DOI: 10.3745/KIPSTC.2010.17C.5.415

H.264/SVC에 대한 접근 제어 방법 및 키 관리 방법

조 태 남[†] · 용 승 림[††]

요 약

CAS는 정당한 사용자만이 콘텐츠를 수신할 수 있도록 하는 수신 제어 시스템으로서, 최근 각광받고 있는 IPTV 등에서도 이를 이용하고 있다. H.264/SVC는 동일한 콘텐츠를 다양한 네트워크 환경과 단말기 성능에 따라 수신할 수 있도록 하는 표준 코딩 기법이다. 이 기법은 사용자가 수신 품질을 결정할 수 있도록 콘텐츠를 계층 구조로 구성한다. 그러므로 콘텐츠 제공자는 사용자가 요금을 지불한 품질의 콘텐츠만을 수신할 수 있도록 제어할 수 있어야 한다. 현재 사용되고 있는 CAS의 구조는 H.264/SVC 콘텐츠 제어에 적합하지 않다. 본 논문에서는 CAS 구조에서 H.264/SVC 콘텐츠 접근을 효율적으로 제어하기 위한 기법과 키관리 기법을 제안한다.

키워드 : 수신제한시스템, H.264/SVC, 키관리

Access Control Method and Key Management Method for H.264/SVC

Taenam Cho[†] · Seunglim Yong[††]

ABSTRACT

CAS is an access control system by which only legal users can access contents. IPTV is a spotlighted system that uses CAS. H.264/SVC is a coding standard that provides a scalable coding method by which users who are in various network environments and have various devices can receive the contents. In this method, the contents are coded in a layered structure to make users choose the quality of the receiving contents. Therefore, contents provider should be able to control users to access only appropriate contents according to their subscriptions. The structure of CAS being employed in many applications is not suitable to control access for H.264/SVC. In this paper, we provide an efficient access control method and a key management method for H.264/SVC contents using CAS.

Keywords : Conditional Access System, H.264/SVC, Key Management

1. 서 론

수신 제한 시스템(CAS: Conditional Access System)은 IPTV와 같이 유료로 콘텐츠를 제공하는 서비스에서 정당한 사용자만이 콘텐츠를 수신할 수 있도록 제어하는데 사용되고 있는 시스템이다[2]. CAS에서는 스크램블링(scrambling) 알고리즘을 이용하여 콘텐츠를 보호한다. 콘텐츠를 스크램블할 때 코드워드(code word)가 사용되는데, 이 코드워드는 콘텐츠를 디스크램블링(descrambling)하는 데도 사용되므로 수신 권한이 있는 정당한 사용자만이 알 수 있도록 한다. 이 코드는 안전성을 위해 주기적으로 갱신되는데, 갱신된 코드워드는 사용자와 서비스 제공자와의 비밀키(secret key)를 이용하여 암호화 통신을 함으로써 보호한다.

사용자들은 케이블, 위성, 모바일 등 다양한 네트워크 환경에서 단말기를 이용하여 음성, 데이터, 영상 서비스를 받는다. 사용자의 단말기는 처리 용량, 스크린 크기 등의 면에서 서로 다른 특성을 가질 수 있다. SVC(Scalable Video Coding)는 비디오 통신 시스템의 다양성 문제를 해결하는 효과적인 방법 중의 하나이다. SVC는 “one-encoding-multiway-decoding” 방식으로 확장성을 제공한다. 즉, 사용자의 기기와 네트워크 환경에 적합하도록 비디오를 다시 인코딩(encoding)하거나 코드를 수정하지 않도록 한다. H.264/AVC에 대한 SVC 표준(H.264/SVC)은 비디오의 세 가지 품질 요소(공간적(spatial), 시간적(temporal), 화질적(quality) 요소)에 대해 확장성을 제공하기 위하여, 세 가지 요소에 대하여 각각 계층 구조로 코딩을 하여 네트워크나 사용자 기기의 환경에 따라 선별적으로 디코딩할 수 있도록 하는 방식이다[3-5]. PPV(Pay Per View)에서는 사용자의 선택(지불한 요금)에 따라 콘텐츠의 품질을 달리할 수 있다. 이를

※ 이 논문은 2010학년도 우석대학교 교내학술연구비 지원에 의하여 연구되었음
† 종신회원 : 우석대학교 정보보안학과 조교수
†† 정 회 원 : 인하공업전문대학 컴퓨터시스템과 전임강사(교신저자)
논문접수 : 2010년 7월 6일
수 정 일 : 1차 2010년 8월 11일
심사완료 : 2010년 8월 27일

위해서는 사용자가 지불한 서비스 품질에 맞는 콘텐츠만 수신할 수 있도록 제한하는 방법이 필요하므로, 각 계층별로 다른 키(key)로 암호화하여 전송하고, 각 사용자는 요금을 지불한 계층의 복호화키를 소유하여 복호화할 수 있도록 할 수 있다[1].

H.264/SVC에서 세 가지 각 요소가 여러 계층으로 이루어지기 때문에 사용자의 편의에 따라 세 요소의 계층을 독립적으로 선택할 경우 매우 다양한 종류의 콘텐츠 품질이 있을 수 있다. 기존의 CAS를 이용하는 시스템에서는 콘텐츠의 보호를 위해 하나의 코드워드가 사용되고 있으며 사용권한이 있는 모든 사용자에게 브로드캐스트로 전송된다. 그러므로 H.264/SVC 콘텐츠에 대한 수신제한을 하기 위해서는 사용자마다 서로 다른 복호화키 집합을 유니캐스트로 전송하여야 한다는 문제가 발생한다. 본 논문에서는 H.264/SVC 콘텐츠에 대하여 효율적으로 수신을 제한하는 방법을 제시한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 기술하고, 3장에서는 관련 연구를 CAS 구조에 적용하여 분석하며 4장에서는 본 논문에서 제안하고자 하는 키 관리 구조를 기술한다. 마지막으로 5장에서는 결론과 향후 연구과제에 대하여 기술한다.

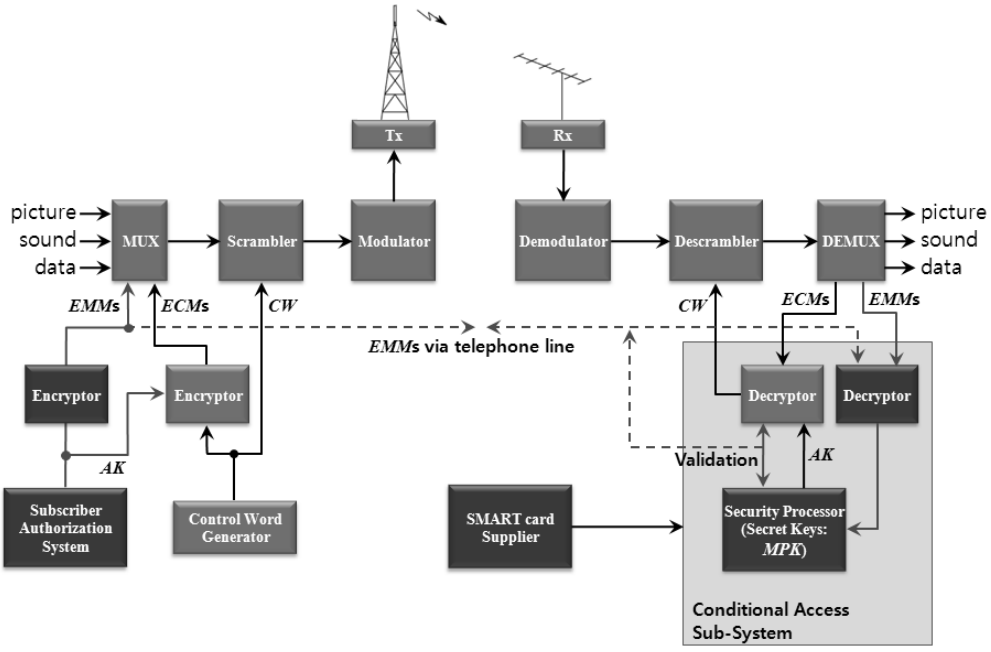
2. 관련 연구

2.1 수신제한 시스템(CAS)

CAS(Conditional Access System)는 스크램블링과 암호화를 이용하여 비인가된 사용자가 데이터를 수신할 수 없도록 하는 시스템이다. [2]에서는 (그림 1)과 같이 수신 제한 시스템 모델을 제시하였다. 서비스 제공자(서버)는 콘텐츠를 CW(CodeWord)로 스크램블(scramble)하여 사용자들에게 브로드캐스트한다. 정당한 사용자는 CW를 소유하도록 하며, 이 값을 소유한 사용자만이 수신한 콘텐츠를 디스크램블(descramble)할 수 있다.

이를 위해 먼저 정당한 사용자 U_i는 오프라인으로 서버와의 비밀키인 MPK_i(Master Private Key)를 공유한다. 이 키는 스마트 카드에 저장되어 사용자측 장치(예: Set-Top Box)에 장착된다. 사용자가 콘텐츠를 수신하여 디스크램블하려면 서버로부터 CW를 전달받아야 한다. (그림 1)에서 보는 바와 같이 CW는 콘텐츠 송신 이전에 사용자 모두가 공유한 비밀키 AK(Authorization Key)로 암호화 되어 ECM(Entitlement Control Message)이라는 메시지를 통해 전송된다. 또한 AK는 사용자 U_i와 서버와의 비밀키인 MPK_i로 암호화되어 EMM(Entitlement Management Message)라는 메시지를 통해 사용자들에게 전송된다. 서버는 안전성을 위하여 주기적으로 CW를 갱신하는데, 갱신된 CW도 AK(Authorization Key)로 암호화한 후 ECM을 통해 전송된다.

정당한 사용자만이 데이터를 수신하도록 하려면 다음과 같은 조건이 만족되어야 한다. 새로 가입한 사용자는 가입 이전의 데이터에 접근할 수 없어야 하며, 탈퇴한 사용자는 탈퇴 이후의 데이터에 대해 접근할 수 없도록 해야 한다. 이를 각각 후방향 안전성(backward secrecy)과 전방향 안전성(forward secrecy)이라고 한다. 이 안전성을 제공하기 위해서는 모든 사용자가 공유하는 그룹키(group key)인 AK가 사용자의 가입/탈퇴마다 새로운 값으로 갱신되어야 한다. 그렇지 않다면 사용자가 탈퇴한 이후에도 AK로 암호화된 CW를 복호화할 수 있고, 새로 가입한 사용자는 가입 후 알게 된 AK를 이용하여 가입 이전에 수신하여 저장한



(그림 1) CAS 구조

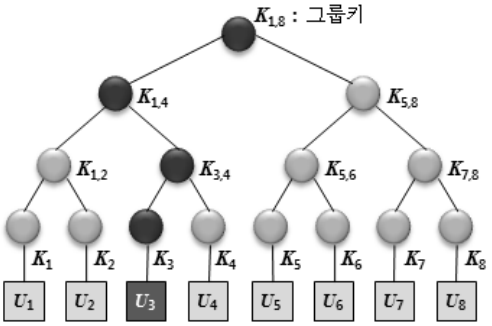
ECM을 복호화함으로써 이전 CW를 알 수 있기 때문이다. 갱신된 AK는 각 사용자와 서버만이 공유한 MPK_i로 암호화하여 EMM이라는 메시지를 통해 전달된다. ECM과 EMM에는 송신자 인증 및 데이터 무결성을 위하여 서버의 서명이 포함되어 있다.

2.2 계층적 그룹키 관리 기법

응용에 따라 후방향 안전성은 고려하지 않거나, 사용자의 가입/탈퇴와 상관없이 그룹키를 주기적으로 변경할 수도 있다[6]. 그러나 전방향 안전성과 후방향 안전성을 모두 만족시키기 위해서는 1절에서 기술한 바와 같이 사용자의 가입이나 탈퇴가 일어날 때마다 그룹키가 갱신되어야 한다. 사용자의 가입에 따른 갱신 방법은 매우 간단하다. 현재의 그룹키를 GK라 하고 갱신된 그룹키를 GK'이라 할 때, GK로 GK'를 암호화한 메시지 $E_{GK}(GK')$를 사용자들에게 브로드캐스트하고, 새로운 사용자에게는 GK'을 유니캐스트 해주면 된다. 사용자 변동이 없을 때 수행하는 주기적 그룹키 갱신에도 동일하게 적용할 수 있다. 그러나 사용자가 탈퇴할 경우에는 이 방법을 사용할 수 없다. 사용자 U_i와 서버와의 비밀키를 MPK_i라 할 때, 탈퇴하는 사용자를 제외한 나머지 각 사용자에게 $E_{MPI_i}(GK')$를 유니캐스트 할 수 밖에 없기 때문에 n_U개의 메시지가 필요하다(n_U는 사용자 수)[6]. 그러므로 사용자의 수가 많고 사용자의 그룹 가입 및 탈퇴가 빈번한 경우에, 보다 효율적인 그룹키 갱신 방법이 필요하다.

Wong 등은 [7]에서 키트리(key-tree)를 이용하여 키 갱신을 $O(\log n_U)$에 수행할 수 있는 LKH(Logical Key Hierarchy)를 제안하였다.

사용자 U_x는 그룹에 가입할 때 안전한 방법으로 서버와 비밀키 K_x를 공유한다. 서버는 사용자들의 비밀키들을 단말 노드로 하는 이진트리를 구성하고, 각 내부노드에 키를 할당한다. 각 사용자는 자신의 단말노드로부터 루트에 이르는 경로상의 키를 소유한다. (그림 2)는 8명의 사용자에 대한 키트리이다. (그림 2)에서 사용자 U_3은 K_3, $K_{3,4}$, $K_{1,4}$, $K_{1,8}$을 소유한다. 루트키는 모든 사용자가 공유하므로 그룹키가 된다. 키트리의 균형이 잡혀 있을 경우, 키트리의 높이는 약 $\log_2 n_U$이고 노드수는 약 $2\times n_U$개이다. 따라서 모든 사용자는 약 $\log_2 n_U$개의 키를 보유하며, 서버는 약 $2\times n_U$개의 키를 보유한다.



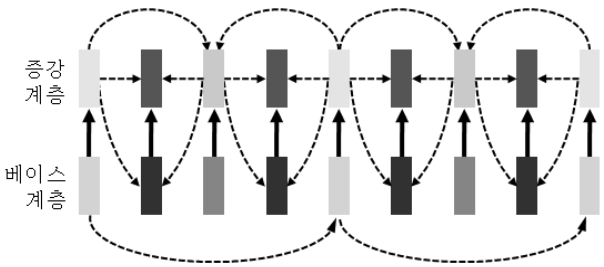
(그림 2) 키트리를 이용한 키관리 구조

사용자가 탈퇴할 경우에, 서버는 그 사용자가 소유하던 키들을 모두 랜덤값으로 갱신하고 갱신된 키는 그 자식 노드키들로 암호화하여 브로드캐스트한다. (그림 2)에서 U_3가 탈퇴한다면 K_3와 $K_{3,4}$는 폐기하고, $K_{1,4}$와 $K_{1,8}$을 $K'_{1,4}$와 $K'_{1,8}$로 갱신하여 메시지 $\{E_{K_{1,2}}(K'_{1,4}),\ E_{K_4}(K'_{1,4}),\ E_{K'_{1,4}}(K'_{1,8}),\ E_{K_{5,8}}(K'_{1,8})\}$을 브로드캐스트한다. 이 메시지를 수신한 사용자들은 자신이 소유한 키로 암호화된 메시지들을 차례로 복호화함으로써 갱신된 키(그룹키 포함)들을 얻을 수 있다. 그러므로 전송되는 키의 수는 $2\times\log_2 n_U$개이며, 각 사용자는 이 메시지로부터최대 $\log_2 n_U$개, 평균 2개의 키를 갱신한다. 사용자가 가입하는 경우에도 사용자가 소유하게 되는 키들을 갱신한다. 갱신된 각 키는 이전 키로 암호화하여 브로드캐스트된다. 따라서 사용자의 가입에 따른 그룹키의 갱신을 위해서는 $\log_2 n_U$개 키가 갱신되고 전송된다. 각 사용자는 이 메시지로부터 최대 $\log_2 n_U$개, 평균 2개의 키를 갱신한다.

2.3 H.264/SVC

H.264/SVC는 다양한 네트워크 환경 및 사용자 단말에 적합하도록 다양한 이미지 크기(spatial), 프레임 수(temporal)와 화질(quality)을 선택적으로 사용할 수 있는 구조를 제공한다[3-5]. 이 세 가지 속성의 다양성을 제공하기 위하여 이미지에 대한 비트 스트림을 계층적으로 구성한다. (그림 3)에서와 같이 베이스 계층(base layer)은 기본적인 해상도와 가장 낮은 품질의 동영상에 대한 비트 스트림이다. 증강 계층(enhancement layer)은 베이스 계층의 해상도와 품질을 높이는데 필요한 데이터로 구성된다. 증강 계층은 여러 개일 수 있다. 각 증강 계층은 하위 증강 계층이나 베이스 계층의 데이터를 참조한다. 이러한 구조로 인코딩하여 콘텐츠를 전송하면, 수신자는 수신환경이나 단말기 성능에 따라 적절한 계층까지만 선별적으로 디코딩한다.

각 계층은 서로 다른 종류의 슬라이스들((그림 3)에서 서로 다른 색의 사각형)로 구성되는데, 이들 슬라이스들이 상호 참조함으로써 동적 이미지를 표현한다. 이 상호 참조의 정도를 조정함으로써 프레임 수를 조정할 수 있다[4].

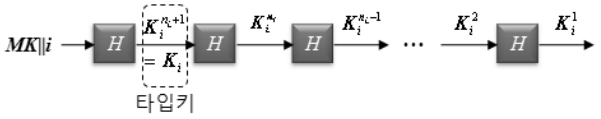


(그림 3) H.264/SVC의 계층 구조

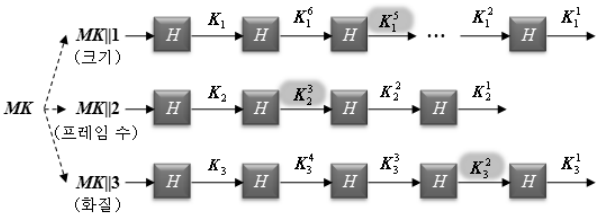
2.4 계층적 접근제어 기법

[1]에서 Bin 등은 H.264/SVC에 대한 사용자의 접근제한 기법을 제안하였다. 이 방법의 핵심 아이디어는 각 계층별로 키를 할당하여 이 계층키로 대응되는 계층의 데이터를 암호화하고, 사용자는 자신이 정당하게 수신할 수 있는 계층에 대한 키만 소유하도록 하는 것이다. 어떤 계층의 품질을 수신하고자 하는 사용자는 그 계층의 모든 하위 계층 정보도 수신할 수 있어야 영상을 복원할 수 있다는 점에 착안하여, 사용자는 자신이 수신할 수 있는 최상위 계층에 대한 키만 서버로부터 수신하도록 하고 하위 계층의 키는 상위 계층으로부터 유도하도록 설계하였다. 하나의 콘텐츠에는 하나의 마스터 키 MK가 대응된다. (그림 4)에서와 같이 MK에 속성 번호 i를 접합하여 해시함수 $H()$를 적용함으로써 각 속성에 대한 타입키 K_i를 생성한다. 이 타입키로부터 해당 속성의 모든 계층키를 유도해낸다. i번째 속성의 j번째 계층키(K_i^j)는 $K_i^j = H(K_i^{j+1}) = H^{n_i+1-j}(K_i)$로 정의된다. 단, n_i는 i번째 속성의 최대 계층 수이며 $K_i = K_i^{n+1}$이다.

(그림 5)는 3개의 속성에 대해 서로 다른 계층을 선택한 예를 보여준다. 화면 크기, 프레임수, 화질의 속성 번호를 각각 1, 2, 3이라 하고, 최대 계층의 수를 각각 6, 3, 4라 할 때 계층키는 (그림 5)와 같이 생성된다. 만약 사용자가 화면 크기 5, 프레임 수는 3, 화질은 2 계층까지 수신하도록 선택하였다면, 서버는 사용자에게 K_1^5, K_2^3, K_3^2를 전송한다. 해시함수의 특성상 상위 계층 키로부터 하위 계층 키는 계산할 수 있지만 하위 계층 키로부터 상위 계층 키는 알아낼 수 없기 때문에, 사용자는 이 키를 이용하여 하위 계층의 키들만 유도할 수 있다. 따라서 사용자는 수신한 영상에 대하여 1~5 계층까지 화면 크기를 복호할 수 있고, 프레임 수는 1~3 계층까지 복호할 수 있으며 화질은 1~2 계층까지 복호할 수 있다.



(그림 4) H.264/SVC용 계층키 생성 방법



(그림 5) 계층키 선택 예

2.5 누적형 키분배 기법

[8]에서 저자들은 H.264/SVC을 사용할 경우 하위 계층만을 필요로 하는 모든 사용자들도 모든 계층을 수신해야 하기 때문에 대역폭을 낭비한다고 지적하고, 수신자가 필요한 계층만을 추출하여 전송하는 방법을 제안하였다. 각 계층에서 서로 다른 키를 사용하기 위해, 서버는 각 속성의 각 계층마다 키트리 $T_i^j (i \leq n_A,\ 1 \leq j \leq n_i,\ n_A$는 속성 수, n_i는 속성 i의 최대 계층 수)를 유지한다. T_i^j의 루트키를 속성 i의 j번째 계층키 K_i^j로 사용한다. 사용자가 i번째 속성의 j계층의 품질을 원한다면 $T_i^k (1 \leq k \leq j)$의 키트리들에 삽입되어야 한다. n_A개의 속성에 대해 서버가 유지해야 하는 키트리의 수는 $n_T = \sum_{i=1}^{n_A} n_i$ 이다. 그러므로 신규 사용자가 가입해야 하는 키트리의 수는 최대 n_T개로서, 사용자가 모든 속성에 대해 최대 계층의 수신을 선택할 때 발생한다. 또한 모든 사용자가 최고의 계층을 선택한다면 모든 키트리의 단말노드수가 n_U이므로, 사용자의 가입과 탈퇴에 따라 갱신되어야 하는 키의 수는 $n_T \times \log_2 n_U$개다. 만약 사용자가 어떤 속성에 대해 k계층에서 $k+d$계층으로 높이고자 한다면, d개의 트리 $T_{k+1}, ..., T_{k+d}$에 가입해야 한다. 만약 k계층에서 $k-d$계층으로 낮추고자 한다면, 역시 d개의 트리 $T_k, ..., T_{k-d+1}$에서 탈퇴해야 한다.

3. 기존 연구에의 CAS 적용

본 장에서는 [1]에서 제안한 계층적 접근제어 기법과 [8]에서 제안한 누적형 키 분배 기법을 CAS 구조에 적용하기 위한 방법을 기술하고 효율성을 분석한다. 이 두 가지 기법뿐 아니라 다음 장에서 제안할 기법은 다양한 사용자 환경을 지원하기 위한 H.264/SVC의 계층적 구조를 이용한다. 이 기법들은 공통적으로 계층마다 다른 키로 계층 데이터를 암호화(스크램블)한다. 즉, 각 계층키는 해당 계층 데이터에 대한 CW이다. 사용자들은 지불한 요금이나 환경에 따라 일부 계층 데이터만을 복호화(디스크램블)할 수 있도록 해당 계층키만을 얻을 수 있도록 하는 것이다. 각 방법들은 이 계층키들을 생성하는 방법과 사용자가의 가입과 탈퇴 등과 같은 이벤트에 대하여 키를 관리하는 방법을 각각 다르게 제시한다. 이 장에서는 기존의 방법들을 CAS 구조에 적용했을 때, 각 이벤트에 대한 키 관리 방법과 관리 메시지인 ECM과 EMM을 정의하고자 한다.

3.1 계층적 접근제어 기법에의 적용

이 방법을 CAS에 적용한다면 서버는 각 사용자와의 비밀키 MPK_i와 MK를 유지해야 하므로 n_U+1개의 키를 유지해야 한다. 사용자는 각 속성별로 수신할 최고 계층키와 MPK_i를 소유해야 하므로 n_A+1개의 키를 소유한다.

이 기법에서는 CAS에서 키관리를 위해 사용자들이 공유하는 그룹키인 AK와 같은 별도의 키를 사용하지 않는다.

AK를 도입한다고 하더라도, 모든 사용자가 하나의 CW를 공유하는 것이 아니라 서로 다른 계층키 부분 집합을 소유해야 하므로 계층키 부분 집합들을 하나의 AK로 암호화해서 전송할 수 없다. 한편 계층키는 그 계층에 속한 사용자들이 공유하고 있으므로 해당 계층 사용자들의 그룹키처럼 사용하여 갱신된 계층키를 암호화하는데 사용할 수 있다. 즉, 계층키는 CAS에서의 AK의 역할도 해야 한다. 따라서, 이를 CAS 구조에 적용하려면 CW를 갱신하는 ECM과 AK를 갱신하기 위한 EMM이 재구성 되어야 한다.

3.1.1 주기적인 CW 갱신

주기적인 CW의 갱신 대신에 모든 계층키를 갱신하여야 하므로 서버는 새로운 마스터키 MK'을 생성하고, 해시함수를 이용하여 K'^{j}_{i}를 생성해야 한다. 새로운 계층키는 기존의 계층키를 아는 사용자만이 수신할 수 있어야 하므로, 각 계층키는 대응되는 기존의 계층키로 암호화하여 전송한다. 그러므로 ECM은 다음과 같이 구성된다. 여기서 $S(\bullet)$는 메시지 $\bullet$에 대한 서버의 서명이다.

$$ECM = M \| S(M),$$
$$M = \{E_{K_i^j}(K'^{j}_{i}) \mid 1 \leq i \leq n_A, 1 \leq j \leq n_i\}$$

이를 수신한 사용자들은 자신이 소유한 계층키를 이용하여 새로운 계층키를 복호화하여 얻고, 이를 이용하여 하위 계층의 계층키를 계산한 후, 이 키들을 이용하여 콘텐츠를 복호화할 수 있다. 그러므로 이를 위해 전송되어야 하는 키의 수는 $\sum_{i=1}^{n_A} n_i$개다.

각 사용자가 갱신해야 하는 키의 수는 각 속성별로 하나의 계층키만 복호화하고 이하의 계층키를 유도해 내면 되므로, n_A개의 키를 복호화한다.

3.1.2 사용자 변동에 따른 키 갱신

AK의 갱신은 사용자의 가입이나 탈퇴가 발생했을 때 변경한다. 사용자 U_x가 가입하면 그 사용자가 선택한 계층 이하의 계층키를 변경해야 한다. 그러나 계층키는 MK로부터 해시함수로 체인을 이루면서 생성되기 때문에 일부 계층키만을 변경할 수 없다. 따라서 3.1.1절에서와 같이 MK를 갱신하여 모든 계층키들을 갱신하고 브로드캐스트한다. 가입자에게는 선택한 계층의 갱신된 계층키들을 MPK_x로 암호화하여 전송한다. 그러므로 CW 갱신과 마찬가지로 브로드캐스트되는 키의 개수는 $\sum_{i=1}^{n_A} n_i$이며 기존 사용자가 갱신해야 하는 키의 수는 n_A개다.

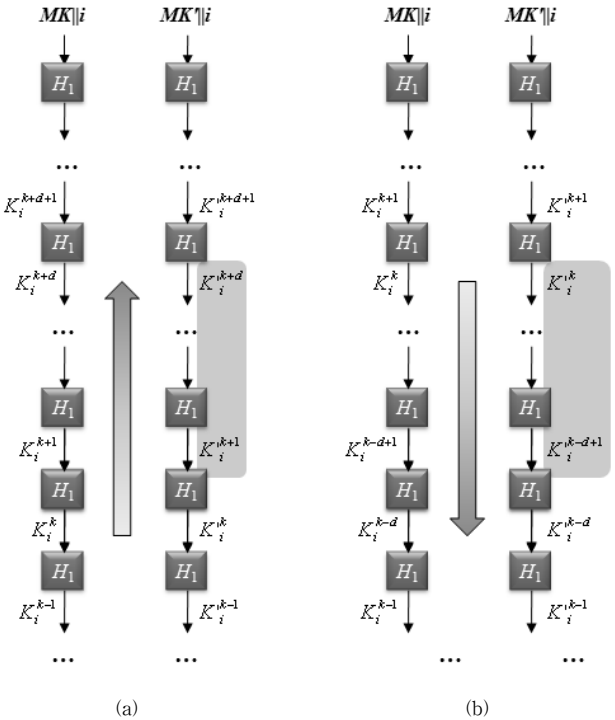
사용자가 탈퇴할 경우에도 모든 계층키를 갱신하여야 한다. 그러나 가입의 경우와는 달리, 갱신된 키를 (탈퇴자가 알고 있는) 기존의 계층키로 암호화하여 전송할 수 없기 때문에, 서버는 각 사용자의 비밀키 MPK_i로 암호화하여 EMM을 통해 전송한다. 그러므로 각 사용자 U_i에게 필요한 n_A개의 갱신된 계층키를 MPK_i로 암호화하여 전송한다. U_i가 각 속성에 대해 선택한 계층이 각각 $j_1, ..., j_{n_A}$라면, U_i에 대한 EMM은 다음과 같이 구성된다.

$$EMM = M \| S(M), \quad M = E_{MPK_i}(K'^{j_1}_{1}, ..., K'^{j_{n_A}}_{n_A})$$

그러므로 전송되는 키의 개수는 $n_A \times n_U$개이며, 각 사용자는 이 메시지로부터 n_A개 키를 복호화한다.

3.1.3 수신 계층 변동에 따른 키 갱신

사용자 U_x가 (그림 6) (a)에서와 같이 i번째 속성에 대해 k계층에서 $k+d$계층으로 높이고자 했을 때나, (그림 6) (b)에서와 같이 k계층에서 $k-d$계층으로 낮추고자 했을 때, 변화되는 계층키(그림에서 음영으로 표시된 키들)를 변경해야 한다. 그러므로 사용자의 가입과 탈퇴 때와 마찬가지로 MK를 갱신하여 모든 속성의 모든 계층키를 갱신하여야 한다. 계층을 향상시킬 때는 갱신된 계층키를 대응되는 기존의 계층키로 암호화해서 브로드캐스트한다. 그러므로 갱신 및 전송되어야 하는 키의 개수는 사용자의 가입 때와 같이 $\sum_{i=1}^{n_A} n_i$개다. 계층을 낮추는 경우에는, U_x가 음영 표시된 키들을 알 수 없어야 하므로 다른 모든 사용자들에게 각자의



(그림 6) 계층 변화에 따른 키의 갱신

비밀키 MPK_i로 암호화하여 전송해야 하고, 나머지 키들은 대응되는 기존의 키로 암호화하여 전송하면 된다. 따라서 갱신되는 키의 수는 $\sum_{i=1}^{n_A} n_i$이고, MPK_i로 암호화되어 전송되는 키는 최대 해당 계층의 수이기 때문에 사용자의 탈퇴 때와 마찬가지로 최대 $n_A \times n_U$개이며, 각 사용자는 n_A개의 계층키를 복호화한다.

3.2 누적형 키분배 기법에의 적용

누적형 기법을 CAS에 적용한다면, 서버는 $\sum_{i=1}^{n_A} n_i$개의 키 트리를 유지하는데 각 키트리는 최대 $2 \times n_U$개의 노드키를 가지므로 $2 \times n_U \times \sum_{i=1}^{n_A} n_i$개의 키를 유지해야 한다. 사용자가 속한 키트리의 단말노드들은 사용자와 서버만의 공유키이므로 CAS에서의 MPK 역할을 한다. 그러므로 각 사용자는 가입한 키트리 수 만큼의 MPK를 소유하는 것과 같다. 여기서도 계층적 기법을 적용했을 때와 같이 각 키트리의 루트키인 계층키가 CW의 역할을 하며, 갱신된 루트키는 기존의 루트키로 암호화되어 전송되므로 CAS의 AK의 기능을 수행해야 한다. 다른 점은 각 계층키들은 서로 연관성 없이 독립적으로 생성된다는 것이며, 계층키를 갱신하기 위한 보조키(키트리의 내부 노드키)들을 가지고 있다는 점이다. 이 방법을 CAS 구조에 적용했을 때에도 ECM과 EMM이 재구성되어야 할 것이다.

3.2.1 주기적인 CW 갱신

이 방법에서도 CAS에서의 주기적인 CW 갱신은 모든 계층키들의 갱신에 해당한다. 누적형 기법에서는 동일한 계층을 선택한 그룹별로 암호화된 계층별 콘텐츠를 전송한다. 이 그룹의 수는 선택한 계층의 조합의 수와 같으므로 $n_G = \prod_{i=1}^{n_A} n_i$개다. 그러나 각 그룹마다 필요한 계층키 집합을 보낸다면 매우 많은 키를 보내야 할 것이다. 따라서 아래와 같이 각 계층키인 키트리의 루트키들을 갱신하고, 대응되는 이전의 계층키로 암호화하여 브로드캐스트하는 것이 효율적일 것이다.

$$ECM = M \| S(M)$$
$$M = \{E_{K_i^j}(K'^j_i) \mid 1 \le i \le n_A, 1 \le j \le n_i\}$$

이를 수신한 사용자들은 자신이 소유한 계층키를 이용하여 새로운 계층키를 복호화하여 얻을 수 있다. 그러므로 이를 위해 전송되어야 하는 키의 수는 $\sum_{i=1}^{n_A} n_i$개이며, 각 수신자가 복호화해야 하는 키의 개수도 역시 최대 $\sum_{i=1}^{n_A} n_i$개다.

3.2.2 사용자 변동에 따른 갱신

사용자 U_x가 가입하면 선택한 계층 이하의 계층키를 변경해야 한다. 각 계층키는 독립적으로 생성되므로, 갱신된 키를 이전의 계층키로 암호화하여 브로드캐스트하고, U_x에게는 선택한 계층의 갱신된 계층키들을 전송한다. 최악의 경우, U_x는 각 속성의 최고 계층을 선택할 것이다. 계층키는 키트리의 루트키로서 가입자가 키트리에 가입하면 루트키 뿐 아니라 트리의 경로상에 있는 모든 키가 갱신되어야 하므로, 이 경우에 사용자가 갱신해야 하는 키 및 전송되는 키의 수는 최대 $\log_2 n_U \times \sum_{i=1}^{n_A} n_i$이다. 사용자 U_x가 탈퇴할 경우에도 최악의 경우 모든 키트리를 갱신해야 하므로, $2 \times \log_2 n_U \times \sum_{i=1}^{n_A} n_i$개의 키가 전송되어야 한다. 사용자의 가입과 탈퇴에 따라 다른 사용자들이 갱신해야 하는 키의 수는 최대 $\log_2 n_U \times \sum_{i=1}^{n_A} n_i$개이다(2.2절 참조).

3.2.3 수신 계층 변동에 따른 갱신

사용자 U_x가 i 번째 속성의 수신 계층을 k에서 $k+d$로 높이고자 한다면, U_x를 T_{k+1}부터 T_{k+d}까지의 키트리에 가입시킴으로써 키트리를 갱신하여야 한다. 계층 k에서 $k-d$ 계층으로 낮추고자 한다면 T_{k-d+1}부터 T_k까지의 키트리에서 삭제함으로서 키트리를 갱신하여야 한다. 그러므로 계층을 높이거나 낮춤으로써 갱신되는 키의 수는 최대 $d \times \log_2 n_U$개이고 전송되는 키의 개수는 각각 $d \times \log_2 n_U$와 $2 \times d \times \log_2 n_U$개이며, 사용자들은 최대 $d \times \log_2 n_U$개의 키를 복호화하여 갱신한다.

4. 제안 방법

CAS에서 콘텐츠 보호에 사용되는 CW는 매우 짧은 주기로 갱신되어 사용자들 모두가 공유하는 그룹키인 AK로 암호화되어 브로드캐스트된다. AK는 CW보다는 긴 주기로 갱신되어 사용자의 개인 비밀키인 MPK_i로 각각 암호화되어 사용자에게 전송된다. 따라서 빈번한 콘텐츠 보호키 갱신 메시지를 효율적으로 설계하는 것이 바람직하다. 또한 전송되는 키는 암호화되어 전송되기 때문에, 전송되는 키의 개수는 메시지의 크기 뿐 아니라 서버와 사용자의 암복호화 횟수에도 관련된다. 본 논문에서는 이러한 요구사항에 부합하는 효율적 키관리 기법을 제안하고자 한다.

4.1 제안 기법

제안 기법에서는 계층적 접근제어 기법이나 누적형 접근제어 기법과 마찬가지로 계층키를 사용하여 각 계층의 데이터를 보호한다. 사용자의 가입과 탈퇴에 따른 키 갱신을 위해서 계층별로 키트리를 유지하며, H.264/SVC의 계층간의

연관성을 이용하여 효율적으로 계층키를 갱신하도록 한다. 또한 가장 빈번하게 발생하는 콘텐츠 보호키(CAS에서의 CW)의 갱신을 효율적으로 수행하기 위하여, 서버가 모든 계층키를 갱신하여 전송하는 것이 아니라 하나의 SV(Seed Value)만을 갱신하여 전송한다. 이를 수신한 각 사용자는 새로운 SV와 소유하고 있던 계층키를 이용하여 새로운 계층키를 생성한다.

4.1.1 키 생성 방법

서버는 각 속성 i의 각 계층 j마다 사용자 관리를 위한 키트리 T_i^j를 유지한다. T_i^j의 루트키 K_i^j는 해당 계층의 데이터를 수신할 수 있는 사용자들이 공유하는 그룹키이다. 서버는 각 속성별 마스터 키인 $MK_i(1 \leq i \leq n_A)$를 생성하고 이 마스터키로부터 해시함수 $H_1()$를 이용하여 식 (1)과 같이 키트리의 루트키들을 생성한다. 키트리의 나머지 키들은 랜덤값이다.

$$K_i^j = H_1(K_i^{j+1}) = H_1^{n_i+1-j}(MK_i) \qquad \text{식 (1)}$$

i번째 속성의 j번째 계층 데이터를 보호하는 데는 계층키 LK_i^j가 사용되며 해시함수 $H_2()$를 이용하여 식 (2)와 같이 생성된다.

$$LK_i^j = H_2(SV \| K_i^j) \quad (1 \leq i \leq n_A, 1 \leq j \leq n_i) \quad \text{식 (2)}$$

이 키들간의 관계를 도식화하면 (그림 7)과 같다.

각 사용자는 각 속성마다 수신할 최상위 계층의 트리에 가입하여 약 $n_A \times \log_2 n_U$개의 키를 소유한다. 소유한 키트리의 루트키 K_i^j로부터 식 (1)을 이용하여 하위 계층에 대한 키트리의 루트키 $K_i^k(1 \leq k < j)$를 유도해 낸다.

키트리의 개수는 $\sum_{i=1}^{n_A} n_i$이다. 사용자는 각 속성마다 하나의 키트리에만 가입하므로 i번째 속성에 대한 키트리들의 모든 단말 노드 수의 합은 n_U이다. 따라서 키트리들의 노드 수의 합은 $2 \times n_U$이며, 서버가 유지하는 키의 수는 $2 \times n_A \times n_U$개다.

키트리의 루트키들은 기존의 CW를 대신하여 컨텐츠 암호화에 사용되는 계층키 생성에 사용되며, 단말 노드키들은 서버와 사용자와의 공유키므로 기존의 MPK_i를 대신한다. 나머지 키들은 갱신된 계층키들을 효율적으로 암호화하여 전송하기 위해 사용되는 키들로서 기존의 AK 역할을 하게 된다.

4.1.2 주기적인 CW 갱신

CW 대신 다음과 같이 SV를 갱신한다.

단계 1: 서버는 SV를 SV'로 갱신하고, 서버의 서명을 추가하여 ECM을 구성한 다음 사용자들에게 브로드캐스트한다.

$$ECM = SV' \| S(SV')$$

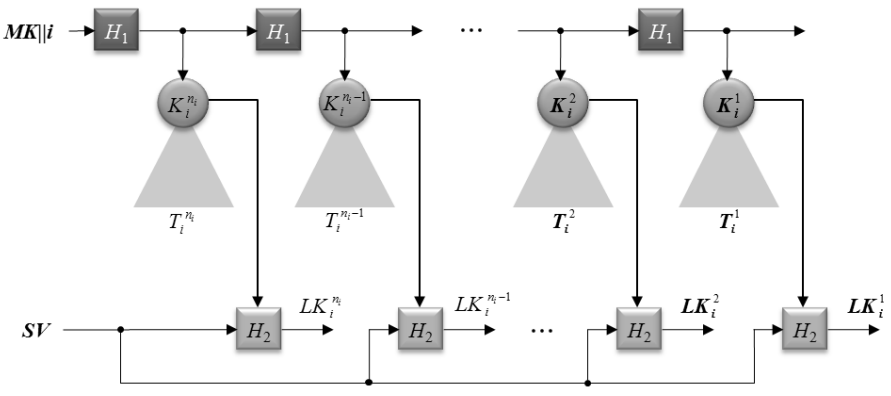
단계 2: ECM을 수신한 사용자는 서명을 검증한 후, SV'와 자신이 이미 알고 있는 루트키들을 이용하여 새로운 계층키들을 계산한다(수식 (1), (2) 참조).

4.1.3 사용자의 변화에 따른 키 갱신

사용자 U_x가 가입하면 서버와 기존의 사용자들은 다음과 같은 과정을 수행한다.

단계 1: 서버는 모든 $MK_i(1 \leq i \leq n_A)$들을 갱신하고 식 (1)을 이용하여 모든 키트리의 루트키를 갱신한다.

단계 2: U_x가 가입해야 하는 키트리 집합을 $T_{new} = \{T_i^{j_i} | 1 \leq i \leq n_A, 1 \leq j_i \leq n_i\}$라 하면, 서버는 이 키트리를 LKH 갱신 방법에 의하여 갱신한다. 단, 루트키는 단계 1에서 생성한 키이다.



(그림 7) 제안 키관리 구조

단계 3: 서버는 T_{new}의 갱신된 키들을 LKH 방식으로 암호화한 메시지 M_1을 생성한다. 이 갱신된 키들은 이전의 키트리에 속한 키들로 암호화되기 때문에 T_{new}에 속한 멤버들만이 복호화할 수 있다. M_1에 포함된 키의 개수는 약 $n_A \times \log_2 n_U$이며, 각 사용자는 M_1으로부터 최대 $n_A \times \log_2 n_U$개의 키를 복호화한다.

단계 4: T_{new}에 속하지 않은 모든 키트리에 대하여, 서버는 다음과 같이 갱신된 루트키 K'^{j}_{i}들을 이전의 루트키 K^{j}_{i}들로 암호화하여 M_2를 생성한다.

$$M_2 = \{E_{K^j_i}(K'^{j}_{i}) \mid T^{j}_{i} \not\subset T_{new}\}$$

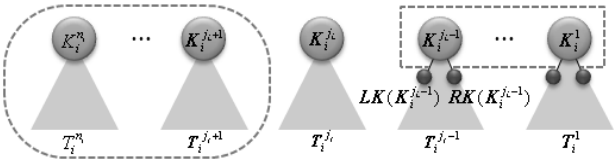
이 키들도 이전의 루트키를 소유한 사용자만이 복호화할 수 있다. 이를 수신한 사용자들은 복호화한 키를 가지고 식 (1)을 이용하여 하위 계층 키트리의 루트키를 생성한다. M_2에 포함된 키의 수는 최대 $\sum_{i=1}^{n_A} n_i - n_A$개며, 각 사용자는 M_2으로부터 n_A개의 키를 복호화한다.

단계 5: 다음과 같이 ECM을 구성하여 사용자들에게 브로드캐스트한다.

$$ECM = M \| S(M), \qquad M = M_1 \| M_2$$

따라서 전송되는 키의 개수는 $(n_A \times \log_2 n_U) + (\sum_{i=1}^{n_A} n_i - n_A)$이며, 사용자는 가입한 계층에 따라 M_1이나 M_2만 필요하므로 최대 $n_A \times \log_2 n_U$개의 키를 복호화한다.

사용자 U_x가 탈퇴할 경우에도 유사한 방법으로 갱신된다. 단, 사용자가 가입해 있던 $T^{j_i}_{i}$ 보다 하위에 있는 키트리에서는 갱신 메시지 구성이 다르다. (그림 8)에서 왼쪽 타원으로 표시된 상위 계층 키트리들은 U_x가 알 수 없는 키들로 구성된다. 하지만 오른쪽 하위 계층 키트리들의 루트키들은 U_x가 알고 있기 때문에, 갱신된 루트키를 이전 루트키로 암호화하여 전송할 수 없다. 따라서 U_x가 알 수 없는 이 루트키들의 자식 노드키들로 암호화하여 전송해야 한다. 처리 절차는 다음과 같다.



(그림 8) 키트리 분류

단계 1: 서버는 모든 $MK_i (1 \leq i \leq n_A)$와 키트리의 루트키를 갱신한다.

단계 2: U_x가 탈퇴해야 하는 키트리 집합을 $T_{old} = \{T^{j_i}_{i} \mid 1 \leq i \leq n_A, 1 \leq j_i \leq n_i\}$라 하면, 서버는 이 키트리들을 LKH 갱신 방법에 의하여 갱신한다. 단, 루트키는 단계 1에서 생성한 키이다.

단계 3: 서버는 T_{old}의 갱신된 키들을 LKH 방식으로 암호화한 메시지 M_1을 생성한다. M_1에 포함된 키의 개수는 약 $2 \times n_A \times \log_2 n_U$이며, 사용자는 M_1으로부터 최대 $n_A \times \log_2 n_U$개의 키를 복호화한다.

단계 4: T_{old}보다 높은 계층의 모든 키트리에 대하여, 서버는 다음과 같이 갱신된 루트키 K'^{j}_{i}들을 이전의 루트키 K^{j}_{i}들로 암호화하여 M_2를 생성한다.

$$M_2 = \{E_{K^k_i}(K'^{k}_{i}) \mid k > j_i\}$$

단계 5: T_{old}보다 낮은 계층의 모든 키트리에 대하여, 서버는 다음과 같이 갱신된 루트키 K'^{j}_{i}들을 두 자식 노드키 $LK(K'^{j}_{i})$와 $RK(K'^{j}_{i})$로 암호화하여 M_3를 생성한다.

$$M_3 = \{E_{LK(K^k_i)}(K'^{k}_{i}), E_{RK(K^k_i)}(K'^{k}_{i}) \mid k < j_i\}$$

M_2와 M_3에 포함된 키의 수는 최대 $2 \times \sum_{i=1}^{n_A} n_i - n_A$개이며, 사용자는 M_2 혹은 M_3로부터 n_A개의 키를 복호화한다.

단계 6: 다음과 같이 ECM을 구성하여 사용자들에게 브로드캐스트한다.

$$ECM = M \| S(M), \qquad M = M_1 \| M_2 \| M_3$$

따라서 전송되는 키의 개수는 $(2 \times n_A \times \log_2 n_U) + (2 \times \sum_{i=1}^{n_A} n_i - n_A)$이, 사용자는 가입한 계층에 따라 M_1, M_2 혹은 M_3만 필요하므로 최대 $n_A \times \log_2 n_U$개의 키를 복호화한다.

4.1.4 수신 계층 변화에 따른 키 갱신

사용자 U_x가 속성 i의 수신 계층을 k계층에서 $k+d$계층으로 높이는 경우에는 다음과 같은 과정이 수행된다.

단계 1: 서버는 MK_i와 루트키들을 갱신한다.

단계 2: U_x를 T^{k}_{i}로부터 탈퇴시키고 T^{k+d}_{i}에 가입시킴으로써 두 키트리를 갱신한다. 단, 루트키는 단계 1에서 생성한 키이다.

단계 3: 서버는 T'^{k}_{i}와 T'^{k+d}_{i}의 갱신된 키들을 LKH 방식으로 암호화하여 메시지 M_1을 만든다. M_1에 속한 키의 개수는 최대 $3 \times \log_2 n_U$이며, 사용자는 M_1으로부터 최대 $\log_2 n_U$개의 키를 복호화한다.

단계 4: 키트리 $T'^{j}_{i} (j \neq k \,\&\, j \neq k+d)$들에 대하여, 서버는 다음과 같이 갱신된 루트키 K'^{j}_{i}들을 이전의 루트키 K^{j}_{i}들로 암호화하여 메시지 M_2를 생성한다. M_2에 속한 키의 수는 $n_i - 2$개이며, 사용자는 M_2으로부터 1개의 키를 복호화한다.

$$M_2 = \{E_{K^j_i}(K'^{j}_{i}) \mid j \neq k \wedge j \neq k+d\}$$

단계 5: 다음과 같이 ECM을 구성하여 사용자들에게 브로드캐스트한다.

$$ECM = M \| S(M), \qquad M = M_1 \| M_2$$

따라서, 계층을 높일 때 전송되는 키의 수는 $(3 \times \log_2 n_U) + (n_i - 2)$개이며, 사용자는 최대 $\log_2 n_U$개의 키를 복호화한다.

사용자가 U_x가 속성 i의 수신 계층을 k계층에서 $k-d$계층으로 낮추는 경우에도 유사한 방법으로 갱신된다. 단, 사용자의 탈퇴에서와 마찬가지로 $K^{k-1}_{i} \sim K^{k-d}_{i}$는 U_x가 알고 있기 때문에 갱신 후 이 키로 암호화해서 전송할 수 없다. 따라서 이 키들은 갱신한 후, 자식 노드키들로 암호화해서 전송해야 한다.

단계 1: 서버는 MK_i와 키트리들의 루트키를 갱신한다.

단계 2: U_x를 T_i^k로부터 탈퇴시키고 T_i^{k-d}에 가입시킴으로써 두 키트리를 갱신한다. 단, 루트키는 단계 1에서 생성한 키이다.

단계 3: 서버는 T'^k_i와 T'^{k-d}_i의 갱신된 키들을 LKH 방식으로 암호화하여 메시지 M_1을 만든다. M_1에 포함된 키의 개수는 최대 $3\times\log_2 n_U$이며, 사용자는 M_1으로부터 최대 $\log_2 n_U$개의 키를 복호화한다.

단계 4: 키트리 $T'^j_i(j>k\vee j<k-d)$들에 대하여, 서버는 다음과 같이 갱신된 루트키 K'^j_i들을 이전의 루트키 K_i^j들로 암호화하여 메시지 M_2를 생성한다. M_2에 속한 키의 수는 n_i-d-1개이다.

$$M_2=\{E_{K_i^j}(K'^j_i)\mid j>k\vee j<k-d\}$$

단계 5: 키트리 $T'^j_i(j>k-d\wedge j<k)$들에 대하여, 서버는 다음과 같이 갱신된 루트키 K'^j_i들 을 자식 노드키들 두 자식 노드키 $LK(K'^j_i)$와 $RK(K'^j_i)$로 암호화하여 M_3를 생성한다. M_3에 속한 키의 수는 $2\times(d-1)$개다. 각 사용자는 M_2나 M_3로부터 1개의 키를 복호화한다.

$$M_3=\{E_{LK(K_i^k)}(K'^k_i),E_{RK(K_i^k)}(K'^k_i)\mid k-d<j<k\}$$

단계 6: 다음과 같이 ECM을 구성하여 사용자들에게 브로드캐스트한다.

$$ECM=M\|S(M),\qquad M=M_1\|M_2\|M_3$$

따라서, 계층을 낮출 때 전송되는 키의 수는 $(3\times\log_2 n_U)+(n_i-d-1+2\times(d-1))$개이며, 사용자는 최대 $\log_2 n_U$의 키를 복호화한다.

4.2 안전성 분석

제안 방법의 안전성은 3가지 측면에서 분석되어야 한다. (1) 정당한 사용자만이 가입한 품질의 콘텐츠를 수신하는가? (2) 신규 사용자가 이전의 콘텐츠를 수신할 수 있는가? 혹은 수신 품질을 높인 사용자가, 높이기 이전에 수신한 콘텐츠에 대해 높은 품질의 콘텐츠를 얻어낼 수 있는가? (3) 탈퇴한 사용자가 탈퇴한 이후에 콘텐츠를 수신할 수 있는가? 혹은 수신 품질을 낮춘 사용자가, 낮춘 후에도 높은 품질의 콘텐츠를 수신할 수 있는가?

첫 번째 측면은 수신제어의 기본적인 목적이고, 2번째 측면은 후방향 안전성이며, 3번째 측면은 전방향 안전성을 의미한다. 응용에 따라 효율성을 위해 후방향 및 전방향 안전성을 지원하지 않거나, 사용자 변동과 상관없이 주기적으로만 키를 갱신할 수도 있다. 주기적으로만 갱신할 경우, 후방향과 전방향 안전성을 위한 키 갱신이 주기적 키 갱신 시점까지 지연되는 것이다. 주기적인 키 갱신을 사용할 경우에는 안전성과 효율성을 고려하여 적절한 주기를 선택해야 한다. 이 주기는 일정 시간 간격이 될 수도 있고, 새로운 콘텐츠마다 변경하도록 할 수도 있을 것이다. 사용자의 변동이 있을 때마다 키 갱신을 수행할 경우, 하나의 콘텐츠 전송 중간에 키가 바뀔 수 있게 된다. 이 경우에는 이후의 컨텐츠를 갱신된 키로 스크램블해야 할 것이다. 이러한 사항은 서비스 제공자의 보안 정책에 따라 달라질 수 있으므로, 본 절에서는 안전성을 위하여 사용자 변동 시마다 키 갱신이 이루어진다고 가정하고 제안 기법의 안전성을 분석할 것이다.

4.2.1 수신 제어

콘텐츠의 각 계층 데이터는 계층키 LK_i^j로 암호화된다. 이 계층키는 주기적으로 전송되는 SV와 K_i^j를 해시하여 계산된다. 비록 SV가 평문으로 전송되지만 서버의 서명이 있으므로 무결성이 보장된다. 그러므로 정당한 사용자만이 K_i^j를 소유한다면 정당한 사용자만이 계층키를 알아낼 수 있다.

사용자가 가입한 최고 계층의 K_i^j는 가입시 안전한 방법(LKH)에 의해 소유한다. 또한 이 키로부터 하위 계층의 키를 유도할 수 있다. 그러나 해시함수의 특성상 상위 계층의 키는 유도할 수 없으므로, 정당한 사용자만이 가입한 계층의 계층키를 알 수 있다.

4.2.2 후방향 안전성

신규 사용자 U_x는 각 속성 i에 대하여, 수신하고자 하는 최고 계층의 키트리 $T_i^{j_i}$에만 가입한다. U_x가 소유하게 되는 $T_i^{j_i}$는 LKH 방식에 의해 갱신된다. 이 방법에 의하면 U_x는 가입한 키트리의 어떤 이전키도 알 수 없다[7]. U_x는 가입한 키트리의 루트 $K_i^{j_i}$로부터 이하 계층의 모든 루트키 K_i^k $(1\le k\le j_i)$를 유도할 수 있다. 그러나 이 유도된 키는 이전의 루트키와는 무관하다. 따라서 U_x 가입 이전의 어떤 키도 알 수 없다.

사용자 U_x가 수신계층을 k에서 $k+d$로 높이는 경우, $1\sim k$ 계층, $(k+1)\sim(k+d)$ 계층, $(k+d+1)\sim n_i$ 계층으로 나누어 생각할 수 있다. 모든 루트키가 갱신되고 U_x는 T_i^{k+d}에 가입하여 갱신된 K'^{k+d}_i을 알게 된다. 이로부터 상위 계층인 $(k+d+1)\sim n_i$ 계층의 어떠한 갱신된 키 혹은 이전 키를 알 수 없다. 하위 계층인 $1\sim k$ 계층에 대해서는 계층 변경과 이전이나 이후에 접근 권한이 있으므로, 문제가 되지 않는다. 마지막으로 $(k+1)\sim(k+d)$ 계층에 대해 이전 키를 얻을 수 있는가 하는 문제가 남는다. U_x 는 T_i^{k+d}에서 갱신된 키만을 알 수 있으며, 이 키로부터 유도된 $(k+1)\sim(k+d-1)$ 계층의 키도 이전의 키와는 연관성이 없다.

따라서 사용자가 가입하거나 수신 계층을 높이더라도, 이전의 키를 알 수 없으므로 후방향 안전성이 보장된다.

4.2.3 전방향 안전성

사용자 U_x가 탈퇴하면, 탈퇴 후에 나머지 사용자가 소유하는 어떠한 키도 알 수 없어야 한다. U_x가 가입되어 있던 키트리 $T_i^{j_i}$로부터 탈퇴하고 키트리를 갱신함으로써 $T'^{j_i}_i$에는 더 이상 U_x가 아는 키가 존재하지 않는다[7]. 이제 고려해야 하는 키는 j_i 하위 계층 키트리의 루트키들 K_i^k

$(1 \leq k < j_i)$의 키들이다. 이 키들은 갱신되어 자식 노드키들로 암호화 전송된다. U_x는 자식 노드키들을 알고 있지 않으므로 이들 키로 암호화된 K'^{k}_{i}를 알 수 없다.

사용자 U_x가 k 계층으로부터 $k-d$ 계층으로 낮출 때는 T_i^k에서 탈퇴하고 T_i^{k-d}에 가입하므로, U_x가 알고 있던 $T_i^k \sim T_i^{k-d}$의 키들이 모두 갱신된다. 갱신된 T_i^k의 키들은 LKH 방식을 이용하므로 안전하다. U_x는 $T_i^{k-1} \sim T_i^{k-d}$의 루트키를 알고 있었으나 갱신된 루트키들이 그 자식 노드키들로 암호화되어 전송되므로, U_x는 이 갱신된 루트키들을 알아낼 수 없다.

따라서 사용자가 탈퇴하거나 수신 계층을 낮추더라도, 이후의 키를 알 수 없으므로 전방향 안전성이 보장된다.

4.3 효율성 비교

이 절에서는 계층적 기법과 누적형 기법을 CAS 구조에 적용했을 때와 제안한 기법에 대하여 최악의 경우에 대한 효율성을 비교한다. 비교 항목은 (1) 서버에 저장되는 키의 개수, (2) 사용자가 보유해야 하는 키의 개수, (3) 주기적인 CW 갱신, (4) 사용자 가입과 탈퇴 및 (5) 수신 계층의 변경에 대하여, 전송되는 암호화된 키의 개수와 사용자가 복호화해야 하는 키의 개수이다. 각 항목에 대한 분석결과는 <표 1>과 같다. 분석에는 상위 계층키로부터 하위 계층키를 유도하기 위한 해시함수의 실행회수는 포함되지 않았다. 해시함수는 가벼운 연산이므로, 네트워크 상에서 전송되는 전송량, 암복호화 수, 그리고 저장해야 하는 키(유도되지 않는 키)들의 수에 대한 비교만 수행하였다. 또한 가입하는 사용자나 계층을 변경하는 사용자에게 전송되는 키들의 수는 나머지 사용자들에게 전송되는 키의 수에 비해 미미하므로, 수식을 간단히 하기 위하여 분석에서 제외시켰다.

속성의 수나 계층의 수는 그 값이 작고 변동의 범위가 작지만, 사용자 수는 매우 큰 값이고 계속 증가할 수 있는 값으로서 효율성에 가장 큰 영향을 미치는 요소이다. 사용자 수의 증가에 따른 효율성을 비교하기 위하여 n_A=3, n_i=5라고 가정하고 n_U=2,000,000~5,000,000으로 변화하면서 각 방법에 대한 효율성을 (그림 9)에서 그래프로 나타내었다. 사용자의 변동과 수신 계층 변화에 따른 효율성은 각 기법별로 차이가 매우 크기 때문에 로그 축척으로 표시하였다.

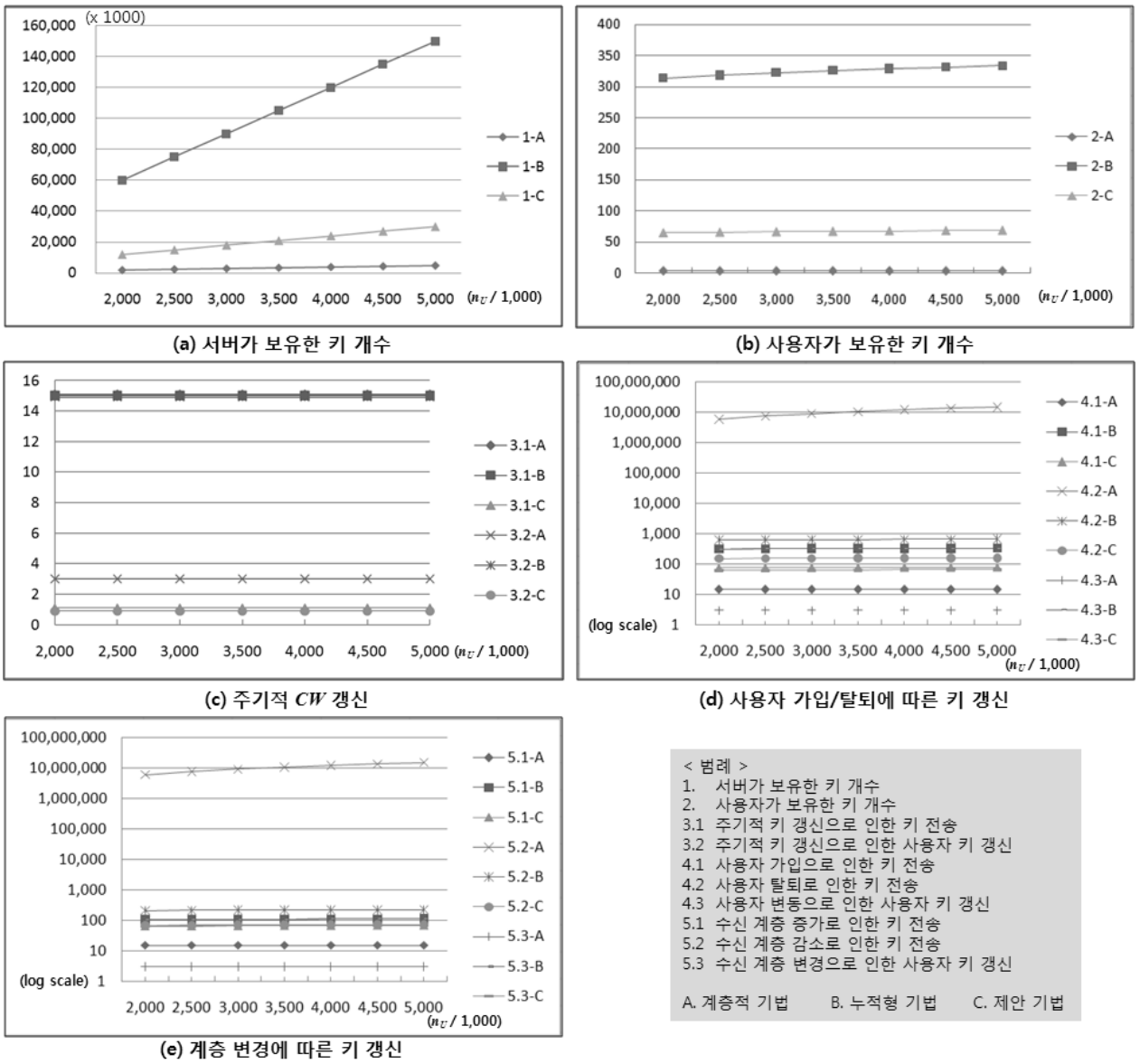
5. 결론 및 향후 연구

본 논문에서는 다양한 네트워크 환경과 사용자 단말기에 적용될 수 있는 H.264/SVC 멀티미디어 콘텐츠 표준을 CAS에 적용하기 위한 키 관리 기법을 제안하였다. 특히 서버로부터 사용자로 네트워크를 통하여 전송되는 키의 개수를 줄이기 위하여 가장 빈번하게 전송되는 메시지의 양을 줄이도록 노력하였다. 전송되는 메시지는 암호화되어 있기 때문에, 전송되는 키의 개수는 전송량뿐 아니라 서버의 암호화 시간과 사용자의 복호화 시간과도 연관이 된다. 제안한 방법은 기존의 방법에 비하여 전송량과 암복호화 수를 감소시켰다.

제안 방법에서 사용한 키트리는 기본적인 LKH를 사용하였으므로, 향후 변형된 LKH 기법들을 적용하기 위한 방법이 연구되어야 할 것이며, 실제 파라미터 값을 적용한 효율성도 분석되어야 할 것이다.

〈표 1〉 최악의 경우에 대한 효율성 비교

			계층적 기법(A)	누적형 기법(B)	제안 기법(C)
서버 보유			n_U+1	$2\times n_U\times\sum_{i=1}^{n_A} n_i$	$2\times n_A\times n_U+n_A+1$
사용자 보유			n_A+1	$\log_2 n_U\times\sum_{i=1}^{n_A} n_i$	$n_A\times\log_2 n_U+2$
주기적 키 갱신	전송		$\sum_{i=1}^{n_A} n_i$	$\sum_{i=1}^{n_A} n_i$	1
	사용자 갱신		n_A	$\sum_{i=1}^{n_A} n_i$	1
사용자 가입/ 탈퇴	전송	가입	$\sum_{i=1}^{n_A} n_i$	$\log_2 n_U\times\sum_{i=1}^{n_A} n_i$	$n_A\times\log_2 n_U+\sum_{i=1}^{n_A} n_i-n_A$
		탈퇴	$n_A\times n_U$	$2\times\log_2 n_U\times\sum_{i=1}^{n_A} n_i$	$2\times n_A\times\log_2 n_U+2\times\sum_{i=1}^{n_A} n_i-n_A$
	사용자 갱신		n_A	$\log_2 n_U\times\sum_{i=1}^{n_A} n_i$	$n_A\times\log_2 n_U$
수신 계층 변경	전송	증가	$\sum_{i=1}^{n_A} n_i$	$d\times\log_2 n_U$	$3\times\log_2 n_U+n_i-2$
		감소	$n_A\times n_U$	$2\times d\times\log_2 n_U$	$3\times\log_2 n_U+2\times(n_i-2)$
	사용자 갱신		n_A	$d\times\log_2 n_U$	$n_A\times\log_2 n_U$



(그림 9) 사용자 수의 변화에 따른 효율성 비교

참 고 문 헌

[1] Bin B.Zhu, Shipeng Li, Ming Feng, "A Framework of Scalable Layered Access Control for Multimedia," IEEE International Symposium, 2005.

[2] EBU Project Group B/CA, "Functional model of a conditional access system," EBU Technical Review, 1995.

[3] Heiko Schward, Mathias Wien, "The Scalable Video Coding Extension of the H.264/AVC Standard," IEEE Signal Processing Magazine, 2008.

[4] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "Overview of the Salable Video Coding Extensions of the H.264/AVC Standard," IEEE transactions on circuits and systems for video technology, Vol.17, No.9, 2007.

[5] ITU-T and ISO/IEC JTC 1, "Advanced Video Coding for Generic Audiovisual Services," ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 8 (including the SVC extension): Consented in July, 2007.

[6] Sanjeev Setia, Sencun Ahu, Susil Jjodia, "A Comparative Performance Analysis of Reliable Group Key Transport Protocols for Secure Multicast," Special issue of Performance Evaluation on the Proceedings of the Performance, 2002.

[7] Wong C.K, Gouda M, and Lam S.S, "Secure Group Communications using Key Graphs," ACM SIGCOMM 98, 1998.

[8] 이정희, 오희국, "IPTV환경에서 누적형 계층 비디오 멀티캐스트를 위한 키 분배 방법", 한국정보보호학회 하계학술대회 논문집, Vol.19, No.1, 2009.

조 태 남

e-mail : tncho@ws.ac.kr

1986년 이화여자대학교 전자계산학과(이학사)

1988년 이화여자대학교 전자계산학과(이학석사)

2004년 이화여자대학교 컴퓨터학과(공학박사)

1988년～1996년 한국전자통신연구원 선임연구원

2004년～2005년 이화여자대학교 컴퓨터학과 전임강사

2005년～현 재 우석대학교 정보보안학과 조교수

관심분야 : 키관리, IPTV, TPM, 암호프로토콜 등

용 승 림

e-mail : slyong@inhatc.ac.kr

1998년 이화여자대학교 컴퓨터학과(공학사)

2000년 이화여자대학교 컴퓨터학과(공학석사)

2006년 이화여자대학교 컴퓨터학과(공학박사)

2006년～2007년 이화여자대학교 전임강사

2007년～2008년 (재)그래픽스연구소 선임연구원

2008년～현 재 인하공업전문대학 컴퓨터시스템과 전임강사

관심분야 : 암호프로토콜, 저작권 보호, RFID 보안 등