情報保護學會論文誌
第19卷 第5號, 2009. 10

DTLS 기반의 안전한 VoIP 컨퍼런스 시스템 구현 및 평가*

강 성 구,[1†] 김 규 영,[1] 김 중 만,[2] 원 유 재,[2] 류 재 철[1‡]
[1]충남대학교, [2]한국인터넷진흥원

Implementation and Evaluation of Secure VoIP Conference System based on DTLS*

Seong-Ku, Kang,[1†] Kyou-Young Kim,[1] Joong-man Kim,[2] Yoo-jae Won,[2]
Jae-Cheol Ryou[1‡]
[1]Chung-Nam National University, [2]Korea Internet & Security Agency

요 약

본 논문에서는 최근 활발하게 서비스되고 있는 인터넷 전화의 컨퍼런스 통화를 안전하게 보호하기 위한 DTLS(Datagram TLS) 기반의 컨퍼런스 시스템을 구현하고 그 시스템의 성능을 측정 및 분석하였다. 컨퍼런스 시스템은 인터넷 전화 기술의 발달에 따라 그 수요가 점차 증가하고 있으며 관련된 기술 또한 날로 발전하고 있다. 하지만 컨퍼런스 서비스를 보호하기 위한 보안 프로토콜은 컨퍼런스 기술에 비해 그 발전 속도가 늦어지고 있는 것이 사실이다. 따라서 본 논문에서는 안전한 컨퍼런스 시스템을 위해 시그널링 채널 보호, 미디어 채널 보호, 그룹키 적용이 모두 가능하고 UDP기반의 VoIP 시스템 구조에 큰 변화 없이 적용 가능한 DTLS를 바탕으로 연구를 진행하였다. 현재까지 컨퍼런스 시스템에 적용 가능한 DTLS 기반의 제안된 보안 프로토콜을 살펴보고 이를 구현 및 적용하였다. 또한 암호화에 따른 오버헤드 및 키 관리 메커니즘에 따른 오버헤드를 측정 및 분석하였다.

ABSTRACT

In this paper, we implemented the conference system based on DTLS for saving securely the VoIP, which is served sprightly in the latest, securely and tested (and also analyzed) the system. As VoIP technology development, demand of conference system is increased and the related technologies are growed. But Security protocol to protect conference service is getting late than conference technology. In this paper, we studied based on DTLS protocol that can provide function of signaling channel security, media channel security and application of group key and apply to VoIP conference system based UDP unchanged. In this paper, we searched suggested security protocols based on DTLS can apply to conference system and implement and apply the protocol to conference system. And we tested (and also alayzed) overhead of encryption and key management mechanism.

Keywords: DTLS, VoIP, VoIP Conference, Group Key

접수일(2009년 2월 3일), 수정일(2009년 6월 9일), 게재확정일(2009년 9월 8일)
* 본 연구는 지식경제부 및 정보통신연구진흥원의 IT성장동력기술개발사업의 일환으로 수행하였음. (2006-S-043-03, VoIP 정보보호기술)
† 주저자, ssabro@cnu.ac.kr
‡ 교신저자, jcryou@cnu.ac.kr

I. 서 론

인터넷이 발달함에 따라 인터넷을 기반으로 한 서비스들이 폭발적으로 증가하고 있으며 그 시장 또한 빠르게 확장되고 있다. 이 중에서도 최근 급격하게 성장하고 있는 서비스 중의 하나가 VoIP(Voice of Internet

Protocol)서비스이다. VoIP 기술은 1994년 이스라엘의 보퀠택(Vocaltec)이 개발에 성공한 이래 기간으로 보면 불과 15년 남짓한 동안 급속한 발전을 거듭하여 왔으며 기존 유선전화 시장을 잠식하고 있다.

VoIP 기술은 문자 그대로 인터넷 프로토콜(IP; Internet Protocol)과 데이터 망을 이용하여 전화를 할 수 있는 기술이다. 초고속 인터넷 망의 보급 확대와 VoIP 기술의 발전에 따라 초창기의 음성 위주 IP 응용 서비스에서 최근에는 음성 뿐 아니라 비디오, 데이터 등 각종 멀티미디어 정보를 통합 전송할 수 있도록 하는 기술로 변화하고 있으며, 이러한 관점에서 MoIP(Multimedia over IP) 혹은 V2oIP (Voice and Video over IP)라 부르기도 한다. VoIP 기술은 차세대통신망(NGN, BcN), 3/4 세대 이동통신망에서 다양한 IP 멀티미디어 응용 서비스의 제공 및 컨버전스화를 실현하는 핵심기술로 부각되고 있다[1].

그러나 VoIP가 이용하는 인터넷은 원래 신뢰성 있는 커뮤니티간의 파일 교환을 목적으로 개발된 연구망이었고 따라서 보안을 염두에 두고 설계된 망이 아니다. 인터넷이 발전되어 기업 활동이나 국가 기간망의 근간으로 자리를 잡게 됨에 따라 보안문제가 가장 큰 선결과제의 하나로 부각되고 있다. 특히 기존의 전용선이나 Dial-up modem 뿐만 아니라 DSL(Digital Subscriber Line)이나 Cable modem, 무선, 위성 등의 다양한 광대역 액세스 망들이 속속 인터넷에 접속되고 이들 위에서 VoIP나 P2P(Peer To Peer) 등 다양한 새로운 응용 서비스들이 제공됨에 따라 보안문제는 그 복잡도가 날로 증가하고 있다.

또한, 저렴한 요금, 유선 전화에 버금가는 통화품질, 다양한 기능의 장점을 지닌 VoIP 에 대한 사용자 요구가 증가함에 따라 사업자들은 앞 다투어 VoIP 관련 서비스들을 개발하여 상용화하고 있어 이와 관련된 표준의 개발이 상대적으로 늦어지고 있는 실정이다. VoIP의 보안 문제 또한 서비스에 급급한 사업자들로 인해 많은 위협이 존재함에도 불구하고 보안에 대한 충분한 고려가 이루어지고 있지 않고 있는 상황이다. 특히 다자간 통화를 위한 VoIP 컨퍼런스와 관련된 보안 표준은 아직 정의된 바가 없어 많은 위협에 취약한 모습을 보이고 있다.

본 논문에서는 보다 안전한 VoIP 컨퍼런스 시스템을 위해 DTLS기반의 보안 프로토콜을 구현 및 공개용 컨퍼런스 서버와 공개용 클라이언트에 적용하여 암호화 오버 헤드 및 성능을 실험 및 측정, 분석하였다.

본 논문은 2장에서는 VoIP 보안 프로토콜의 특징을 살펴보고 3장에서는 구현에 적용한 컨퍼런스 시스템에 대해 설명한다. 또한 4장에서는 VoIP 컨퍼런스 시스템을 위한 보안 프로토콜을 구현한 내용을 설명하고 5장에서는 구현 시스템의 성능평가 내용을 설명한다. 마지막으로 6장에서는 결론을 기술하였다.

II. VoIP 보안 프로토콜

2.1 시그널링(SIP[12]) 보안

프록시 서버, 방향 재지정 서버, 등록 서버들과 UA간에 시그널을 보호하기 위해서 RFC 3261은 TLS[13]사용을 권고하고 있다. TLS의 경우 인증서 기반으로 동작하여 무결성, 기밀성, 리플라이 어택 방지 기능을 제공한다. 하지만 TLS의 경우 TCP 기반에서 동작하므로 일반적인 UDP 기반의 시그널링에 적용할 수가 없다. TLS외에도 IPsec을 이용하여 SIP 시그널링 보안을 제공받을 수 있다. IPsec은 IP 계층에 해당되는 프로토콜이므로 TCP 및 UDP의 구분 없이 모두 사용할 수 있는 장점이 있다. 하지만 RFC 3261은 IPSec의 사용을 언급하지 않고 키관리 실현방법 또는 사용할 IPSec 헤더와 모드에 대해 주어진 요구사항이 없으며 키 관리용 프로토콜로 IKE (Internet Key Exchange)가 유일하다.

또한 IPSec에 이어 DTLS(Datagram TLS)[11]를 시그널링에 적용시킬 수 있는 DTLS-SIP (SIP over DTLS)가 제시되고 있다. SIP 메시지 전송 시 SIP 메시지 헤더에 어떤 전송 프로토콜을 통하여 전송하는지 나타내는 transport 파라미터를 두어 표시하는데 DTLS-SIP의 경우는 현재 정의되어 있는 값 외에 'DTLS-UDP'와 'DTLS-DCCP'를 추가하여 SIP 메시지가 DTLS에 의하여 전송됨을 표시한다. 변경된 SIP 표준의 ABNF(Augmented Backus-Naur Form)은 [표 1]과 같다[2].

[표 1] transport 파라미터의 확장

"transport = " ("UDP" / "TCP" / "TLS" / "STCP" / "TLS-SCP" / "DTLS-DCCP" / "DTLS-UDP" / other-transport)

2.2 미디어(RTP) 보안

2.2.1 종단점에서의 암호화

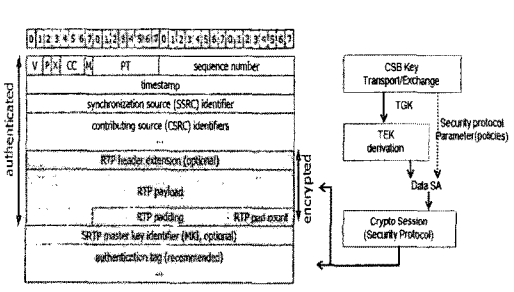
VoIP 네트워크에서 암호화 쟁점으로 인한 라우터

에서의 병목현상을 풀 수 있는 한 가지 해결책은 전적으로 종단점을 기준으로 암호화/복호화를 취급하는 것이다. 이 방법을 적용할 때 한 가지 고려할 것은 암호화 메커니즘을 취급할 수 있을 만큼 종단점이 계산적으로 강력해야 한다는 점이다. 그러나 일반적으로 종단점은 다중 클라이언트를 기반으로 하드웨어 가속을 이용하는 게이트웨이보다 덜 강력하다. 존속하는 모든 hop에서 VoIP 패킷 암호화를 유지하는 것이 이상적이지만, 소프트웨어능력 또는 계산력이라는 점에서 거의 힘을 발휘하지 못하는 단순한 IP 폰에게는 개연성이 부족하다. 이 경우, 종단점과 라우터(또는 그 반대) 간의 데이터를 암호화하는 것이 바람직할 수 있으며, LAN 기반의 암호화하지 않은 트래픽은 인터넷기반의 암호화되지 않은 트래픽에 비해 손상정도가 다소 덜하다. 다행스럽게도, 새로 나온 폰은 프로세싱 능력이 강화되어 종단점에서의 암호화 쟁점을 줄이고 있다. 이 밖에도, SRTP 및 MIKEY가 IETF의 RFC 표준지침으로 제시되어지고 있으며 이 표준지침의 문제를 해결하기위한 다양한 초안들이 제시되어 지고 있다.

2.2.2 SRTP(Secure RTP)

인터넷 전화(Internet Telephony) 애플리케이션에서 RTP는 일반적으로 실시간 오디오/비디오 데이터 전송에 사용된다. 보호하지 않을 경우 IP 기반의 전화 대화가 쉽게 도청 당할 수 있으므로 RTP의 안전을 보장하기 어렵다. 더구나 수정된 RTCP 데이터 역시 협상된 서비스품질의 비인가 변경을 가능하게 함으로써 RTP 스트림프로세싱을 교란할 수 있다.

이러한 문제 해결을 위해 SRTP 프로토콜이 제안되었다. SRTP는 [그림 1]과 같은 구조를 가지며 기밀성뿐 아니라 메시지 인증, 그리고 RTCP를 비롯하여 RTP 트래픽에 재생방지를 제공하는 RTP 프로파일이다. SRTP는 국제인터넷표준화기구(IETF)의



[그림 1] SRTP 패킷 구조 및 암호화

AVT 연구그룹이 표준화 작업을 수행하여 2004년 3월에 'RFC 3711'이라는 타이틀로 발표한 바 있다.

SRTP는 같이 RTP 및 RTCP 스트림 암호화 및 메시지 인증을 위한 프레임워크를 제공한다. SRTP는 높은 처리량과 낮은 패킷 확장을 달성할 수 있다. SRTP는 특정 RTP 스택구현과 무관하며 특정의 키 관리 표준과도 관계없다[3].

2.3 미디어 보안을 위한 키 관리

2.3.1 MIKEY(Multimedia Internet KEYing)

SRTP는 암호화, 인증 및 무결성 보호 세션 키에서 파생한 일련의 협상 파라미터를 사용한다. MIKEY는 실시간 멀티미디어 시나리오(예를 들면 SIP Call 및 RTSP Session, Streaming, Unicast, Group, 멀티캐스트)를 다루는 키관리 체계로서 현재 IETF 내의 MSEC 그룹이 표준화 작업을 진행하고 있다. 중점은 이질적인 환경 요구사항을 충족할 수 있도록 키관리 및 업데이트, 보안 정책 데이터 등을 비롯하여 멀티미디어 세션 안전을 확보하기 위한 SA를 구성하는데 목적이 있다. MIKEY는 또한 단일 및 다중 암호 세션 협상을 지원한다. 이것은 특히 SRTP에 키관리를 적용할 경우에 유용한데, 그 이유는 RTP와 RTCP를 독자적으로 보안할 수 있기 때문이다. MIKEY 배치 시나리오는 'Peer-to-Peer', 'Simple One-to-Many', 'Small-size Interactive Group'으로 구성된다.

MIKEY는 하나 이상의 보안 프로토콜에 대해 암호키와 SP(Security Parameter) 협상을 지원한다. 이것은 암호 세션 묶음 개념으로 이어지며, 이 묶음은 공통적인 TEK(Traffic Encryption 키) TGK(Traffic Generation 키)와 속성 세션 보안 파라미터를 가질 수 있는 암호 세션 모음을 뜻한다[4].

MIKEY는 다음의 중요한 속성이 있다.

MIKEY는 멀티미디어 통신프로토콜에 용이하게 통합할 수 있는 독립적인 소프트웨어라이브러리로 구현할 수 있다. 또한 특정 통신프로토콜(SIP, H.323 등)의 독립성을 제공한다.

- 2-Way 핸드셰이크 내에서 실시간 멀티미디어 시나리오에 가장 적합한 키 자료를 설정한다.
- 네 가지 키 할당 옵션이 있다.
  - Preshared-key
  - 공용키 암호화(Public-키 Encryption)

- 공용키 암호화가 보호하는 Diffie-Hellman 키 교환
- Preshared-key와 키ed Hash 기능[MIKEY 확장(DHHMAC)을 이용하는]이 보호하는 Diffie-Hellman 키 교환

• 키 재생성(Re-KEYing) 지원
• 멀티캐스트 지원(One Sender)

2.3.2 SDES(Session Description Protocol (SDP) Security Descriptions for Media Streams)

SDES[5]은 미디어 스트림을 위해 SDP의 키 전송 확장을 정의한다. 이는 시그널링을 하고 일반적인 경우에는 미디어 스트림을 위한 그리고 특별한 경우에는 SRTP를 위한 암호 키와 다른 세션 파라미터들을 협상하기 위한 하나의 방법을 제공한다. "crypto"라고 불리는 속성은 각 소스가 유일한 암호 키를 가지면 두 당사자 사이의 Unicast 미디어 스트림으로 제한된다. SRTP를 위한 crypto 속성은 다음과 같이 정의된다.

a = crypto :
〈tag〉〈crypto-suite〉〈key-params〉
[〈session-params〉]

tag는 속성 식별자로 사용되는 십진수이고 crypto-suite는 SRTP에 사용되는 암호화와 인증 알고리즘을 정의한다. key-params는 crypto-suite를 위해 하나 이상의 암호 키를 〈key-method〉: 〈key-info〉로 명기한다. 키 자체는 평문에 포함되어야 하는 경우에, 키 교환을 위해 지원되는 오직 하나의 방법은 inline: 이다.

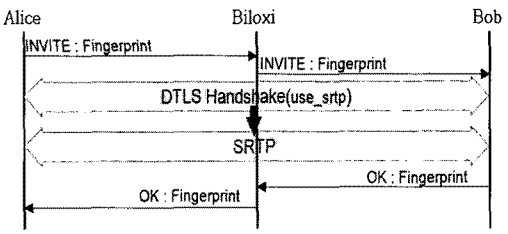
키가 SIP 메시지의 SDP 첨부에 직접적으로 포함되기 때문에 SIP는 메시지가 전송 계층에서 보호되도록 보장해야 한다. SIP에서 전송 계층 보호는 (만일 전송 계층이 TCP라면) TLS 또는 S/MIME을 사용해서 이루어 질 수 있다는 사실만 알면 충분하다. TLS는 프락시의 체인 상에서 단대단(end-to-end) 보호를 제공하지 않기 때문에 TLS의 사용은 좋지 않게 여겨진다. 게다가 SIP 프락시 체인에서 다음 홉을 신뢰한다고 가정한다. S/MIME은 반면에 MIME으로 Encode된 SDP 페이로드를 위한 단대단 기밀성과 인증을 제공한다.

S/MIME은 어떤 replay 보호도 제공하지 않는다. 따라서 S/MIME이 SDP 페이로드를 보호하기 위해 사용된다면, 응용프로그램은 replay 공격에 대한 개별적인 방어를 제공해야 한다. 일반적으로, 이를 위해서는 상태 유지(state maintenance)/느슨한 시간 동기화(loose clock synchronization)를 필요로 하기 때문에, 대부분의 응용프로그램들은 replay 보호를 제한하고 있다[6].

2.3.3 DTLS-SRTP

DTLS의 기본 디자인 개념은 "TLS over datagram" 이다. 데이터그램 환경에서 TLS를 적용시키지 못하는 직접적인 이유는 단순히 패킷이 유실되거나 재배열 될지도 모르기 때문이다. TLS는 비신뢰성을 처리하기 위한 기능을 가지고 있지 않기 때문에 데이터그램 전송에서 다시 호스트가 되는 경우 중단되게 된다. DTLS의 목적은 이와 같은 문제점을 보완하기 위한 TLS의 변경시의 요구사항을 최소로 하는 것이다. 전체적으로 살펴보았을 때 DTLS와 TLS는 거의 일치한다. S치한다. S치한키 교환 프로토콜로 [그림 2]와 같이 DTLS-SRTP가 제안되었다. DTLS-SRTP 구조는 SRTP에는 키관리 메커니즘을 제공하고 DTLS에는 새로운 RTP 데이터의 보안기능을 제공한다. 이를 위한 표준 제안에서는 DTLS 핸드셰이크 과정에 확장필드의 추가와 SRTP를 확장, 수정하는 방식을 사용하여 SRTP 데이터 전송에 DTLS를 사용할 수 있도록 정의하고 있다.

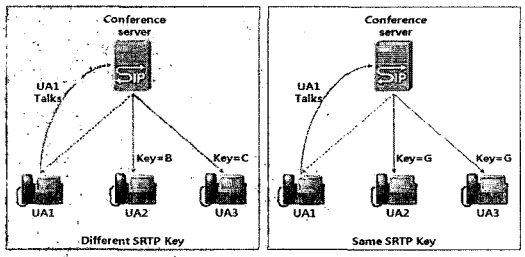
RTP 데이터 전송에 DTLS를 사용하는 과정은 먼저 일반적인 DTLS 핸드셰이크 과정이 일어나고 여기서 협상된 키를 사용하여 SRTP 키를 생성한다. 그 후 오가는 데이터는 DTLS 보안이 적용된 패킷이 아닌 SRTP 보안이 적용된 패킷이 오가게 된다. 이때 SRTP가 아닌 다른 콘텐츠 타입은 일반 DTLS Record Protocol을 사용하여 패킷이 오가게 된다[7].



[그림 2] DTLS에서의 SRTP 사용

2.4 미디어 보안을 위한 그룹키 관리

멀티미디어 컨퍼런스 구조의 미디어 보안을 위한 키 대안으로 [그림 3]과 같이 단일 키(Unique 키) 컨퍼런스 구조와 그룹키(Group 키) 혹은 공유 키(Shared 키) 컨퍼런스 구조를 가질 수 있다.



[그림 3] 단일 키 및 그룹키 컨퍼런스 구조

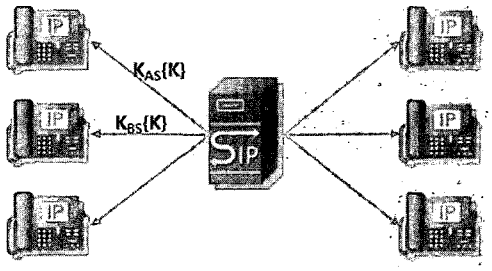
단일 키 컨퍼런스 구조의 경우 포커스는 각각의 컨퍼런스 참여자의 개인키를 잘 처리해야 한다. 이 방법은 위에 소개한 미디어를 위한 키관리 방법으로 지원하기 쉽고, 참여자들 간의 인증과 멤버십의 취소(Revocation)등의 보안적인 관리가 쉽다. 하지만 이 방법은 포커스가 송신자로부터 수신한 데이터를 각각의 참여자의 키로 데이터를 암호화하여 각각의 해당 참여자로 전송해야 하므로 참여자의 수만큼 비용이 생길 수 있다.

그룹키 컨퍼런스 구조의 경우 포커스는 참여자들에게 그룹키를 제공해야 하며 같은 키로써 모든 수신자의 트래픽을 보호해야 한다. 이 방법을 이용할 경우 암호화에 있어 성능의 개선과 비용을 줄일 수 있는 효과를 가질 수 있다. 하지만 참여자의 추가 혹은 참여자의 탈퇴 시 그룹키에 대한 취소 및 키 갱신(Rekey) 등의 정책이 필요하며 그룹키를 이용하여 참여자들 간의 인증을 수행할 수 없다.

2.4.1 중앙 집중형 그룹키 확립 프로토콜

1) Simple 키 Distribution Center - SKDC

SKDC는 가장 전통적인 그룹키 확립 프로토콜이다. [그림 4]와 같이 중앙 서버가 키를 배포하고 각각의 멤버를 인증하여 서버와 멤버사이에 안전한 채널을 통해 그룹키는 부여하고, 보내고자 하는 데이터를 그룹키로 암호화하여 메시지를 전달한다. 멤버들이 가입하고 탈퇴하는 경우 이후의 전송 메시지를 보호하기 위하여 키를 새로 변경한다. 이때 새로 변경된 키를



[그림 4] 기본적인 중앙 집중 그룹키 확립 프로토콜

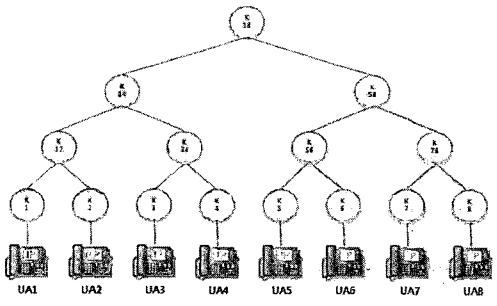
각 멤버의 개인키로 암호화하여 전송된다. 이 경우 멤버의 수가 N 이라면 변경된 키를 N번 전송해야하기 때문에 키 변경 시 전송해야 하는 메시지의 수는 O(N)가 된다. 따라서 이 방법은 멤버의 수가 늘어날수록 전송해야 하는 메시지의 수도 비례하기 때문에 멤버의 수가 커질 경우 규모 확장성 문제에 봉착하게 된다.

2) Logical 키 Hierarchy - LKH

위에서 설명한 SKDC의 확장성의 문제를 개선하기 위해 논리적 키 계층구조 방식을 제안되었다. 중앙 서버는 [그림 5] 와 같은 논리적인 트리를 유지한다. 이 트리는 다음과 같은 특성을 가지고 있다.

- 각 노드마다 하나의 키가 유지된다.
- 트리의 루트 노드에 있는 키가 그룹키가 된다.
- 각 사용자마다 하나의 단말 노드와 연관되며, 사용자는 그 노드와 그 노드의 조상 노드에 있는 모든 키를 받아야 한다. 예를 들어 사용자 UA(User Agent)1은 K1, K12, K14, K18을 유지해야한다.

따라서 초기 설정 프로토콜은 초기 가입된 사용자들을 이용하여 논리적 키 계층구조를 형성하고 서버는 각 사용자에게 각 사용자가 유지해야 하는 키를 각 사용자와 공유된 비밀키로 전달하여 준다[8].



림 5] 논리적 키 계층구조

2.4.2 DTLS-SRTP 키 전송 프로토콜

컨퍼런스에서 그룹키를 사용하기 위한 방법으로 DTLS-SRTP-KTR에 관한 제안이 진행되고 있다. 이는 DTLS-SRTP 방법을 이용해 각 사용자들과 키 교환을 수행한 뒤 그룹키를 DTLS 채널을 이용해 전송하여 암·복호화에 사용한다[9].

1) 시나리오

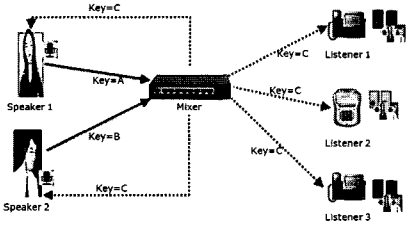
DTLS-SRTP-KTR(이하 KTR)은 믹서와 영상 스위처가 여러 개의 SRTP 키를 이용하여 여러 번 각 패킷을 암호화 하는 오버헤드를 다자간의 참여자와 공유하게 되는 하나의 SRTP 키를 가지고 SRTP 패킷을 받게 함으로써 감소시킨다. 아래 시나리오는 KTR으로 인해 발생되는 몇 가지 SRTP 시나리오를 기술한 것이다.

▪ 믹서 모델을 사용한 점 대 다중점

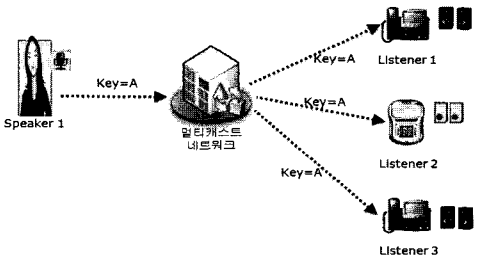
[그림 6]에서 점선은 KTR이 이용된 세션을 보여주고, 실선은 단순한 DTLS-SRTP가 필요한 곳을 보여준다. 이 위상에서, 수신자만이 KTR 지원을 필요로 하기 때문에 스위처와 수신자들이 KTR의 장점을 취할 수 있다. KTR을 이용하는 이 시나리오에서, 믹서는 그룹키 서버의 추가적인 역할을 가정하고 모든 수신자에게 공통 그룹 SRTP 키(여기에서는 "C")를 제공한다. 이 그룹 SRTP 키는 모든 수신자 사이에 공유된다. 하지만 두 송신자는 유일한 스트림을 받지만, 재전송 공격을 예방하기 위해서는 그들의 내용은 다른 SRTP 키("D" 와 "E")로 암호화된다.

▪ 멀티캐스트를 사용하는 점 대 다중점

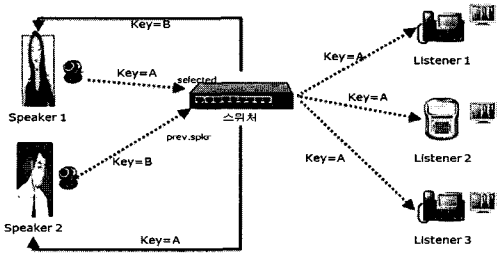
이 시나리오는 [그림 7]과 같이 동일한 키가 다중 수신자에게 제공될 수 있기 때문에 지원가능하다. 그룹 크기가 송신자가 모든 수신자에 대한 키 서버 기능을 수행할 수 있을 정도로 충분히 작을 때 적합하다.



[그림 6] KTR을 이용한 점 대 다중점 믹서



[그림 7] 키 전송 기반 멀티캐스트를 사용한 점 대 다중점



[그림 8] KTR을 이용한 점 대 다중점 비디오 스위칭

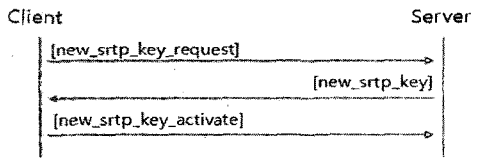
▪ 비디오 스위칭 MCU를 사용한 점 대 다중점

[그림 8]에서 영상 스위처는 퇴장하거나 입장하는 수신자를 인식한다. 본 장에서 기술하는 프로토콜은 스위처가 새로운 암호키를 이용하는 것을 지시하도록 송신자에게 지시하는 것을 허용한다. 이것은 스위처가 스위처의 정책을 근거로 하는 보안을 실시하도록 허용한다. 이것은 DTLS 영상 스위처가 "your_new_srtp_키" 메시지를 보냄으로 수행된다. 송신자는 동일한 키를 가지는 DTLS "new_srtp_키" 메시지로 응답한다. "new_srtp_키" 메시지가 스위처에 의해 수신자에게 중계된다.

2) 프로토콜

KTR을 사용하기 위해 DTLS 핸드셰이크 시 "키_transport"라는 새로운 협상을 위한 확장을 추가하며, "use_srtp" 확장과 결합하여 이용되어야만 한다. DTLS 서버는 "키_transport"를 "ServerHello" 메시지에 포함하여 키_transport에 대한 지원을 나타낸다. DTLS 클라이언트가 "키_transport"를 "ClientHello"에 포함하였지만 "ServerHello"에 "키_transport"가 없다면, DTLS 클라이언트는 "srtp_키_transport" 콘텐츠 형태로 DTLS 패킷을 전송하면 안 된다.

그룹키를 교환하기 위해, 어느 때라도 DTLS 클라이언트/서버는 키_transport_message를 보낼 수



〔그림 9〕 새로운 키 메시지 흐름

있다. 〔그림 9〕와 같이 new_srtp_키 메시지 송신자는 새로운 키로 SRTP 패킷 전송을 즉시 시작할 수 있다. 그러나 "new_srtp_키"를 이용한 메시지의 손실을 감안하기 위해 송신자는 "new_키_activate" 메시지나 "new_키_activate" 메시지에 대한 대기 타임아웃까지 새로운 SRTP 키의 변경을 기다려야 한다. 타임아웃의 지속시간은 콘텐츠의 민감도에 따라 다를 수 있다. "new_srtp_키" 메시지는 "new_키_activate" 메시지의 수신에 의해 확인될 때까지 재전송된다.

III. 컨퍼런스 시스템

3.1 구성

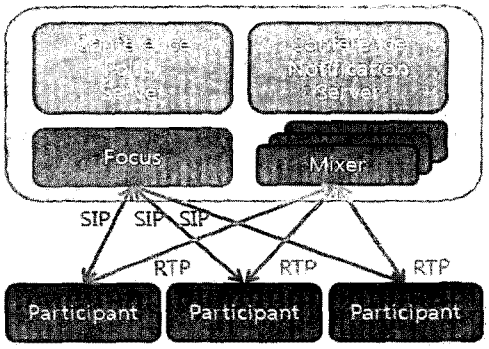
컨퍼런스 시스템은 크게 컨퍼런스 서버와 참여자들이 사용하는 클라이언트로 구성되며, 컨퍼런스 서버에는 클라이언트들의 호를 제어하는 포커스, 참여자들의 멤버십과 미디어를 제어하는 컨퍼런스 정책 서버, 오디오나 비디오 데이터를 전송하는 미디어 서버 그리고 컨퍼런스 관련 정보를 참여자들에게 통보해 주는 서버로 구성된다. 클라이언트와 서버간의 호는 SIP를 이용하여 설정되거나 해지되며, 오디오, 비디오 정보는 RTP를 통해 전송되는 것을 기본으로 하고 있다.

3.2 모델

VoIP에서 다자간 컨퍼런스를 구성하는 데 있어 크게 다음 세 가지 모델이 사용될 수 있다[9].

- Loosely coupled 컨퍼런스
- Fully distributed 다자간 컨퍼런스
- Tightly coupled 컨퍼런스

Loosely coupled 컨퍼런스는 다자간 컨퍼런스에 참여한 모든 참여자 사이에 SIP *dialog*를 통해 호를 설정하는 것이 아니라 멀티캐스트 주소를 이용하여 컨퍼런스를 구성하는 방식이다. 이 방법의 경우 컨퍼런스 서버나 제어를 위한 포커스 등이 존재하지 않는다.



〔그림 10〕 중앙 집중형 서버

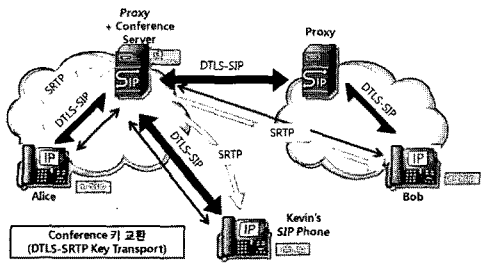
Fully distributed 다자간 컨퍼런스는 각 참여자가 SIP를 이용하여 다른 모든 참여자와 호를 설정하여 사용하는 방식이다. 이 경우도 loosely coupled 컨퍼런스 방식과 마찬가지로 제어를 위한 포커스가 존재하지 않으며 컨퍼런스의 제어를 위한 기능은 모든 참여자에게 분배되어진다.

〔그림 10〕과 같이 Tightly coupled 컨퍼런스는 제어를 위한 포커스가 존재하는 방식이다. 각 참여자는 포커스에 통화를 시도함으로써 컨퍼런스에 참여하게 된다. 이 방식을 사용할 경우 포커스에서는 컨퍼런스 제어 등 다양한 기능을 사용할 수 있으며 통화 또한 이 포커스를 통해 전달되기 때문에 미디어를 믹싱하는 기능이 필요하다.

앞에서 설명한 3가지 모델 중 모든 참여자를 중앙에서 제어하여 서비스를 제공 받는 사용자의 인증과 과금 관리가 용이한 Tightly coupled 컨퍼런스가 표준화를 위해 고려되고 있으며 RFC 4353(A Framework for Conferencing with the Session Initiation Protocol)에 다양한 모델이 제시되어 있다. 본 논문에서도 Tightly coupled 컨퍼런스를 기준으로 연구를 진행하였으며, 그 중에서도 하나의 집중된 서버를 사용하는 컨퍼런스를 기준하였다.

IV. 컨퍼런스 보안 모듈 개발

보안 모듈을 구현함에 있어 기본 토대로 사용한 것이 DTLS 이다. 본 연구에서 구현하여 적용하고자 하는 SIP 기반의 컨퍼런스 시스템은 호 설정을 위해 SIP를 사용하고 미디어 채널은 RTP를 사용한다. SIP는 UDP, TCP 위에서 모두 동작하지만 RTP의 경우 UDP 위에서만 동작하기 때문에 SIP와 RTP에 모두 보안을 적용할 수 있는 DTLS를 사용하였다.



〔그림 11〕 DTLS 기반의 VoIP 컨퍼런스 시스템 전체 구성도

구현한 전체적인 시스템 구조는 〔그림 11〕과 같다. 각 구성간의 시그널 채널은 DTLS-SIP를, 미디어 채널은 SRTP를, 미디어 키 교환의 경우 MIKEY와 SDES, DTLS-SRTP를 적용하였으며 그룹키 교환 프로토콜로 DTLS-SRTP-KTR를 적용하였다. DTLS는 OpenSSL 0.9.8 〔14〕버전에 기본적으로 구현되어 있지만 그 외 DTLS-SRTP 및 DTLS-SRTP-KTR은 기존 OpenSSl을 바탕으로 직접 구현하여 해당 핸드셰이크 및 키 교환, 데이터 전송이 가능하도록 수정하였다. 이를 바탕으로 Proxy Server 및 User Agent에 적용이 용이하도록 〔표 2〕와 같이 API를 구현하였다.

V. 성능분석

5.1 환경 구성

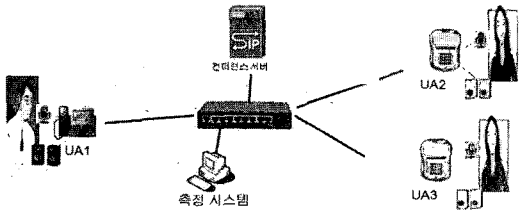
성능 측정을 위한 환경을 〔그림 12〕와 같이 구성하였다. 보안 모듈이 OpenSSL 0.9.8h 버전을 토대로 구현되었기 때문에 각각의 장비에는 OpenSSL 0.9.8의 수정 버전을 설치하였으며, 위에서 구현한 DTLS-SIP, DTLS-SRTP, DTLS-SRTP-KTR(SKDC, LKH) 보안 모듈을 또한 적용하였다.

Open Source로 제공되는 Proxy server와 User Agent 중 컨퍼런스 시스템을 구성하기 위한 프로그램으로 각각 Asterisk〔15〕와 Minisip〔16〕을 선택하였다. Asterisk는 VoIP에서 Telephony 엔진과 툴킷들 중 가장 많이 사용되고 있는 오픈소스기반의 PBX시스템이다. Zaptel 패키지를 이용해 컨퍼런스 기능을 지원하며 Audio 및 Video 믹싱 모듈과 연동할 수 있다. Minisip은 User Agent로 사용 가능한 Open Source 중 가장 많은 보안 기능을 지원하며 Asterisk와 가장 연동이 잘되는 프로그램이다.

이들 두 프로그램을 이용해 기본적인 SIP 기반의 컨퍼런스 시스템을 구축하였다. 컨퍼런스 시 각 참여

〔표 2〕 구현한 DTLS 통합 라이브러리 리스트

함수명	설명	비고
dtls_new	DTLS 클라이언트 환경설정에 관련된 변수를 할당한다.	공통
dtls_setup	DTLS 클라이언트 환경설정에 관련된 변수를 설정한다.	공통
dtls_free	DTLS 클라이언트 환경설정에 관련된 변수를 반환한다.	공통
dtls_srtp_get_키	SRTP키를 획득한다.	공통
dtls_set_timeout	타임아웃을 설정한다.	공통
dtls_recv	데이터를 수신한다.	공통
dtls_send	데이터를 송신한다.	공통
dtls_renegotiate	서버와 새로 세션을 연결한다.	공통
dtls_srtp_handshake	DTLS-SRTP를 위한 핸드셰이크를 한다.	DTLS-SRTP
dtls_srtp_키_transport_handshake	DTLS-SRTP KTR를 위한 핸드셰이크를 한다	DTLS-SRTP-KTR
dtls_srtp_키_transport_handshake_server_mode	DTLS-SRTP KTR를 위한 핸드셰이크를 서버모드로 한다.	DTLS-SRTP-KTR
client_send_new_srtp_키_server_mode	새로운 SRTP키를 서버모드인 경우 전송한다.	DTLS-SRTP-KTR
dtls_srtp_키_transport_handshake_client_mode	DTLS-SRTP KTR를 위한 핸드셰이크를 클라이언트모드로 한다.	DTLS-SRTP-KTR



〔그림 12 성능 분석을 위한 환경 구성

자는 컨퍼런스 서버의 1:1로 호 채널 및 미디어 채널을 사용하게 되며 컨퍼런스 서버는 음성 및 영상 믹싱을 통해 각 참여자에게 다른 참여자의 미디어를 전송하게 된다.

5.2 성능 측정 및 분석

본 연구에서 구현한 보안 모듈과 웹상에 공개되어 있는 다른 보안 모듈과 가장 큰 차이점은 컨퍼런스 시스템모듈과 함께 있어 그룹키의 사용 여부이다. 현재 VoIP 보안을 위해 많이 사용되고 있는 MIKEY의 경우 컨퍼런스 시스템모듈과 할 경우 그룹키가 아닌 각각의 개인키로 암·복호화를 수행하기 때문에 컨퍼런스 서버에 오버헤드가 발생하게 된다. 반대로 그룹키를 사용하는 DTLS-SRTP-KTR의 경우에는 그룹키의 사용으로 인해 컨퍼런스 서버의 암호화 오버헤드는 줄어들지만 그룹키 유지를 위한 오버헤드가 발생하게 된다.

이 두 가지 경우를 고려하여 컨퍼런스 참여자를 1~3명까지 변화시켜가며 각 보안 방식에 따른 소요시간을 측정하였다. 소요시간은 측정 시작위치와 측정 종료 위치에 gettimeofday 함수를 사용하여 시간을 기록하였고 테스트 결과 gettimeofday 함수를 실행하는데 걸리는 시간은 1 μsec 미만으로써 성능측정에 거의 영향을 미치지 않는 것으로 나타났다. 구체적인 평가 방법은 아래와 같다.

5.2.1 키 전송 시간 측정

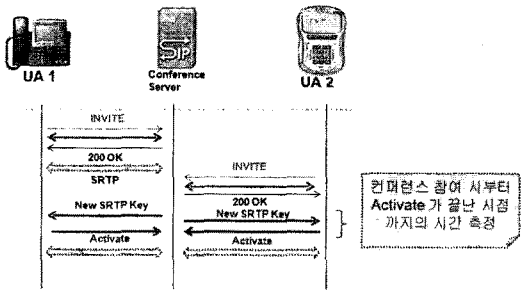
보안 프로토콜을 적용하는 데 있어 가장 신중하게 고려해야 하는 부분이 키 교환이다. 키 교환은 보안 프로토콜의 강도에 영향을 미칠 뿐만 아니라 키 교환에 많은 시간이 소요될 경우 미디어 품질에 영향을 미칠 수가 있다. 특히 그룹 키를 사용하는 프로토콜의 경우 키 교환이 빈번하게 발생할 수 있기 때문에 이와 관련된 측정이 필요하다.

키 교환 시간 측정의 대상은 SIP 메시지를 이용하는 SDES/MIKEY와 DTLS 핸드셰이크를 사용한 DTLS-SRTP, 그룹키 전송에 사용되는 DTLS-SRTP-KTR 이며 각 프로토콜이 키 교환 방법이 다르기 때문에 각각 다른 방법을 사용하여 키 교환 시간을 측정하였다.

SDES나 MIKEY의 경우 INVITE와 200 OK 메시지를 통해 서로의 키를 교환한다. 따라서 이 두 방법의 경우에는 INVITE 전송 시간부터 200 OK 수신 시간까지 소요된 시간을 측정하였으며 DTLS-SRTP의 경우에는 미디어 채널의 DTLS 핸드셰이크를 통해 키 교환이 수행되므로 핸드셰이크에 소요되는 시간을 키 교환 시간으로 측정하였다. DTLS-SRTP-KTR의 경우 〔그림 13〕과 같이 SKDC와 LKH 방법 모두 1차적으로는 DTLS-SRTP와 같은 방법을 이용하여 키 교환을 수행하고, 그룹키 교환 시에만 추가적인 시간이 소요된다. 따라서 이 두 프로토콜은 새로운 참여자가 컨퍼런스 룸에 들어온 시간부터 activate 메시지를 수신한 시간까지를 그룹키 교환 시간으로 계산하였다.

약 10,000회의 시도에 대한 시간 측정한 결과 그 평균값은 〔표 3〕과 같다.

SDES 및 MIKEY의 경우 SIP 메시지의 SDP 부분에 그 키값을 설정하여 전송하게 된다. 해당 메시지를 UDP로 전송함으로 전체 수행 시간 및 평균 송수신 시간이 DTLS-SRTP 및 DTLS-SRTP-KTR



〔그림 13〕 DTLS-SRTP-KTR의 키 교환 시간 측정

〔표 3〕 각 프로토콜별 키 교환 시간 측정 결과

프로토콜	전체 수행 시간(sec)	평균 송수신 시간(usec)
SDES/MIKEY	12.58792	1285.455
DTLS-SRTP	60.013657	6055.358
DTLS-SRTP-KTR	59.989123	6035.128

과 많은 차이를 보이고 있다. DTLS-SRTP 및 DTLS-SRTP-KTR의 경우 핸드셰이크 과정을 통해 키 값을 얻어오는 과정에서 SDES 및 MIKEY 보다 상대적으로 더 많은 과정이 필요함으로 송신에서 응답까지의 시간이 상대적으로 크게 측정된다. 또한 DTLS-SRTP 및 DTLS-SRTP-KTR는 키 교환을 위한 절차가 매우 유사하므로 서로 비슷한 결과값을 보이고 있다. DTLS-SRTP 및 DTLS-SRTP-KTR 경우 핸드셰이크로 인한 오버헤드, 송. 수신 측에서 암호화 및 복호화 과정이 각각 일어남으로 인해 소비되는 오버헤드로 생각되어 질 수 있다. 하지만 SDES 및 MIKEY와 같이 UDP기반의 전송에서는 데이터가 목적지로 전송중 Loss되는 경우가 발생되거나 데이터의 순서가 바뀔 수 있는 문제가 생길 수 있다.

5.2.2 키 교환 시간 측정

컨퍼런스 시스템에 그룹키를 사용하는 것은 암호화 시간을 줄임으로써 장점으로 작용하지만 그룹키 관리를 위한 오버헤드를 발생시키는 단점도 가지고 있다. 이에 컨퍼런스의 인원을 증가시켜 가며 키 관리에 소요되는 시간을 측정하였다.

측정은 새로운 참여자가 호 설정을 통해 컨퍼런스에 참여한 후의 시점부터 New SRTP 키 메시지를 전송한 후 Activate 메시지를 받는 데 걸리는 시간을 측정하였다. Activate 메시지는 1 명에게 받을 때마다 시간을 기록하여 각 참여자 들이 New SRTP 키 메시지를 처리하는 시간도 함께 알아보았다. 측정은 100번을 실시하여 평균한 값을 데이터로 사용하였다.

또한 DTLS-SRTP-KTR의 두 가지 버전의 차이를 고려하여 각각 측정을 실시하였다. 측정의 결과는 아래 〔표 4〕, 〔표 5〕, 〔표 6〕, 〔표 7〕과 같다.

〔표 4~7〕의 측정 결과로 볼 때 전체 시간으로 놓고 보자면 참여자 증가 시 Activate 도착에 많은 시간이 소요되는 것으로 나타났지만 원인을 살펴볼 때 마지막 참여자로부터의 응답이 다른 참여자에 비해 현저하게 늦게 도착하는 것을 알 수 있다. 이는 마지막 참여자가 호설정이 끝남과 동시에 그룹 키 교환을 실시하기 때문에 미디어 채널의 초기화를 위한 시간이 많이 소요되기 때문으로 판단된다. 이와 같은 이유로 참여자 감소 시 키 교환 소요 시간은 참여자 증가 시에 비해 적게 측정되었다.

또한, 위 〔표 6, 7〕로부터 SKDC와 LKH 키 관리 메커니즘에 따른 키 전송 및 적용, Activate 메시지

〔표 4〕 DTLS-SRTP-KTR(SKDC)의 경우 참여자 수에 따른 키 교환 소요시간

참여자 수 / 소요 시간	1명 ⇨ 2명	2명 ⇨ 3명	3명 ⇨ 4명
UA1 Acticate 도착 소요 시간	634 μsec	564 μsec	539 μsec
UA2 Acticate 도착 소요 시간	280,591 μsec	843 μsec	750 μsec
UA3 Acticate 도착 소요 시간		279,765 μsec	883 μsec
UA4 Acticate 도착 소요 시간			279,676 μsec

〔표 5〕 DTLS-SRTP-KTR(LKH) 참여자 수에 따른 키 교환 소요시간

참여자 수 / 소요 시간	1명 ⇨ 2명	2명 ⇨ 3명	3명 ⇨ 4명
UA1 Activate 도착 소요 시간	766 μsec	697 μsec	723 μsec
UA2 Activate 도착 소요 시간	230,101 μsec	890 μsec	790 μsec
UA3 Activate 도착 소요 시간		279,650 μsec	911 μsec
UA4 Activate 도착 소요 시간			232,898 μsec

〔표 6〕 DTLS-SRTP-KTR(SKDC)에서의 모든 Activate 도착 시간

참여자 수 / 소요 시간	4명 ⇨ 3명	3명 ⇨ 2명	2명 ⇨ 1명
모든 Activate 도착 소요 시간	1,031 μsec	761 μsec	558 μsec

〔표 7〕 DTLS-SRTP-KTR(LKH)에서의 모든 Activate 도착 시간

참여자 수 / 소요 시간	4명 ⇨ 3명	3명 ⇨ 2명	2명 ⇨ 1명
모든 Activate 도착 소요 시간	958 μsec	706 μsec	573 μsec

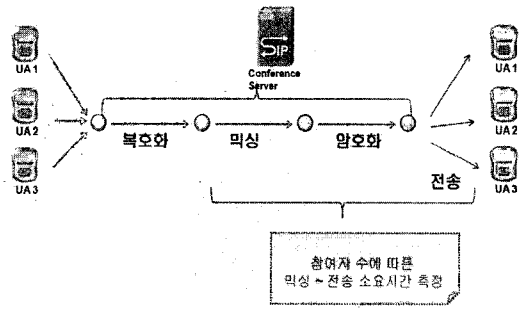
수신까지의 소요시간이 SKDC에 비해 LKH가 더 적은 시간을 볼 수 있다. 이는 SKDC 와 LKH간에 새로운 키 전송에는 차이가 없지만 키 전송 데이터의 암호화의 횟수가 SKDC는 참여자의 수만큼 되는 반면 LKH는 최대 트리 높이만큼의 횟수가 필요하므로 암호화에 걸리는 시간차이로 분석될 수 있다.

5.3 컨퍼런스 시스템 내부 처리시간 측정

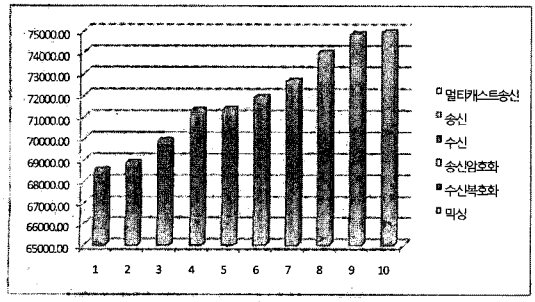
본 절에서는 보안 프로토콜의 적용이 컨퍼런스 시스템 내부에서 발생할 수 있는 음성 및 비디오 데이터 수신과 다시 컨퍼런스 사용자에게 전송까지 발생되는 지연시간과 오버헤드를 비암호화의 경우와 SDES등과 같은 개인키로의 암호화 경우, 또한 그룹키를 사용했을 경우에 따른 처리 시간 오버헤드를 [그림 14]와 같이 측정 및 분석 하였다.

측정을 위해 1명에서 10명의 가상 사용자를 RTP Generator를 수정 및 이용하여 구성하였으며 컨퍼런스 시스템의 복호화 과정을 제외한 나머지 모듈을 분리하여 실험을 실시하였다. 분석을 위해 1명에서 10명의 사용자를 차례대로 컨퍼런스에 참여 시켰으며 각 시점에서 약 10,000개(200초)의 패킷을 수집 및 이용하여 측정하였다.

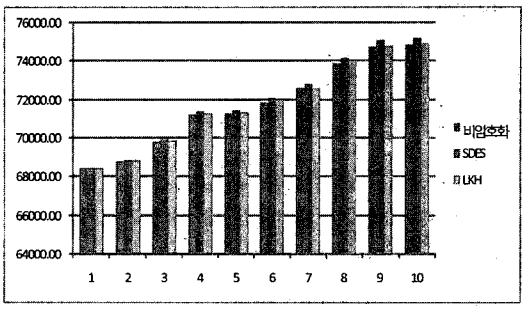
[그림 15]를 볼 때 서버 내부에서의 총 처리 시간의 대부분은 믹싱과정이 차지하고 있으며 이 시간은 서버에서의 가장 큰 오버헤드로 생각될 수 있다. 믹싱의 오버헤드로 [그림 16]과 같이 비암호화 및 SDES, DTLS-SRTP-KTR(LKH)간에 총 처리시간의 구분을 가지기 어렵다. 하지만 믹싱과정의 시간을 제외한 나머지 데이터인 DTLS-SRTP-KTR를 기반한 그룹



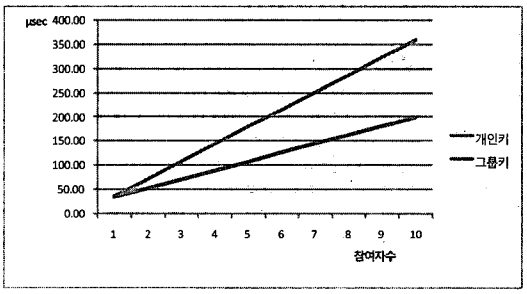
[그림 14] 컨퍼런스 시스템 내부 처리 시간 측정



[그림 15] 각 기능별 소요 시간



[그림 16] 사용자 추가로 인한 각각의 처리 시간



[그림 17] 그룹키 사용 및 개인키 사용의 처리 시간 비교

키의 사용과 SDES와 같이 개인키를 사용하였을 경우의 암호화 및 복호화 과정의 처리 시간만을 비교하면 [그림 17]과 같다. 개인키 및 그룹키의 사용의 경우 두 가지 방법 모두 선형적 증가의 모습을 띄지만 그 차이는 참여자의 수가 증가 할수록 그 차이가 현저히 커짐을 볼 수 있다. 이는 그룹키를 사용하면 데이터 송신시 한번의 암호화만이 일어나기 때문이다. 위 내용들을 비추어 봤을 때 믹싱의 오버헤드를 현저히 줄이고 그룹키를 사용하여 암호화에 적용한다면 컨퍼런스 서비스에 필요한 처리 시간 단축에 개인키 사용보다 좋은 효과를 가질 수 있다.

VI. 결 론

인터넷 사용으로 인해 네트워크상에서 발생할 수 있는 침해 및 공격은 VoIP 서비스에 쉽게 적용되어진다. VoIP 음성 패킷의 경우 압축이나 인코딩이 수행되어 기존 전화에 비해 분석이 어렵다는 견해도 있으나, Wireshark, Cain등과 같이 오픈된 다양한 네트워크 분석 소프트웨어를 이용하여 쉽게 해석되어진다. 이로 인해 VoIP에는 기존 전화망에 비해 보다 높은 보안이 요구되어지며 다양한 보안 프로토콜들이 제시되고 있다.

그러나 대부분의 보안 프로토콜들은 1:1 통신을 위해 제안되었으며 컨퍼런스 시스템을 위한 보안 표준은 아직 제시되지 않은 상황이다. 본 연구에서는 멀티미디어 컨퍼런스 시스템을 구축하여 DTLS기반의 보안 프로토콜 외에도 기존의 1:1 통화에 사용된 보안 프로토콜들과 컨퍼런스 시스템에 적용 가능한 보안 DTLS-SRTP-KTR 프로토콜을 적용하여 성능을 측정하였다.

측정 결과 암호화가 컨퍼런스 서버에 미치는 영향은 믹싱에서의 오버헤드에 비해 매우 낮았으며 이로 인해 각 보안 프로토콜들의 믹싱부터 전송에 걸리는 시간은 보안이 적용되지 않았을 때와 크게 차이가 나지 않았다. 또한 믹싱이 다른 서버에서 수행되어 믹싱으로 인한 오버헤드를 고려하지 않을 경우 미디어 암호화에 그룹키를 사용하는 프로토콜이 개인키를 사용하는 프로토콜에 비해 미디어 전송 시 시간이 적게 소요됨을 알 수 있었다. 이는 참여자 수에 관계없이 미디어 암호화 횟수를 1번으로 줄임으로써 암호화에 따른 오버헤드를 감소시킨 결과이다

그룹 키의 사용에 있어 또 한 가지 고려해야 할 사항은 키 관리 매커니즘이다. 본 연구에서 구현하여 성능을 측정한 SKDC와 LKH의 경우 본 테스트 환경에서는 다소 큰 차이를 보이지 않지만 그룹의 참여자가 늘어남에 따라 LKH의 키 관리 매커니즘이 좋은 성능을 나타낼 것으로 예상되며 이는 큰 그룹일수록 성능을 발휘하는 그룹 키를 사용하는 프로토콜과 맞물려 좋은 성능을 나타낼 수 있을 것이다.

향후에는 좀 더 많은 컨퍼런스 참여자를 이용한 테스트를 실시하여 그룹키의 사용 여부 및 키 관리 방안에 따른 성능 개선 정도를 정확하게 수치화할 필요가 있으며 컨퍼런스 서버의 최대 수용 인원의 변화 또한 연구할 가치가 있을 것으로 판단된다.

참 고 문 헌

[1] VoIP 표준화 로드맵, IT Standard Weekly 2006-31호, 2006년 8월.

[2] C. Jennings, Cisco Systems, and N. Modadugu, "Session Initiation Protocol (SIP) over Datagram Transport Layer Security (DTLS)," draft-jennings-sip-dtls-05.txt, Oct. 2007.

[3] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The Secure Real-time Transport Protocol(SRTP)," RFC 3711, Mar. 2004.

[4] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, "MIKEY: Multimedia Internet KEYing," RFC 3830, Aug. 2004.

[5] B. Ramsdell, "S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)," RFC 3853, July 2004.

[6] F. Andreasen and M. Baugher, "Session Description Protocol (SDP) Security Descriptions for Media Streams," RFC 4568, July 2006.

[7] D. McGrew and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)," draft-ietf-avt-dtls-srtp-07.txt, Feb. 2009.

[8] D. Wallner, E. Harder, and R. Agee, "Key Management for Multicast: Issues and Architectures," RFC 2627, June 1999.

[9] D. Wing and Cisco, "DTLS-SRTP Key Transport," draft-wing-avt-dtls-srtp-key-transport-02.txt, July 2008.

[10] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol," RFC 4353, Feb. 2006.

[11] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security," RFC 4347, Apr. 2006.

[12] J. Rosenberg, H. Schulzrinne, G. Camarillo, A.R. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol. IETF," RFC 3261, June 2002.

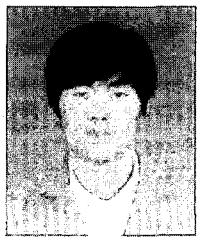
[13] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol," RFC 4346, April. 2006.

[14] OpenSSL, http://www.openssl.org

[15] Asterisk, http://www.asterisk.org

[16] Minsip, http://minisip.org/

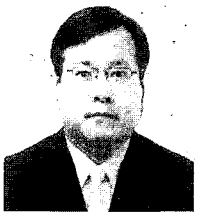
〈著者紹介〉



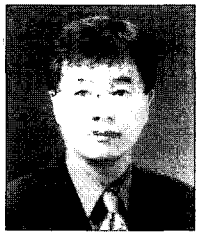
강 성 구 (Seong-Ku Kang) 학생회원
2008년 2월: 충남대학교 컴퓨터공학과 졸업
2008년 3월 ~ 현재: 충남대학교 컴퓨터공학과 석사과정
〈관심분야〉 VoIP 보안, 네트워크 보안, 암호프로토콜

김 규 영 (Kyu-young Kim) 학생회원
2007년 2월: 충남대학교 컴퓨터공학과 졸업
2009년 2월: 충남대학교 컴퓨터공학과 석사 졸업
〈관심분야〉 VoIP 보안, Ad-hoc 보안

김 중 만 (Joong-man Kim)
2002년 2월: 한국과학기술원 수학과 졸업
2004년 2월: 한국정보통신대학원 컴퓨터공학과 석사
2005년 6월 ~ 현재 : 한국인터넷진흥원 소속
〈관심분야〉 암호프로토콜, 네트워크 보안(VoIP 보안)



원 유 재 (Yoo-jae Won)
1998년 8월: 충남대학교 전산학과 박사
1987년 2월 ~ 2001년 2월: 한국전자통신연구원(ETRI) 팀장
2001년 3월 ~ 2004년 8월: 안랩유비웨어, 안철수연구소 CTO
2004년 9월 ~ 현재: 한국인터넷진흥원 IT기반보호단장
〈관심분야〉 정보보호, PKI, IPv6
〈관심분야〉 VoIP 보안, IPv6 보안, 멀티캐스트 보안, 무선 인터넷 보안, PKI 등



류 재 철 (Jae-Cheol Ryou) 종신회원
1988년 5월: Iowa State University 전산학과 석사
1990년 12월: Northwestern University 전산학과 박사
1991년 ~ 현재: 충남대학교 정보통신공학부 교수
1997년 ~ 현재: 한국정보보호학회 이사
2001년 ~ 현재: 국가정보원 정보보호시스템 인증위원회 위원
2003년 ~ 현재: 인터넷침해대응기술연구센터 센터장
〈관심분야〉 네트워크 보안(VoIP 보안), 인증이론 및 시스템, 유·무선 인터넷 보안