되게 SMM 패킷을 보내는 것은 네트워크 자원의 낭비를 초래한다. MoM protocol[6]에서는 이 문제를 해결하기 위하여 DMSP(Designated Multicast Service Provider)란 개념을 도입해 오직 하나의 tunnel로부터 멀티캐스트 데이터를 받는 것을 제안했다. 그러나 이동보안멀티캐스트에서는 각각의 Indirect Mode를 제공하는 패킷을 암호화한 키가 다르기 때문에 〔6〕과 같은 방법을 사용할 수는 없다. 따라서 본 논문에서 제안한 SMM에서는 위와 같은 문제를 고려해서 SMMSA에서 SMM 패킷을 처리하도록 하였다(〔그림 5〕).



〔그림 5〕 SMMSA에서 SMM 패킷 처리

〔그림 5(a)〕는 Direct Mode를 보여준다. SMMSA A는 SMMGC로부터의 SMM 패킷을 수신하여 이를 복호화 한 후, 자신의 서비스 영역키로 암호화한 뒤 IP 헤더의 source address부분에 자신의 IP 주소를 적어 넣은 다음, 이 패킷을 서비스 영역 내에 형성된 SMM 멀티캐스트 트리를 따라 멀티캐스팅 해 줌으로써, 자신에게 등록된 이동 단말에게 SMM 서비스를 제공한다. 〔그림 5의(b)〕는 SMMSA B가 SMMSA A에게 SMM 패킷을 tunneling하여 Indirect Mode를 제공하고 있는 상황을 보여 준다. SMMSA B는 자신의 Indirect Service List에서 하나 이상의 이동 단말이 서비스 영역 A에 존재하는 것을 확인한 후, SMMSA A에게 하나의 SMM 패킷만을 tunneling 하여 보내준다. 이 패킷을 받은 SMMSA A는 외부 IP헤더를 제거하고, 이를 자신의 서비스 영역에 형성된 멀티캐스트 루트를 따라 보낸다. 이 패킷의 source address부분은 SMMSA B의 IP 주소가 적혀있고 또한 이 패킷은 SMMSA B의 서비스 영역키로 암호화되어 있다. 따라서, 이동 단말은 SMM 패킷을 수신한 후, 패킷의 source address부분을 확인하고, 이 source address가 자신의 SMMSP와 일치하는 패킷만을 복호화하여 수신하고, 일치하지 않는 패킷은 폐기함으로써, 네트워크 자원을 효율적으로 사용할 수 있다.

3.6 Batch Process에 의한 키관리

앞 절에서 이동 단말에게 SMM 패킷의 낮은 전송지연과 끊김 없는 SMM 서비스를 보장하기 위한

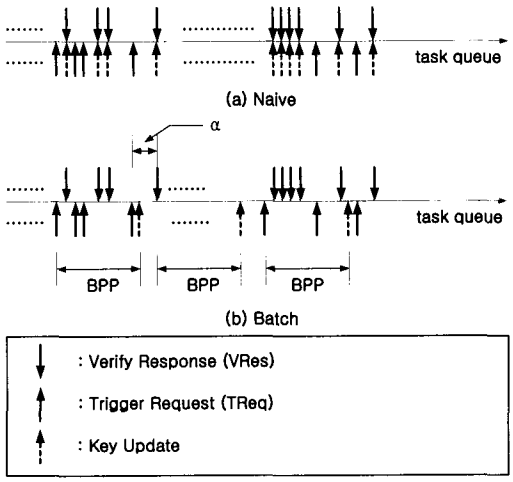
방안으로, SMM 구조를 Direct Mode와 Indirect Mode로 적응적으로 적용하였다. 이와 함께 본 절에서는 SMM 프로토콜에 사용되는 키관리 방법에 대해서 설명한다.

2장에서 소개한 계층적 키 트리 방법은 일정 수 이상의 멤버들을 관리하는데 아주 효율적인 방법이다. 비록 넓은 지역에 퍼져있는 다수에 대해서는 적용하기는 키갱신의 비일관성이나 잦은 키갱신 등의 문제점을 갖지만, 일정 크기의 그룹에 적용하기는 효율적인 방법이 될 수 있다. 따라서, 본 논문에서는 서비스 영역에서의 키갱신 방법으로 기본적으로 계층적 키 트리를 적용한 키갱신 구조를 채택하였다.

하지만, 이동 환경에서는 멤버들이 새로운 서비스 영역으로 이동해서 그 서비스 영역에 〈*Registration*〉 요청을 할 경우에도 그 서비스 영역은 효율적인 보안 유지를 위하여 키갱신이 요구된다. 즉, 멤버의 가입과 탈퇴에 기인한 키갱신 외에 멤버의 이동에 기인한 키갱신도 요구되므로 빈번한 키갱신이 발생한다. 만일 단지 하나의 서비스 영역이라는 제한된 구역에서 계층적 키 트리를 적용하는 경우에는, 빈번한 키갱신의 비일관성은 크게 문제가 되지 않는다. 그러나, 설사 이동성이 빈번한 경우에도, 가입이나 탈퇴, 등록에 관계없는 다른 멤버들까지 더불어 잦은 키갱신을 수행하는 것은 매우 비효율적이다. 이 경우 잦은 키갱신은 멤버들이 키갱신 메시지를 놓칠 확률과 보안상 키를 노출할 확률을 높일 수 있다. 더욱이 잦은 키갱신은 SMMSA의 시스템 및 네트워크 자원 사용 비중을 높이므로, SMM 패킷 전송이 더욱 지연되는 결과를 낳을 수 있다. 따라서 멤버의 가입이나 탈퇴, 혹은 등록을 위하여 키갱신이 요구될 때, 이를 곧바로 모두 처리해주는 Naive한 방법[11]은 부적합하다.

본 논문에서는 이를 해결하기 위해서 Batch Process에 의한 키갱신을 사용하고자 한다. 이는 Kronos[11]의 주기적 키갱신 방법을 확장한 개념이다. Kronos에서는 항상 주기적으로 키가 갱신되지만, 제안된 방법은 최초의 키갱신 트리거 메시지(Key Update Trigger Message)가 도착한 때를 기준으로 일정 시간인 BPP(Batch Process Period) 동안 키갱신을 모아서 한꺼번에 처리하는 Batch Process에 의한 키갱신을 수행한다.

키갱신 트리거 프로토콜은 키갱신을 유발시키기 위하여 *Service Join Request, Service Leave Request, Registration Request, Indirect Stop Request* 등의 TReq(Trigger Request)에 관한 메시지들과 이러한 메시지들에 대응하는 *Service Join Verify Response, Service Leave Verify Response, Registration Verify Response* 등의 VRes(Verify Response)에 관한 메시지들로 이루어져 있다.



〔그림 6〕 Batch Process에 의한 키갱신

SMMSA는 *Service Join Request* 등의 TReq 메시지를 받으면 이 메시지에 대한 위조여부와 해당 이동 단말이 서비스 해주기 적합한 조건인지를 판단하기 위해서 이동 단말의 Home SMMSA에게 *Service Join Verify Request* 등의 VReq(Verify Request)를 보낸다. Home SMMSA는 이동 단말이 보낸 메시지에 대한 검증과 현재 이동 단말이 서비스 받기 유효한지에 대해 판단한 뒤 해당 SMMSA에게 *Service Join Verify Response* 등의 VRes를 보낸다. SMMSA는 VRes를 수신하고, 이에 따라서 키를 갱신한다.

〔그림 6〕은 Batch Process에 의한 키갱신 방법에 대한 설명이다. 〔그림 6(a)〕는 Naive한 방법으로 이동단말로부터 키갱신을 위하여 TReq를 수신하고, 이에 대응하는 VRes를 Home SMMSA로부터 받을 때마다 키갱신을 수행하는 방법이다. 이러한 방법의 단점은 그림에서도 볼 수 있듯이 키갱신 트리거 메시지들의 증가에 비례해서 너무 빈번한 키갱신을 수행하여 키갱신의 비효율성을 낳게 된다. 따라서 본 논문에서는 이를 해결하기 위해서 Batch Process에 의한 키갱신을 이용한다. 〔그림 6(b)〕와 같이 키갱신 후에 최초의 TReq나 VRes로부터 BPP만큼의 시간 후에 키갱신을 수행하게 된다. 즉, BPP동안 TReq들과 VRes들을 모아서 한번의 키갱신으로 처

리한다. 최초의 TReq나 VRes를 기준으로 BPP를 설정함으로써 특정상황에서 α값의 증가에 의한 Join Latency, Leave Latency, Registration Latency가 너무 길어지는 것을 방지한다.

Batch Process를 이용한 키갱신 방법에서 키갱신 시간을 식으로 나타내면 (1)과 같다.

$$T_{Key\_Update} = T_{First\_Key\_Update\_Trigger} + BPP \quad (1)$$

한편, Join Latency, Leave Latency, Registration Latency에 관한 식들은 아래의 식 (2)~(7)과 같다.

$$Join\ Latency = T_{Key\_Update} - T_{Join\_Request} \quad (2)$$

$$Leave\ Latency = T_{Key\_Update} - T_{Leave\_Request} \quad (3)$$

$$Registration\ Latency = T_{Key\_Update} - T_{Registration\_Request} \quad (4)$$

$$\alpha < Join\ Latency \le \alpha + BPP \quad (5)$$
$$(\alpha = T_{Join\_Verify\_Response} - T_{Join\_Request})$$

$$\alpha < Leave\ Latency \le \alpha + BPP \quad (6)$$
$$(\alpha = T_{Leave\_Verify\_Response} - T_{Leave\_Request})$$

$$\alpha < Registration\ Latency \le \alpha + BPP \quad (7)$$
$$(\alpha = T_{Registration\_Verify\_Response} - T_{Registration\_Request})$$

즉, Join Latency, Leave Latency, Registration Latency는 각각 α+BPP 값을 넘지 않는다. BPP 값을 높일수록 그만큼 키갱신 빈도가 줄겠지만, 반대로 Join Latency, Leave Latency, Registration Latency가 늘어나 이동 단말에 대한 QoS가 떨어진다. 또한 BPP값을 낮출수록 이동 단말에 대한 QoS는 좋아지겠지만, 키갱신 빈도가 너무 잦아질 수 있으므로, BPP값은 해당 SMM 서비스의 특성에 맞추어 선정하여야 한다.

Ⅳ. SMM 프로토콜의 구현

지금까지 기본적인 SMM 프로토콜의 구성에 대하여 알아보았다. 본 절에서는 SMM 프로토콜의 각 부분에 대하여 상세히 알아보고, 이동 단말이 SMM 그룹에 가입해서 탈퇴하기까지의 과정을 예를 통해 알아본다.

SMM 프로토콜은 〈*Group Join*〉, 〈*Service Join & Direct Mode*〉, 〈*Movement & Indirect Mode*〉, 〈*Registration & Direct Mode*〉, 〈*Service Leave*〉, 〈*Group Leave*〉 등의 6개의 과정으로 구분해 볼 수 있다.

위의 과정 중 〈*Group Join*〉, 〈*Service Join*〉, 〈*Group Leave*〉, 〈*Service Leave*〉의 구분과 기능을 명확히 하기 위해 실시간 동영상 서비스를 적용하여 설명하고자 한다. 유료 서비스인 실시간 동영상 서비스를 받고자 하는 사용자는 인터넷을 통하여 가입 신청서를 작성하여 이 서비스를 제공하는 업체에 가입하게 된다. 이 과정을 〈*Group Join*〉이라고 할 수 있다. 한편, 사용자는 이러한 동영상 서비스를 항상 받고 싶어하기보다는 일반적으로 자신이 원할 때(즉, 보고 싶어하는 영화가 방송될 경우 등)만 이러한 서비스의 제공을 원한다. 이때, 해당 동영상 서비스를 받기 위해서는 ID와 password 등을 제시한 후에 서비스를 받아 보게 되는데, 이를 〈*Service Join*〉이라고 할 수 있다. 또한, 자신이 원하는 동영상을 다 본 뒤에 더 이상 보고싶은 동영상이 없다면 이 서비스를 중단하게 되는데, 이를 〈*Service Leave*〉라 할 수 있다. 마지막으로 사용자는 그 서비스를 해지할 수 있는데, 이를 〈*Group Leave*〉라 할 수 있다.

이를 위한 SMM 프로토콜의 표현을 위해 본 논문은 다음과 같은 표기법을 사용한다.

- ID_X : 이동 단말 X의 ID(Identification)
- $Home\_SMMSA\_X$: X의 Home SMMSA
- PW_X : X의 password
- $Information_X$: X의 인증서를 포함한 SMM 그룹 가입에 필요한 정보
- $TS_{\#}$: #번째 Time Stamp(replay attack 방지)
- K_X : 이동 단말 X와 SMMSA들 사이에 암호화를 위하여 사용되는 비밀키. 이동 단말 X가 〈*Group Leave*〉 할 때까지 Home SMMSA가 관리하고 X가 〈*Group Leave*〉하면 Home SMMSA는 이 키를 폐기한다. 또한, X에게 SMM 서비스를 제공해 주는 SMMSP들은 자신의 service list에 X가 존재하는 동안만 이 키를 관리한다.
- $K_{X,Y}$: X와 Y가 공유하는 비밀키
- $K_{A\_Sub_X}$: [3]의 계층적 키트리 방식의 그룹키

$\langle 1 \rangle\ a \rightarrow SMMSA\_A: E_{KU_{SMMSA\_A}}[ID_a || PW_a || Information_a || TS_1 || F_{Group\_Join\_Request}]$

$\langle 2 \rangle\ SMMSA\_A \rightarrow a: E_{KU_a}[TS_1 || TS_2 || K_a]$

[그림 7] 〈Group Join〉 프로토콜

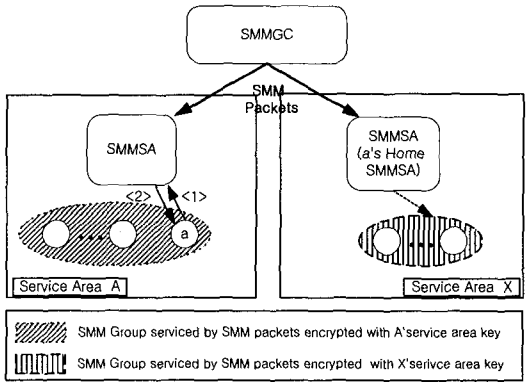
(본 논문에서는 서비스 영역 키) 갱신에 사용되는 서비스 영역 A에서 이동 단말 X를 위한 서브키들. X에게 SMM 서비스를 제공해 주는 SMMSP들은 자신의 service list에 X가 존재하는 동안만 이 키를 관리한다.

- KU_X : X의 public key
- KR_X : X의 private key
- $E_K[C]$: 키 K를 통한 C의 암호화
- $KeyUpdate_A$: 서비스 영역 A의 서비스 영역키 갱신
- $IndirectMode_{A,B}$: 서비스 영역 A에서 서비스 영역 B로의 Indirect Mode 서비스
- F_R : 패킷 R에 대해 설명해주는 플래그(flag)
- SMMSA_A : 서비스 영역 A의 SMMSA
- || : concatenation
- $H(M)$: M에 대한 해쉬

4.1 Group Join

〈*Group Join*〉은 이동 단말이 처음 SMM 그룹에 가입하는 과정이며, SMM 서비스의 제공자에 따라 off-line에서 이를 처리하는 경우도 있을 수 있지만, on-line의 경우와 off-line의 경우 모두 기본적으로는 동일한 기능을 수행하는 과정이다. 본 절에서는 on-line에서 이를 처리하는 상황으로 가정한다.

SMMSA는 주기적으로 *SMMSA Advertisement*를 브로드캐스팅함으로써 자신의 서비스 영역에 위치한 이동 단말에게 자신의 존재를 알린다. 이 *SMMSA Advertisement* 패킷에는 해당 SMM 서비스를 잘 나타내주는 ID와 이에 따른 멀티캐스트 주소, SMMSA의 ID, SMMSA의 public key를 담고 있는 인증서 등을 포함한다. SMM 그룹에 가입하고자 하는 이동 단말은 이 *SMMSA Advertisement* 패킷에 담긴 내용을 이용하여 *Group Join Request*를 하게 된다. [그림 7]은 〈*Group Join*〉에 해당하는 프로토콜을 설명하고 있으며, [그림 8]은 이에 대한 과정이다. 각각의 메시지는 다음과 같은 역할을 수행한다.



[그림 8] 〈Group Join〉 과정

〈1〉 (*Group Join Request*): SMM 서비스에 가입하려는 이동 단말은 자신의 ID와 SMM 서비스에 사용될 password, 그 외 인증서를 포함한 SMM 그룹 가입에 필요한 정보들을 *SMMSA Advertisement* 패킷의 인증서에서 추출한 SMMSA의 public key로 암호화하여 SMMSA_A에게 보낸다.

〈2〉 (*Group Join Grant*): 〈1〉의 메시지를 받은 SMMSA_A는 이동 단말이 제출한 정보를 토대로 이동 단말이 SMM 서비스에 적합한지를 판단하고, 적합하면 이동 단말의 정보를 자신의 데이터 베이스에 저장하고 자신이 이동 단말의 Home SMMSA가 되어 이동 단말에게 SMM 그룹에 가입되었음을 알리는 메시지를 보낸다. 이 메시지는 이동 단말과 각 SMMSA간의 프로토콜로, 암호화에 쓰일 키를 담고 있고 a의 public key로 암호화되어 보내진다.

4.2 Service Join & Direct Mode

이동단말이 해당 SMM 서비스를 받기 위해 〈*Service Join*〉하고 Direct Mode로 서비스를 받는 과정이다. [그림 9]는 〈*Service Join & Direct Mode*〉 서비스에 전달되는 메시지를 나타내고, [그림 10]은 이에 대한 과정이다. 각각의 메시지는 다음과 같은 기능을 수행한다.

$\langle 1 \rangle\ a \rightarrow SMMSA\_A: ID_a \| F_{Service\_Join\_Request} \| TS_1 \| ID_{Home\_SMMSA\_a} \| E_{K_a}[M \| E_{KR_a}[H(M)]]$

$M = ID_a \| PW_a \| TS_1 \| F_{Service\_Join\_Request} \| ID_{SMMSA\_A}$

$\langle 2 \rangle\ SMMSA\_A \rightarrow Home\_SMMSA\_a:$

$E_{K_{SMMSA\_A,Home\_SMMSA\_a}}[ID_{SMMSA\_A} \| ID_a \| TS_1 \| F_{Service\_Join\_Request}] \| E_{K_a}[M \| E_{KR_a}[H(M)]]$

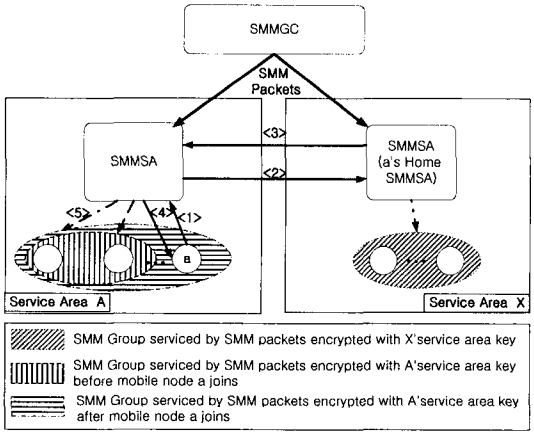
$M = ID_a \| PW_a \| TS_1 \| F_{Service\_Join\_Request} \| ID_{SMMSA\_A}$

$\langle 3 \rangle\ Home\_SMMSA\_a \rightarrow SMMSA\_A: E_{K_{Home\_SMMSA\_a,SMMSA\_A}}[TS_1 \| TS_2 \| ID_a \| F_{Service\_Join\_Grant} \| K_a]$

$\langle 4 \rangle\ SMMSA\_A \rightarrow a: E_{K_a}[TS_1 \| TS_3 \| K_{SMMSA\_A,a} \| K_{A\_Sub_a} \| F_{Service\_Join\_Grant}]$

$\langle 5 \rangle\ KeyUpdate_A$

〔그림 9〕 〈Service Join & Direct Mode〉 프로토콜



〔그림 10〕 〈Service Join & Direct Mode〉 과정

〈1〉 (*Service Join Request*): 이 메시지는 크게 SMMSA_A와 a의 Home SMMSA를 위한 서브 메시지로 구성된다. 먼저 SMMSA_A에게는 자신의 ID를 보내주고 〈*Service Join*〉을 요청한다. 하지만 이 경우 발원지의 인증과 a가 현재 서비스 받기 적합한 자격인지를 판단하기 위해서, a의 Home SMMSA에게 이를 확인하는 절차가 필요하다. 즉, SMMSA_A는 a의 Home SMMSA로부터 이를 검증 받고 난 후에 a에게 서비스를 제공한다. 이렇게 Home SMMSA에게 보내어질 메시지는 a의 ID와 password, 또한 SMMSA_A에 〈*Service Join*〉을 요청하는 내용을 담고있다. 이 내용은 해쉬된 후 a의 private key로 서명된 뒤 함께 보내어져 a를 인증 하는데 사용된다. 이런 검증과정을 위하여 a의 Home SMMSA의 ID도 SMMSA_A에게 함께 보낼 필요성이 있다.

〈2〉 (*Service Join Verify Request*): a의 Home SMMSA에게 메시지 〈1〉에 대해 검증을 요청하는 과정이다. 〈1〉의 메시지를 받은 SMMSA_A는 a의 Home SMMSA에게 a가 자신에게 Service Join Request 한 것을 알리고, a에 대해 검증을 의뢰하기 위하여 〈1〉에서 받은 메시지 중 Home SMMSA를 위한 메시지를 같이 보내준다.

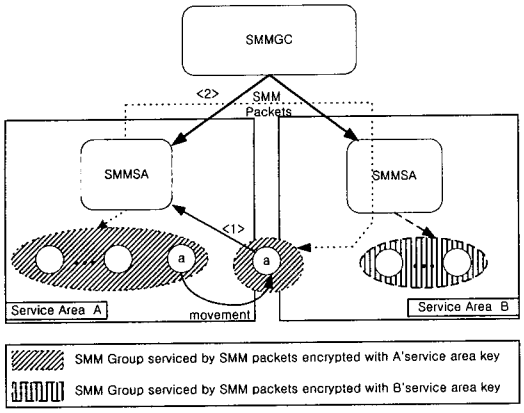
〈3〉 (*Service Join Verify Response*): 메시지 〈2〉를 받은 a의 Home SMMSA는 보내온 메시지를 검증하고 데이터 베이스를 살펴서, a가 SMM 서비스에 유효한 상태이면 SMMSA_A에게 a를 서비스해도 좋다는 메시지를 보내준다. 이 메시지에는 SMMSA_A가 a와의 처음 통신에 쓰일 키 를 포함하고 있다.

〈4〉 (*Service Join Grant*): SMMSA_A는 a에게 〈*Service Join*〉 기능이 완료됨을 알리고, 서비스 영역키 갱신 때 a가 서비스 영역키를 받아 볼 수 있도록 SMMSA_A와 a가 공유하는 비밀키와 서브키 를 준다. 이 메시지는 〈3〉에서 받은 키로 암호화되어 보내진다.

〈5〉 (*Key Update*): Batch Process에 의한 키갱신을 통해 SMMSA_A의 서비스 영역키를 받은 이동 단말은 SMMSA_A로부터 Direct Mode SMM 서비스를 받는다.

4.3 Movement & Indirect Mode

Direct Mode로 서비스 받던 이동 단말이 새로운 서비스 영역으로 이동할 경우 당분간 Indirect Mode로 서비스 받는 과정이 필요하다. 이동 단말은 새로운 *SMMSA Advertisement* 패킷을 수신하고 현재 자신의 SMMSP에게 Indirect Mode 서비스를 요



〔그림 11〕 〈Movement & Indirect Mode〉 과정

청하게 된다. 〔그림 11〕은 이 과정을 메시지와 함께 나타낸 것이고, 〔그림 12〕는 메시지 형태를 보여준다.

〈1〉 (*Indirect Mode Request*): SMMSA_B의 *Advertisement* 패킷을 받은 이동 단말 a는 현재 SMMSP인 SMMSA_A에게 *Indirect Mode Request* 메시지를 보낸다. 이 메시지는 a 자신의 ID와 Indirect Mode로 서비스 받기 원하는 서비스 영역의 SMMSA의 ID를 포함하고 있다.

〈2〉 (Indirect Mode): 서비스 영역 A의 SMMSA는 서비스 영역 B로 Indirect Mode로 서비스 해준다.

4.4 Registration & Direct Mode

Indirect Mode로 서비스 받던 이동 단말 a는 더 이상 SMMSA_A의 SMMSA Advertisement 메시지를 받지 못하고 SMMSA_B의 SMMSA Advertisement 메시지만 받게 되면 자신이 완전히 서비스 영역 B로 이동해 왔다고 인지한다. 이때, Registration Request를 전송해서 SMMSA_B로부터 Direct Mode로 서비스 받는다. Indirect Mode와 Direct Mode의 패킷 시퀀스 번호를 비교하여, Indirect Mode로부터 받은 패킷의 시퀀스 번호가 Direct Mode로 받았던 패킷의 시퀀스 번호보다 크거나 일치하면 Indirect Mode Stop Request 메시지를 SMMSA_A로 보내고, 아니면 시퀀스 번호가 일치할 때까지 두 가지 모드로 서비스 받는다. Indirect Mode Stop Request 메시지를 받은 SMMSA_A는 Batch Process에 의해 서비스 영역 키갱신을 수행한다. 〔그림 13〕은 이러한 〈Registration & Direct Mode〉의 일련의 과정의 프로토콜을 설명하고 있고, 〔그림 14〕는 이에 대한 과정을 메시지와 함께 그림으로 보여주고 있다.

〈1〉 (*Registration Request*): a가 SMMSA_B로 *Registration Request* 메시지를 보낸다. 이는 〈*Service Join*〉의 메시지 〈1〉과 유사하다.

〈1〉 $a \rightarrow SMMSA\_A: ID_a \| E_{K_a}[ID_a \| ID_{SMMSA\_B} \| TS_1 \| F_{Indirect\_Mode\_Request}]$

〈2〉 $IndirectMode_{A,B}$

〔그림 12〕 〈Movement & Indirect Mode〉 프로토콜

〈1〉 $a \rightarrow SMMSA\_B: ID_a \| F_{Registration\_Request} \| TS_1 \| ID_{Home\_SMMSA\_a} \| E_{K_a}[M \| E_{KR_a}[H(M)]]$

$M = ID_a \| PW_a \| TS_1 \| F_{Registration\_Request} \| ID_{SMMSA\_B}$

〈2〉 $SMMSA\_B \rightarrow Home\_SMMSA\_a:$

$E_{K_{SMMSA\_B,Home\_SMMSA\_a}}[ID_{SMMSA\_B} \| ID_a \| TS_1 \| F_{Registration\_Request}] \| E_{K_a}[M \| E_{KR_a}[H(M)]]$

$M = ID_a \| PW_a \| TS_1 \| F_{Registration\_Request} \| ID_{SMMSA\_B}$

〈3〉 $Home\_SMMSA\_a \rightarrow SMMSA\_B: E_{K_{Home\_SMMSA\_a,SMMSA\_B}}[TS_1 \| TS_2 \| ID_a \| F_{Registration\_Grant} \| K_a]$

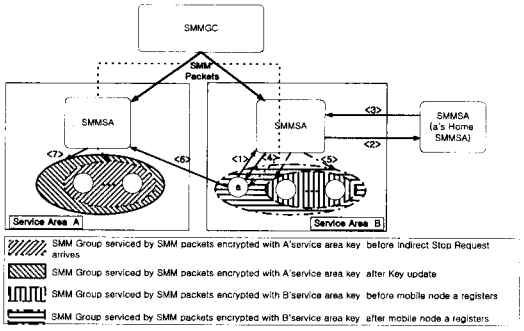
〈4〉 $SMMSA\_B \rightarrow a: E_{K_a}[TS_1 \| TS_3 \| K_{SMMSA\_B,a} \| K_{B\_Sub_a} \| F_{Registration\_Grant}]$

〈5〉 $KeyUpdate_B$

〈6〉 $a \rightarrow SMMSA\_A: E_{K_a}[ID_a \| ID_{SMMSA\_B} \| TS_4 \| F_{Indirect\_Mode\_Stop\_Request}]$

〈7〉 $KeyUpdate_A$

〔그림 13〕 〈Registration & Direct Mode〉 프로토콜



〔그림 14〕 〈Registration & Direct Mode〉 과정

〈2〉 (*Registration Verify Request*): 〈1〉의 메시지를 받은 SMMSA_B가 a의 Home SMMSA에게 a의 검증을 요청하는 과정이다. 이는 〈*Service Join*〉의 메시지 〈2〉와 유사하다.

〈3〉 (*Registration Verify Response*): 〈2〉에 대한 검증을 마치고 이에 대한 응답을 SMMSA_B에게 전해주는 과정이다. 이는 〈*Service Join*〉의 메시지 〈3〉과 유사하다.

〈4〉 (*Registration Grant*): SMMSA_B는 a에게 〈*Registration*〉이 수행되었음을 알려준다. 이는 〈*Service Join*〉의 메시지 〈4〉와 유사하다.

〈5〉 (*Key Update*): Batch Process에 의한 서비스 영역 B의 서비스 영역키가 갱신되면서 a는 이때부터 비로소 Direct Mode 서비스를 받게 된다.

〈6〉 (*Indirect Mode Stop Request*): Direct Mode와 Indirect Mode 양쪽으로 서비스를 받던 a가 패킷의 시퀀스 번호를 비교해서 Indirect Mode의 시퀀스 번호가 Direct Mode보다 크거나 일치하면 Indirect Mode 서비스가 더 이상 필요하지 않음을 알린다.

〈7〉 (*Key Update*): 〈6〉을 받은 SMMSA_A는 a를 자신의 'service list'에서 삭제하고 Batch Process에 의한 키를 갱신한다.

4.5 Service Leave

〈*Service Join*〉 후 SMM 서비스를 받던 이동 단말이 〈*Service Leave*〉를 수행하는 과정이다. 〔그림 15〕는 〈*Service Leave*〉에 대한 프로토콜을 설명하고 있고, 〔그림 16〕은 이에 대한 과정을 메시지와 함께 그림으로 보여준다.

〈1〉 (*Service Leave Request*): a는 SMMSA_B로 *Service Leave Request*를 한다. 이는 〈*Service Join*〉의 메시지 〈1〉과 유사하다.

〈2〉 (*Service Leave Verify Request*): a의 Home SMMSA에게 검증을 요청하는 과정이다. 이는 〈*Service Join*〉의 메시지 〈2〉와 유사하다.

〈3〉 (*Service Leave Verify Response*): a의 Home SMMSA는 검증을 마치고 이에 대한 응답을 SMMSA_B에게 전해주는 과정이다.

〈4〉 (*Service Leave Grant*): a에게 〈*Service Leave*〉 수행결과를 알려준다.

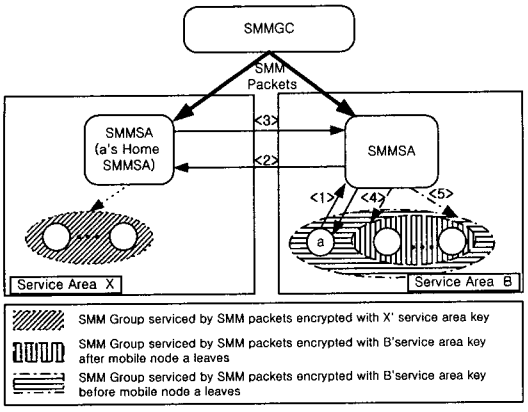
〈5〉 (*Key Update*): a가 〈*Service Leave*〉되었으므로, Batch Process에 의한 키갱신을 하여 a를 서비스에서 제외시킨다.

4.6 Group Leave

SMM 서비스를 받던 이동 단말이 SMM 그룹을 완전히 탈퇴하는 과정이다. 〈*Group Leave*〉 과정은 〈*Group Join*〉 과정과 마찬가지로 off-line에서 이뤄지는 경우도 있을 수 있지만, off-line에서 행하

$$\langle 1\rangle\ a \rightarrow SMMSA\_B:\ ID_a \| F_{Service\_Leave\_Request} \| TS_1 \| ID_{Home\_SMMSA\_a} \| E_{K_a}[M \| E_{KR_a}[H(M)]]$$
$$M = ID_a \| PW_a \| TS_1 \| F_{Service\_Leave\_Request} \| ID_{SMMSA\_B}$$
$$\langle 2\rangle\ SMMSA\_B \rightarrow Home\_SMMSA\_a:$$
$$E_{K_{SMMSA\_B,Home\_SMMSA\_a}}[ID_{SMMSA\_B} \| ID_a \| TS_1 \| F_{Service\_Leave\_Request}] \| E_{K_a}[M \| E_{KR_a}[H(M)]]$$
$$M = ID_a \| PW_a \| TS_1 \| F_{Service\_Leave\_Request} \| ID_{SMMSA\_B}$$
$$\langle 3\rangle\ Home\_SMMSA\_a \rightarrow SMMSA\_B:\ E_{K_{Home\_SMMSA\_a,SMMSA\_B}}[TS_1 \| TS_2 \| ID_a \| F_{Service\_Leave\_Grant}]$$
$$\langle 4\rangle\ SMMSA\_B \rightarrow a:\ E_{K_a}[TS_1 \| TS_3 \| F_{Service\_Leave\_Grant}]$$
$$\langle 5\rangle\ KeyUpdate_B$$

〔그림 15〕 〈Service Leave〉 프로토콜



〔그림 16〕 〈Service Leave〉 과정

여지는 〈*Group Leave*〉 과정도 본 절에서 설명된 on-line에서 행하여지는 〈*Group Leave*〉와 유사한 과정을 수행하게 된다. 따라서, 사용자와 관리자 편의를 위해 on-line 메시지 수행을 우선적으로 고려하여 다음과 같이 〈*Group Leave*〉를 고안하였다.

On-line에서 행하여지는 〈*Group Leave*〉 과정은 다음과 같다. 이동 단말의 Home SMMSA는 *Group Leave Request* 메시지를 검증한 후 정말로 이동 단말이 SMM 그룹을 탈퇴하려는 지를 확인하고 맞는다면 해당 이동 단말을 자신의 데이터 베이스에서 삭제한다.

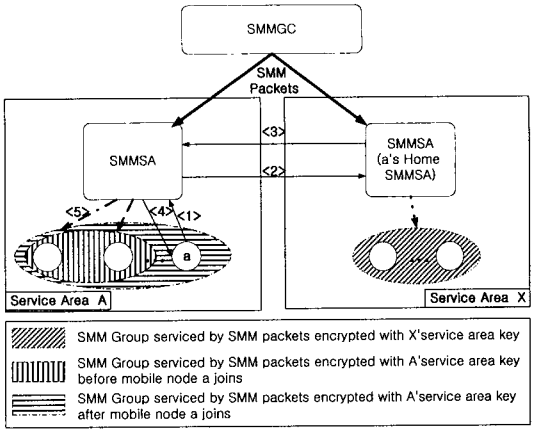
〔그림 17〕은 〈*Group Leave*〉 과정에 대한 프로토콜을 보여주고 있고, 〔그림 18〕은 이 과정을 그림으로 보여주고 있다.

〈1〉 (*Group Leave Request*): a는 어디에 위치하든지 자신의 Home SMMSA에게 Group Leave Request를 한다.

〈2〉 (*Group Leave Grant*): a의 Home SMMSA는 〈1〉의 메시지를 검증하고, 검증이 끝난 뒤 a에게 〈*Group Leave*〉가 완료되었음을 알려주고, a를 데이터 베이스에서 삭제한다.

4.7 SMM 프로토콜의 예

본 절에서는 앞 절에서 설명된 SMM 프로토콜



〔그림 18〕 〈Group Leave〉 과정

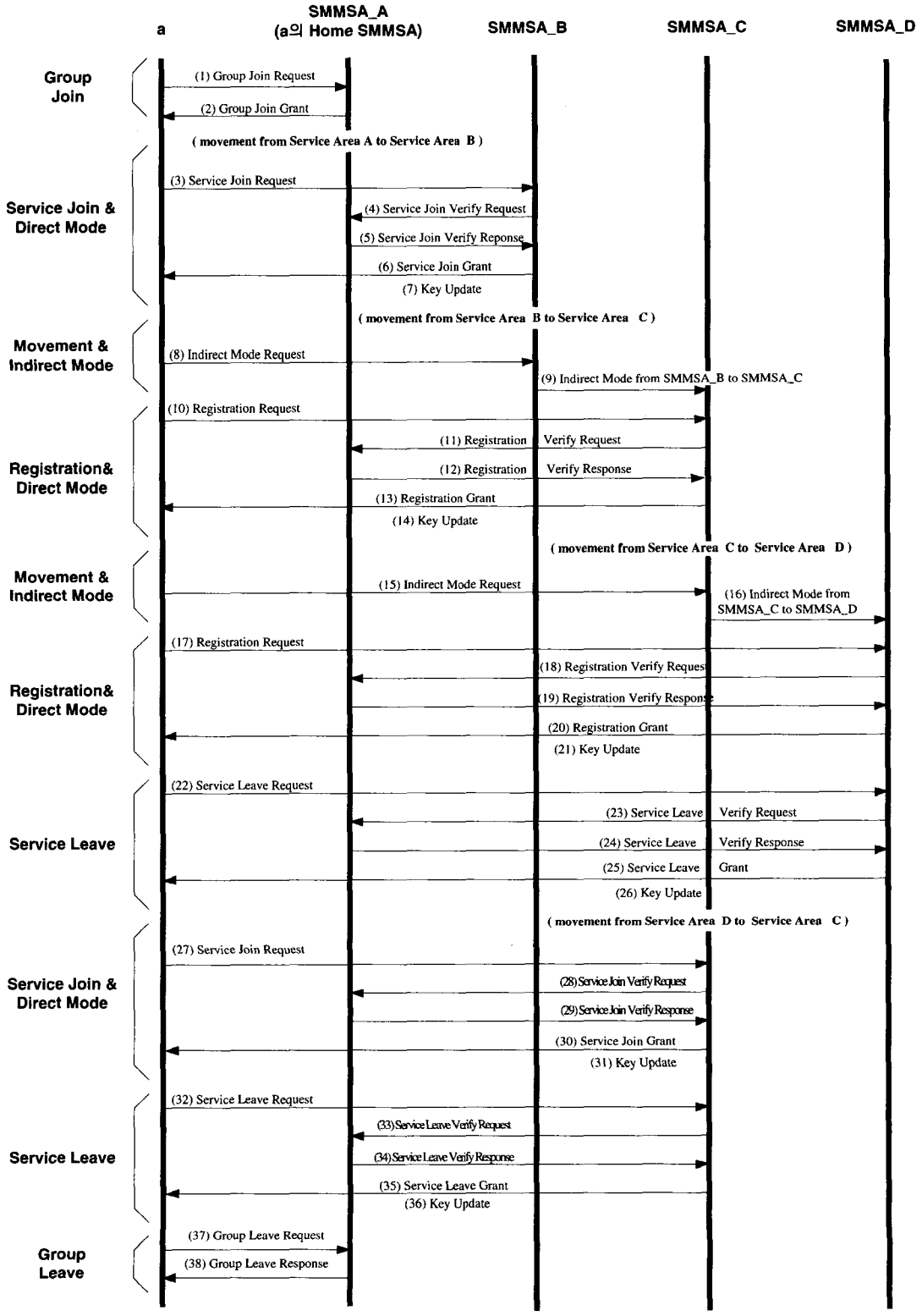
과정들을 종합해 이동 단말이 SMM 서비스 과정이 발생할 수 있는 상황의 예를 설정하여 프로토콜 과정을 보여준다(〔그림 19〕).

〔그림 19〕의 시나리오는 다음과 같다.

- 이동 단말 a는 SMMSA_A에게 *Group Join Request*를 하여 〈*Group Join*〉을 수행한다.
- a는 서비스 영역 B로 이동하여 〈*Service Join*〉 수행 후 SMMSA_B로부터 Direct Mode 서비스를 받는다.
- a는 서비스 영역 B에서 C로 이동한다. 따라서 SMMSA_B로부터 Indirect Mode 서비스를 받다가, 영역 C에 등록 후 SMMSA_C로부터 Direct Mode 서비스를 받는다.
- a는 서비스 영역 C에서 D로 이동한다. 따라서 SMMSA_C로부터 Indirect Mode 서비스를 받다가, 새로운 영역 D에 등록하여 SMMSA_D로부터 Direct Mode 서비스를 받는다.
- a는 서비스 영역 D에서 *Service Leave Request*를 하여 〈*Service Leave*〉를 수행한다.
- a는 서비스 영역 C로 이동하여 〈*Service Join*〉 한 후 SMMSA_C로부터 Direct Mode 서비스를 받는다.
- a는 서비스 영역 C에서 *Service Leave Request*

$$\langle 1\rangle\ a \rightarrow Home\_SMMSA\_a:\ E_{K_a}[M \| E_{KR_a}[H(M)]]$$
$$M = ID_a \| PW_a \| TS_1 \| F_{Group\_Leave\_Request} \| ID_{SMMSA\_A}]$$
$$\langle 2\rangle\ Home\_SMMSA\_a \rightarrow a:\ E_{K_a}[TS_1 \| TS_2 \| F_{Group\_Leave\_Grant}]$$

〔그림 17〕 〈Group Leave〉 프로토콜



〔그림 19〕 SMM 프로토콜의 예

를 하여 〈*Service Leave*〉를 수행한다.

- a는 *Group Leave Request*를 Home SMMSA에 보내어 〈*Group Leave*〉를 최종적으로 수행한다.

따라서, 본 논문에서 제시하는 SMM 프로토콜이 이동성이 고려된 시나리오에 잘 적용됨을 알 수 있다.

V. 시뮬레이션 및 결과

5.1 시뮬레이션

제안된 SMM 구조에서의 SMM 프로토콜의 성능을 평가하기 위하여 이동 event를 처리할 수 있는 Discrete Event Simulator를 구현하였다. 시뮬레이션에 사용된 주요 파라미터들은 〔표 1〕과 같다.

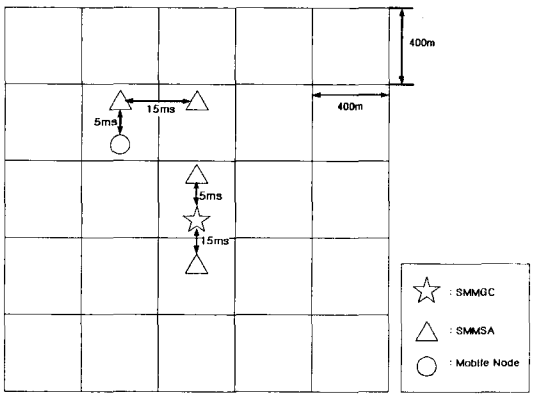
시뮬레이션 환경은 〔그림 20〕과 같이 5-by-5로 구성된 서비스 영역들로 이루어져 있다. 각 서비스 영역은 400m의 길이를 가진 정사각형 모양으로 구성한다고 가정하였으며, 각각의 서비스 영역은 meter (m) 단위의 이차원 좌표를 갖는다. 인접한 SMMSA 사이의 전송 지연은 15ms의 평균값과 uniform distribution에 의해 0에서 2ms사이의 추가적인 전송지연을 갖는 것으로 설정하였다.

또한, 같은 서비스 영역에 위치한 SMMSA에서 이동 단말까지 전송지연은 5ms의 평균값과 uniform distribution에 의해 0에서 0.5ms 사이의 추가지연을 갖도록 설정하였다. 모든 서비스 영역에 균등하게 SMM 패킷이 도달하는 상황을 가정하기 위하여 SMMGC는 중앙의 서비스 영역에 위치한다고 가정하였다. SMMGC에서 같은 서비스 영역에 위치한 SMMSA까지의 전송 지연은 SMMSA와 이동 단말 사이의 전송지연과 같은 값을 적용하였고, SMMGC와 인접한 서비스 영역의 SMMSA와의 전송지연은 인접한 SMMSA들 사이의 전송지연을 적용하였다.

이동 단말의 이동에 대한 모델링은 다음과 같이 설정하였다. 본 시뮬레이션에서 이동 단말은 항상 고정되어 있는 상황과 도보로 걷는 상황, 버스를 타고 가는 상황, 지하철을 타고 가는 상황, 고속의 차를 타고 달리는 상황을 기반으로 하여 모델링한다. 이를 위해 각각의 속도를 0, 5, 40, 70, 100Km/h로 설정한다. 또한 각 이동 단말의 움직임은 1초마다 샘플링된다. 즉,

〔표 1〕 시뮬레이션 파라미터

시뮬레이션 파라미터	값 (내용)	비고
시뮬레이션 시간	70sec	
서비스 영역 수	25	5 X 5
SMM 서비스에 참여한 이동 단말의 수	1250개	50 X 25
서비스 영역의 면적	160,000㎡	400m X 400m
이동 단말의 속도	0, 5, 40, 70, 100Km/h	각 20%
SMM source	CBR 128Kbps	60초 동안 발생
SMM 패킷 크기	500bytes	
Control 패킷 크기	32bytes	
인접한 SMMSA 사이의 전송 지연	15ms	(+0~2ms)
SMMSA에서 이동 단말까지의 전송 지연	5ms	(+0~0.5ms)
Encryption Algorithm	DES-EDE (3DES)	4.748Mbytes/sec
Hash Algorithm	MD5	100.738Mbytes/sec
Sign Algorithm	RSA 512	1.92Mbytes/sec
Sign Length	512bits	
RSA sign (512bits)	1.92ms	
RSA verify (512bits)	0.13ms	
서비스 영역 키갱신 오버헤드	13.2ms	〔3〕의 User-oriented
이동 단말의 서비스 가입·탈퇴 기간	8~12sec	
Background traffic	λ = 10000 (packets/sec)	$T_{Processing}$: 0.1ms
SMMSA의 라우팅 시간	0.01ms	



〔그림 20〕 시뮬레이션 환경

1초마다 상, 하, 좌, 우, 정지 5개의 상황을 일정한 확률로 분포하여 1초에 각각 0, 1.4, 11.1, 19.4, 27.8m 만큼 이동하게 한다.

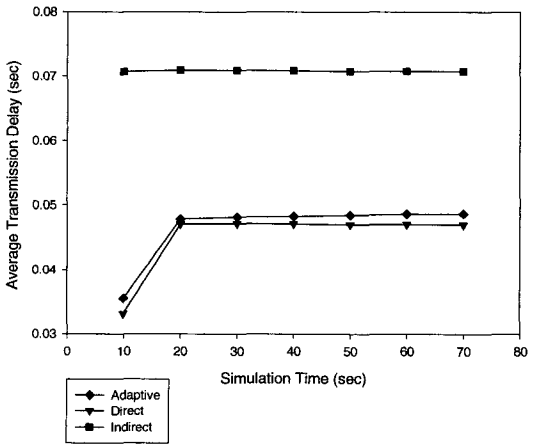
시뮬레이션은 총 70초 동안 진행된다. SMM 패킷은 500byte의 크기로 총 60초 동안 CBR(Constant Bit Rate) source인 SMMGC로부터 128Kbps로 발생된다. 이동 단말은 1부터 25까지의 서비스 영역에 일정하게 분포되지만, 기본적으로 한 서비스 영역 당 50개의 이동 단말을 가정하여 총 1250개의 이동 단말이 SMM 서비스를 받는 상황을 설정했다. 한편 SMM에 쓰일 암호화 및 해쉬, 서명 알고리즘으로 3DES, MD5, RSA를 설정했다. 암호화 및 해쉬, 서명에 소요되는 시간에 대한 비용은 〔18〕을 참조하였다. 또한, 서비스 영역키 갱신에 소요되는 시간에 대한 비용은 〔3〕의 User-oriented 방법을 참조하였다. 한편 시뮬레이션에 이동 단말의 빈번한 가입과 탈퇴의 상황을 적용하기 위하여, 일정기간이 지나면 이동 단말이 번갈아 가며 가입과 탈퇴하게 하였다. 이 일정기간은 8에서 12초 동안 일정하게 분포한다. 또한, 각 서비스 영역의 SMMSA의 Queue에 Poisson 분포를 갖는 background traffic을 삽입한다. 이 background traffic은 평균적으로 초당 10000개가 발생하게 하였고, SMMSA가 이 패킷을 처리하기 위해서는 0.1ms의 처리 시간을 갖게 하였다. 한편 SMMSA에서 일반적인 패킷의 라우팅에는 0.001ms가 걸리게 설정하였다.

본 시뮬레이션은 Direct Mode와 Indirect Mode를 적응적으로 사용하는 SMM 구조 및 프로토콜에 대한 성능을 평가가 주목적이다. 이를 위해 Direct Mode만 사용했을 경우, Indirect Mode만 사용했을 경우, 마지막으로 Direct Mode와 Indirect Mode를 적응적으로 사용한 Adaptive Mode를 가질 경우의 전송지연을 비교했다. 또한, Naive한 키갱신 방법과 Batch Process에 의한 키갱신 방법의 키 갱신 빈도를 비교하였다. 마지막으로, Direct Mode, Indirect Mode, Adaptive Mode의 패킷 손실률을 비교하여 Adaptive한 방법으로 SMM이 이동 단말에게 끊김 없는 서비스와 패킷 손실을 방지해주는 것을 보여준다. 이때, 손실률은 다음과 같이 계산된다. SMM 패킷은 CBR 128Kbps이므로, 500byte의 패킷이 초당 32번 발생하게 된다. 따라서 60초 동안 발생된 SMM 패킷의 총수는 60 X 32=1920개이다. 이 때 이동 단말이 시뮬레이션 시간동안 받은 총 패킷의 수를 $N_{Received}$ 라고 하여 손실률을 계산하며, 이를 위하여 20%의 이동 단말은 시뮬레이션 시작할 때부터 계속 가입되어 탈퇴하지 않게 설정하였다. 이동 단말이 받은 총 패킷들을 평균 내어 $N_{Received}$ 값으로 하여 최종 손실률을 계산한다.

5.2 결과

〔그림 21〕은 BPP(Batch Process Period)가 1일 때 각 모드에 따른 SMM 패킷의 평균 전송 지연을 보여준다.

먼저, Direct Mode는 가장 낮은 전송 지연을 갖는 것을 볼 수 있다. Indirect Mode는 3가지 모드 중 가장 높은 전송 지연을 갖는 것을 볼 수 있다. Adaptive Mode는 Direct Mode와 거의 근접하게 낮은 전송지연을 갖는 것을 볼 수 있다. 〔표 2〕는 이 3가지 모드에 대한 최종 전송 지연을 보여준다. 각 모드는 BPP값에 관계없이 Direct Mode,



〔그림 21〕 각 모드의 평균 전송 지연

〔표 2〕 각 모드의 최종 평균 전송 지연 (단위 : sec)

	BPP=1	BPP=2	BPP=3	BPP=4
Adaptive	0.048699	0.049909	0.049147	0.049012
Direct	0.046999	0.045431	0.047682	0.047265
Indirect	0.079540	0.080052	0.081365	0.080993

〔표 3〕 Naive Vs Batch 키갱신 빈도 (단위 : sec)

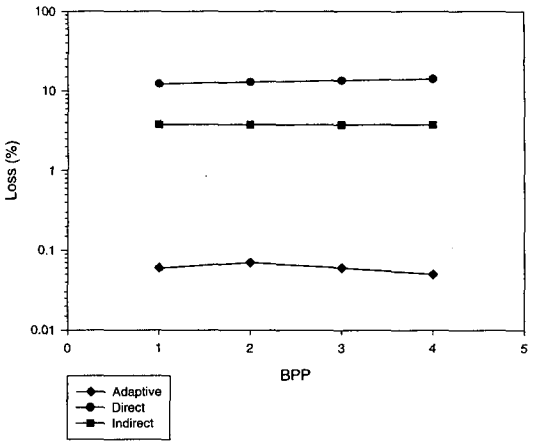
		키갱신 주기
Batch Process	BPP=1	1.273450
	BPP=2	2.190471
	BPP=3	3.314121
	BPP=4	4.205257
Naive		0.246461

Adaptive Mode, Indirect Mode 순으로 전송 지연이 낮았다. 한편 BPP값의 변화에 따른 각 모드의 전송 지연의 차이는 미미했다.

Naive한 키관리 방법과 Batch Process에 의한 키관리 방법의 키갱신 빈도를 비교하기 위하여 Naive한 키관리 방법의 Adaptive Mode를 만들어서 이를 비교해 보았다. 〔표 3〕은 이 결과를 보여준다. Batch Process에 의한 키갱신 주기는 각각의 BPP값에 기인하는 것을 볼 수 있다. 하지만 Naive한 키갱신 방법을 쓴다면 약 0.2초마다 키갱신을 해줘야 하는 상황이 발생한다. 이러한 짧은 주기로 키갱신을 하는 것은 SMM그룹의 멤버들이 키갱신 메시지를 놓칠 확률을 증가시키고 키갱신 메시지를 노출시킬 확률을 증가시킨다. 또한 이러한 잦은 키갱신은 SMMSA에도 많은 부담이 될 수 있다.

〔그림 22〕는 BPP에 따른 각 모드의 데이터 손실률을 보여주고, 〔표 4〕는 이에 대한 값을 정리하였다.

Direct Mode의 경우 가장 높은 패킷 손실률을 보였다. 이는 주로 Registration Latency와 패킷 시퀀스의 비일관성에 기인한다. BPP가 높아질수록 Registration Latency가 커지는데, 이 기간동안은 서비스가 끊기게 되므로, 패킷 손실도 높아진다. Indirect Mode의 경우는 약 3.8%의 패킷 손실률을 보이는데 이는 대부분 패킷 시퀀스의 비일관성에 기인한다. 한편 Adaptive Mode의 경우에는 패킷 손실률이 거의 0%인 것을 볼 수 있었다. Adaptive Mode에서는 이동 단말이 새로운 서비스 영역으로 이동해서 등록하는 동안 이동 단말에게 Indirect Mode로 서비스 해줌으로써, Registration Latency



〔그림 22〕 BPP에 따른 각 모드의 패킷 손실률

〔표 4〕 BPP에 따른 각 모드의 패킷 손실률 (단위 : %)

	BPP=1	BPP=2	BPP=3	BPP=4
Adaptive	0.060000	0.070000	0.060000	0.050000
Direct	12.250000	12.900000	13.030000	13.470000
Indirect	3.790000	3.760000	3.750000	3.760000

로 인한 서비스 끊김을 막고 패킷 비일관성을 보정해 줌으로써 패킷 손실을 방지하였다.

지금까지 시뮬레이션 결과를 종합적으로 고찰해보면 Adaptive Mode는 거의 Direct Mode와 비슷한 수준의 낮은 전송지연을 갖으면서도, 패킷 손실을 방지해주는 효율적인 구조 및 프로토콜임을 검증할 수 있다.

Ⅵ. 결 론

본 논문은 이동 환경에서 안전한 멀티캐스팅을 구현하는 보안멀티캐스트 구조와 프로토콜을 제시하였다. 제안된 이동보안멀티캐스트(SMM) 구조는 이동 단말의 움직임에 따라 적응적으로 서비스를 제공해 줌으로써 멀티캐스트 데이터의 낮은 전송 지연을 보장해 주었고, 이동 단말의 움직임으로 인한 보안 멀티캐스트 서비스의 끊김과 데이터 손실을 막아주어 서비스 QoS를 높였다. 한편, 분산구조를 채택하여 이동 단말에 대한 데이터 베이스를 분산함으로써 효율화를 높였다.

SMM 프로토콜의 성능은 이동 event를 고려한 Discrete Event Simulator에 의해 성능을 평가하였다. 시뮬레이션 결과를 통해서, SMM은 멀티캐

스트 데이터의 낮은 전송 지연을 보장해주고, 이동 단말의 이동으로 인한 서비스 끊김 및 데이터 손실을 방지하는 효율적인 구조 및 프로토콜로 사용될 수 있는 가능성을 보여주었다.

제안된 이동보안멀티캐스트 구조 및 프로토콜은 이동 무선 인터넷 환경에서 실시간 유료 시청 및 실시간 증권정보 등의 다양한 보안 어플리케이션에 유용하게 사용될 수 있을 것으로 기대된다.

향후 연구 방향으로는 도래할 무선 인터넷 시대를 맞이하여 IMT-2000이나 PDA 등의 이동 단말에 SMM 서비스가 잘 적용될 수 있도록, 각 이동 단말의 특성 및 망의 구조 또한 서비스의 특성을 고려하여 보안 QoS 향상을 위한 연구를 계속 진행하고자 한다.

참 고 문 헌

[1] J.Y. Ahn, J.B. Gu and S.H. Park, "A Secure Mobile Multicast(SMM) Protocol for Efficient Key Management and Low Transmission Delay", 'Five minutes presentation' *IEEE Symposium on the Security and Privacy*, 2001.

[2] M.J. Moyer, J.R. Rao and P. Rohotgi, "A Survey of Security Issues in Multicast Communications", *IEEE Network*, November/December 1999.

[3] C.K. Wong, M. Gouda, S.S. Lam, "Secure Group Communication Using Key Graphs", *Proceedings of ACM SIGCOMM'98*, 1998.

[4] S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting", *Proceedings of ACM SIGCOMM'97*, 1997.

[5] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas, "Multicast security: A Taxonomy and Some Efficient Constructions", *Proceedings of the IEEE INFOCOM'99*, 1999.

[6] T.G. Harrison, C.L. Williamson, W.L. Mackrell, R.B. Bunt, "Mobile Multicast (MoM) Protocol: Multicast Support for Mobile Hosts", *Proceedings of ACM/IEEE MOBICOM'97*, September 1997.

[7] C.R. Lin and K.M Wang, "Mobile Multicast Support in IP Networks", *Proceedings of the IEEE INFOCOM2000*, 2000.

[8] G. Xylomenons and G.C. Polyzos, "IP Multicast for Mobile Hosts", *IEEE Communication Magazine*, January 1997.

[9] C. Perkins, "IP Mobility Support", RFC 2002, October 1996.

[10] C. E. Perkins, "Mobile IP", *International Journal of Communication Systems*, 1998.

[11] S. Setia, S. Koussih, S. Jajodia and E. Harder, "Kronos: A Scalable Group Re-Keying Approach for Secure Multicast", *Proceedings of Security and Privacy 2000*, 2000.

[12] L.H. Sahasrabuddhe and B. Mukherjee, "Multicasting Routing: Algorithms and Protocols: A Tutorial", *IEEE Network*, January/February 2000.

[13] C. Perkins, "IP Encapsulation within IP", RFC 2003, October 1996.

[14] S. Deering, "Host Extensions for IP Multicasting", RFC 1112, August 1989.

[15] W. Fenner, "Internet Group Management Protocol, Version 2" , RFC 2236, November 1997.

[16] S. Paul, "Multicasting on the Internet and Its Applications", *KLUWER ACADEMY PUBLISHERS*, 1998.

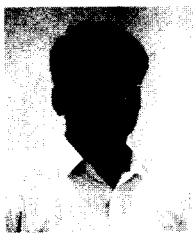
[17] R. Wittmann and M. Zitterbart, Multicast Communication Protocols and Applications, *MORGAN KAUFMANN PUBLISHERS*, 2000.

[18] Crypto++4.0 speed benchmark(www.eskimo.com/~weidai/benchmarks.html)

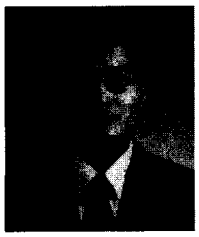
〈著者紹介〉

안 재 영 (Jae Young Ahn)
1998년 : 학사, 중앙대학교 전자공학과
1998년 1월~1999년 4월 : 삼성전자 연구원
2001년 : 석사, 중앙대학교 전자공학과
2001년 7월~현재 : 한국정보보호진흥원 연구원
〈관심분야〉 이동 통신 보안, 네트워크 보안

구 자 범 (Ja Beom Gu)
2000년 : 학사, 중앙대학교 전자공학과
2002년 : 석사, 중앙대학교 전자전기공학부
2002년 2월~현재 : 박사과정, 중앙대학교 전자전기공학부
〈관심분야〉 이동 통신 보안, 유·무선 PKI, 네트워크 보안



이 재 일 (Jae Il Lee)
1986년 : 학사, 서울대학교 계산통계학과
1988년 : 석사, 서울대학교 계산통계학과
1996년 : 한국 IBM 소프트웨어연구소
1996년~현재 : 한국정보보호진흥원 전자서명인증관리센터장
〈관심분야〉 유·무선 PKI, 전자상거래 보안



박 세 현 (Se Hyun Park) 종신회원
1986년 : 학사, 중앙대학교, 전자공학과
1988년 : 석사, 중앙대학교, 전자공학과
1998년 : 컴퓨터 공학 박사, University of Massachusetts at Amherst, ECE Dept.
1988년 2월~1999년 2월 : 한국전자통신연구원 선임 연구원
1999년 3월~현재 : 중앙대학교 전자전기공학부 조교수
〈관심분야〉 이동 통신 보안, 무선인터넷보안, 유·무선 PKI, New Trust Model