情報保護學會論文誌
第17卷 第2號, 2007. 4

멀티캐스트 분배트리 접근제어를 위한 Authenticated IGMP*

강 현 선,[1‡] 박 창 섭[1†]
[1]단국대학교

Authenticated IGMP for Controlling Access to Multicast Distribution Tree

Hyun-sun Kang,[1‡] Chang-Seop Park[1†]
[1]Dankook University

요 약

본 논문에서는 IGMP(Internet group management protocol)를 부당하게 사용함으로써 발생하는 DoS(Denial-of Service) 공격으로부터 멀티캐스트 분배트리를 보호하기 위해 IGMP의 보안관련 기능을 확장시킴으로써 수신자 접근제어 기법을 제안하였다. IP 멀티캐스트 애플리케이션의 상업적인 적용을 위해 채택된 특정 네트워크와 비즈니스 모델을 기반으로, CP(Content Provider), NSP(Network Service Provider), 그룹멤버(group member)에 대한 회계 및 청구와 함께 제안 접근제어기법의 부트스트래핑(bootstrapping)을 위해 키 관리기법 또한 제시하였다.

ABSTRACT

Receiver access control scheme is proposed to protect multicast distribution tree from DoS(Denial-of Service) attack induced by unauthorized use of IGMP(Internet group management protocol), by extending the security-related functionality of IGMP. Based on a specific network and business model adopted for commercial deployment of IP multicast applications, key management scheme is also presented for bootstrapping the proposed access control as well as accounting and billing for CP(Content Provider), NSP(Network Service Provider), and group members.

Keywords : *Authenticated IGMP, Access control, DoS attack, Multicast*

Ⅰ. 서 론

IP 멀티캐스트(multicast) 통신은 인터넷 애플리케이션(internet application)에서 다수의 그룹멤버(group members)에게 데이터를 전송하기 위한 효율적인 방법이다. 하지만, 멀티캐스트 애플리케이션을 신속히 상업적으로 적용하기 위해서는 멀티캐스트와 관련한 안전성 문제가 해결되어야 한다. 멀티캐스트 안전성은 멀티캐스트 콘텐츠(contents) 보호와 멀티캐스트 인프라(infrastructure) 보호, 두 영역으로 구분한다. 전자는 주로 그룹키(group key) 관리를 다루는 반면, 후자는 송신자로부터의 멀티캐스트 데이터를 다수의 멀티캐스트 그룹멤버에게 전달하기 위한 몇몇의 멀티캐스트 라우터(routers)로 구성된 경로인 멀티캐스트 분배트리를 주로 다룬다. 멀티캐스트 분배트리는 PIM-SM과 같은 멀티캐스트 라우팅 프로토콜(routing protocol)을 통해 생성되며, 호스트는 직접 연결된 멀티캐스트 라우터와

접수일: 2006년 8월 1일; 채택일: 2006년 12월 1일
* 본 연구는 2005학년도 단국대학교 대학 연구비의 지원으로 연구되었음.
† 주저자, csp0@dankook.ac.kr
‡ 교신저자, sshskang@dankook.ac.kr

IGMP(Internet Group Management Protocol)[1][2]를 사용함으로써 멀티캐스트 분배트리에 연결될 수 있다. 즉, 호스트는 멀티캐스트 라우터에게 가입을 원하는 멀티캐스트 그룹을 알리기 위해서 IGMP를 사용한다.

멀티캐스트 분배트리에 대한 공격에는 수신자 공격과 송신자 공격이 있으며, 해당 공격들은 기본적으로 멀티캐스트 라우팅 프로토콜과 IGMP의 보안성 부족에 기인한다. IP 멀티캐스트는 익명의 수신자 모델을 기반으로 하기 때문에, 말단 멀티캐스트 라우터는 멀티캐스트 그룹에 가입중인 호스트에 대한 어떠한 신원확인 정보도 유지하지 않으며, 이는 서브넷(subnet) 레벨에서 그룹 멤버쉽에 대한 접근제어(access control)가 수행될 수 없음을 의미한다. 따라서 서브넷 상의 어떠한 호스트도 멀티캐스트 분배트리의 확장과 해당 서브넷으로 멀티캐스트 트래픽(traffic)의 유입을 위해 IGMP를 사용함으로써 멀티캐스트 그룹에 가입할 수 있다. 만약 공격자가 전송되는 멀티캐스트 콘텐츠를 사용할 의도도 없이 동시에 몇몇의 멀티캐스트 그룹에 가입한다면, 이와 같은 수신자 공격은 결국 네트워크 자원(resource)과 영향을 받는 모든 라우터 내부의 상태(state)를 소모시키는 DoS(Denial-of Service) 공격이 발생하게 된다. 반면, 공격자는 유효한 멀티캐스트 주소를 가진 가짜의 패킷을 멀티캐스트 분배트리로 전송할 수도 있다. 이와 같은 송신자 공격 또한 해당 패킷이 모든 그룹멤버에게 전송되기 때문에 네트워크 대역폭(bandwidth)을 낭비한다. PIM-SM의 경우, 모든 패킷이 멀티캐스트 분배트리를 통해 모든 그룹멤버에게 전달되기 전에 우선 RP(Rendezvous Point)로 보내지기 때문에 송신자 공격은 RP에 병목현상을 야기 시킬 수도 있다.

[3]에서 처음 멀티캐스트 분배트리에 대한 접근제어의 필요성이 언급된 이후, 대부분의 송신자와 수신자의 접근제어와 관련한 연구는 주로 IETF와 IRTF 워킹그룹의 주도아래 진행되었다. 본 논문에서는 접근제어와 관련한 키 관리 기법과 함께, [4]에서 소개된 네트워크와 비즈니스 모델을 기반으로 수신자 접근제어를 위해 IGMPv3[2] 기능의 확장하고 일부 수정하였다. 회계(accounting)와 청구(billing) 문제 또한 논의되었다. 2장에서는 수신자 접근제어와 관련한 기존연구가 소개되며, 3장과 4장에서는 본 논문의 접근제어기법을 제안하고 적용된 설계원리가 소개된다. 5장에서는 제안기법을 분석하고 기존기법과 성능을 비교하며, 6장에서 결론을 맺는다.

Ⅱ. 연구배경과 기존연구

2.1 IGMP의 안전성 문제점

*S*는 멀티캐스트 트래픽의 소스 주소이고, *G*는 멀티캐스트 주소라고 하자. 만약 호스트가 *Join Group* 메시지를 서브넷의 말단 멀티캐스트 라우터 (Querier)로 보내면, 호스트는 서브넷의 멀티캐스트 그룹 (*S*, *G*)의 그룹멤버가 된다. 그 후, Querier는 (*S*, *G*)에 대한 상태를 생성하고, *S*에 의해 보내어진 *G*로 향하는 멀티캐스트 트래픽을 서브넷으로 전달한다. Querier는 자신이 유지하고 있는 멀티캐스트 그룹이 여전히 활동 중인가를 검사하기 위해 *General Query Interval* (e.g. 125 초)마다 서브넷으로 *General Query* 메시지를 멀티캐스트 한다. *General Query Interval*은 Querier가 송신하는 *General Query* 메시지의 송신 간격이다. 각 그룹멤버가 멀티캐스트 트래픽을 지속적으로 수신하기 위해서는, *Report* 메시지로 응답해야 한다. (*S*, *G*)를 탈퇴할 때, 그룹멤버는 Querier로 *Leave Group* 메시지를 보낸다. 서브넷에 여전히 (*S*, *G*)의 그룹멤버가 존재하는지를 검사하기 위해, Querier는 서브넷으로 *Group and Source Specific Query* 메시지를 멀티캐스트 한다. 만약 서브넷의 호스트로부터 *Report* 메시지가 없으면, 서브넷에 더 이상의 그룹멤버가 존재하지 않음을 의미하고, Querier는 (*S*, *G*)와 관련한 멀티캐스트 트래픽을 서브넷으로 전달하지 않는다. IGMPv3에서는 *Join* / *Leave Group* 메시지를 Report 메시지로 통합하였지만, 본 논문에서는 설명의 용이함을 위해 *Join* / *Leave Group* 메시지를 *Report* 메시지와 구별한다.

IGMP에는 인증 메커니즘이 포함되어 있지 않기 때문에, 위조된 *Query* / *Join* / *Leave* / *Report Group* 메시지로부터 DoS 공격과 자원 고갈(resource exhaustion) 공격 등이 발생할 수 있다. 위조된 *Join Group* 메시지는 네트워크 대역폭의 고갈을 위해 서브넷으로 다수의 대량 멀티미디어 스트림을 유도할 수 있다. 한편, *Leave Group* 메시지는 *Group and Source Specific Query* 메시지를 발생시키기 때문에, 위조된 *Leave Group* 메시지는 서브넷 상에 불필요한 다량의 시그널링(signaling) 메시지를 유도할 수 있으며, 특히 IGMPv3에서 해당 메시지는 다시 서브넷에 남아있는 그룹멤버의 *Report* 메시지를 발생시킨다. 탈퇴하는 서브넷의 (*S*, *G*) 그룹멤버가 마지막 그룹멤버일 때에는,

Leave Group 메시지를 전송하고, Querier는 *Group and Source Specific Query* 메시지로 검사한 후, (*S*, *G*)와 관련한 멀티캐스트 트래픽을 전달하지 않는다. 하지만, 공격자가 서브넷으로 멀티캐스트 트래픽을 지속적으로 유입하기 위해 *Report* 메시지를 위조할 수 있다. 또한, 현재의 Querier 보다 낮은 IP 주소를 가진 기계로부터의 위조된 *Query* 메시지는 공격자를 새로운 Querier로 지정되도록 할 수도 있다. 이 경우, 만약 공격자가 *Leave Group* 메시지를 무시한다면 일정 시간이 흐른 뒤, 멀티캐스트 트래픽은 해당 서브넷으로 밀려들 것이다. 위의 모든 안전성 문제는 IGMP에 키 관리와 함께 인증기능의 부족으로 인해 발생한다.

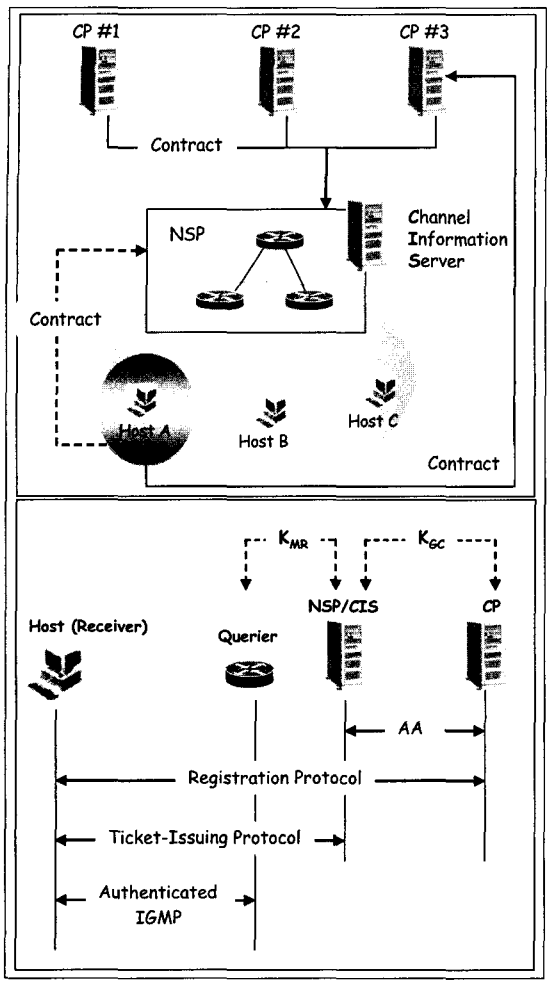
2.2 기존연구

Querier와 호스트 사이에 교환되는 IGMP와 관련한 메시지의 인증을 위해, 사전에 대칭키나 공개키가 분배되어야 한다. [5, 6]에서, 호스트의 인증과 키 정보 분배를 위해 접근토큰(access token) 개념이 사용되었다. 접근토큰은 가입 호스트와 호스트에게 할당된 키에 대한 정보를 포함하며, 해당 정보는 TA(trusted authority)에 의해 전자서명 된다. [5]에서 제안된 기법은 접근토큰이 일회용이기 때문에 매우 제한적이다. 따라서 호스트는 Querier로 *Report* 메시지를 보낼 때마다 TA와 접속해야 한다. [6]에서는 호스트의 공개키가 접근토큰에 포함되며, *Report* 메시지는 호스트에 의해 전자서명 되어 접근토큰과 함께 Querier로 전송된다. [7]에서는 호스트와 가입을 원하는 멀티캐스트 주소에 대한 정보가 포함된 capability가 TA에 의해 전자서명 되어 호스트에게 발행된다. 호스트는 멀티캐스트 그룹에 가입할 때마다 Querier에게 capability를 제시하여야 한다. capability는 일종의 고정된 패스워드로, 유효기간 포함되어 있지만 여러 차례 사용할 수 있다. capability는 Querier로 전송될 때 전혀 보호되지 않기 때문에, 다른 안전성 문제를 발생시킬 수 있다. [8]의 기법에서는 새로운 G-CBA(Group Cryptographically Based Address) 개념이 사용되었다. 하지만, 기본적으로 호스트가 *Report* 메시지를 서명하고 서명확인을 위한 공개키와 함께 Querier로 전송한다는 점에서 [6]과 동일하다. 위의 [5,6,7,8] 기법들은 *Report* 메시지를 전자서명 후 전송하고, Querier가 해당 서명을 공개키로 확인해야 하기 때문에 DoS 공격을 받을 수 있다. 만약 공격자 그룹이 가짜이거나 도청해 두었던 다수의 *Report* 메시지를 연속적으로 Querier로 전송한다면, Querier의 정상적인 동작에 피해를 입을 것이다. IGMP의 메시지 인증을 위한 또 다른 연구 [9,10,11,12]에서는 인증서버(AS)와 연결된 호스트와 Querier 사이에 공유한 대칭키를 기반으로 한 시도-응답(challenge-response) 메커니즘이 사용된다. 가입 호스트로부터 *Report* 메시지 수신 시, Querier는 호스트에게 시도값을 전송한다. 호스트는 공유키를 기반으로 계산한 응답값을 전송한다. 그 다음에, Querier는 시도값과 응답값의 확인을 위해 해당 값들을 AS로 전송한다. [10]에서는 수신자의 회계와 접근제어를 위해 RADIUS 서버의 사용을 제안하였다. [9,10,11,12]에서는 *Report* 메시지가 전자서명으로 보호되지 않기 때문에, DoS 공격은 피할 수 있다. 하지만, 다음의 상황을 고려해보자. *General Query* 메시지가 서브넷으로 멀티캐스트 될 때, 서브넷의 모든 그룹멤버는 *Report* 메시지를 전송해야 한다. 이 경우, Querier는 각 그룹멤버와 시도-응답 프로토콜을 개별적으로 수행해야 한다. 만약 서브넷에 현재 참여 중인 그룹멤버가 매우 많다면, Querier의 성능은 저하될 것이다. 대부분의 기존연구에서는 [5,13]을 제외하고는 대칭키의 분배에 대한 언급이 없다. 공통의 대칭키 분배를 위해서는 Needham-Schroeder 프로토콜[14]와 같이 티켓(ticket)이나 토큰을 사용한다. 하지만, 앞서 언급한 것과 같이 [5]는 실제 구현에 대해서는 매우 비효율적이며, [13]은 IGMP와의 상호작용에 대한 상세한 설명 없이 기본적인 키 분배 개념만을 제시한다.

Ⅲ. 설계원리

3.1 네트워크 아키텍처와 비즈니스 모델

기본적으로, 본 논문의 설계는 상업적인 멀티캐스트 서비스를 위한 네트워크 접근제어와 회계에 초점을 두고 있다. 본 논문에서는 [4]에서 제시한 네트워크와 비즈니스 모델을 사용하였으며, 해당 모델에서는 CP(Content Provider), NSP(Network Service Provider), Host(receiver)가 주요 구성요소이다. NSP는 다수의 CP와 Host 간의 채널(channel)을 제공한다. NSP는 다수의 CP가 제공하는 콘텐츠의 멀티캐스팅을 위한 인프라를 구축하고 유지하고, CP와 Host는 네트워크 자원을 사용하기 전에 NSP와 계약을 체결해야 한다.



〔그림 1〕 (a) 네트워크 아키텍처 (b) 관련 프로토콜

마찬가지로, Host는 CP의 콘텐츠를 수신하기 위해 CP와 또 다른 계약을 체결해야 한다. CP는 Host로부터 징수한 금액의 정해진 일부를 NSP에게 지불해야 한다. NSP는 CP와 관련된 채널정보를 유지하기 위해 CIS(Channel Information Server)를 작동한다. 특히, CIS는 회계와 청구를 목적으로 CP와 Host의 네트워크 자원 사용 기록을 유지한다.

3.2 CP와 NSP와의 관계

〔그림 1〕(b)는 본 논문에서 제안하는 접근 제어기법과 관련한 프로토콜의 집합을 나타낸다. 멀티캐스트 콘텐츠를 판매하거나 분배하기 위해서, CP는 AA(Address Allocation) 프로토콜을 통해 NSP로 멀티캐스트 주소할당을 요청한다. 본 논문에서는 해당 프로토콜이 어떻게 동작하는지에 대해서는 논의하지 않는다. 멀티캐스트 서비스의 기간과 타입과 같은 멀티캐스트 콘텐츠에 대한 정보를 기반으로 NSP는 멀티캐스트 주소 *maddr*을 유효시간 *exp*와 함께 할당한다. CP가 네트워크 자원을 사용하기 위해 NSP와 계약을 맺을 때, 대칭키 K_{GC}를 포함한 SA(Security Association)가 CP와 NSP 사이에 공유될 수 있다. 이제, session announcement protocol(SAP)[17]을 통해 CP는 판매를 원하는 멀티캐스트 콘텐츠에 대한 채널정보를 광고한다.

3.3 Host와 CP와의 관계

SAP를 통한 광고 후, CP는 정당한 Host만이 사용 가능한 그룹키로 암호화된 콘텐츠를 멀티캐스트 한다. 따라서 멀티캐스트 콘텐츠의 수신에 관심이 있는 Host는 CP와 계약을 맺고, Host에게 그룹키와 인증티켓(Authentication ticket)을 제공하기 위한 목적의 등록 프로토콜(registration protocol)을 수행한다. 그룹키는 암호화된 멀티캐스트 콘텐츠의 복호화를 위해 필요하고, 인증티켓은 NSP에 의해 동작되는 멀티캐스트 네트워크의 사용을 위한 일종의 credential로써 수신자 접근제어를 위해 사용된다. 즉, CP는 NSP와 Host의 중재자 역할을 수행한다. IPSEC이나 TLS 기반의 등록 프로토콜은 GDOI[18]와 같은 그룹 키관리 프로토콜의 일부로서, 본 논문에서는 해당 프로토콜의 동작에 대해서 설명하지 않을 것이며, 단지 인증티켓이 등록 프로토콜을 통해 Host에게 안전하게 전달됨만을 가정한다.

3.4 Host와 NSP와의 관계

Host가 NSP와 기본적인 네트워크 접근을 위해 계약을 맺었음에도 불구하고, 멀티캐스트 서비스는 NSP와 CP에 의해 제공되는 일종의 부가가치 서비스이기 때문에 Host에 대한 또 다른 접근제어를 수행해야 한다. 더욱이 만약 서브넷 상에서 수신자에 대한 적절한 접근제어가 수행되지 않으면 멀티캐스트 서비스는 DoS 공격의 근원이 될 수 있다. 티켓발행 프로토콜(Ticket-Issuing protocol)은 CP에 의해 제공된 인증티켓을 기반으로 Host를 인증하기 위한 목적으로 Host와 NSP 사이에 수행된다. 만약 인증에 성공하면 NSP는 Host에게 멀티캐스트 트래픽을 Host가 위치한 서브넷으로 유입하기 위

한 목적의 인증된 IGMP(Authenticated IGMP)를 위한 접근티켓(Access ticket)을 발행한다. 접근티켓을 사용하여, Host는 서브넷 상의 기본 멀티캐스트 라우터인 Querier와 SA를 공유할 수 있다. 다수의 멀티캐스트 라우터는 하나의 NSP에 의해 관리되기 때문에, 멀티캐스트 라우터들 사이에는 공유된 대칭키 K_{MR}를 포함하는 SA가 있음을 가정한다.

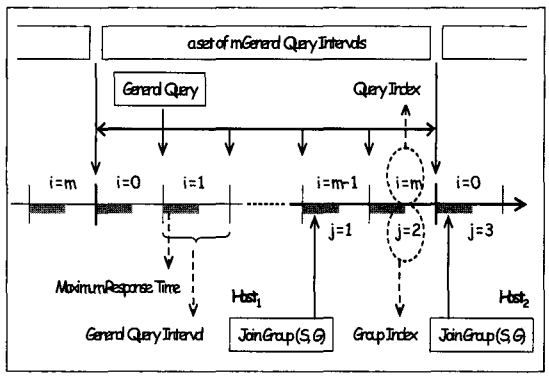
Ⅳ. 수신자 접근제어 기법

이번 장에서, 멀티캐스트 네트워크에 대한 수신자 접근을 제어하기 위한 인증된 IGMP 기반의 접근 제어기법을 제안한다. 수신자 접근제어 기능의 포함을 위해 IGMPv3의 기능이 확장되고 수정되었다. *Join Group / Leave Group / Report* 메시지 수신 시, Querier는 해당 메시지가 정당한 호스트가 보낸 유효한 메시지인지를 검사한다. Querier가 보낸 *Query* 메시지 또한 인증된다. 간단한 설명을 위해 다음과 같은 표기법을 정의한다.

- *CP*, *NSP*, *Host* : CP, NSP, Host의 IP 주소
- $h(\ )$: 일방향 해쉬함수
- $MAC(K)$: K로 계산한 MAC 값
- $[m_1, m_2]K$: 데이터 m_1과 m_2를 대칭키 K로 암호화한 값
- K_{GC}, K_{Net}, K_{MR}, K_{Host} : CP와 NSP, Host와 NSP, NSP와 Querier, Host와 Querier의 대칭키
- (P_Q, S_Q),(P_C, S_C) : Querier, CP 공개키/개인키
- $Sig(S)$: 선행하는 모든 데이터에 대해 개인키 S로 서명한 값
- *subprefix* : Host의 IP 서브넷 프리픽스
- *channel* : (*subprefix*, *CP*, *maddr*, *exp*)
- n : group hash chain의 길이
- x_0 : group hash chain의 루트값
- TS_x : X에 의해 생성된 timestamp
- *Auhentication_Ticket* : $[Host, channel, K_{Net}, (n, x_0, Sig(S_C)), P_C]K_{GC}$
- *Access_Ticket* : $[Host, channel, K_{Host}, (n, x_0)]K_{MR}$

4.1 인증을 위한 일방향 해쉬체인

Host와 Querier는 서로 간의 송수신 메시지의 인증을 위해 일방향 해쉬체인[16]을 사용한다. Querier는 서브넷 상에서 관리하고 있는 각 멀티캐스트 그룹에 지역멤버가 있는지를 알아보기 위해 주기적으로 *General Query* 메시지를 멀티캐스트 한다[1,2]. 해당 메시지는 서브넷 상의 모든 그룹멤버에 의해 인증되어야 한다. 메시지 인증과 소스 인증은 전자서명과 함께 일방향 해쉬체인을 기반으로 수행된다. [그림 2]와 같이, i=0으로 시작하는 query index는 Querier에 의해 제공된 query hash chain 기반의 상응하는 query hash value h(y_i)와 함께 *m General Query Interval* 내의 각 *General Query Interval*마다 할당된다.



[그림 2] Group and Query index

$$y_{i-1} = h(y_i) \text{ for } i = 1, 2, 3, \cdots, m$$

위에서 y_m은 *m General Query Interval*을 위해 임의로 선택되며, 루트값 y_0은 Querier에 의해 서명된다.

Querier는 매 *General Query Interval*마다 순차적으로 각 query hash value와 함께 *General Query* 메시지를 멀티캐스트 한다. 즉, 각 query hash value는 소스 인증을 위해 사용되어 지는 일종의 일회용 패스워드이며, *General Query Interval* 동안 유효하다. 특히, (0, y_0, $Sig(S_Q)$)는 첫 *General Query* 메시지가 보내어 질 때 서브넷 상으로 멀티캐스트 된다.

한편, 멀티캐스트 그룹 (*S*, *G*)가 있을 때, (*S*, *G*)의 각 그룹멤버에 의해 전송되는 *Join / Leave / Report* 메시지 또한 Querier에 의해 인증되어야 한다. 다음의 group hash chain은 CP가 임의로 선택한 seed 값 x_n에 기반하여 생성된다. 서명되어 Host로 보내어지는 루트값 x_0의 계산에 channel을 포함시키는 이유는 (*CP*, *maddr*, *exp*) 관련한 멀티캐스트 트래픽의 요청을 위한 group hash chain의 사용을 서브넷 프리픽스가 *subprefix*인 서브넷으로 제한하기 위해서이다.

$x_0 = h(x_1, channel)$

$x_{j-1} = h(x_j)$ for $j = 2, 3, \cdots, n$

위의 group hash chain의 seed 값 x_n은 3.3절에서 언급한 등록 프로토콜을 통해 Host에게 전달되며, 등록 프로토콜이 성공적이면, Host는 전달받은 x_n을 기반으로 n개의 group hash value $x_0, x_1, \cdots, x_{n-1}$을 계산할 수 있게 된다.

Host가 멀티캐스트 콘텐츠를 서브넷 상으로 계속 서비스 받기 위해서는 *General Query* 메시지에 대한 응답으로 일정한 간격으로 각 해쉬값 x_j를 Querier로 전송해야 한다. 이 경우, group hash chain은 서브넷의 (S, G) 그룹멤버의 해쉬값 여부를 확인하기 위한 그룹 인증을 위해 사용되는 일회용 패스워드이다. 또한, group hash chain은 서브넷의 멀티캐스트 트래픽의 측정에 적절히 사용될 수 있다. Host는 NSP에 의해 관리되는 몇몇의 서브넷에 분산된다. 만약 Host가 CP와 계약을 맺고 멀티캐스트 콘텐츠를 수신 받는다면, 네트워크 대역폭과 라우터의 작업부하 등 네트워크 리소스가 소모될 것이다. 만약 멀티캐스트 라우터가 해당 서브넷 내에서 멀티캐스트 세션을 얼마나 지속적으로 사용하였는지를 측정할 수 있다면, NSP는 해당 정보를 CP로의 회계정보로 사용할 수 있다. 멀티캐스트 라우터는 CP로의 청구를 위해 Host로부터 수집된 해쉬값의 수를 NSP로 보낸다. 해쉬체인과 관련된 청구문제는 5.5절에서 더 논의될 것이다.

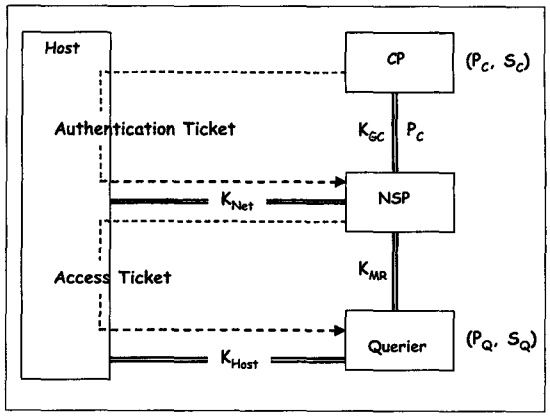
4.2 인증티켓과 접근티켓

Host는 *Authentication_Ticket*을 CP와의 등록 프로토콜 과정에서 발행 받는다.

$[Host, channel, K_{Net}, (n, x_0, Sig(S_C)), P_C]K_{GC}$

*Authentication_Ticket*은 "K_{Net}은 Host와 NSP 사이에 공유된 대칭키이고, *Host*는 *channel*과 관련한 멀티캐스트 콘텐츠의 수신을 원한다. x_0은 해쉬체인의 루트값, n은 group hash chain의 길이, $Sig(S_C)$는 P_C로 확인할 수 있는 (n, x_0)의 전자서명 값을 나타내며, 이 값들을 기반으로 Host에 대한 접근제어를 수행할 수 있다."를 의미한다.

$K_{Net}, (n, x_n), Authentication\_Ticket$



〔그림 3〕 Host, CP, NSP, Querier 간의 공유키

티켓발행 프로토콜은 Host와 NSP가 교환하는 두 개의 메시지로 구성된다. 해당 프로토콜의 주요 목적은 Host와 Querier의 대칭키 K_{Host} 공유에 있다. 먼저, Host가 다음의 메시지를 NSP로 전송한다.

Access_Ticket_Request : *CP*, *Authentication_Ticket*

NSP가 Host로부터 인증티켓을 수신하면, CP와 공유한 대칭키 K_{GC}로 복호화 할 수 있고, 인증티켓 내의 서명은 CP의 공개키 P_C로 확인할 수 있다.

그 후, NSP는 Host의 접근티켓 요청에 대해 기록하고, 인증된 IGMP에서 사용할 *Access_Ticket*을 생성한 후, 다음의 *Access_Ticket_Grant* 메시지를 Host로 보낸다.

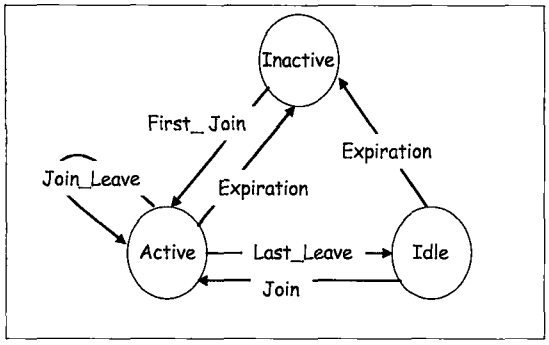
$Access\_Ticket = [Host, channel, K_{Host}, (n, x_0)]K_{MR}$

$Access\_Ticket\_Grant{:}Access\_Ticket, [K_{Host}, P_Q, TS_{NSP}]K_{Net}$

K_{Host}는 Host와 Querier가 공유한 대칭키로서, 해당 키는 서브넷 프리픽스가 *subprefix*인 서브넷 상에서 *exp*까지 유효하다. P_Q는 Querier에 의해 전자서명 된 *Query* 메시지를 확인하기 위한 Querier의 공개키이다. TS_{NSP}는 *Access_Ticket_Grant* 메시지의 freshness를 위한 값이다. 해당 메시지를 수신하면, Host는 Querier와 인증된 IGMP에 참여할 수 있게 된다. Host는 *channel*과 함께 다음과 같은 정보를 유지한다.

$K_{Host}, (n, x_n), P_Q, Access\_Ticket$

〔그림 3〕은 여러 개체 사이에 공유된 다양한 키들을 나타낸다. CP에 의해 발행된 인증티켓은 Host와 NSP 사이에 K_{Net}을 공유하기 위하여 사용되며, NSP에 의해 발행된 접근티켓은 Host와 Querier 사이에 K_{Host}를 공유



〔그림 4〕 멀티캐스트 그룹에 대한 상태 전이도

하기 위해 사용된다.

4.3 멀티캐스트 그룹의 상태 전이도

Querier는 만기가 지나지 않은 각 멀티캐스트 그룹 (*S*, *G*)에 대해서는 상태정보를 유지한다. 서브넷 상의 각 멀티캐스트 그룹은 〔그림 4〕에서 보는 것과 같이 3개의 다른 상태를 가진다.

First_Join은 정당한 Host가 서브넷에서 처음으로 멀티캐스트 그룹 (*S*, *G*)에 가입할 때를 나타내며, 멀티캐스트 그룹이 생성되는 상태로서 "Inactive"에서 "Active"로 상태가 변경된다. 즉, Querier가 멀티캐스트 트래픽을 해당 서브넷 상으로 유입하기 시작함을 의미한다. Host는 언제나 가입 또는 탈퇴할 수 있으며, 이를 Join_Leave로 나타낸다.

서브넷 상의 멀티캐스트 그룹에 적어도 한명의 Host가 남아있고 멀티캐스트 그룹의 만기가 지나지 않은 경우, 멀티캐스트 그룹은 여전히 "Active" 상태를 유지한다. Last_Leave는 서브넷 상의 멀티캐스트 그룹의 마지막 그룹멤버인 Host가 탈퇴하는 경우를 나타내며, "Active"에서 "Idle"로 상태가 변경된다. 이 경우, Querier는 서브넷으로 대응 멀티캐스트 트래픽의 유입을 중단한다. 하지만, 멀티캐스트 그룹의 만기가 지나지 않았음으로 그룹에 대한 상태정보를 완전히 제거하지는 않는다. 언제든 Host가 멀티캐스트 그룹에 가입하게 되면 상태는 "Idle"에서 "Active"로 변경될 수 있으며, 이를 Join으로 나타내며, Querier는 다시 멀티캐스트 트래픽을 서브넷 상으로 유입하기 시작한다. 현재 멀티캐스트 그룹이 "Active" 또는 "Idle" 상태일지라도 만기가 지나면 "Inactive"로 상태가 변경되며, 이를 Expiration으로 나타내며, 이 후로 멀티캐스트의 상태정보가 완전히 제거된다.

4.4 멀티캐스트 그룹 가입

Join Group 메시지의 주요 목적은 다음과 같이 세 가지로 나눌 수 있다. 첫 번째, 정당한 Host가 현재 활동중이지 않은 멀티캐스트 그룹 (*S*, *G*)에 가입을 원할 때, Querier가 해당 서브넷으로 멀티캐스트 트래픽을 유입하기 위해서이다. 두 번째, 서브넷 상의 특정 멀티캐스트 그룹에 가입한 Host의 목록을 유지하기 위해서이다. 세 번째, 정당한 가입 호스트에게 현재 *General Query Interval*에 대한 query index와 group index를 알림으로써 Host가 Querier와 query index와 group index를 동기화할 수 있도록 하기 위해서이다. 서브넷 상의 Host가 (*S*, *G*)에 처음 가입할 때, 멀티캐스트 그룹 (*S*, *G*)에 대해 j=1로 시작하는 group index는 각 *General Query Interval*에 성공적으로 설정된다. group index는 현재 *General Query Interval*에 대한 group hash value를 선택하기 위해 사용된다. $Host_l$이 〔그림 2〕와 같이 *General Query Interval*이 i=m-1일 동안 서브넷에서 처음으로 (*S*, *G*)=(*CP*, *maddr*)에 가입할 때, group index j=1이 *General Query Interval*에 설정되고, 다음과 같은 *Join Group* 메시지를 Querier로 보낸다.

$$Join(S,\ G),\ TS_{Host_l},\ MAC(K_{Host_l}),\ Access\_Ticket$$

Join(*S*, *G*)는 *Join Group*과 관련한 파라미터의 집합으로서, 특히, *S*는 *CP*로 설정이 되고, *G*는 *maddr*로 설정이 된다. TS_{Host_l}는 (*S*, *G*)에 가입 시 $Host_l$이 생성한 타임스탬프이다. $MAC(K_{Host_l})$는 $Host_l$의 IP 주소와 함께 *Join*(*S*, *G*)와 TS_{Host_l}을 기반으로 계산된 값이다. 해당 메시지를 수신하면, Querier는 먼저 NSP와 공유한 K_{MR}을 기반으로 접근티켓을 복호화하여 *Access_Ticket*에 포함되어 있는 K_{Host_l}을 구함으로써 $MAC(K_{Host_l})$을 확인할 수 있다. 또한 *Access_Ticket*에 포함되어 있는 *channel*이 유효한지를 검사한다. 즉, (*S*, *G*)와 (*CP*, *maddr*)가 동일한지, 현재 연결되어 있는 Querier의 서브넷 프리픽스와 *subprefix*가 동일한지, *exp*는 지나지 않았는지를 확인한다. 만약 모든 검사가 성공적이면, Querier는 해당 그룹을 서브넷의 멀티캐스트 그룹 멤버쉽 리스트에 추가한다. 그렇지 않은 경우는 단지 메시지를 드롭한다. 특히 Querier는 각 멀티캐스트 그룹에 대한 상태정보를 생성한다.

$$<channel,\ (x_0,\ j{=}1,\ x_j),\ \{(Host_l,\ K_{Host_l},\ TS_{Host_l},\ \mathbf{0/1})\}>$$

상태비트 0/1은 $Host_1$이 해당 서브넷 상에서 활동 중인지를 나타낸다. 적어도 한명의 활동 중인 그룹멤버가 있기 때문에 멀티캐스트 그룹 (*CP*, *maddr*)은 "Active" 상태이다. 만약 $Host_1$이 서브넷 상의 멀티캐스트 그룹을 탈퇴하면 상태비트는 **0**/1으로 변경된다. TS_{Host1}을 유지하는 이유는 재생공격에 대응하기 위해서이다. 이후 *Join Group* 메시지에 대한 응답인 *Group and Source Specific Query* 메시지는 〔그림 5〕에서 보는 것과 같이 $Host_1$에게만 보내진다.

$Query(S, G), (i, y_i, j, x_j), MAC(K_{Host1})$

$Query(S, G)$는 *Group and Source Specific Query*와 관련한 파라미터의 집합이다. 〔그림 2〕에 기반하여, $i = m-1$이고 $j = 1$이다. 이러한 *Query* 메시지는 *Maximum Response Time*은 물론 group index, query index와 같은 현재 *General Query Interval*의 상태를 $Host_1$에게 알리기 위한 목적이기 때문에, $Host_1$은 이러한 *Query* 메시지에 응답하지 않는다. 이제, $Host_1$은 다음의 상태정보를 유지한다.

$K_{Host}, (n,x_n j=1), (i=m-1,y_i), P_Q, TS_{Querier}$, *Access_Ticket*

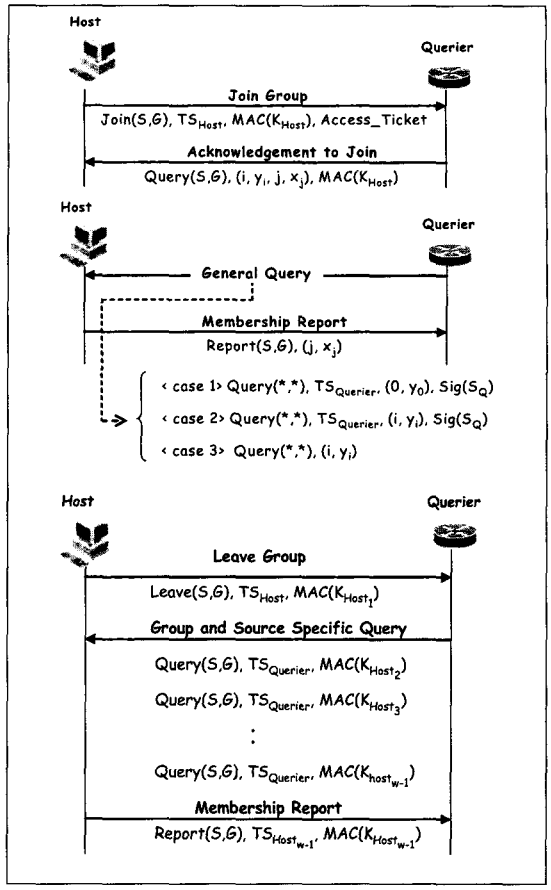
다음의 〔그림 2〕와 같이 $Host_2$가 *General Query Interval* $i = 0$일 때 동일한 그룹 (*S*, *G*)에 성공적으로 가입할 때, 새로운 그룹멤버의 IP 주소와 대칭키가 (*S*, *G*)에 대한 상태정보에 포함된다.

또한 $Host_2$로 현재의 group index와 query index가 포함된 응답 메시지를 전송한다. 이제, (*S*, *G*)에 대한 상태정보는 다음과 같이 유지된다.

$<channel, (x_0, j=3, x_j), \{(Host_1, K_{Host1}, TS_{Host1}, \mathbf{0/1}), (Host_2, K_{Host2}, TS_{Host2}, \mathbf{0/1})\}>$

4.5 멀티캐스트 그룹 유지

General Query 메시지가 보내질 때, *Query* 메시지가 처리되는 방식을 기반으로 다음의 세 가지 다른 경우가 있다. 첫 번째 경우(case 1)는 Querier가 새로운 *m General Query Interval*을 시작하기 위한 경우이다. 임의로 선택한 y_m을 기반으로 새로운 hash chain $y_{i-1} = h(y_i)$, for j=1, 2, 3, ⋯, m을 생성하고, hash chain의 루트 값 y_0은 *Maximum Response Time*과 같은 다른 파라미터들과 함께 전자서명 된다.



〔그림 5〕 인증된 IGMP 프로토콜

$Query(*, *), TS_{Querier}, (0, y_0), Sig(S_Q)$

$TS_{Querier}$는 *Query* 메시지의 freshness를 위한 것이며, $Query(*, *)$는 *General Query*와 관련한 파라미터들의 집합이다. 해당 메시지 수신 시, 서브넷의 각 그룹멤버는 P_Q를 이용하여 서명을 확인한 후, 파라미터들과 (0, y_0)과 같은 상태 정보를 저장한다.

두 번째 경우(case 2)는 *Maximum Response Time*과 같이 *General Query*와 관련한 몇몇의 파라미터를 변경하기 위한 경우이다. 예를 들면, 만약 서브넷 상에 그룹멤버의 수가 증가한다면, Querier는 *Maximum Response Time*을 다음의 *m General Query Interval*의 시작할 때까지 즉시 증가시켜야 한다. 이러한 변경 또한 Querier의 전자서명으로 보호될 수 있다.

$Query(*, *), TS_{Querier}, (i, y_i), Sig(S_Q)$

i는 현재 *General Query Interval*에 대한 query in-

dex이다. 메시지 수신 시, 각 그룹멤버는 저장해 놓은 y_{i-1}이 y_i를 기반으로 계산한 $h(y_i)$와 일치하는지를 검사한다. 또한 서명의 확인에 성공하면, 그룹멤버는 query 파라미터를 변경하고 $(i\text{-}1, y_{i-1})$을 (i, y_i)로 대체한다.

세 번째 경우 (case 3)는 멀티캐스트 그룹이 여전히 참여 중인지를 검사하기 위한 전형적인 *General Query* 메시지의 경우이다. 이 경우, 각 그룹멤버는 query index와 query hash value가 유효한지를 검사하고, index와 hash value를 새로운 값으로 수정한다.

Query(*, *), (i, y_i)

General Query 메시지를 수신하였을 때, Host가 아래의 *Report* 메시지로 응답하기 전에 Host는 이미 그룹 (S, G)에 가입하였고, (S, G)에 대한 지연 타이머(delay timer)인 *Max_Response_Time*을 (0, *Max_Response_Time*) 범위에서 선택된 임의의 수를 기반으로 설정한다.

Report(S, G), (j, x_j), where j=2, 3, ⋯, n

타이머가 만료되면, Host는 *Report* 메시지를 Querier로 보내고, 현재 group hash value와 group hash index를 갱신한다. *Report* 메시지 내의 x_j를 기반으로 $h(x_j)$를 계산한 후, Querier는 유지하고 있던 x_{j-1}이 $h(x_j)$와 동일한지를 검사한다. 만약 확인에 성공하면, Querier는 해쉬값을 갱신, 즉 j는 물론 x_{j-1}을 x_j로 대체한다.

4.6 멀티캐스트 그룹 탈퇴

만약 Host가 (S, G)를 탈퇴한다면, 다음의 *Leave Group* 메시지를 Querier로 전송한다.

Leave(S, G), TS_{Host}, $MAC(K_{\text{Host}})$

Querier는 (S, G)에 대한 상태 정보에서 Host를 확인하고, Host의 IP 주소와 함께 *Leave*(S, G), TS_{Host}를 기반으로 계산된 $MAC(K_{\text{Host}})$를 확인한다. 만약 확인에 성공하면, Querier는 유지하고 있던 Host의 상태정보의 상태비트를 **0**/1으로 변경한다. 그렇지 않은 경우는 단지 메시지를 드롭한다. 만약 (S, G)에 대한 모든 호스트들의 상태비트가 **0**/1으로 설정되면, 멀티캐스트 그룹은 "Idle" 상태로 변경되고 대응하는 멀티캐스트 트래픽은 차단된다. 만약 *exp*가 지나기 전에 그룹멤버가 (S, G)에 재가입을 원할 경우, 새로운 *Join Group* 메시지를 송신하고, 만약 메시지가 유효하면, 그룹멤버의 상태비트가 다시 **0/1**로 설정될 것이다.

몇몇의 그룹멤버가 *Leave Group* 메시지를 전송하지 않고 묵시적으로 (S, G)를 탈퇴했다고 가정하자. 그러면, 만약 (S, G)의 마지막 그룹멤버가 *Leave Group* 메시지를 전송하고 탈퇴한다면, 묵시적으로 탈퇴한 그룹멤버의 상태비트가 여전히 0/1로 설정되어 있기 때문에, 이어지는 두 번의 *General Query Interval* 동안 해당 서브넷으로 불필요한 트래픽은 유입된다. 따라서 Querier는 멀티캐스트 트래픽의 차단여부를 결정하기 위해 *Group and Source Specific Query* 메시지를 전송해야 한다. [그림 5]에서와 같이, Host_1이 *Leave Group* 메시지 전송 후 (S, G)를 탈퇴하였다고 가정하자. L = { Host_2, Host_3, ⋯, Host_w }은 Querier가 유지하고 있는 (S, G)의 활동 중인 그룹멤버라고 가정하자. 그 중, M = { Host_2, Host_3, ⋯, Host_{w-2} }은 묵시적으로 (S, G)를 탈퇴한 그룹멤버의 집합으로, 해당 상태비트는 여전히 **0/1**로 설정되어 있다고 하자. 그러면, Querier는 유효한 *Report* 메시지를 수신할 때까지 *Group and Source Specific Query* 메시지를 L의 각 그룹멤버에게 차례로 유니캐스트(unicast) 한다. M의 각 그룹멤버는 이미 (S, G)를 탈퇴했기 때문에 M에 포함된 그룹멤버로부터의 *Report* 메시지는 전송되지 않는다. 결국, Querier는 L에서 M을 삭제한다. 하지만, Host_{w-1}은 활동 중인 그룹멤버이기 때문에 *Report* 메시지를 전송할 것이고, (S, G)에 적어도 한명의 그룹멤버가 존재함을 의미하고, Querier는 그룹멤버가 존재하므로 *Group and Source Specific Query* 메시지를 Host_w에게 전송하지 않는다. 즉, *Group and Source Specific Query* 메시지의 주요 목적은 메시지 전송 없이 탈퇴한 그룹멤버를 찾는 것이 아니라, (S, G)가 여전히 활동 중인가를 결정하기 위함이다.

V. 분석과 논의

5.1 위조된 Query 메시지에 대한 안전성

서브넷에 인증된 *General Query* 메시지를 멀티캐스트하기 위한 직관적인 방법은 Querier와 그룹멤버들 간에 공유한 일종의 그룹키 기반의 MAC을 사용하는 것이다. 그룹키를 사용하는 원리는 서브넷 상의 많은 그룹멤버에게 단 한번의 *General Query* 메시지를 보내는 것이다. 하지만, 서브넷 상의 그룹멤버는 그룹키를 알기

때문에, 부당한 행동을 할 수 있다. 즉, 위조된 *Query* 메시지를 송신할 수 있다. 따라서 그룹키 기반의 MAC은 위조된 *Query* 메시지에 대한 대응책으로는 적당하지 않다. 또 다른 방법은 각 *Query* 메시지마다 직접 전자서명 하는 방식이 있지만, 전자서명으로 인해 발생하는 큰 계산량으로 인해 DoS 공격에 노출될 수도 있다는 문제점이 존재한다. 공격자가 *Query* 메시지에 서명값으로 임의의 쓰레기 데이터를 첨부하고, 서브넷으로 지속적으로 멀티캐스팅 한다고 가정하자. 해당 메시지가 유효하지 않다고 판명되어도, 그룹멤버는 해당 서명의 확인을 위한 작업으로 상당한 시간을 소비하기 때문이다. 이와 같은 문제점을 고려하여, 본 논문에서는 *Query* 메시지의 인증을 위해 query hash chain의 사용을 제안하였다. 각 query hash chain의 루트값은 *Query* 파라미터와 함께 Querier에 의해 전자서명 되기 때문에, 이어지는 query hash value는 해쉬체인의 일방향 속성으로 인해 MAC이나 전자서명으로 보호할 필요가 없다. query hash value는 Querier의 소스 인증을 위한 일회용 패스워드와 같은 역할을 한다. 각 그룹멤버는 새로운 query hash chain이 생성될 때, 전자서명으로 보호된 *Query* 파라미터를 유지하고 갱신한다. 하지만, 만약 현재 query hash chain을 완전히 사용하기 전에 *Query* 파라미터를 수정할 필요가 있을 경우에는 다음 *General Query Interval*에 보내져야 할 *Query* 메시지는 또한 전자서명으로 보호되어야 한다.

$Query(*, *), TS_{Querier}, (i, y_i), Sig(S_Q)$

이와 같은 경우, 공격자는 현재 *Query interval*의 query index와 hash value (i, y_i)를 모르기 때문에 위에서 언급한 DoS 공격이 불가능하다. 따라서, 각 그룹멤버는 서명을 확인하기 전에 먼저 (i, y_i)가 유효한지 검사해야 한다. Host가 (S, G)에 가입하고, Querier로부터 다음과 같은 응답 메시지를 받는다고 가정하자.

$Query(S, G), (i, y_i, j, x_j), MAC(K_{Host})$

Host가 Querier에 의해 전자서명 된 query hash chain의 루트값 y_0을 가지고 있지 않더라도, MAC 값이 유효하다면 query hash value y_i를 수락할 수 있다. K_{Host}는 Querier와 Host 자신만이 알고 있기 때문이다. *Join Group* 메시지에 대한 응답인 *Group and Source Specific Query* 메시지에는 현재 *General Query Interval*에 대한 group hash index와 함께 group hash value가 포함되어야 한다. 그렇지 않으면, Querier에 의해 수집된 group hash value는 청구와 회계의 기초가 될 수 있기 때문에 Querier가 부당한 행동할 수 도 있다. 만약 *Group and Source Specific Query* 메시지가 group hash value와 현재의 group index j를 포함하지 않는다고 가정하자. 만약 Querier가 임의의 Host로부터 *Join Group* 메시지 수신할 때 다음과 같은 메시지를 전송한다면, Host는 다음 *General Query Interval* 동안 $(j+1, x_{j+1})$ 대신 $(j+e+1, x_{j+e+1})$을 전송할 것이고, 이를 통해 부당하게 e개 더 많은 group hash value를 취할 것이다.

$Query(S, G), (i, y_i, j+e), MAC(K_{Host})$

하지만, 만약 group hash value가 *Group and Source Specific Query* 메시지에 포함된다면, group hash chain의 일방향 속성으로 Querier에 의한 이러한 공격은 발생할 수 없게 된다.

5.2 위조된 Join / Leave 메시지에 대한 안전성

$MG = \{(S_1, G_1), (S_2, G_2), \cdots\}$를 현재 서브넷 상에서 활동 중인 멀티캐스트 그룹의 집합이라고 하고, HG_j, $j=1, 2, 3, \cdots$,를 (S_j, G_j)에 가입한 정당한 그룹멤버의 집합이라고 하자. 공격자가 해당 서브넷으로 새로운 멀티캐스트 트래픽을 유입시키기 위해서, 즉 새로운 멀티캐스트 그룹 $(S, G) \notin MG$에 가입하기 위해서는 유효한 *Access_Ticket*과 유효한 *Join Group* 메시지를 생성하기 위한 K_{Host}가 있어야 한다. 따라서 유효한 *Access_Ticket* 없이는 DoS 공격은 불가능하다. 한편, 공격자가 정당한 그룹멤버가 보낸 다음과 같은 *Join Group* 메시지를 관측했다고 가정하고, Host는 서브넷 상에서 (S, G)의 마지막 그룹멤버이고, (S, G)를 탈퇴한다고 가정하자.

$Join(S, G), TS_{Host}, MAC(K_{Host}), Access\_Ticket$

Querier는 서브넷 상으로 대응하는 멀티캐스트 트래픽의 유입을 중단할 것이다. 이 때, 공격자는 해당 서브넷으로 다시 트래픽이 유입될 수 있도록 관측한 메시지를 기반으로 DoS 공격을 시도할 것이다. 하지만, 해당 메시지에는 TS_{Host} 파라미터를 포함하고 있기 때문에 저장된 메시지의 재전송은 불가능하다. (S, G)의 만기가 지나지 않는 한, Querier는 TS_{Host}를 포함한 (S, G)의 상태정보를 유지한다.

어떤 j에 대해 $|HG_j| > 0$를 가정하자. 만약 공격자가 HG_j에 포함되어 있는 모든 그룹멤버가 생성한 것과 같은 *Leave Group* 메시지를 생성할 수 있다면, (S_j, G_j)와 관련한 멀티캐스트 트래픽은 차단될 것이다. 하지만, 이러한 DoS 공격은 HG_j에 포함되어 있는 Host의 K_{Host}를 모르면 불가능하다.

$Leave(S, G), TS_{Host}, MAC(K_{Host})$

5.3 Host의 묵시적인 가입

Host가 $|HG_j| > 0$인 서브넷 상의 (S_j, G_j)에 가입을 원한다고 가정하자. (S_j, G_j)와 관련한 멀티캐스트 트래픽은 이미 서브넷으로 유입되고 있기 때문에, Host는 (S_j, G_j)으로 명시적인 가입 없이도 해당 트래픽에 접근할 수 있다. 하지만, 만약 HG_j의 모든 그룹멤버가 탈퇴하면, 잠시 후에 멀티캐스트 트래픽은 Host가 명시적으로 (S_j, G_j)에 가입할 때까지 임시적으로 차단될 것이다. 따라서, 그룹멤버가 지속적으로 멀티캐스트 서비스를 수신하기 위해서는 명시적인 가입이 바람직하다.

5.4 Host의 묵시적인 탈퇴

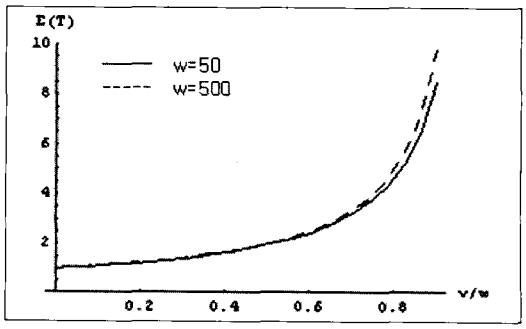
임의의 Host는 *Leave Group* 메시지의 송신 없이 묵시적으로 탈퇴할 수 있으며, 다음의 두 경우가 있을 수 있다. 첫 번째, Host가 서브넷 상의 (S_j, G_j)의 마지막 그룹멤버이고, 묵시적으로 탈퇴한다고 가정하자. Querier는 그것을 인식할 수 없기 때문에, 멀티캐스트 트래픽은 이어지는 두 번의 *General Query Interval*이 지날 때까지 해당 서브넷으로 지속적으로 유입된다. 두 번째, 어떤 Host가 묵시적으로 멀티캐스트 그룹을 탈퇴하고, 그 후 또 다른 Host가 명시적으로 탈퇴한 경우를 가정하자. 이와 같은 경우 Querier는 자신이 보낸 *Group and Source Specific Query* 메시지에 대한 응답인 *Report* 메시지가 수신되지 않으면 멀티캐스트 트래픽을 즉시 차단할 수 있다. 반면, 만약 *Report* 메시지가 수신되면, 멀티캐스트 트래픽은 4.6절에서 설명한 것과 같이 차단되지 않을 것이다.

Group and Source Specific Query 메시지는 Host와 Querier가 공유한 키인 K_{Host}에 기반하여 개별적으로 인증된다. 악의를 가진 그룹멤버가 서브넷 상에 다수의 의미 없는 *Report* 메시지를 유도하기 위하여, 위조된 *Group and Source Specific Query* 메시지를 생성할 수 있으므로 모든 그룹멤버가 공유한 그룹키에 기반한 그룹인증은 사용할 수 없다. 5.1절에서 언급한 것과 같이 인증을 위해서 전자서명을 사용하는 것 또한 DoS 공격이 발생할 수 있기 때문에 적당하지 않다.

다음에서는, (S, G)의 활동 중인 그룹멤버로부터 *Report* 메시지를 수신하기 전까지 전송되어야 하는 *Group and Source Specific Query* 메시지 수를 계산한다. w를 (S, G)의 활동 중인 그룹멤버의 수라고 하고, 이후 어떤 Host가 *Leave Group* 메시지를 송신 후 (S, G)를 탈퇴한다고 가정하자. 또한, v를 묵시적으로 (S, G)를 탈퇴한 그룹멤버라고 가정하자. 만약 Querier가 서브넷 상에서 활동 중인 남아있는 그룹멤버를 선택할 수 있다면, $\{(w-v)/w\}$의 확률로 오직 하나의 *Group and Source Specific Query* 메시지를 송신한 후, 해당 그룹멤버로부터 *Report* 메시지를 수신할 수 있을 것이다. *Report* 메시지를 수신하기 전 두 개의 *Group and Source Specific Query* 메시지를 송신할 확률은 $\{v/w\}\cdot\{(w-v)/(w-1)\}$이다. 일반적으로 Querier는 $\{v/w\}\cdot\{(w-1)/(w-1)\}$ … $\{(w-i+2)/(w-i+2)\}\cdot\{(w-v)/(w-i+1)\}$의 확률로 $i(\geq 2)$ 개의 *Group and Source Specific Query* 메시지를 송신 후 *Report* 메시지를 수신할 수 있다. 따라서, 전송 *Group and Source Specific Query* 메시지의 기대치는 다음과 같이 계산될 수 있다.

$$E(T)=1\cdot\left(\frac{w-v}{w}\right)+\sum_{i=2}^{v+1} i\cdot\left(\frac{v}{w}\right)\left(\frac{v-1}{w-1}\right)\cdots$$
$$\left(\frac{v-i+2}{w-i+2}\right)\left(\frac{w-v}{w-i+1}\right)$$

[그림 6]은 위의 식을 기반으로 한 수치적인 결과로서, 전송될 *Group and Source Specific Query* 메시지



[그림 6] *Group and Source Specific Query* 메시지의 기대치

의 기대치에 대한 (v/w)의 영향에 대해서 알 수 있다. (v/w)가 증가하면 전송 *Group and Source Specific Query* 메시지의 수가 점차적으로 증가하며, w의 수치와는 거의 무관함을 알 수 있다. [그림 6]에서 본 것과 같이, Querier는 *Report* 메시지를 수신하기 위해 그다지 많지 않은 *Group and Source Specific Query* 메시지만을 필요로 한다. 즉, (v/w)가 0.4 이하일 때, 1개 또는 2개의 *Group and Source Specific Query* 메시지이면 충분하다.

5.5 회계와 청구문제

CP가 멀티캐스트 그룹멤버에 대한 group hash chain을 생성할 때, 멀티캐스트 세션의 기간과 group hash chain의 길이를 연관 지을 수 있다. 예를 들면, *General Query Interval*이 2분이고 멀티캐스트 세션이 3시간 동안 지속된다면, hash chain의 길이 n은 90(≈180/2)+α로 계산된다. α는 실제 세션 시작 시간과 처음 Host가 서브넷의 멀티캐스트 그룹에 가입한 시간과의 시간차를 고려한 여분이다. 3시간 멀티캐스트 세션에 대해 지불할 때, Host는 group hash chain이 포함된 인증티켓을 얻을 수 있다. 멀티캐스트 그룹의 각 그룹멤버는 멀티캐스트 트래픽의 수신을 위해 하나의 *General Query Interval*에 대해 하나의 group hash value를 사용하기 때문에, Querier가 그룹멤버로부터 수집한 group hash value로 서브넷의 멀티캐스트 트래픽으로 인해 사용한 네트워크 리소스 금액을 부과하는 근거가 될 수 있다. CP는 NSP가 유지하는 네트워크 리소스를 사용함으로써 비즈니스를 경영하기 때문에, CP는 서로간의 미리 동의된 계약에 따라 NSP에게 지불해야 한다.

NSP는 인터넷 연결과 같은 기본적인 네트워크 서비스에 대해 Host에게 청구할 수 있다. 멀티캐스트 서비스는 일종의 부가가치 서비스이기 때문에, NSP는 멀티캐스트 서비스를 요구하는 Host에게 추가적인 요금의 징수를 원한다. 하지만, NSP가 직접적으로 멀티캐스트 네트워크 접근을 위한 이더넷(Ethernet)과 같은 네트워크 미디어의 공유에 대해 Host에게 청구하기는 쉽지 않다. 한 번에 서브넷 미디어의 한 Host가 인증 받고, Querier는 멀티캐스트 데이터그램(datagram)을 서브넷으로 전달을 시작한다. 그 결과, 서브넷의 다른 Host들은 인증 없이도 멀티캐스트 데이터그램을 수신할 수 있다. 이러한 문제에 대한 가장 간단한 해결책은 [9]에서 언급한 것과 같이 Querier가 멀티캐스트 데이터그램을 서브넷의 각 그룹멤버에 대해 각각 다시 암호화하는 것이다. 하지만, 그 방법은 Querier에 엄청난 계산적 부담을 발생시킨다. 결과적으로, CP가 등록과정에서 Host로부터 징수한 요금의 일부를 NSP에게 지급하는 것이 합리적이다.

5.6 메모리와 계산적 오버헤드

Querier에 대한 메모리 요구량은 현재 서브넷에 참여중인 그룹멤버에 대한 정보이다. 각 멀티캐스트 그룹 (S, G)는 다음의 데이터를 유지해야 한다.

channel, (x_0,j,x_j), {list of pairs $(Host_1,K_{Host},TS_{Host},0/1)$}

$h(\ )$가 128비트 해쉬값을 생성하고 대칭키 또한 128비트라고 가정하자. w를 서브넷의 (S, G)에 가입하는 그룹멤버의 수라고 하면, 멀티캐스트 그룹에 대해 각각 $26+(16+2+16)+w(4+16+16+1) = 60+37w$ 바이트가 필요하다. query hash chain y_{i-1} 또한 Querier에 의해 유지된다. *General Query Interval*이 120초이고 m=180이라고 가정하자. 그 다음, Querier는 case 1의 *Query* 메시지를 위해 매 6시간마다 서명 작업을 수행한다. Querier가 case 2와 case 3의 *Query* 메시지를 위해 필요한 y_i를 계산하기 위해서는 두개의 극단적인 방법이 있다. 하나는 y_0에서 y_m까지의 모든 해쉬값을 저장하는 것이고, 다른 하나는 단지 y_m만을 저장하고 m-1번은 일방향 해쉬함수를 적용함으로써 y_i를 계산하는 것이다. 만약 m이 큰 값이라면 두 방법 모두 비효율적이다. 대안으로, 만약 intermediate hash value(e.g. y_{10}, y_{20}, …, y_{180} when m=180)가 미리 계산, 저장된다면 Querier 상에서 y_i를 계산하기 위한 부담과 메모리 부담이 균형을 이룰 것이다. 따라서, 추가 메모리의 $16 \cdot (m/c)$ 바이트는 확보해야 하며, y_i를 계산하기 위한 평균 $c/2$ 해쉬작업이 수행된다. c는 두개의 연속적인 intermediate query hash index의 차이를 나타낸다. 각 *Join Group* 메시지에 대해 Querier는 한번의 *Access_Ticket* 복호화와 $MAC(K_{Host})$ 계산을 위한 한번의 MAC 작업이 수행한다. *Leave Group* 메시지 수신 시, Querier는 우선 그룹 멤버쉽 목록에 있는 K_{Host}를 검색한 후, $MAC(K_{Host})$ 확인을 위한 MAC 작업을 수행한다. 한편, Host에 대한 메모리 요구량은 가입한 멀티캐스트 그룹에 대한 정보이다.

〔표 1〕 Memory and Computation Overheads for a Multicast group

	Memory (bytes)	General Query	Join Group	Leave Group
Querier	(60+37w)+ 16 • (m/c)	(c/2) Hash + [1 Sig]*	1 Dec + 1 MAC	2 MAC + E(T)
Host	230 + 16 • (n/d)	(d/2) Hash + [1 Ver]**	1 MAC	1 MAC

[1 Sig]* : *Query*(case1과 case2)에 대한 서명
[1 Ver]** : *Query*(case1과 case2)에 대한 서명확인
E(T) : Group and Source Specific Query 메시지 수(기대치)
Hash=hash operation, MAC=MAC operation
Dec=decryption operation,

channel, K_{Host}, (n, x_n, j), (i, y_i), P_Q, *Access_Ticket*

query hash chain으로서, intermediate group hash value x_j는 (n, x_n, j) 대신 유지할 수 있다. 예를 들면, n=100이고 두개의 연속적인 intermediate group hash index 간의 차 d=5일 때, x_5, x_{15},x_{20}, …, x_{100}이 미리 계산되고 저장된 해쉬값이다. 따라서, group hash chain에 대해 16 • (n/d) 바이트가 필요하다.

P_Q를 위한 128바이트와 함께 *channel*을 위한 26바이트, *Access_Ticket*을 위한 42바이트, TS_{Host}를 위한 16바이트, (i, y_i)를 위한 18바이트가 필요하다고 가정할 때, Host는 가입된 멀티캐스트 그룹에 대해 각각 16 • (n/d)+230바이트 메모리를 확보해야 한다. 〔표 1〕은 IGMP 관련 메시지 처리 시, Host와 Querier에 대한 메모리와 계산 요구량을 나타낸다.

5.7 기존 기법과의 비교

〔표 2〕는 본 논문에서 제안한 기법과 함께 다양한 참조 수신자 접근제어기법의 특징과 안전성을 나타낸다. 재생공격은 정당한 그룹멤버에 의한 메시지를 저장하였다가 다시 보내는 공격방식이다. 해당 공격으로 인해 불필요한 멀티캐스트 트래픽이 서브넷으로 유입될 수 있다. DoS 공격은 Querier 또는 그룹멤버를 분주하게 만드는 공격방식이다. 해당 공격은 인증방식을 부적절히 사용함으로써 유도가능하다. 5.1절에서 언급한 것과 같이, 〔6,7,8〕에 대한 보완책 없이 만약 *Report* 메시지 인증을 위해 서명이 사용된다면, Querier는 엄청난 무의미한 계산을 수행해야 한다. 〔9,10,12〕에서, 시도-응답 인증방식이 사용되었다. *Report* 메시지 수신 시, Querier는 시도값을 Host로 보내고 응답을 기다리는 동안 Host와 관련한 상태를 생성한다. 만약 공격자가 시도에 대한 응답 없이 대량의 위조된 *Report* 메시지를 보낸다면, Querier는 DoS 공격에 당하게 된다.

〔표 2〕 기존 수신자 접근제어기법과의 비교

	Authentication method	Replay Attack	DoS Attack	Protection for Query	Key Distribution Method
〔1〕 〔2〕	None	insecure	insecure	No	None
〔5〕	MAC	secure	secure	No	Token
〔6〕	signature	insecure	insecure	No	Token
〔7〕	signature	insecure	insecure	No	PKI
〔8〕	signature	secure	insecure	Yes	CBA
〔9〕 〔10〕	CR	secure	insecure	No	Pre-shared
〔11〕	EAP	secure	secure	No	Pre-shared
〔12〕	CR	secure	insecure	No	Pre-shared
Ours	Hash Chain	secure	secure	Yes	Ticket

Query 메시지를 보호하기 위한 방법의 기술 없이, 기존에 제안된 대부분의 기법들은 안전한 *Report* 메시지를 위한 방법만을 소개한다. 만약 *Query* 메시지가 보호되지 않는다면, 2.2절에서 언급한 것과 같이 DoS 공격이 발생가능하다. 회계와 청구와 관련한 방법으로 〔10〕에서는 Host의 네트워크 접근 기간을 측정하기 위해 RADIUS 서버를 사용한다. Host가 인증받자마자, 타이머가 작동된다. 보안적인 관점에서, Host와 NSP 사이의 청구의 분쟁은 이와 같은 간단한 방법으로는 조정될 수 없다. 본 논문에서 제안하는 기법은 전자서명으로 보호된 일방향 해쉬체인을 기반으로 하기 때문에, 그러한 분쟁이 발생하지 않는다.

Ⅵ. 결론

멀티캐스트 애플리케이션의 안전성 문제는 빠른 상업적인 적용에 있어서 하나의 장애요인이다. 멀티캐스트 보안과 관련한 대부분의 연구는 그룹키 관리가 주를 이루며, 반면 서브넷 레벨에서의 멀티캐스트 서비스에 대한 접근제어와 관련한 연구는 IETF 단체 외에서는

주목받지 않는다. 본 논문에서는 자원고갈 공격과 DoS 공격에 대응적인 안전하고 효율적인 서브넷 상에서의 수신자 접근제어기법을 제안하였다. 또한, 네트워크와 비즈니스 모델을 기반으로 회계와 청구와 관련한 근거 있는 데이터 수집을 위한 방법을 제안하였다.

참고문헌

[1] W. Fenner, "Internet Group Management Protocol, Version 2," *RFC 2236*, Nov.

[2] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "InternetGroup Management Protocol, Version 3," *RFC 3376*, Oct. 2002.

[3] L. Gong and N. Shacham, "Elements of trusted multicasting," *in Proceedings of2nd ACM Conference on Computer and Communications Security*, Fairfax,1994, pp. 176-183.

[4] T. Hayashi, H. He, H. Satou, H. Ohta, S. Vaidya, "Accounting, Authentication and Authorization Issues in Managed IP Multicasting Services", Internet Draft, *draft-hayashi-maccnt-02.txt*, Feb. 2005

[5] T. Hardjono and B. Cain "Key Establishment for IGMP Authentication in IPECUMN," France, Oct. 2000, pp. 247-52.

[6] H. He, T. Hardjono, and B. Cain, "Simple Multicast Receiver Access Control," *Internet draft*, draft-irtf-gsec-smrac-00.txt, Nov. 2001.

[7] P. Judge and M. Ammar, "Gothic: A Group Access Control Architecture forMulticast and Anycast," *IEEE INFOCOM*, New York, June 2002, pp. 1547-56

[8] C. Castelluccia and G. Montenegro, "Securing Group Management in IPv6 withCryptographically Based Addresses," *Proc. 8th IEEE International Symposium on Computer andCommunication*, Turkey, July 2003, pp. 588-93.

[9] N. Ishikawa, N. Yamanouchi, O. Takahashi, "IGMP Extension for Authentication of IP Multicast," *Internet Draft*, draft-ishikawa-igmp-auth-01.txt, Aug. 1998.

[10] N. Yamanouchi, N. Ishikawa, Takahashi, "RADIUS Extension for Multicast Router Authentication," *Internet Draft*, draft-yamanouchi-radius-ext-00.txt, Mar. 1998.

[11] H. Ueno, H. Suzuki, N. Ishikawa, and O. Takahashi, "A Receiver Authenticationband Group Key Delivery Protocol for Secure Multicast," *IEICE Trans.* onvol. E88-B, no. 3, Mar. 2005, pp. 1139-1148.

[12] T. Hayashi, D. Andou, H. He, W. Tawbi, and T. Niki, "IGMP for user Authentication Protocol (IGAP)," *Internet Draft*, draft-hayashi-igap-00.txt, Oct. 2002.

[13] B. Coan, V. Kaul, S. Narain, W. Stephens, "HASM: Hierachical Application-Level Secure Multicast," *Internet Draft*, draft-coan-hasm-00.txt, Nov. 2001.

[14] R. M. Needham and M. D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Communications of the ACM*, vol.21, 1978, pp. 993-999.

[15] M. Baugher, R. Canetti, L. Dondeti, and F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture," *RFC 4046*, Apr. 2005.

[16] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no.11, 1981, pp. 770-772.

[17] M. Handley, C. Perkins, and E. Whelan, "Session Announcement Protocol," *RFC 2974*, Oct. 2000.

[18] M. Baugher, B. Weis, T. Hardjono, H. Harney, "The Group Domain of Interpretation," *RFC 3547*, July 2003.

[19] T. Hayashi, H. He, H. Satou, H. Ohta, S. Vaidya, "Issues Related to Receiver Access Control in the Current Multicast Protocols," *Internet Draft*, draft-ietf-mboned-rac-issues-00.txt, July 2005.

〈著者紹介〉

강 현 선 (Hyun-sun Kang)
2002년 2월: 단국대학교 전자계산학과 졸업
2004년 2월: 단국대학교 전자계산학 석사
2007년 2월: 단국대학교 전자계산학 박사
2007년 3월~현재: 단국대학교 인재개발원 강의전임 강사
<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