韓國 컴퓨터情報學會 論文誌
第11卷 第6號, 2006. 12

2006-11-6-1-5

키 관리시스템의 부하절감을 위한 향상된 키 분배 메커니즘과 보안프로토콜

전 정 훈*

An advanced key distribution mechanism and security protocol to reduce a load of the key management system

Jeon Jeong Hoon *

요 약

유비쿼터스 환경에서 각종 서비스와 장비의 발전은 멀티캐스트 사용자의 급증과 멀티캐스트 키의 다양한 해킹공격을 예상케 하고 있다. 이러한 멀티캐스트 사용자의 급증과 보안프로토콜의 적용은 중앙 키 관리시스템의 부하를 증가시켜 성능을 저하시킨다. 따라서 본 논문에서는 멀티캐스트 서비스의 효율성과 안정성을 극대화하기 위해 키 관리 메커니즘의 기능성을 향상시키고자 한다. 기존 키 관리 메커니즘의 비교와 시뮬레이션을 통해 문제를 분석하고, 문제해결을 위한 제안방법으로 소그룹화와 키 길이제어, 새로운 보안프로토콜의 적용을 통해 기존의 메커니즘의 기능성을 향상시킨 SMKD(Secure Multicast Key Distribution)메커니즘을 제안하고자 한다. 본 논문의 SMKD는 중앙 키 관리시스템의 키 분배와 암호화 수행에 따르는 부하를 경감하고, 효율적인 키 관리를 통해 시스템에 안정성을 보장한다.

Abstract

In an Ubiquitous Environment, the growth of various services and equipment is forecasted to increase both the multicast users and diverse hacking attacks of the multicast key. Rapid increasing of multicast users and application security protocols reduce the performance of the Central key management system. Accordingly, We propose to elevate the functionality of the key management mechanism for greater efficiency and stability of the multicast services, in this paper. The existing key management mechanism comparison and simulation will analyze these problems. We propose the advanced SMKD (Secure Multicast Key Distribution) mechanism application of the small group and key length control, new security protocol by methods to solve these problems. The SMKD Model in this paper will help reduce loading the key distribution and encryption execution of a central key management system, and this model can also ensure stability to a central key management system by efficient key management.

▸ Keyword : 키 분배시스템(Key Distribution System), 부하(Load), 메커니즘(Mechanism), 암호와 해쉬 알고리즘(Encryption and Hash Algorithm), 멀티캐스트 키(Multicast Key)

• 제1저자 : 전정훈
• 접수일 : 2006.11.01, 심사일 : 2006.12.12, 심사완료일 : 2006. 12.20
* 동덕여자대학 정보학부 컴퓨터학과 전임강사
※ 이 논문은 2005년도 동덕여자대학교 교내학술연구비 지원에 의해 수행된 것임.

Ⅰ. 서론

통신의 대부분은 유니캐스트(Unicast)를 사용하지만, 다중 사용자의 경우에는 멀티캐스트(Multicast) 통신이 유용하다. 유니캐스트 통신은 전송데이터의 높은 서비스 의존도와 대역폭 관리, 트래픽 지연, 중복작업수행 등의 문제로 인해 시스템 부하를 초래하기 때문이다. 반면 멀티캐스트 통신은 그룹단위의 관리체계로 메시지를 전송한 후, 멤버들에게 메시지의 복사본을 전송하므로 대역폭의 효율성을 높일 수 있다. 향후 멀티캐스트 통신은 유비쿼터스 네트워크 환경에서 인터넷방송이나 소프트웨어 분배 및 공유, 원격 화상회의, 온라인 게임, 센서네트워크 등의 멀티미디어 통신과 다수의 사용자 관리에 유용하게 사용될 것이다. 그러나 문제는 급격히 증가하는 서비스 및 단말장비의 보급으로 다중그룹을 관리해야하는 키 관리메커니즘과 관리시스템의 부하에 있다. 그룹멤버들의 빈번한 가입과 탈퇴는 네트워크의 불필요한 트래픽을 초래하고, 이에 따르는 키 관리와 보안문제는 키 관리시스템에 많은 부하를 야기 시켜, 시스템 효율성을 저하시키기 때문이다. 따라서 본 논문에서는 효율적인 메커니즘과 키 관리시스템의 부하를 경감시키고, 보안상의 취약점을 보완하며, 최적화할 수 있는, SMKD(Secure Multicast Key Distribution)를 제안하고자한다. 2장에서는 키 관리 메커니즘에 대한 관련연구와 3장은 SMKD에 대한 키 생성 및 분배 방법과 보안방법에 대한 기술, 4장은 SMKD의 기능성 및 보안성에 대한 비교분석과 효과, 5장은 제안하고 있는 SMKD에 대한 결론부분으로 구성되어 있다.

Ⅱ. 관련연구

본 절에서는 여러 키 관리 모델들을 알아보고 각 모델의 특징 및 장단점을 비교분석하였다. 그리고 멀티캐스트의 보안을 위한 영역구분과 관련 보안프로토콜에 대해 기술하였다.

2.1 키 분배 모델

(1) 키 분배센터(Key Distribution Center) 모델

키 분배센터 모델은 하나의 키 분배센터로부터 그룹 키를 부여받는다. 멤버의 수가 적은 그룹에서 구현하기 쉽고, 멤버관리가 용이하지만, 모든 멤버들이 하나의 키 분배센터에 의존하고 있으므로 멤버의 수가 증가하면, 시스템 지연과 트래픽 증가 현상이 발생한다. 그리고 동적인 측면에서 멀티캐스트 세션관리는 비효율적이며, 확장성이 떨어지는 문제점이 있다.

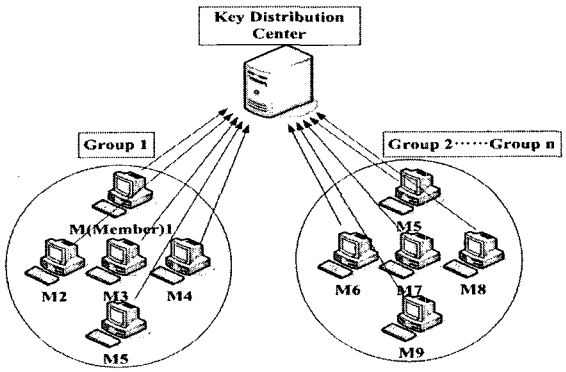


그림 1. 키 분배센터 모델
Fig 1. Key Distribution Center Model

그림1에서처럼 5명의 멤버로 구성된 2개의 그룹 키 관리는 키 분배센터 단독으로 처리해야만 한다. 결과적으로 키 분배센터 모델은 구현이 쉽고, 소규모에 효율적인 장점이 있으나, 그룹의 규모가 커지고 그룹 멤버의 변동이 많은 환경에서는 비효율적이다.

(2) GKMP(Group Key Management Protocol) 모델

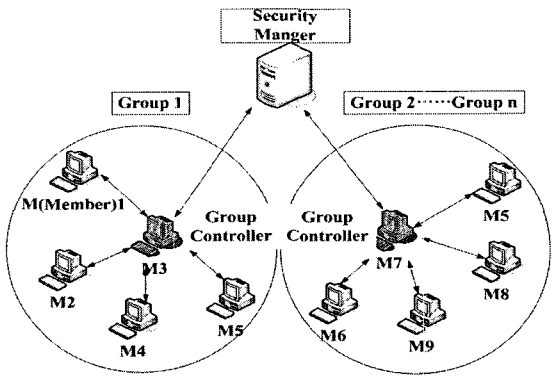


그림 2. GKMP 모델
Fig 2. GKMP Model

GKMP[1]모델은 키 분배센터모델의 중앙관리방식과는 달리, 그룹통제시스템(Group Controller System)을 두고, 그림2와 같이 보안관리자(Security Manager)로부터 키 관

리에 대한 권한을 위임받는다. 그룹통제시스템은 권한을 위임받아 멤버들에게 키를 분배하고, 그룹에 속한 멤버들만 관리한다. 따라서 키 분배센터 모델 보다 키 관리시스템의 부하를 경감시킬 수는 있지만, 그룹 내에서 그룹통제시스템의 선출방법이 고려되어야만 한다.[2] 또한 GKMP모델은 중앙 키 분배영역(Central Key Distribution Site)이 필요 없기 때문에 유리하지만, 그룹별 인증정보를 수집 및 관리를 위해서 보안 관리자를 두어야 하는 단점도 있다.

(3) Iolus 모델

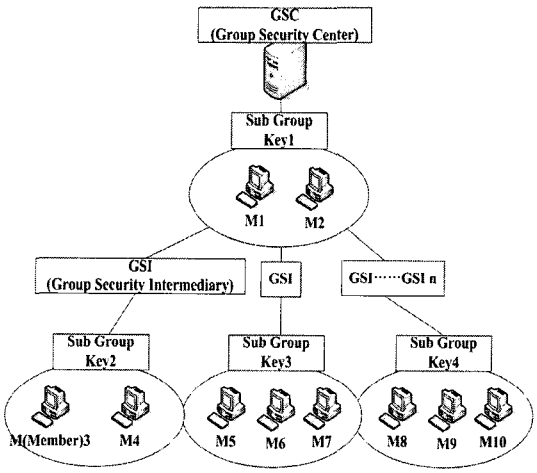


그림 3. Iolus 모델
Fig 3. Iolus Model

Iolus 모델은 그룹을 독립적인 소그룹으로 나누고, 소그룹 단위로 관리하는 구조이다.[3][16] 각 소그룹들 간, 그룹보안센터(Group Security Center)와 그룹보안중재자(Group Security Intermediary)가 존재하며, 소그룹 간에 전달되는 메시지의 경우 목적지의 소그룹 키로 암호화를 수행한다. 새로운 멤버가 가입하거나 탈퇴할 경우, 멤버가 속한 소그룹의 키만 변경하면 되기 때문에 키를 갱신하는데 효율적이다. 그리고 그룹 내 멤버의 가입 및 탈퇴의 경우, 키 변경 작업을 소그룹으로 최소화할 수 있다. 특정 그룹보안중재자가 서비스에 실패할 경우, 그룹보안중재자의 하부 소그룹과 멤버들만이 서비스에 실패함으로써 다른 그룹에 영향을 미치지 않는다. 그러나 그룹의 규모가 커질 경우, 여러 개의 그룹보안중재자를 두어야 함으로 구성이 복잡해지고, 그룹 간 발생하는 암·복호화에 따른 부하로 성능이 저하된다. 또한 그룹보안센터에 문제가 발생할 경우, 모든 하위그룹에 영향을 미치는 단점이 있다.

(4) Scalable Multicast Key Distribution 모델

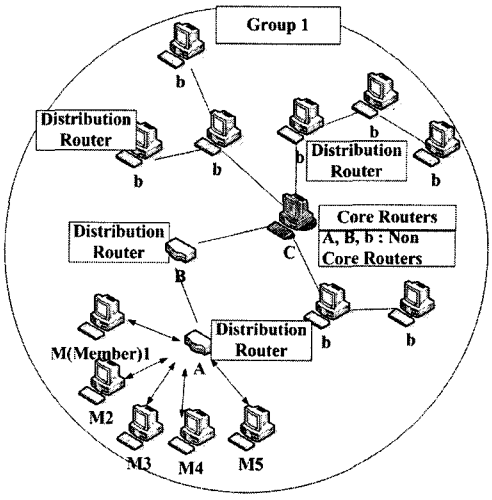


그림 4. Scalable 멀티캐스트 키 분배모델
Fig 4. Scalable Multicast Key Distribution 모델

Scalable Multicast Key Distribution[4] 모델은 멀티캐스트 라우팅 알고리즘 중, 하나인 코어기반트리(Core Based Tree)를 응용하고 있다.[4] 멤버들의 인증정보를 코어에 저장하고, 새로운 멤버가 그룹에 가입할 경우, 코어에서 새로운 멤버에 대한 인증이 이뤄진다. 그리고 멤버에 대한 인증결과를 전송할 때, 전송경로 상에 있는 라우터에 전달한다. 또한 동일 전송경로를 통한 멤버의 가입요청이 있을 경우, 코어까지 전송하지 않고, 분배라우터(Distribution Router)에서 가입처리를 수행한다. 그림4에서 A와 B는 새로운 그룹 키 분배센터(Group Key Distribution Center)가 된다. 이 모델은 키 분배센터의 역할을 분산하여, 멤버의 가입 및 탈퇴에 따른 부하는 경감되었으나, 코어기반트리를 기반으로 하고 있어, 다른 멀티캐스트 알고리즘의 적용이 어려운 단점이 있다.

(5) SDP(Session Description Protocol) 모델

SDP는 세션정보를 추가멤버에게 통보하도록 고안된 프로토콜이다.[5] SDP는 단지 세션정보를 기술하는데 사용되고, 기술된 정보[6]는 SAP(Session Announcement Protocol), SIP(Session Initiation Protocol)를 이용해 전달된다. SDP가 세션에 대한 정보를 기술하는데 사용된다면, SAP는 정보전달을 위해 사용되며, SIP[8]는 특정 멤버를 초대하기 위해 사용되는 프로토콜이다. 멤버는 특정 SAP 주소를 주기적으로 점검해야하고, 세션에 대한 정보를 수정

및 삭제해야한다. 또한 SAP[7] 패킷에는 세션정보(Session Description)외에, 선택적인 인증헤더를 포함하고, 세션정보를 암호화할 수 있다.[5][6]

(6) 계층적 트리 구조(Hierarchical Tree Architecture) 모델

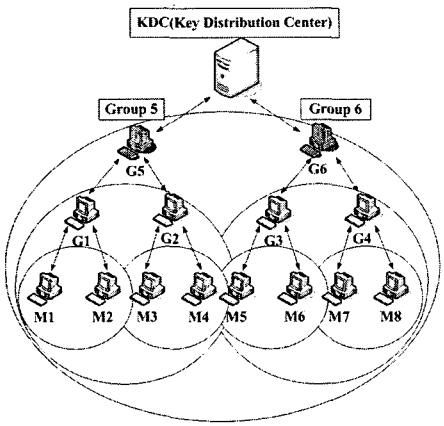


그림 5. 계층적 트리 구조 모델
Fig 5. Hierarchical Tree Structure Model

계층적 트리 구조[17]모델은 최상위 루트(Root)인 키 분배센터(KDC)에서 그룹 키를 관리하고, 그림5의 G1~G6은 2진 트리구조에서 소그룹 관리자의 역할을 대행하게 된다. 또한 멤버는 멀티캐스트 서비스를 받고자 하는 사용자 시스템으로서 leaf 노드가 되며, 자신으로부터 루트까지 모든 그룹관리시스템들의 키를 저장해야만 한다. 계층적 루트 구조는 키의 재분배가 발생할 경우, 다른 멤버에게 영향을 적게 미치기 때문에 적은 량의 메시지 트래픽만으로도 키의 재분배가 가능하고, 확장성도 뛰어나다.(그림5와 같이 M1~M4의 가입과 탈퇴는 M5~M8에 미치는 영향이 적다.) 그러나 멤버에서 루트까지 Key+1개 키가 요구되기 때문에 공격에 의한 키 유출 및 오류에 치명적이며, 그림5에서처럼 멤버는 자신의 키와 G1, G5, 키 분배센터까지의 모든 Key를 알고 있어야만 하는 단점이 있다.

2.2 키 보안

(1) 멀티캐스트 보안(Multicast Security)

안전한 멀티캐스트 통신을 위한 보안 영역은 표1[16]과 같이 핵심 문제영역(Core Problem Area)과 기반구조 문제영역(Infrastructure Problem Area), 혼합 애플리케이션영역(Complex Application Area)으로 구분할 수 있다.

표 1. 멀티캐스트 보안영역
Table 1. Multicast Security Area

구분 영역	내 용
Core Problem Area	- 멀티캐스트데이터의 기밀성, 무결성, 인증 - 멀티캐스트 그룹 키 관리 - 그룹 보안과 관련된 정책의 설정
Infrastructure Problem Area	- 멀티캐스트 라우팅 프로토콜 상의 보안 - 멀티캐스트 프로토콜의 신뢰성
Application Problem Area	- 그룹 키 생성과 분배 - 멤버들 간의 안전한 통신 - 그룹과 그룹 멤버의 인증(Authentication) - 멤버의 부인봉쇄(Non-repudiation) - 멤버의 익명성 - 공격에 대한 강건성(Robustness)

표1에서의 핵심보안영역과 애플리케이션영역은 유니캐스트와 멀티캐스트 통신방법에 따라 차이가 있다. 유니캐스트 통신은 두 사용자간 공유키를 이용해 기밀성과 인증성을 제공하는 반면, 멀티캐스트 통신은 그룹의 모든 멤버들이 암· 복호화와 인증에 필요한 키를 가져야만 하기 때문에 별도의 관리가 필요하다. 그리고 그룹 멤버의 가입과 탈퇴가 빈번할 경우, 보안정책에 따라 허가된 멤버로의 접근이 제한되어야만 한다. 멤버들 간, 키 보호를 위해 기밀성(Confidentiality)과 무결성(Integrity)이 제공되어야 하며,[9][10] 인가된 멤버들만이 그룹 키에 접근할 수 있도록 인증성(Authentication)도 함께 제공되어야 한다.

(2) 멀티캐스트 키 보안 프로토콜(Multicast Key Security Protocol)

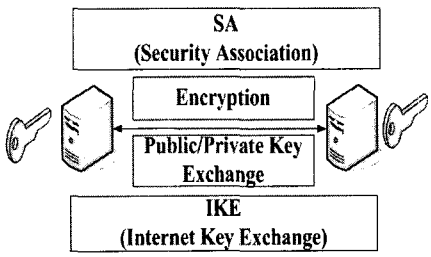


그림 6. IPsec 프로토콜
Fig 6. IPsec Protocol

중앙관리시스템과 소그룹관리시스템, 멤버들 간의 보안성 제공을 위해 보안프로토콜을 적용하고 있다. 대표적인 보안 프로토콜로 IPsec을 예로 들 수 있는데, UDP 기반의 IPS

ec은 그림 6에서처럼 보안성 부분에 뛰어나지만 보안협상 및 연결설정관리, 정책관리로 인해 관리시스템에 많은 부하가 발생한다. 그리고 다수 멤버관리의 경우, 추가 세션을 위한 정책관리가 필요하며, 시스템 부하는 더욱 증가하게 된다. 그러나 Stunnel[18]은 임의의 특정 포트를 통해 동작하며, 전송데이터의 암호화와 메시지 인증까지 함께 수행할 수 있는 프로토콜이다.

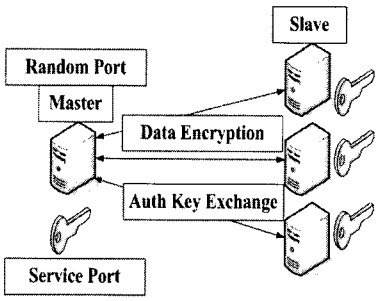


그림 7. Stunnel 프로토콜
Fig 7. Stunnel Protocol

그림7에서 Stunnel은 TCP기반의 SSL프로토콜에 의한 메시지 가공방식을 사용하며, 기밀성, 무결성, 인증성 모두를 제공한다. 또한 빠른 연결협상을 위해 초기 협상과정을 생략하였으며, IPSec의 설정파일방식과는 달리, 일괄 설정이 가능하기 때문에 키 관리시스템의 컴퓨팅자원을 보다 효율적으로 사용할 수 있다.

2.3 키 관리 메커니즘의 비교

앞서 관련연구에서 기술했던 키 관리메커니즘들과 제안하는 SMKD모델의 장단점을 표2로 정리하였고, 각 모델들은 관리시스템의 기능성과 키 분배의 효율성, 보안성에 대해 분석하였다.

표 2. 키 관리메커니즘 비교
Table 2. Key Management Mechanism Comparison

	장 점	단 점
키분배 센터모델	• 소그룹에 유리.	• 확장성이 떨어지고, 대규모에 비효율적
GKMP모델	• 키 관리시스템의 부하 경감.	• 인증정보 수집 및 관리하는 보안관리자를 두어야 함. • 그룹통제시스템을 선출해야 함.
Iolus모델	• 키 갱신 효율적. • 다른 그룹에 영향을 미치지 않음.	• 암호화 부하로 구성이 복잡. • 루트가 실패할 경우 모두 하위그룹 에러 발생.
Scalable Multicast Key Distribution 모델	• 키 분배센터역할을 분산하여 키 관리시스템의 부하경감. • 가입 및 탈퇴에 따른 부하 경감.	• 다른 멀티캐스트 알고리즘의 적용이 어려움.
계층적 트리 구조 모델	• 다른멤버에 영향이 적음 • 확장성이 뛰어남.	• 많은 키가 요구됨. • 공격에 의한 유출 및 오류에 치명적 • 모든 키를 알고 있어야 함.
SMKD모델	• 소그룹화로 키 관리시스템의 오버헤드경감. • 키 갱신 효율적. • 다른 그룹과 멤버에 영향을 미치지 않음. • 키에 보안성제공. • 키 길이의 제한으로 시스템부하절감. • 확장성이 용이함.	• SSL기반의 보안프로토콜 업그레이드. • 다른 멀티캐스트 알고리즘의 적용이 어려움. • 소그룹관리자 선출에 대한 부하.

III. 제안하는 SMKD(Secure Multicast Key Distribution)

제안모델은 기존의 키 관리메커니즘의 장단점을 보완한 것으로 중앙 키 관리시스템의 부하를 경감시키고, 급증하는 멤버를 효율적으로 관리할 수 있도록 한다. SMKD는 소그룹으로 운영이 되며, GMEK(Group Message Encryption Key)와 SG_MEK(Small Group Message Encryption Key) 두 가지 키를 암호화와 인증에 사용하고, 소그룹과 중앙관리시스템간의 보안을 위해 Stunnel을 사용함으로써, 암호화수행에 따른 시스템 부하 경감과 소그룹화를 통한 키 분배센터 모델의 확장성문제를 개선한다. 그리고 GKMP모델의 보안관리자(Security Manager)를 두어야 하는 문제점과 Iolus 모델의 암· 복호화의 부하 문제를 해결하고, GMEK와 SG_MEK, SG_Key를 사용하여 계층적 구조모델의 키 공격과 유출 및 오류에 대해 대응한다.

3.1 중앙관리시스템의 키(GMEK, SG_MEK) 관리

SMKD모델은 소그룹의 중앙관리를 위해 두 가지의 키가 사용된다. 하나는 키 암호화에 사용되는 GMEK와 다른 하나는 소그룹 인증에 사용되는 SG_MEK이다. 각각의 키는 소그룹의 멤버로부터 생성되어 최종적으로 중앙관리시스템

에 보고되어지고, 중앙관리시스템은 소그룹으로 키 관리 역할을 위임한다.

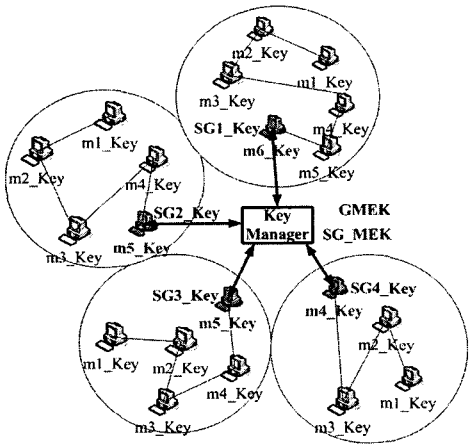


그림 8. SMKD 흐름도
Fig 8. StMKD Flow

그림8은 SMKD모델의 전체 흐름을 나타낸 것으로, 소그룹 내에서 선발된 멤버를 키 관리시스템으로 지정하고, 멤버 키(m_Key)로 생성된 소그룹 키(SG_Key)를 중앙관리시스템에 보고한다. 그리고 소그룹들은 중앙관리시스템으로부터 통제를 받으며, 보안성과 인증성에 적용할 키를 분배받는다. 중앙 키 관리방식은 키의 생성을 제한할 뿐만 아니라, 역추적 조사를 통해 문제해결을 수행할 수 있다. 따라서 제안하는 SMKD모델은 키 분배센터의 중앙 키 관리방식과 Iolus의 키 재생성 및 소그룹관리 방식을 적용하고 있으며, 기존 키 분배메커니즘의 문제점을 보완한다.

3.2 소그룹관리시스템의 키 관리

(1) 멤버 키(m_Key)

수식의(1)~(5) 표현을 위해서 다음과 같은 기호를 사용한다.

H( ) : Hash 함수
|| : Catenation
n : 멤버번호
PID: 프로세스 식별번호
Time : 키 생성 시 시간
Process Number : 사용자 식별번호(UID)
m_Key : 멤버를 식별하기 위한 멤버 키
SG_Key : 멤버 키들의 결합으로 이뤄지며, 소그룹을 식별하기 위한 키
GMEK : 소그룹들 간에 사용되는 암호 키
SG_MEK : 소그룹의 인증 및 암호에 사용

멤버 키는 멤버의 식별과 소그룹의 인증 키 생성에 사용되며, 키의 생성은 PID와 Time, Process Number등 시스템 정보를 이용한다. 생성된 멤버 키는 마지막 참가한 멤버에게 보내지며, 마지막 참가한 멤버가 소그룹 관리자의 권한을 위임받아 각 멤버들이 생성한 키들을 조합한다. 멤버 키의 생성은 다음 수식(1)과 같다.

$$mn\_Key = H(PID||time||Process\ Number) \quad \cdots\cdots (1)$$

(2) 소그룹 키(SG_Key)

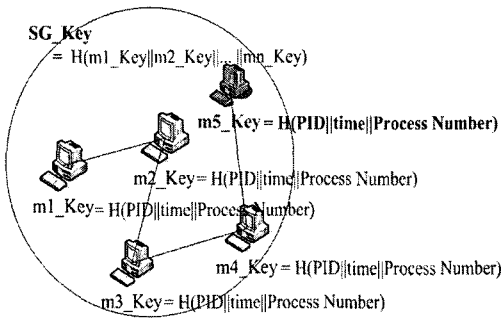


그림 9. m_Key, SG_KEY생성
Fig 9. m_key, SG_KEY Creation

소그룹 키는 소그룹 내에서 멤버들 간의 인증키와 중앙관리시스템의 소그룹 식별에 사용된다. 소그룹의 각 멤버들은 수식(1)을 통해 m_Key를 생성하고, 소그룹 관리시스템은 수식(2)을 통해 SG_Key를 생성한다. 소그룹 내에 관리시스템 선정은 마지막 참가한 멤버로 정하며, SG_Key는 소그룹의 관리시스템으로 선정된 마지막 참가자에 의해 중앙 관리시스템으로 업데이트된다. 그림9는 m_Key와 SG_Key의 생성과정을 나타내고 있다. 멤버들은 자신의 m_Key를 소그룹의 마지막 참가한 멤버에게 전달하고, 마지막 멤버(m5_Key)는 관리시스템의 역할을 위임받아, SG_Key를 생성하게 된다. 이때 SG_Key는 해쉬(Hash)알고리즘으로 정해진 키 길이로 생성되며, 모든 멤버들에게 통보된다. 생성에 필요한 인자 값으로 "프로세스식별자와 생성시간, 프로세스번호, 멤버들의 m_Key"를 조합하여 공격자가 유추할 수 없도록 한다. 또한 SG_Key는 소그룹 내에서 각 멤버간의 암호화키로도 사용되며, 키 생성은 수식(2)과 같다.

$$SG\_Key = H(m1\_Key||m2\_Key||...||mn\_Key) \quad \cdots\cdots (2)$$

(3) 새로운 멤버의 가입(Join)

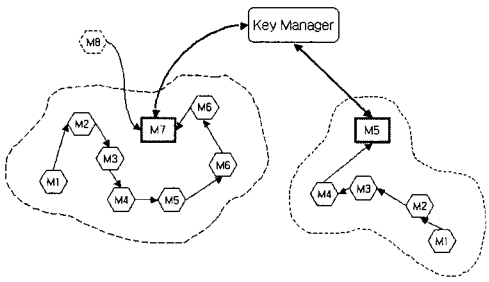


그림 10. SMKD 멤버 가입
Fig 10. SMKD Member Join

새로운 멤버가 가입을 요청하면, 소그룹관리시스템은 저장하고 있던 멤버들의 m_Key를 새로운 멤버들에게 전송한다. 소그룹관리시스템은 새로운 멤버로부터 전달받은 m_Key와 기존 SG_Key를 연산하고, 새로운 SG_Key를 생성하여 멤버들에게 전송한다. 그림10은 새로운 멤버의 가입을 나타낸 것이다. M8은 M7이 갖고 있던 SG_Key를 전달받아, 자신의 m_Key와 연산하고, 새로운 SG_Key를 생성함으로써 키 생성시간을 단축시킬 수 있다. 다음 수식(3)은 갱신 이전에 생성한 SG_Key와 새로운 멤버구성에 따른 m_Key를 해쉬하여 새로운 SG_Key를 생성한다. 새로운 소그룹관리시스템인, M8은 중앙관리시스템으로 갱신된 SG_Key를 전송한다.

갱신된 SG_Key=H(초기 SG_Key||m_Key) ····· (3)

(4) 기존 멤버의 탈퇴

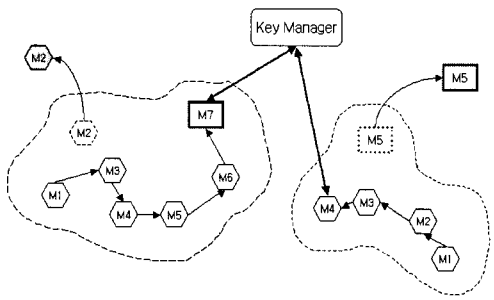


그림 11. SMKD 멤버 탈퇴
Fig 11. SMKD Member Secession

그림11은 멤버의 탈퇴과정을 나타낸 것으로, 두 가지 경우를 예로 들 수 있다. 첫 번째로 그림11의 왼쪽 그룹처럼 일반 멤버인 M2가 탈퇴할 경우, 소그룹관리시스템인 M7에 탈퇴를 통보하고, M7은 탈퇴멤버의 m_Key를 제외한 나머지 멤버의 m_Key로 새로운 SG_Key를 생성한다. 두 번째로 소그룹관리시스템인 M5가 탈퇴할 경우, 멤버 중 소그룹관리시스템에 마지막으로 m_Key를 전달한 M4가 소그룹관리시스템으로서의 역할을 위임받게 된다. 소그룹관리시스템이었던 M5는 새로운 소그룹관리시스템인 M4에게 탈퇴를 통보하고, M4는 새로운 SG_Key를 중앙관리시스템에 전송한다.

3.3 중앙관리시스템의 인증키

SG_MEK는 소그룹의 인증 및 암호에 사용되며, 중앙관리시스템으로부터 분배받는다. 멤버들은 수식(1)에 의해 m_Key를 생성하고, 소그룹관리시스템은 수식(2)에 의해 SG_Key를 생성한다. 생성된 SG_Key는 중앙관리시스템으로 보고되며, 중앙관리시스템은 수식(4)에 의해 SG_MEK를 생성하여 소그룹들에게 분배한다. 소그룹 자신의 SG_Key를 제외하고, 생성한 SG_MEK는 각 소그룹관리시스템에게 전송되며, 인증키로 사용된다.

SG1_MEK=H(SG2_Key||SG3_Key||SG4_Key) ····· (4)
SG2_MEK=H(SG1_Key||SG3_Key||SG4_Key)
SG3_MEK=H(SG1_Key||SG2_Key||SG4_Key)

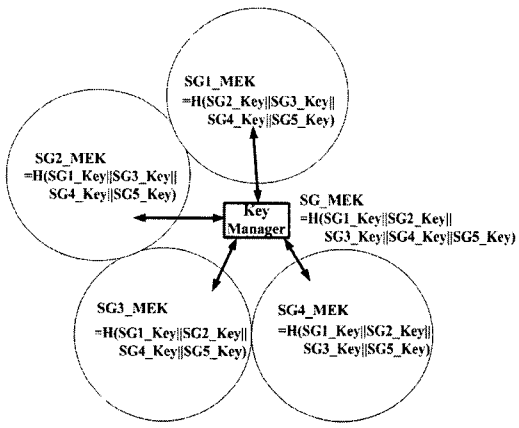


그림 12. SMKD 멤버 탈퇴
Fig 12. SMKD Member Secession

그림12는 중앙관리시스템으로부터 소그룹들이 인증을 위한 SG_MEK를 분배받고 있다.

3.4 중앙관리시스템의 키(GMEK) 생성

SMKD 모델은 중앙관리를 위해 두 가지 키를 생성한다. 각 소그룹들 간에 사용되는 암호 키인 GMEK와 인증키인 SG_MEK이다. GMEK는 모든 SG_Key로 생성된 SG_MEK를 말하며, 소그룹의 SG_MEK는 자신의 SG_Key를 제외한 SG_MEK로써 GMEK와는 서로 다르다. GMEK의 생성은 다음 수식(5)과 같다.

$$GMEK=H(SG1\_Key||SG2\_Key||...SGn\_Key) \cdots\cdots\cdots (5)$$

기존 키 분배모델들은 멤버의 수가 증가할수록 길이가 길어지는 단점을 갖고 있기 때문에 긴키를 사용하는 것은 키 관리문제와 메시지 암·복호화에 따른 부하, 대역폭 낭비와 같은 문제들을 초래하게 된다. 따라서 키 길이의 간략화가 필요하며, 소그룹관리시스템들 간의 독립성이 유지되어야한다.

3.5 그룹 간 통신보안

MSEC(Multicast Security Working Group)의 멀티캐스트 키 전송은 IPSec[12],[13]을 사용하여 접근통제와 무결성, 기밀성을 보장하는 그룹 간 보안통신방법을 제안하고 있다. 그러나 다중 접속과 빈번한 멤버의 가입과 탈퇴는 키 관리시스템에 많은 부하를 발생시킨다. 그리고 멤버의 수가 증가할 경우, 키 암호화와 해쉬 및 인증으로 많은 지연이 발생하는 단점이 있다. 따라서 가입과 탈퇴로 인한, 시스템 부하를 절감하고 그룹 간 교환되는 키의 안전을 위해 또 다른 보안프로토콜[20],[21]인 Stunnel을 적용한다. SSL 기반의 Stunnel은 임의의 지정 포트를 통해 서비스하고, 데이터에 대한 인증을 수행한다. 그리고 SSL의 초기연결(Initializing)과정을 생략함으로써, 연결협상과정에서 발생하는 지연과 재협상에 따른 지연, 지속적인 연결 관리 문제 등을 해결할 수 있으며, 인가되지 않은 사용자로부터 데이터를 보호하고, 안전성 확보를 목적으로 하고 있다. 결과적으로 멤버의 빈번한 가입과 탈퇴로 발생하는 많은 트래픽을 고려해본다면, 보안성이 강력한 IPSec에 비해 관리시스템의 기능성과 관리측면에서 Stunnel이 보다 효율적이다.

IV. SMKD 모델의 분석 및 효과

본 절에서는 SMKD모델의 성능분석을 위해 키 길이 최소화와 소그룹별 키 관리운용, 시스템 암호화에 따른 부하를 경감에 대한 기존 키 관리모델과 비교분석 한다.

실험을 위한 시나리오로는 해쉬 알고리즘의 처리속도를 측정하여, 해쉬되는 메시지의 크기에 따른 시스템 성능을 측정한다. 단 메시지의 크기는 1Mbyte크기의 단위 메시지 사이즈로 측정한다. 또 다른 시나리오는 해쉬 알고리즘의 키 길이 변화에 따른 시스템 성능을 측정함으로써 키 길이가 키 분배에 미치는 부하를 측정한다. 또한 IPsec과 SSL 프로토콜의 데이터 전송속도의 효율성을 측정한다.

실험을 위한 환경으로 운영체제는 리눅스 레드햇 8.0과 시스템 사양으로는 Pentium 4, 2.4 GHz, 512 RAM, HASH알고리즘(SHA1, SHA256, SHA512), IPsec, SSL 프로토콜, IPerf 툴이 설치된 2대의 시스템과 노트북2대를 사용한다.

4.1 키 길이 최소에 따른 키 관리 효과

SMKD모델은 기존의 키 생성방식과 달리 멤버 키로부터 소그룹 키를 생성하고, GMEK와 SG_MEK를 생성한다. 초기 생성된 GMEK와 SG_MEK키는 SMKD의 키 관리자가 보관하며, 해쉬 값으로 소그룹관리자들에게 부여한다. 그리고 생성된 키는 해쉬 알고리즘에 의해 고정길이를 유지하게 된다. 이러한 이유는 멤버의 급증으로 키 생성이 증가할 경우, 키 분배시스템에 부하를 초래하기 때문이다.

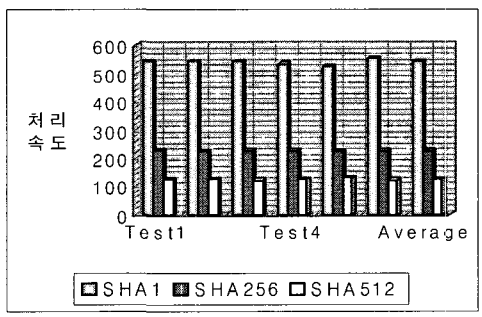


그림 13. Hash 알고리즘 Speed(160, 256, 512bit)
Fig 13. Hash Algorithm Speed(160, 256, 512bit)

그림13은 키 길이가 시스템에 미치는 연관성을 알아보기 위해 1Mbyte 크기의 메시지를 SHA 알고리즘으로 키 길이를 160(SHA1), 256(SHA256), 512(SHA512)bit로 변화하여 생성한 시뮬레이션 결과이다. 그림13을 통해 키 길이에 따라 처리속도에 차이가 있음을 알 수 있으며, "메시지 길이(Message Length) / 시간 = 키 생성속도" 수식을 통해 키 길이는 연산속도와 반비례하고, 생성시간이 증가함을 알 수 있다. 결과적으로 키 생성시간의 증가는 시스템 부하와 비례하고, 시스템 성능과 반비례하며, 생성 시마다

늘어나는 키의 길이가 시스템 부하를 증가시키고 있음을 알 수 있다. 따라서 키 길이의 최소화와 해쉬 키의 사용은 멤버의 증감에 따른 키 관리 부하를 경감시키고, 중앙관리시스템의 부하를 경감시키는 효과가 있다. 중앙 집중관리방식을 응용하고 있는 SMKD모델은 키 길이에 따른 문제를 보완함으로써 중앙관리시스템의 기능성을 향상시킬 수 있다. 다음 그림14는 그림13의 처리속도를 시간으로 환산하여 메시지 해쉬에 소요되는 처리시간을 나타낸 것이다.

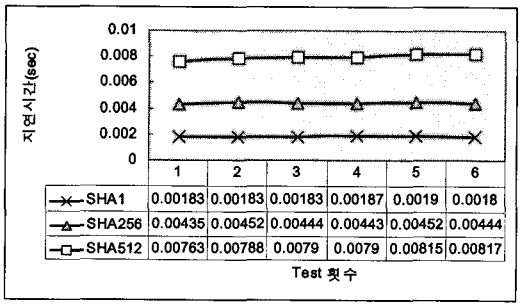


그림 14 지연시간비교
Fig 14. Latency Time Comparison

여기서 키 길이의 증가에 따라 처리시간이 증가하고, 처리시간은 시스템부하와 비례하기 때문에 키 분배시스템 측면에서 그림14는 해쉬 처리시간이 아닌, 시스템의 지연시간으로 나타내었다.

4.2 소 그룹화에 따른 키 관리 효과

소 그룹화는 키 생성과 분배 및 키 보안으로 인한 부하를 분산관리 할 수 있다. 그리고 중앙관리시스템의 키 분배 횟수를 줄이고, 멤버의 가입과 탈퇴를 효율적으로 관리할 수 있다. 표3은 키 분배센터, Iolus[16], 계층적 트리 구조[17], SMKD의 중앙관리시스템에서 분배 및 관리해야할 키를 수식으로 표현하고, 최소한의 생성 및 분배를 해야 하는지 비교할 수 있도록 하였다. 표3의 중앙관리시스템의 키 분배횟수 비교에서는 다음과 같은 기호를 사용한다.

n : 멀티캐스트 멤버의 수,
g : 서브그룹의 수,
L : 계층적 트리 구조의 각 계층(Depth)
E : 암호화로 인한 지연시간으로 모든 메커니즘에 동일 값이라 전제
2L : 2진 트리의 멤버 수로 중앙관리시스템의 키 교환 횟수,
n/g : 소그룹 당 멤버의 수
조건 : 자식노드의 최상의 경우(Best Case)를 가정하기위해 2진 트리를 전제

표 3. 중앙관리시스템의 키 분배 횟수 비교
Table 3. Key Management Mechanism's comparison the number of Key Distribution time Comparison

	키 분배센터	Iolus	Tree	SMKD
System Load	n	2×g	(n/g)×2L	g
Session Manage	n	2×g	(n/g)×2L	g
Member Manage	n	2×g	(n/g)×2L	g
Re-keying	n	2×g	(n/g)×2L	g
Encryption Load	E×n	E×(2×g)	E×(n/g)×2L	E×g
Key Recovery	no	no	no	yes
Manage. System	1	1	1	g+1

표3은 네트워크 관점에서의 키 분배에 따른 각 메커니즘의 부하를 알아보기 위해 키의 전달과정에 거쳐 가는 모든 시스템을 노드(Node)로 간주하고, 노드간의 필요한 키 교환 횟수를 비교한다. 그리고 각 모델별 수식은 다음과 같다. 키 분배센터 모델은 중앙에서 모든 관리를 수행하기 때문에 멀티캐스트 멤버의 수(n)가 분배 및 관리되는 키의 수가 된다. 같은 중앙 집중방식인 Iolus는 그룹보안센터에 소그룹 당 그룹보안 중재자가 존재하고 소그룹 키와 암호화키를 관리해야 하기 때문에 중앙의 그룹보안센터에서 관리되는 키의 수는 "2×g"가 된다. 계층적 구조는 2진 트리를 전제조건으로 하여, 모든 멤버의 수는 2L이 되고 소그룹 당 멤버의 수가 중앙관리 시스템에서 관리되어야 할 키의 수가 된다. 마지막으로 SMKD의 중앙관리시스템은 소그룹 관리시스템만을 관리 하기 때문에 그룹의 수가 관리해야할 키의 수가 되며, 아래 그림15는 표3의 암·복호화수식에 멤버의 수와 그룹 수, 계층(Depth)을 동일 조건으로 전제하고, 각 모델별로 총 키 분배횟수를 6차례 테스트를 하였다.

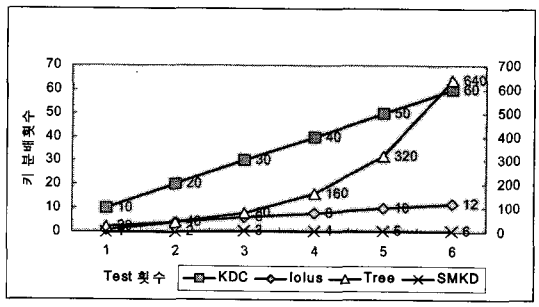


그림 15. 중앙관리시스템의 키 분배 횟수 비교
Fig 15. Central Management System's comparison the number of Key Distribution time(160, 256, 512bit)

그림15의 결과를 통해 SMKD가 다른 키 교환메커니즘에 비해 상대적으로 적은 키 교환이 이뤄짐을 알 수 있고, 중앙관리시스템의 키 교환과 관리에 따른 부하를 소그룹화로 분산 관리하는 것이 효율적임을 보여준다. 따라서 키 분배센터모델과 계층적 구조 모델, GKMP 모델, Iolus 모델의 그룹화에 따르는 키 교환문제를 보완하여 교환횟수를 줄이고, 소그룹단위의 관리를 수행함으로써 중앙 관리시스템을 효율적으로 운용할 수 있다. 그림15에서 좌우 스케일이 다르게 표현된 것은 편차가 큰 수치 결과를 하나의 그래프로 나타내기 위해서이다.

4.3 보안프로토콜에 따른 부하 절감효과

키에 대한 보안성제공을 위해 SMKD모델은 보안프로토콜을 적용하고 있다. MSEC(Multicast Security)[14][15]에서는 멀티캐스트 보안을 위해 UDP기반의 IPsec[12][13]적용을 제안하고 있지만, SMKD모델은 SSL기반의 Stunnel을 사용한다. 그러나 보안성 제공은 보안프로토콜에 따라 중앙 키 관리시스템에 많은 부하를 초래하여 시스템의 효율성을 저하시킨다. 따라서 시스템 부하와 보안성을 비교하기 위해 MSEC의 IPsec과 SMKD모델의 Stunnel에 대해 오버헤드 크기(Overhead Size)와 세션관리에 따르는 지연을 비교분석한다. 표4, 표5, 표6, 표7, 표8의 결과는 2대의 리눅스 시스템에 IPsec과 Stunnel을 각각 설치하고, Iperf로 시뮬레이션 한 결과이다. 시험환경은 다음과 같다. 운영체제는 Red Hat 8(Kernel 2.4.20-20.8), Pentium 4, 2.4 GHz, RAM 512MB, NIC 100Mbps and 1000Mbps, 100Mbps and 1000Mbps Switching Hub, Super FreeS/WAN .99-8, Stunnel3.26, Ethereal 0.9.16, Iperf1.7.0으로 시스템의 CPU의 소모율과 처리속도를 측정하였다.

(1) 오버헤드 사이즈(Overhead Size)

표4[19]는 IPsec과 Stunnel의 오버헤드를 비교한 것이다. 오버헤드 크기는 중앙시스템에서 생성된 키 길이 외에 추가되기 때문에 오버헤드 크기가 클수록 시스템에 부담을 가중시키게 된다. 따라서 오버헤드의 변화가 큰 IPsec과 비교해 볼 때, 메시지의 길이가 시스템 부하와 비례하는 4.1절의 결과에 따라 오버헤드 크기를 최소화하여야 한다. 반면, Stunnel은 비교적 작은 오버헤드 크기를 제공하고 있어, 다중 그룹관리를 고려해 볼 때, Stunnel의 사용이 키 관리시스템에 효율적이다. 표3에 의해 키 관리메커니즘의 키 분배횟수와 오버헤드 크기를 함께 고려해 본다면, Stunnel을 사용하는 SMKD모델은 다른 모델에 비해 시스템 부하를 감소시킬 수 있다.

표 4. 오버헤드 크기(Overhead Size)
Table 4. Overhead Size

프로토콜(Protocol)	모드(Mode)	바이트(Byte)
IPsec Tunnel 모드	ESP	32
	ESP + AH	44
IPsec Transport 모드	ESP	36
	ESP + AH	48
SSL(Stunnel)	HMAC-MD5	21
	HMAC-SHA-1	25

(2) 세션관리의 효율성

이 절에서는 시스템부하와 세션설립시간의 상관관계를 알아보기 위해, 연결(Connection)이전 상태까지의 세션설립시간과 키 재생성에 소요되는 시간을 비교해 보고자 한다. 시간 측정을 위한 테스트 환경으로 IPsec과 SSL이 설치된 2대의 리눅스 서버를 상호 연결하고, 1Mbyte 크기의 메시지를 전송하여 IPerf로 핸드세이크 시간을 측정하였다.

표 5. IPsec 핸드세이크(Handshake) 시간
Table 5. IPsec 핸Handshake Time

Mode	세션설립시간
Main Mode(PSK)	97msec
Aggressive Mode(PSK)	56msec
Main Mode(RSA)	170msec

표 6. SSL(Stunnel) 핸드세이크(Handshake) 시간
Table 6. SSL Handshake Time

Mode	세션설립시간
서버인증	41.7msec
클라이언트인증	74.8msec
서버인증(Diffie-Hellman)	66.1msec
클라이언트인증(Diffie-Hellman)	118.6msec

표5[19]와 표6[19]에서 두 가지 보안프로토콜에 대한 세션설립시간(Session Establishing Time)을 비교하고 있다. 세션설립시간은 연결(Connection) 이전 단계로 시스템 자원을 소모하는 요인이 되며, 설립시간이 길어질수록 성능과는 반비례하고, 시스템 부하와는 비례한다. 그러나 SSL기반의 Stunnel은 연결협상과정이 생략되어 표5의 결과보다도 비교적 적은시간이 소요되어, 키 전송에 따르는 관리시스템의 부담을 절감할 수 있다. 표3에서 세션관리(Session Manage)에 관한 키 분배횟수를 비교해볼 때, SMKD모델에서 제안하는 SSL이 세션관리에 보다 효율적이다.

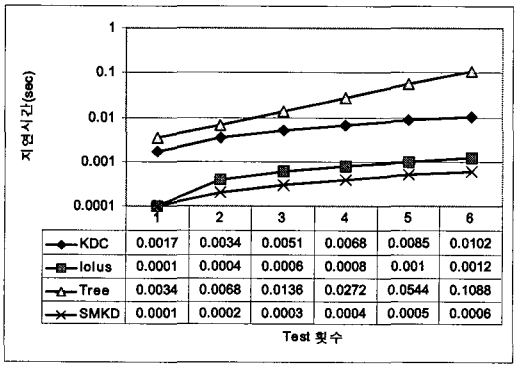


그림 16 IPsec Main Mode(RSA) 170msec
Fig 16. IPsec Main Mode(RSA) 170msec

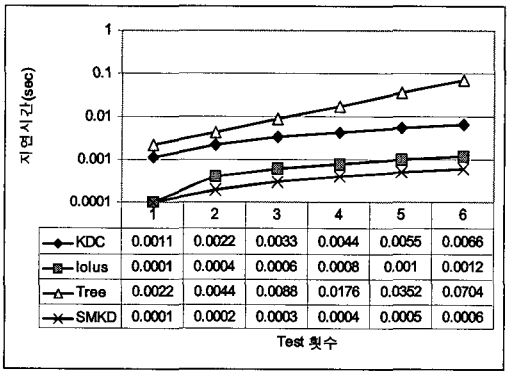


그림 17 클라이언트인증(Diffie-Hellman) 118.6msec
Fig 17. Client Authentication(Diffie-Hellman) 118.6msec

그림16과 17은 IPsec과 SSL(Stunnel)의 핸드세이크 시간을 표3의 수식을 통해 키 교환모델 별로 지연시간을 계산한 것으로, 그림15에서 표현하고 있는 키 분배횟수에 노드(node)당 소요되는 핸드세이크 시간을 지연시간으로 표현하였다. IPsec, SSL 모두 최악의 상황(Worst Case)을 고려하여, 핸드세이크 시간을 각각 170msec와 118.6msec로 계산하여, 각 모델별 차이를 쉽게 알아보기 위해 로그(Log)단위의 그래프로 표현하였다.

표 7. IPsec 키 재생성(Re-keying)시간
Table 7. IPsec Re-keying Time

Mode	시간
Main Mode(PSK)	26msec
Aggressive Mode(PSK)	
Main Mode(RSA)	

표 8. SSL(Stunnel) 키 재생성(Re-keying)시간
Table 8. SSL(Stunnel) Re-keying Time

Mode	시간
서버 인증	1.3msec
클라이언트 인증	
서버 인증(Diffie-Hellman)	
클라이언트 인증(Diffie-Hellman)	

IPerf로 측정한 키 재생성시간을 표7[19]과 표8[19]에서 비교하고 있다. 키 재생성 시, 키 생성의 지연과 연결실패가 반복된다면, 지속적인 시스템자원의 사용으로 이어져 관리시스템에 많은 부담을 주게 된다.

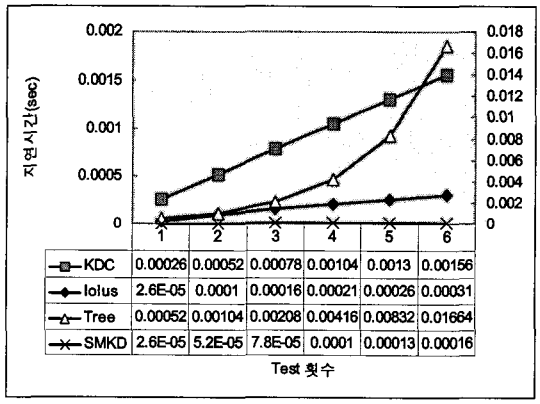


그림 18 IPsec 키 재생성(Re-keying)시간(26msec)
Fig 18. IPsec Key Re-Keying Time(26msec)

그림 18과 19에서 키 재생성 시간을 비교해 볼 때, IPsec은 연결설정관리로 세션마다 지연이 발생하고, 인증처리의 반복으로 시스템 부하를 가중시킨다.

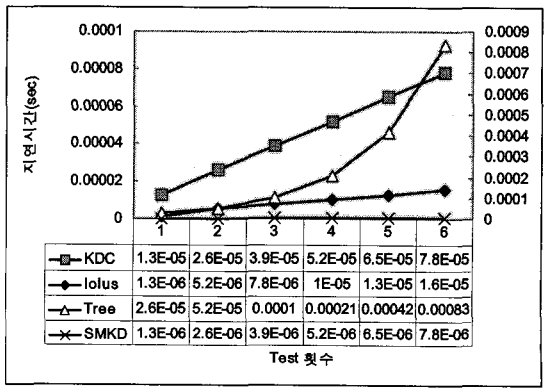


그림 19 SSL 키 재생성(Re-keying)시간(1.3msec)
Fig 19. SSL Key Re-Keying Time(1.3msec)

그러나 Stunnel은 키 관리시스템에서 멀티유저로 일괄 전송됨에 따라 추가적인 관리가 불필요하며, 멤버로 인해 발생하는 키 관리시스템의 비효율적인 자원관리를 개선할 수 있어IPsec 보다 효율적이다. 결과적으로 키 유출 및 위변조, 오류에 대해 보안성을 강화하고, 암·복호화에 따른 중앙관리시스템의 부하를 절감시킬 수 있다. 표3의 수식에 의해 키 재생성에 소요되는 지연시간을 산출하여 그림18과 19와 같이 표현하였다.

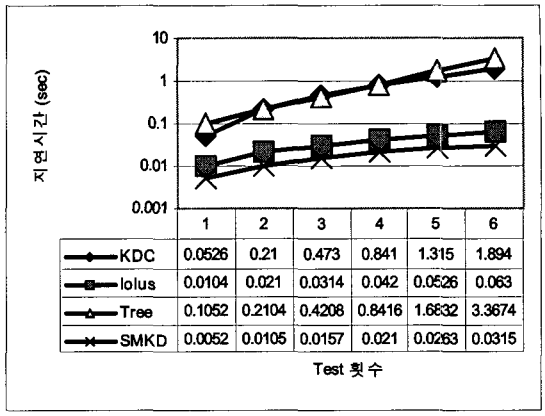
4.4 키 보안 효과

(1) 그룹과 멤버간의 인증성 제공

SMKD모델은 SG_MEK라는 인증키를 이용해 소그룹관리시스템의 인증을 수행한다. 소그룹들은 SG_Key를 생성하고, 중앙관리시스템은 각 소그룹들의 SG_Key를 조합하여, SG_MEK를 생성한다. 인증키의 생성은 3.3절의 수식(4)으로 생성되며, 중앙시스템으로 집중되는 멤버의 인증부하를 소그룹으로 분산한다. 그리고 소그룹 간 인증에 SG_MEK를 사용하며, 각 소그룹들은 SG_Key를 이용해 멤버 인증을 수행한다. 따라서 허가받지 않은 멤버나 소그룹관리시스템을 위장한 공격에 대응한다. 실험을 위해 Pentium 4, 2.4 GHz, 메모리 512Mbyte 환경에서 1Mbyte의 데이터를 3DES알고리즘으로 9차례 암·복호화 과정을 반복하여 평균 암호화 시간을 측정하였고, IPsec과 SSL을 통해 동일 알고리즘으로 데이터 암호화에 소요되는 시간을 I Perf로 측정하여 두 가지 경우의 평균값을 표3의 암·복호화에 수행되는 키 교환 횟수 수식으로 그림20의 결과를 산출하였다.

(2) 그룹과 멤버간의 기밀성제공

그림20의 시뮬레이션은 표3의 암호화 부하(Encryption Load) 수식에 3DES 128bit 암호화 알고리즘의 9차례의 실행 결과의 평균 암호화지연시간(E)을 지연시간으로 하여, 모델별 암호화에 따르는 키 관리시스템의 부하를 계산하였다. SMKD모델은 GMEK라는 암호 키를 이용해 전송데이터의 암·복호화를 수행한다. 중앙관리시스템은 각 소그룹들에게 GMEK를 생성하고 분배한다. 암호 키의 생성은 3.4절의 수식(5)로 생성되며, 중앙관리시스템과 소그룹 간에 기밀성을 제공하고, 소그룹 내의 암호 키로 SG_Key를 사용한다. 중앙관리시스템과 소그룹 관리시스템에서 사용하는 암호화키를 달리 함으로써 암호에 따르는 시스템 부하를 분산시켜 키의 독립성을 유지한다. 표3의 암호화에 따른 부하



	1	2	3	4	5	6
KDC	0.0526	0.21	0.473	0.841	1.315	1.894
Iolus	0.0104	0.021	0.0314	0.042	0.0526	0.063
Tree	0.1052	0.2104	0.4208	0.8416	1.6832	3.3674
SMKD	0.0052	0.0105	0.0157	0.021	0.0263	0.0315

그림 20 중앙관리시스템의 암호화 부하비교
Fig 20. Central Management's Encryption Load Comparison

를 메시지 교환횟수와 연관시켜 볼 때, SMKD모델은 소그룹 수만큼 분배함으로써 부하를 절감할 수 있다.

V. 결 론

앞으로의 멀티캐스트 통신은 광범위한 활용범위와 다양한 서비스로 급격한 변화가 예상되고 있다. 따라서 수요의 급증을 고려해 볼 때, 안정적인 서비스와 관리는 매우 중요하다 할 것이다. 본 논문에서는 전형적인 중앙 키 관리 모델의 키 분배 및 관리와 보안문제로 인한 성능저하를 경감시킬 수 있는 SMKD모델을 제안하였다. 제안 모델은 키 관리시스템의 성능개선에 초점을 맞추었고, 소그룹과 멤버의 성능개선은 고려하지 않았다. 소그룹과 멤버의 성능이 중앙 키 관리 시스템에 성능을 저하시킬 수 있기 때문에 암호화 모듈의 하드웨어 화와 소프트웨어의 업그레이드 및 호환성에 대해 추가적인 연구가 필요할 것이다. 또한 다양한 서비스의 증가로 인해 무선 서비스와 같은 다수의 멤버들을 관리해야 함에 있어, 현실성을 고려한 연구가 지속되어야 할 것이다. 향후 유비쿼터스 환경에서의 멤버 증가에 따른 확장성과 서로 다른 그룹멤버들 간의 통신에 안전한 서비스 제공이 기대된다.

참고문헌

[1] 라영주, 전정훈 "멀티캐스트의 효율적 키 분배 및 보안성 향상 구축" 한국정보처리학회 vol. 10 no. 1 pp.2205-2208 2003.

[2] H. Harney, C. Muckenhirn "Group Key Management Protocol(GKMP) Architecture" RFC2094 July 1997.

[3] Suvo Mittra. " Framework for Scalable Secure Multicasting" ACM SIGOOMM '97, Cannes, France, pp. 277-288, 1997.

[4] A. Ballardie "Scalable Multicast Key Distribution" Experimental RFC1949 May 1996.

[5] Eunsook Kim, SeokJoo Koh, ShinGak Kang, Juyoung Park, "Extensions of SAP and SDP for Tight Membership Management" ETRI November 2001.

[6] B. Quinn "Session Description Protocol" RFC4570 July 2006.

[7] Mark Handley Draft-ietftmmusic-sap-v2-02 "Session Announcement Protocol" RFC2974 October 2000.

[8] Mark Handley "Session initiation Protocol" RFC2543 March 1999.

[9] P. S. Kruus and J. P. Macker, "Techniques and Issues in Multicast Security, Proc." IEEE MILCOM, 1998.

[10] Ran Canetti et al "Multicast Security: A Taxonomy and Some Efficient Constructions" Proc. IEEE INFOCOM'99, vol. 2, pp. 708-716, New York, NY, March 1999.

[11] H. Harney, C. Muckenhirn, "Group Secure Association Key Management Protocol", RFC4535, June 2006.

[12] S.Kent "IP Authentication Header" RFC4302 December 2005.

[13] S.Kent, R Atkinson "IP Encapsulating Security Payload" RFC2406 November 1998.

[14] Hardjono & Weis "The Multicast Group Security Architecture" RFC3740 March 2004.

[15] Baugher, et al "MSEC Group Key Management Architecture" RFC4046 April 2005.

[16] T.Hardjono, G. Tsudik "IP Multicast Security: Issues and Direction" Annales de Telecom, 2000 [5] S. Mittra, Iolus: A Framework for Scalable Secure Multicasting Proceedings ACM SIGCOM, pp.277-288, 1997.

[17] D. Wallner, E.Harder, R.Agee "Key management Multicast: Issue and Architectures" RFC4046 January 2006.

[18] http://www.stunnel.org/

[19] AbdelNasir Alshamsi Takamichi Saito "A Technical Comparison of IPSec and SSL" Tokyo University of Technology vol. 2 pp395-398 March 2005.

[20] 안광수 창원전문대학 "ATM망에서의 IP 멀티캐스트 지원 메커니즘" 한국컴퓨터정보학회 논문지 vol. 8 no. 4 pp117-125 2003.

[21] 김성선, 이상순, 정영목, 가천길대학 "소규모 멀티캐스트를 기반으로 한 멀티캐스트 보안구조" 한국컴퓨터정보학회 논문지 vol. 6 no. 2 pp116-122 2001.

저 자 소 개

전 정 훈

숭실대학교 컴퓨터공학석사

숭실대학교 컴퓨터공학박사수료

동덕여자대학교 전임강사

관심분야 : 네트워크 보안, 디지털 포렌식, 인증, 무선보안