신뢰성 있는 브로드캐스트 암호화를 위한 자가 키 복구 기법

(A Self-Recovering Key Management Scheme for Reliable Broadcast Encryption)

허 준 범 † 윤 현 수 ††
(Junbeom Hur) (Hyunsoo Yoon)

요 약 브로드캐스트 서비스의 사용자가 많아질수록 확장성 있는 접근 제어는 효율적인 권한 부여에 필수적인 요구사항이라고 할 수 있다. 접근 제어는 합법적인 사용자들에게만 비밀키를 안전하게 전달함으로써 이루어지게 된다. 그러나 브로드캐스트 환경에서는 채널의 안정성이 보장되지 않기 때문에 비밀키를 전달하는 과정에서 키 전달 메시지가 사라지거나, 사용자의 단말이 네트워크에 접속되어 있지 않은 경우 메시지가 사용자에게 제대로 전달되지 못하는 경우가 빈번하게 발생할 수 있다. 본 연구에서는 합법적인 사용자가 서버로부터 다수의 세션동안 키 전달 메시지를 수신하지 못한 경우라도 자신의 이전키와 현재 세션에 전송되는 힌트 메시지를 이용해서 현재 세션의 그룹키를 복구할 수 있는 효율적인 그룹키 자가복구 기법을 제안한다. 성능 분석 결과 제안한 기법은 기존의 신뢰성 있는 키 분배 기법에 비해 확장성 및 효율성 측면에서 더 향상되었다는 것을 알 수 있다. 제안한 기법은 사용자로부터 서버로의 역방향 채널이 존재하지 않는 브로드캐스트 네트워크 환경에서 제한수신시스템 등에 활용될 수 있다.

키워드 : 접근 제어, 브로드캐스트 암호화, 제한수신시스템, 키 복구 기법

Abstract One of the principal impediments to the achievement of a scalable access control for a large number of subscribers in a public broadcast is to distribute key update messages reliably to all stateless receivers. However, in a public broadcast, the rekeying messages can be dropped or compromised during the transmission over an insecure broadcast channel, or transmitted to the receivers while it was off-line. In this study, we propose a novel group key management scheme that features a mechanism that allows the legitimate receivers to recover the current group key even if they lose key update messages for long-term sessions using short hint messages and member computation. The performance analysis result shows that the proposed scheme has advantages of the scalable and efficient rekeying compared with the previous reliable group key distribution schemes. The proposed key management scheme targets a conditional access system in a media broadcast where there is no feedback channel from receivers to the broadcasting station.

Key words : access control, broadcast encryption, reliable key distribution, stateless receiver

· This research is supported by the Ubiquitous Computing and Network(UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy(MKE) in Korea as a result of UCN's subproject 09C1-T1-20S, and the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (No. R01-2007-000-20865-0).

† 학생회원 : 한국과학기술원 전산학과
jbhur@nslab.kaist.ac.kr
†† 종신회원 : 한국과학기술원 전산학과 교수
hyoon@nslab.kaist.ac.kr

논문접수 : 2009년 8월 6일
심사완료 : 2009년 10월 7일

정보과학회논문지: 정보통신 제36권 제6호(2009.12)

1. 서 론

그룹 통신 기반의 애플리케이션은 무선 통신의 브로드캐스트 특징을 활용함으로써 다수의 사용자 간에 보다 효율적인 정보 교환을 가능하게 한다. 이러한 대다수의 그룹 기반 애플리케이션에서는 권한이 없는 사용자가 그룹 통신에 접근하지 못하게함으로써 그룹 데이터를 알 수 없도록 하는 접근제어(access control)를 가장 중요한 보안 요구 사항 중 하나로 요구하고 있다[1].

안전한 그룹통신은 오직 합법적인 사용자들 만이 공유하고 있는 안전한 비밀키(그룹키)를 기반으로 이루어지게 된다[2]. 따라서, 대규모의 가입자가 존재하는 브로드캐스트 서비스의 경우 가입자 간에 확장성(scalable)있고 신뢰성(reliable)있는 방법으로 그룹키를 갱신하는 것은 안전한 그룹통신에 있어서 가장 중요한 문제 중 하나이다. 특히 대규모의 사용자가 참여하는 멀티미디어 브로드캐스트 환경에서 그룹통신에 대한 접근제어를 위한 키 관리는 다음의 특징들로 인해 더욱 복잡해진다: (1) 다수의 사용자가 아무때나 네트워크에 가입 또는 탈퇴할 수 있는 동적 그룹, (2) 특정 세션에 그룹키 갱신 메시지를 전달받지 못하면 그 세션 이후로는 현재 세션에 대한 그룹키를 갱신할 수 없는 stateless 수신자[3], (3) 브로드캐스트 서버로부터 수신자로의 단방향 통신 채널.

기존의 그룹키 분배 기법들에서는 stateless 키갱신 기법들이 stateless 수신자에 대한 안정적이고 신뢰할 수 있는 키 분배 문제를 해결할 수 있다[3]. stateless 키갱신 알고리즘은 합법적인 그룹 멤버가 이전 세션에 키갱신 메시지를 수신하지 못하더라도 현재 세션의 그룹키를 복호화할 수 있도록 한다. 그러나 이러한 stateless 키갱신 기법들은 키갱신 메시지가 네트워크에서 탈퇴한 사용자의 수에 비례해서 증가하기 때문에 확장성이 떨어지는 단점이 있다. 이와는 반대로, stateful 키갱신 알고리즘은 키갱신 메시지가 비교적 적기 때문에 확장성 측면에서 장점이 있지만 안정적인 통신 채널을 가정하고 있기 때문에 실제 무선 통신 환경에서는 키갱신 메시지의 재전송으로 인한 통신 비용이 더 커질 가능성이 높다[4-6]. 최근에 제안된 ELK[7] 기법은 키복구 매커니즘을 이용해서 키갱신의 신뢰성을 향상시켰지만 가장 마지막의 한번의 세션에 대해서만 키복구를 허용하기 때문에 수신자가 두 번 이상 연속적으로 키갱신 메시지를 못받는 경우 키복구를 할 수 없게되는 문제가 있다.

본 연구에서는 위에서 언급한 모든 제한사항들이 존재하는 멀티미디어 브로드캐스트 환경에서의 그룹통신에 대한 접근제어를 위해 신뢰성있는 w-RGK(w-session reliable group key distribution) 그룹키 분배 및 복구 기법을 제안한다. 제안하는 기법은 매 세션마다 적은양의 키갱신 메시지와 더불어 짧은 힌트 메시지를 전송함으로써 stateless 수신자가 w 세션동안 키갱신 메시지를 못받더라도 현재 세션의 키를 복구할 수 있는 매커니즘을 제공한다. 제안하는 w-RGK 기법은 w 값에 따라 안정성 및 확장성 측면에서 매우 유연하게 응용될 수 있으며, stateful 방식과 stateless 방식의 장점들, 즉 효율적인 키갱신 및 stateless 수신자를 위한 안정적인 키복구의 특징들을 모두 갖게 된다. 분석 결과에 따르면 제안한 기법은 높은 안정성(reliability)과 안전성(security)을 보장함과 동시에 같은 수준의 안정성을 만족시키기 위해서 필요한 통신 비용이 기존의 제안된 기법들에 비해 더 적은 장점이 있다.

2. 관련 연구

브로드캐스트 암호화는 [2]에서 처음 연구 및 제안되었고, 이 연구를 기반으로 브로드캐스트 암호화에 대한 연구는 [8,9] 등의 연구를 통해 더욱 확장되게 되었다. 브로드캐스트 키 분배 과정에 있어서 키분배서버(Key Distribution Center, KDC)는 사용자 인증 및 권한위임 등의 기능을 수행하며, 사용자가 브로드캐스트 정보에 접근할 수 있도록 그들에게 키 정보를 전송한다. 키 정보는 암호화된 브로드캐스트 정보를 복호화하는데 사용되는 그룹키와 각 세션에 그룹키를 복호화하는데 사용되는 키암호화키(Key-encrypting key, KEK)로 구성된다. 그룹키 분배 문제는 합법적인 그룹 멤버들에게 KEK를 전송하는 문제로 생각할 수 있는데 다음의 안전성을 만족시켜야한다[10,11]: 그룹키 안전성(group key secrecy), 순방향 안전성(forward secrecy), 역방향 안전성(backward secrecy). 그룹키 안전성은 그룹멤버가 아닌 외부의 사용자가 그룹키에 대한 정보를 얻지 못해야함을 의미한다. 순방향 안전성은 탈퇴한 사용자가 탈퇴한 세션 이후의 그룹 통신을 복호화할 수 없어야 함을 의미한다. 역방향 안전성은 그룹에 새로 가입한 사용자가 가입 이전 세션의 그룹통신을 복호화 할 수 없어야 함을 의미한다.

그룹키 분배 알고리즘은 크게 stateful 알고리즘과 stateless 알고리즘으로 구분할 수 있다[12]. stateless 방식은 사용자가 서비스에 등록할 당시부터 저장하고 있는 고유키를 이용해서 그룹키를 암호화한 후 전송하기 때문에 이전 세션의 키갱신 메시지와 독립적으로 현재 세션의 그룹키를 갱신할 수 있다. 따라서 안정적인 통신 채널이 없는 환경에서도 신뢰성있는 키갱신을 가능하게 한다[3,13,14]. 이러한 stateless 알고리즘은 그룹멤버가 이전의 키갱신 메시지를 수신하지 못한 상황에서도 현재 세션의 그룹키를 갱신할 수 있게 만든다는 점에서 장점이 있지만 탈퇴한 멤버의 수에 비례해서 키

갱신 메시지가 증가하기 때문에 확장성 측면에서 단점이 있다고 할 수 있다.

이와 반대로, logical key hierarchy(LKH)[4]와 one-way function tree(OFT)[6] 등의 stateful 알고리즘은 $O(\log N)$의 짧은 키갱신 통신비용을 요구하기 때문에 확장성 측면에서 장점이 있다. 그러나 stateful 키관리 알고리즘은 안정적인 통신채널을 기반으로 각 키갱신 메시지가 이전 세션의 KEK 혹은 키트리의 다른 노드의 KEK로 암호화된 후 전송되기 때문에 만일 사용자가 오프라인이거나 키갱신 메시지를 못받는 경우, 그 이후 세션부터의 메시지는 복호화할 수 없는 문제가 있다. 따라서 stateful 알고리즘은 stateless 수신자 문제를 해결하는데 적합하지 않다.

Perrig는 stateful 방식의 ELK 그룹키 분배알고리즘을 제안하였는데 이 기법은 stateless 수신자 문제를 해결하기 위해서 힌트 메시지를 키갱신 메시지에 포함시킴으로써 사용자가 마지막 한번의 세션에대해 잃어버린 키를 복구할 수 있도록 하였다[7]. 그러나 힌트 메시지 안에는 오직 과거 한 세션에 대한 키복구 정보만 포함되어 있기 때문에 두번 이상의 세션동안 키갱신 메시지를 못받는 경우 단일전송(unicast)을 통해 키를 재전송해야하는 문제가 있다. 이러한 문제는 대규모의 사용자가 존재하는 브로드캐스트 서비스 환경에서 심각한 확장성 문제를 야기시킬 수 있다.

본 연구에서는 세션 정보를 이용해 과거 w 세션 동안 변경된 키트리를 복구하게 함으로써 stateless 사용자가 장기간에 걸친 키갱신 누락의 경우에 효율적으로 대응할 수 있게 한다. 제안한 기법은 다수의 가입자가 동적으로 빈번하게 변화하고, 브로드캐스트 서버로부터 수신자로의 안정적이지 않은 단방향 채널만 존재하는 브로드캐스트 서비스 환경을 위한 효율적이고 신뢰성 있는 그룹키 분배를 가능하게 한다.

3. 시스템 설계 및 정의

본 논문에서 제안하는 키복구 과정의 명확한 설명을 위해 w-세션 안정성(w-*session reliability*)의 개념을 도입한다. w 세션 이하의 세션동안 키갱신 메시지를 전달받지 못한 합법적인 사용자가 현재 세션의 키를 복구할 수 있는 경우 그러한 키분배 기법은 w-세션 안정성을 만족시킨다고 정의할 수 있다.

본 절에서는 제안하는 기법의 시스템 설계 및 정보이론 모델을 이용한 w-세션에 안정적인 w-RGK 그룹키 관리 기법을 정의한다. 엔트로피 함수 H를 포함한 많은 정보이론 기호들은 [15]에서 정의한 기호를 따른다.

3.1 시스템 설계

$\mathbb{U}=\{u_1,\ldots,u_n\}$는 전체 사용자, i는 세션 인덱스를 의미한다. GM은 그룹 매니저, $G_i \subset \mathbb{U}$ 는 i 세션에 그룹 서비스에 가입한 합법적인 사용자들의 집합으로 정의한다. sk_t^i는 사용자 $u_t \in \mathbb{U}$ 가 i 세션에 저장하고 있는 비밀키 집합, GK^i는 i 세션의 그룹키를 가리킨다. 초기 셋업 단계(세션 0)에 모든 사용자 $u_t \in G_0$는 GM으로부터 안전하게 sk_t^0와 GK^0를 받는다. 한 사용자가 i 세션에 그룹에 가입 혹은 탈퇴할 경우 세션은 i에서 $i+1$로 증가하고 순방향 및 역방향 안전성을 위해 각 사용자의 비밀키 및 그룹키는 새로운 키로 갱신된 후 키 갱신 프로토콜(4.2절)을 통해서 합법적인 그룹 멤버들에게 전달된다.

u_t가 $i-1$ 세션에 그룹에 가입(또는 탈퇴)할 경우 i 세션이 시작됨과 동시에 GM은 GK^i를 계산하고 $\forall u_t \in G_i$가 sk_t^i와 GK^i를 얻게하기 위해 키갱신 메시지 D^i와 힌트 메시지 B^i를 브로드캐스트한다. GK^i, SK_t^i, D^i, 그리고 B^i는 각각 GK^i, sk_t^i, D^i, 그리고 B^i에 대한 확률변수(random variable)를 의미한다.

3.2 w-RGK(w-session reliable group key management) 정의

정의 1 (w-RGK 키관리 기법): 다음의 조건들을 만족시키는 알고리즘 A(w,$\mathbb{U}$)를 w-RGK 키관리 기법으로 정의한다.

1) 정확성: $\forall u_t \in \mathbb{U}$ 에 대해서 다음을 만족한다.
$H(\mathrm{GK}^i, \mathrm{SK}_t^i | \mathrm{D}^i, \mathrm{SK}_t^{i-1}) = 0$, if $u_t \in G_i$.

2) w-세션 안정성: $\forall u_t \in \cap_{l=j}^{i} G_l$에 대해서 다음을 만족한다.
$H(\mathrm{GK}^i, \mathrm{SK}_t^i | \mathrm{B}^i, \mathrm{SK}_t^j) \le b$, if $i-w \le j < i$. (b는 u_t가 전수조사를 통해서 현실적인 시간 안에 찾을 수 있을 정도로 작은수.)

3) 그룹키 안전성 및 순방향/역방향 안전성: $\mathfrak{D}=\{D^0, D^1, \ldots\}$와 $\mathfrak{B}=\{B^0, B^1, \ldots\}$로 정의할 때, 다음을 만족한다.

3a) 그룹키 안전성: $u_t \notin G_i$에 대해서 $H(\mathrm{GK}^i | \mathfrak{D}, \mathfrak{B}) = H(\mathrm{GK}^i)$.

3b) 순방향 안전성: $u_t \notin G_l$에 대해서 $H(\{\mathrm{GK}^{i'}\}_{i'>i} | \mathfrak{D}, \mathfrak{B}, \{\mathrm{SK}_t^{i'}\}_{i'<i}, \{\mathrm{GK}_t^{i'}\}_{i'<i},) = H(\{\mathrm{GK}^{i'}\}_{i'>i})$, $l > i$.

3c) 역방향 안전성: $u_t \in G_l$에 대해서 $H(\{\mathrm{GK}^{i'}\}_{i'<i} | \mathfrak{D}, \mathfrak{B}, \{\mathrm{SK}_t^{i'}\}_{i' \ge i}, \{\mathrm{GK}_t^{i'}\}_{i' \ge i},) = H(\{\mathrm{GK}^{i'}\}_{i'<i}), l \ge i$.

그룹키 안전성과 순방향/역방향 안전성은 안전한 그룹 통신을 위한 접근제어를 위한 가장 기본적인 안전성

요구사항이다[10]. 따라서 w-세션 안정성은 기존의 전통적인 키관리 프로토콜로부터의 차별성을 나타내는 가장 중요한 특징이라고 할 수 있다.

4. w-RGK 그룹키 관리 기법

w-RGK 프로토콜은 멤버의 가입 및 탈퇴에 따른 그룹키 갱신과 복구 매커니즘으로 구성된다. 연속된 두 멤버의 변화 사이의 기간을 *세션*으로 정의한다. 제안한 프로토콜에서는 멤버의 가입 및 탈퇴에 따라 세션이 변화하고 한 세션에 단 한번의 멤버 변화만 허용한다.

w-RGK 기법의 주요 아이디어는 키트리를 상향식(bottom-up) 방식으로 구성하고 세션정보를 힌트 메시지로 사용하는 것이다. 세션정보는 stateless 수신자로 하여금 오프라인에 있는 동안 수신할 수 없었던 지난 세션에 대한 키계층에 대한 키갱신 정보를 알 수 있게 한다. 특히, 세션정보는 w 세션동안 한번 이상 자신의 키상태 정보를 갱신할 수 있는 w-세션 stateless 맴버로 하여금 최근 w 세션동안 키갱신 메시지를 수신하지 못하더라도 현재 세션의 비밀키의 유효성을 확인할 수 있도록 한다. 이것은 이후에 증명하겠지만 세션이 증가함에 따라 힌트 메시지 안의 키검증(key verification) 정보의 증가율을 줄여준다. 프로토콜 설명을 위해 다음과 같은 기호 및 함수를 정의한다.

- n: 키 비트 크기.
- K_j^i: i 세션에 키트리의 노드 v_j에 할당된 KEK.
- $\mathrm{PRF}^{\langle l \to m \rangle}(M)$: l 비트 메시지 M을 입력으로 받아 m 비트의 의사난수를 생성하는 단방향 의사난수함수.
- $L(x), R(x)$: $\mathrm{PRF}^{\langle n \to 2n \rangle}(x)$의 결과값에 대한 왼쪽, 오른쪽 절반 비트 정보 ($|L(x)| = |R(x)| = |x| = n$).
- $\{M\}_k$: 키 k로 암호화된 메시지 M.
- $a|b$: 문자열 a와 b의 연쇄 문자열(concatenation).

본 절에서는 키서버 측면에서 키갱신 과정, 그리고 사용자 측면에서 키 복구 과정에 대해서 설명한다.

4.1 초기 셋업

제안한 기법은 그림 1과 같은 이진 키트리를 사용한다. 키트리의 각 노드 v_j는 i 세션에 KEK K_j^i를 가지고 있고, 키트리의 루트(root)노드의 KEK는 그룹키가 된다. 따라서, 그림 1에서 K_1^i는 세션 i의 그룹키 GK^i를 의미한다. 그룹 멤버는 키트리의 말단(leaf)노드에 할당되고, 각 멤버는 자신의 말단노드로 부터 루트노드에 이르는 모든 패스 상의 키를 저장하고 있다. 이러한 키를 패스키라 부른다. $u_t \in G_i$에 대해서 sk_t^i는 그룹키를 제외한 u_t의 패스키를 가리킨다. 예를 들어, 그림 1에서 $u_2 \in G_i$일 경우, $sk_2^i = \{K_9^i, K_4^i, K_2^i\}$가 된다.

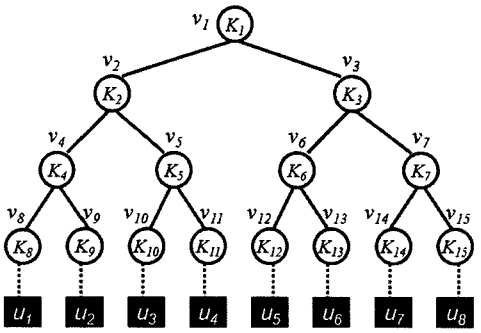


그림 1 키트리

키트리의 초기 셋업 단계에서(세션 0), 키트리는 다음과 같이 구성된다.

1) 모든 멤버는 트리의 말단노드에 할당되고, 각 말단노드에 임의의 n 비트 키를 생성한 후 할당한다.
2) 키트리의 각 K_j는 자신의 두 자식 노드의 KEK인 K_{2j}와 K_{2j+1}로부터 상향식 방식으로 생성된다. 초기 세션의 K_j는 $K_j^0 = L(R(\mathrm{PRF}^{\langle n \to n \rangle}(C_{LR})))$로 생성된다. (여기서 $C_{LR} = \mathrm{PRF}^{\langle n \to n/2 \rangle}(K_{2j}^0) | \mathrm{PRF}^{\langle n \to n/2 \rangle}(K_{2j+1}^0)$ 이다.)
3) 각 멤버 $u_t \in G_0$는 서버로부터 키트리의 말단노드로부터 루트노드에 이르는 자신의 모든 패스키를 안전하게 전달받는다.

초기 셋업 단계 이후 키트리는 그룹 멤버에 변화가 생길 경우 다음 절에서 설명할 그룹키 갱신 알고리즘에 의해서 관리 및 갱신된다.

4.2 키 갱신

4.2.1 멤버 가입

멤버 가입 시, 서버는 가입한 멤버를 빈 말단노드에 할당하고 해당 말단노드와 루트노드 상의 모든 키를 PRF를 이용해서 갱신한다. 그 후, 서버는 새로운 임의의 키를 생성하고 새로 가입한 멤버의 말단 노드에 할당한 후 갱신된 패스키와 함께 새 멤버에 안전하게 전송한다. 예를 들어, 그림 1에서 만약 u_8이 i 세션에 새롭게 가입한 멤버라면, K_7^{i-1}은 $K_7^i = PRF^{\langle n \to n \rangle}(K_7^{i-1})$로 갱신된다. 마찬가지로, K_3^{i-1}와 $K_1^{i-1}(= GK^{i-1})$ 또한 단방향 PRF 함수를 이용해서 K_3^i과 $K_1^i(= GK^i)$로 갱신된다. K_7^{i-1}, K_3^{i-1}, K_1^{i-1}를 가지고 있는 모든 멤버는 자체적으로 새로운 키들로 계산할 수 있다.

4.2.2 멤버 탈퇴

멤버 탈퇴시, 탈퇴한 멤버가 탈퇴 후 그룹통신을 알 수 없도록 그 멤버의 모든 패스키는 각 노드의 자식 노드의 키 값을 이용해 갱신된다. 멤버 u가 $i-1$ 세션에

탈퇴할 경우(세션 i 시작), 키트리 갱신 알고리즘은 다음과 같다.

1) 난수를 생성하고 u의 말단노드에 할당한다. 루트 노드를 제외한 u의 모든 패스노드의 형제(sibling) 노드키는 $K_l^i = PRF^{\langle n \to n \rangle}(K_l^{i-1})$로 갱신된다.
2) 키트리에서 u의 모든 패스노드 v_j에 대해서, 노드의 레벨이 d인 경우 새로운 시드값 $r_d^i = PRF^{\langle n \to n \rangle}(C_{LR})$을 생성한 후 v_j에 할당한다(여기서 $C_{LR} = PRF^{\langle n \to n/2 \rangle}(K_{2j}^i) | PRF^{\langle n \to n/2 \rangle}(K_{2j+1}^i)$. 노드 v_j의 키는 $K_j^i = L(R(r_d^i))$가 되며 K_1^i는 그룹키 GK^i가 된다. u의 모든 패스노드와 그 형제노드를 제외한 다른 모든 노드의 키는 $K^i = K^{i-1}$가 된다.
3) 서버는 v_j의 $R(r_d^i)$를 K_{2j}^i와 K_{2j+1}^i를 이용해 암호화한 후 브로드캐스트 한다(키갱신 메시지).

예를 들어, 그림 2에서 u_2가 그룹을 탈퇴할 경우, 서버는 새로운 K_9^i를 v_9에 할당하고, u_2의 모든 패스키의 형제키 K_8^{i-1}, K_5^{i-1}, K_3^{i-1}를 PRF를 이용해서 갱신한다. 그리고 서버는 v_4의 자식키들인 $K_8^i (= PRF^{\langle n \to n \rangle}(K_8^{i-1}))$과 K_9^i를 이용해서 r_3^i를 만들고 v_4노드에 할당한 후, $K_4^i (= L(R(r_3^i)))$를 생성한다. r_2^i는 K_4^i와 $K_5^i (= PRF^{\langle n \to n \rangle}(K_5^{i-1}))$를 이용해서 생성된 후 v_2 노드에 할당되고, 서버는 $K_2^i (= L(R(r_2^i)))$를 생성한다. 마지막으로, K_1^i 역시 같은 방법으로 생성된다. $R(r_1^i)$, $R(r_2^i)$, 그리고 $R(r_3^i)$ 값들은 중간의 KEK들을 갱신하기 위해 그들의 자식 KEK들로 암호화된 후, 키갱신 메시지로 서버에 의해 브로드캐스트된다. u_2를 제외한 나머지 멤버들은 각각 $L(R(r_1^i))$, $L(R(r_2^i))$, 그리고 $L(R(r_3^i))$를 계산함으로서 새로운 KEK인 K_1^i, K_2^i, 또는 K_4^i를 얻는다.

4.2.3 키 복구

합법적인 멤버가 키갱신 메시지를 최대 w 세션동안 수신하지 못한 경우, 그 멤버는 잃어버린 세션 동안 갱신된 키, 즉 현재 세션의 그룹키를 포함한 자신의 패스키를 힌트 메시지와 자신이 유지하고 있는 유효한 패스키를 이용해서 복구할 수 있다. 본 논문에서 각 멤버는 $2^{n/2}$의 계산을 현실적인 시간 안에 할 수 있고, 마지막으로 갱신한 세션 및 패스키를 유지할 수 있다고 가정한다. 그러면 현재 세션과 어떤 키가 갱신되었는지에 대한 정보가 주어지면, 멤버는 말단 노드로부터 루트 노드에 이르는 현재 갱신된 각각의 패스키를 $2^{n/2}$에 대한 PRF 전수조사(brute-force)를 통해서 복구할 수 있다.

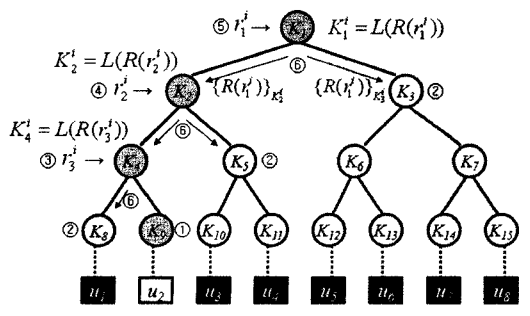


그림 2 멤버 u_2 탈퇴

따라서 w-세션 안정성 보장을 위한 힌트 메시지는 앞서 언급한 최근 w 세션에 대한 (1) 세션정보(session information): {(*session, event, index*)}와 (2) 최근 w 세션동안 멤버 탈퇴 이벤트에 의해 갱신된 노드 v_j의 KEK들에 대한 키검증(key verification): $\{PRF^{\langle n \to m \rangle}(K_j)\}$, $m < n$으로 구성된다. 세션정보에서 *event*는 이벤트의 종류, 즉 가입 또는 탈퇴를 가리키는 1 비트 정보이고, *session*은 그 이벤트의 세션 정보를 가리킨다. 만약 *event*가 탈퇴 이벤트를 가리킬 경우, *index*는 시드 $r_{\log N}^i$가 위치하고 있는 노드의 인덱스를 가리키고, 가입 이벤트일 경우 *index*는 키트리에서 갱신된 노드의 패스 중에서 log$N$레벨의 인덱스를 나타낸다.

합법적인 멤버는 힌트 메세지를 가지고 자신의 상태를 갱신하지 못한 (w 세션 이하의) 과거 세션 동안 갱신된 현재 세션의 패스키와 그룹키를 복구할 수 있다. 키복구 과정은 다음과 같은 키트리복구 단계와 키복구 단계로 구성된다.

1) 키트리 복구: 세션정보와 *TreeRecovery* 알고리즘을 이용해 현재 키구조상태를 재구성한다.
2) 키복구: *KeyRecovery* 알고리즘을 이용해 힌트 메시지 안의 키검증과 계산된 해당 후보키(candidate key)의 키검증을 비교함으로써 갱신된 패스키들을 복구한다. 후보키는 유효한 패스키의 각 형제키에 대한 $2^{n/2}$ 전수조사를 통해 얻을 수 있다.

$S_i^{i+\Delta}$와 $V_i^{i+\Delta}$를 각각 i부터 $i+\Delta$까지의 세션에 대한 힌트 메시지 안의 세션정보와 키검증을 가리킨다고 하자. 그러면 키복구 과정은 다음의 *TreeRecovery*와 *KeyRecovery* 알고리즘으로 설명될 수 있다.

그림 3은 i 세션부터 $i+1$ 세션까지 그룹에 u_1 탈퇴, u_7 가입, 그리고 u_6 탈퇴 순서로 멤버 변화가 이루어진 경우의 키복구 시나리오의 예를 보여준다. u_3이 그 세션동안 키갱신 메시지를 못받았고 현재 세션이 $i+2$이라고 가정하자. 힌트 메시지는 세션정보 { (i, leave, 4), ($i+1$, join, 7), ($i+2$, leave, 6)}와 키검증 $\{PRF^{\langle n \to m \rangle}(K_1^{i+2})$,

Procedure 1 $TreeRecover(u, v_j, \mathcal{S}_i^{i+\Delta})$

// i 부터 $i+\Delta$ 세션 까지 키갱신 메시지를 수신하지 못한 v_j에 할당된 사용자 u가 $\mathcal{S}_i^{i+\Delta}$로 부터 키구조상태 재구성

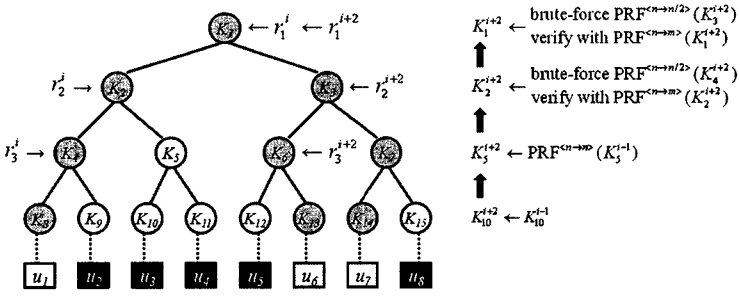
```
for 모든 세션정보 (session, event, index) ∈ S_i^{i+Δ} do
  if event = join then
    temp1 ← ⌊j/2⌋, temp2 ← index, K_j^i ← K_j^{i-1}
    while temp1 > 0 and v_temp1 ≠ null do
      if temp1 = temp2 then
        K_temp1^i ← PRF^{<n→n>}(K_temp1^{i-1})
      else
        K_temp1^i ← K_temp1^{i-1}
      end if
      temp1 ← ⌊temp1/2⌋, temp2 ← ⌊temp2/2⌋
    end while
  else
    if event = leave then
      temp1 ← j, temp2 ← index
      while temp1 > 0 do
        if ⌊temp1/2⌋ = temp2 then
          K_temp1^i ← PRF^{<n→n>}(K_temp1^{i-1})
          while temp1 > 0 do
            temp1 ← ⌊temp1/2⌋, K_temp1^i ← null
          end while
          break while loop
        else
          K_temp1^i ← K_temp1^{i-1}
        end if
        temp1 ← ⌊temp1/2⌋, temp2 ← ⌊temp2/2⌋
      end while
    end if
  end if
end for
```

Procedure 2 $KeyRecover(u, v_j, \mathcal{V}_i^{i+\Delta})$

// i 부터 $i+\Delta$ 세션 까지 키갱신 메시지를 수신하지 못한 v_j에 할당된 사용자 u가 $\mathcal{V}_i^{i+\Delta}$로 부터 패스키 복구

```
for 모든 키검증 ∈ V_i^{i+Δ} do
  temp ← j
  while temp > 0 do
    if K^{i+Δ}_⌊temp/2⌋ ≠ null then
      temp ← ⌊temp/2⌋
    else
      if temp mod 2 ≠ 0 then
        for 2^{n/2}개의 PRF^{<n→n/2>}(K^{i+Δ}_{temp-1}) = C̃_L do
          r̃^{i+Δ}_⌊temp/2⌋ ← PRF^{<n→n>}(C̃_LR) with C̃_LR = C̃_L|PRF^{<n→n/2>}(K^{i+Δ}_temp)
          if PRF^{<n→m>}(L(R(r̃^{i+Δ}_⌊temp/2⌋))) = PRF^{<n→m>}(K^{i+Δ}_⌊temp/2⌋) (∈ V_i^{i+Δ}) then
            K^{i+Δ}_⌊temp/2⌋ ← L(R(r̃^{i+Δ}_⌊temp/2⌋))
          end if
        end for
      else
        for 2^{n/2}개의 PRF^{<n→n/2>}(K^{i+Δ}_{temp+1}) = C̃_R do
          r̃^{i+Δ}_⌊temp/2⌋ ← PRF^{<n→n>}(C̃_LR) with C̃_LR = PRF^{<n→n/2>}(K^{i+Δ}_temp)|C̃_R
          if PRF^{<n→m>}(L(R(r̃^{i+Δ}_⌊temp/2⌋))) = PRF^{<n→m>}(K^{i+Δ}_⌊temp/2⌋) (∈ V_i^{i+Δ}) then
            K^{i+Δ}_⌊temp/2⌋ ← L(R(r̃^{i+Δ}_⌊temp/2⌋))
          end if
        end for
      end if
      temp ← ⌊temp/2⌋
    end if
  end while
end for
```

$PRF^{\langle n\to m\rangle}(K_2^i)$, $PRF^{\langle n\to m\rangle}(K_3^{i+2})$, $PRF^{\langle n\to m\rangle}(K_4^i)$, $PRF^{\langle n\to m\rangle}(K_6^{i+2})\}$로 구성된다. u_3은 세션정보를 통해서 그림 3의 키구조의 현재 상태를 재구성할 수 있다. (예를 들어, v_3은 마지막으로 $i+2$ 세션에 갱신, v_7은 $i+1$ 세션에 갱신, 그리고 v_5는 i 세션 이후로 갱신이 안되었음). u_3은 그 후 $K_5^i(=PRF^{\langle n\to n\rangle}(K_5^{i-1}))$가 i 세션 이후로 갱신되지 않았음을 알기 때문에 K_4^i의 값에 대한 추정을 시작할 수 있다. u_3은 $PRF^{\langle n\to n/2\rangle}(K_4^i)\}$에 대한 모든 $2^{n/2}$개의 가능성들을 전수조사 하고 키검증 $PRF^{\langle n\to m\rangle}(L(R(\tilde{r}_2^i)))$과 $PRF^{\langle n\to m\rangle}(K_2^i)$의 비교를 통해 최종 추측 $\tilde{r}_2^i = PRF^{\langle n\to n\rangle}(\tilde{C_{LR}})$을 검증한다(여기서 $\tilde{C_{LR}} = PRF^{\langle n\to n/2\rangle}(K_4^i)|PRF^{\langle n\to n/2\rangle}(K_5^i)$). 만약 동일하게 검증될 경우, 그 키는 후보키가 된다. 그리고 u_3은 후보키 $K_2^{i+2}(=PRF^{\langle n\to n\rangle}(K_2^i))$와 키검증 $PRF^{\langle n\to m\rangle}(K_1^{i+2})$를 가지고 동일한 검증 과정을 통해 $PRF^{\langle n\to n/2\rangle}(K_3^{i+2})$에 대한 $2^{n/2}$의 가능성을 전수조사를 한다.



(a) Tree recovery (b) Key recovery by u_3 with sk_3^{i-1}

그림 3 키복구 과정

이 절차는 올바른 키를 복구시킬 수 있지만, 하나 이상의 후보키(false positive)를 생성할 수도 있다. 만약 $m=n/2$일 경우, 한 멤버는 레벨 당 하나의 추가적인 false positive 키를 얻게 된다. 그러나 m을 한 비트 증가시킴으로써 false positive 키를 절반으로 줄일 수 있다. 따라서, m은 false positive 키생성을 줄이기 위해서 $n/2$ 보다 크게 정해져야 한다. 그러나 키검증 정보의 크기는 계산과 통신 비용 간의 trade-off를 고려해서 설정되어야 한다.

5. 성능 측정 및 분석

본 절에서는 제안한 기법의 성능을 분석하고 기존 키 관리 알고리즘과 성능을 비교 한다. 성능 분석은 통신, 계산, 그리고 저장 비용 측면에서 이루어지며 각 기준은 다음과 같이 정의한다[4,10].

- 저장 비용: 각 멤버가 저장해야 하는 키 개수
- 통신 비용: KDC가 전송하는 키갱신 메시지의 크기
- 계산 비용: 각 멤버가 수행하는 복호화 및 PRF 계산량

표 1은 기존 그룹키 관리 기법 및 제안한 기법의 성능을 비교한다. 표 1은 키복구를 지원하는 프로토콜, 키갱신 및 힌트 메시지의 크기, 그리고 각 멤버의 저장 비용을 보여준다. 또한 각 멤버가 키를 갱신 및 복구하는데 필요한 계산량을 보여준다. 제안한 기법에서 힌트 메시지에 대한 통신 비용은 최악의 경우 과거 w 세션동안 R 명의 멤버가 그룹을 탈퇴한 조건에서 w-세션 안정성을 보장하는데 필요한 힌트 메시지의 크기를 의미한다. 반면에, ELK의 통신 비용은 오직 1-세션 키복구만을 위한 힌트 메시지의 크기를 나타낸다. 제안한 기법이 ELK와 동일한 보안 매개변수를 사용한다고 가정할 때 ($m=n_3$, $n_1=n_2=n/2$), ELK는 w-세션 안정성을 보장하기 위해서 키검증 정보를 위한 $mR\log N$의 통신 비용과 현재 세션의 키를 복구하기 위한 $\frac{R}{2}CP_{worst}$의 PRF 연산을 요구한다.

5.1 통신 비용

본 분석에서는 대부분의 키관리 기법과 동일하게 완전 이진 키트리(complete binary tree)를 가정한다. 제안한 기법에서 KDC가 전송하는 힌트 메시지는 최대 $Comm_k + Comm_s \le m((\lceil \log R \rceil - \lfloor \log R \rfloor + 1)2^{\lfloor \log R \rfloor} + (\log N - \lceil \log R \rceil)R - 1) + w\log wN$ 비트를 요구한다. 또한 제안한 기법에서 전송되는 키갱신 메시지의 크기는 멤버 가입 경우 $\log N$ 비트, 그리고 멤버 탈퇴 경우 $2n\log N$ 비트가 된다. 이러한 비용은 CS와 SD 기법이 각각 $nR\log(N/R)$, $2n(R-1)$ 비트의 키갱신 메시지를 전송하는 것과 비교할 때, 탈퇴하는 멤버가 증가할수록 기존 기법에 비해 보다 확장성 있다고 볼 수 있다.

분석한 전체 키갱신 및 힌트 메시지의 전송량은 다른 기법들과 그림 4에서 비교되고 있다. 그림 4 에서 x-축은 세션을 나타내고, y-축은 현재 세션에 전송되는 전체 키갱신 및 힌트 메시지의 크기를 나타낸다. 이 시뮬레이션에서는 공정한 비교를 위해 오직 멤버 탈퇴에 의해서만 세션이 바뀌고 힌트 메시지가 각 세션마다 전송된다고 가정한다. 시뮬레이션 결과는 서로 다른 키 크기에 따라 200-세션 안정성 을 보장하기 위해 각 세션 당 전송되는 메시지의 크기를 나타낸다. CS와 SD 기법들은 언제나 신뢰성 있는 전송을 보장하므로 w 값에 영향을 받지 않는다. 그러나 이 기법들의 통신 비용은 탈퇴한 멤버의 수에 비례해서 증가하게 된다. 반면에 ELK는 오직 1-세션 안정성만을 지원할 수 있기 때문에 w-세션 안정성을 보장하기 위해서는 ELK는 각 세션에 w 개의 힌트 메시지를 전송해야만 한다.

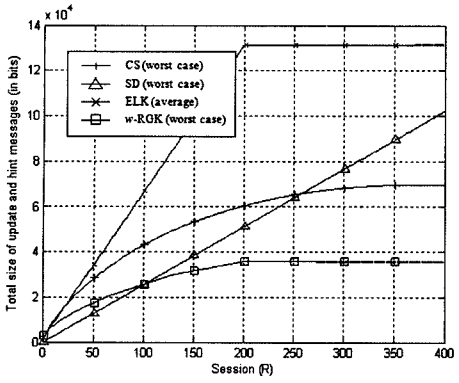
5.2 계산 비용

여기서는 제안한 기법의 키복구 과정에 필요한 계산

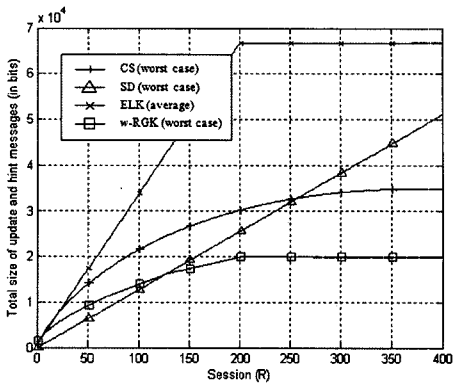
표 1 그룹키 관리 프로토콜 비교

	키복구	저장 비용	계산 비용		통신 비용	
			갱신	복구	갱신	힌트
LKH [4]	n/a	$n\log N$	$D\log N$	–	$2n\log N$	–
ELK [7]	1-세션	$4n\log N$	가입: $H\log N$ 탈퇴: $(D+H)\log N$	$H2^{n_3}\{(1+\frac{1}{2^{n_3-n_1}})^{\log N}-1\}$	가입: 0 탈퇴: $(n_1+n_2)\times\log N$	키검증: $n_3\log N$ 자식 KEK 정보: $(n_2-n_1)\log N$
CS [3]	∞-세션	$n\log N$	$D+P\log\log N$	0	$nR\log\frac{N}{R}$	–
SD [3]	∞-세션	$n(\frac{1}{2}\log^2 N+\frac{1}{2}\log N+1)$	$D+P\log\log N$ $+H\log N$	0	$n(2R-1)$	–
w-RGK	w-세션	$n\log N$	가입: $H\log N$ 탈퇴: $(D+H)\log N$	$H2^{m+1}\{(1+\frac{1}{2^{m-\frac{m}{2}}})^{\log N}-1\}$	가입: $\log N$ 탈퇴: $2n\log N$	키검증: $Comm_k$ 세션정보: $Comm_s$

N: 그룹 크기, R: 탈퇴한 멤버의 수, D: 복호화 연산, H: 해쉬 또는 PRF 연산, P: 비교 연산.
$Comm_k = m((\lceil \log R \rceil - \lfloor \log R \rfloor + 1)2^{\lfloor \log R \rfloor} + (\log N - \lceil \log R \rceil)R - 1)$, $Comm_s = w\log wN$.



(a) 200-세션 안정성 (n = 128-bit)



(b) 200-세션 안정성 (n = 64-bit)

그림 4 200-세션 안정성을 위해 전송되는 키갱신 및 힌트 메시지 크기 ($N=1024; m=n_3, n_1=n_2=n/2$ in ELK)

비용에 대해서 분석한다. 키검증의 크기(m)와 안정성을 보장할 수 있는 최대 세션(w)의 측면에서 키갱신 알고리즘에 요구되는 계산량을 분석한다. 각 세션은 멤버 탈퇴에 의해 다음 세션으로 진행된다고 가정한다.

w 세션 이전에 생성된 패스키와 그룹키를 가지고 있는 사용자 u가 키복구 알고리즘을 수행한다고 가정하자. 본 절에서는 u가 현재 세션의 모든 패스키와 그룹키를 복구하는 데에 필요한 PRF 연산의 수를 분석한다.

*unit*을 전수조사를 통해 하나의 패스키 또는 그룹키를 찾아내기 위해 수행해야 하는 PRF 연산의 수라고 하자. 키복구 과정에서 r_R(또는 r_L)을 알고 있는 합법적인 사용자 u는 각 $r_L \in \{0,1^{n/2}\}$ (또는 $r_R \in \{0,1^{n/2}\}$)에 대해서 $4\times PRF$ 연산을 수행해야 하므로 $unit = 4\times 2^{n/2}$이 된다. 먼저, 복구해야 하는 키의 개수가 d일 때, 요구되는 PRF 연산의 수(A_d)를 분석하면 다음의 식이 성립한다.

$$A_d = 2^{m-n/2-1} \cdot \left\{ \left(1+\frac{1}{2^{m-n/2}}\right)^d - 1 \right\} \cdot unit. \qquad (1)$$

위의 분석에 따라, 사용자에 필요한 가장 많은 PRF 연산은 다음과 같다.

$$CP_{worst} = A_{\log N}. \qquad (2)$$

다음은 평균 계산량에 대해 분석한다. w 세션동안 탈퇴한 사용자가 각 세션에 걸쳐 고르게 선택된다고 가정한다. 위에서 언급한 것처럼 u가 수행해야할 계산량은 과거 w 세션동안 탈퇴한 다른 사용자와 공유하고 있던 키의 최대 개수에 의존한다. S를 그러한 최대 공유키의 개수라고 하자. 그러면 평균 계산 요구량 (CP_{avg})은 다음과 같이 계산될 수 있다.

$$CP_{avg} = \sum_{i=1}^{\log N} \Pr[S=i] \cdot A_i. \qquad (3)$$

$\Pr[S=i] = \Pr[S \geq i] - \Pr[S \geq i+1]$과 $\Pr[S \geq i] = 1-(1-\frac{2^{\log N-(i-1)}-1}{N-1})^w$이 성립하기 때문에, CP_{avg}는 w가 증가할수록, $A_{\log N}$에 수렴한다. 따라서 식 (2)와 (3)으로 부터 w가 증가할수록 CP_{avg}는 CP_{worst}로 수렴한다는 것을 알 수 있다.

6. 안전성 분석

이번 절에서는 제안한 기법의 안전성을 분석한다. 안전성 분석은 정보이론의 엔트로피 모델을 바탕으로 그룹 멤버가 아닌 사용자가 그룹키를 유도해내는 데에 필요한 계산 복잡도를 분석한다. 이번 절에서는 3절에서 정의한 동일한 기호를 사용한다.

정리 1 (그룹키 안전성): 임의의 사용자 $u_t \notin G_i$에 대해서, 다음이 성립한다.

$$H(\mathrm{GK}^i | \mathfrak{D}, \mathfrak{B}) = H(\mathrm{GK}^i)$$

정리 2 (역방향 안전성): $l \geq i$인 경우, $u_t \notin G_l$에 대해서 다음이 성립한다.

$$H(\{\mathrm{GK}^{i'}\}_{i'<i} | \mathfrak{D}, \mathfrak{B}, \{\mathrm{SK}_t^{i'}\}_{i' \geq i}, \{\mathrm{GK}_t^{i'}\}_{i' \geq i},) = H(\{\mathrm{GK}^{i'}\}_{i'<i}).$$

정리 3 (순방향 안전성): $l > i$인 경우, $u_t \notin G_l$에 대해서 다음이 성립한다.

$$H(\{\mathrm{GK}^{i'}\}_{i'>i} | \mathfrak{D}, \mathfrak{B}, \{\mathrm{SK}_t^{i'}\}_{i'<i}, \{\mathrm{GK}_t^{i'}\}_{i'<i},) = H(\{\mathrm{GK}^{i'}\}_{i'>i}).$$

각 정리의 증명은 부록에 기술한다.

7. 구 현

PRF 함수는 Crypto++ 5.5 라이브러리[16]에서 제공하는 VMAC(AES)-64[17]로 구현했다. 브로드캐스트 암호화가 소비자로 하여금 영속 데이터를 저장하게끔 하는 것 보다 실시간 접근제어에 사용되는 경향이 더

크므로 [18], $n=64$는 공공 브로드캐스트 환경에서 계산 비용과 보안 레벨을 고려했을 때 실용적인 선택이 될 수 있다.

7.1 복구 시간

그림 5는 복구해야 하는 패스키의 개수에 따른 키복구 시간을 나타낸다. 5.2절에서 분석했듯이, 계산량은 멤버가 키복구 알고리즘을 이용해서 복구하려고 시도하는 패스키와 그룹키에 대한 false positive 비율과 밀접한 연관이 있다. 그림 5는 키복구 시간에 대한 false positive 비율의 영향을 보여준다. False positive 비율이 줄어들수록, 그로부터 야기되는 추가적인 키복구 시간은 0으로 수렴하게 된다. 그러나 키검증 정보의 크기를 무한대로 늘리지 않는 한 false positive 키 생성을 완전하게 막을 수는 없다. 따라서 시스템의 계산 및 통신 비용 사이의 trade-off를 고려해서 키검증 정보의 크기를 결정할 필요가 있다.

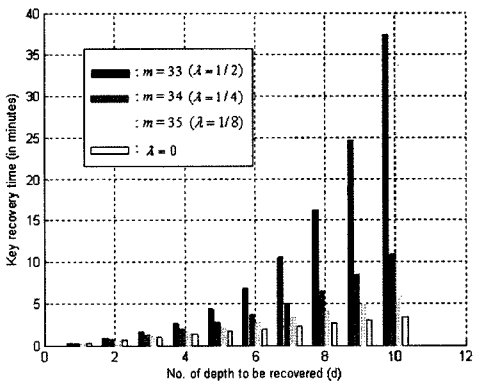


그림 5 키복구 시간($N=2^{10}$, $n=64$)

7.2 안정성

이제 우리는 주어진 키복구 시간에 브로드캐스트 시스템이 지원할 수 있는 안정성의 레벨(w)에 대해서 분석한다. 그림 6은 시스템이 사용자에 요구되는 키복구 시간에 따라 지원할 수 있는 안정성 레벨을 보여준다. 만약 시스템이 사용자로 하여금 그룹키를 5분 안에 복구하기를 요구한다면, $m=35, 34, 33$일 경우 시스템은 각각 400-세션, 50-세션, 10-세션 안정성을 지원할 수 있게 된다.

8. 결 론

본 논문에서는 w-세션 안정성을 보장하는 새로운 w-RGK 그룹키 관리 기법을 제안하였다. 제안한 w-RGK 기법은 w 세션동안 키갱신을 하지 못한 stateless 수신자로 하여금 현재 세션의 키를 복구할 수 있도록 한다. 분석 및 실험 결과에 따르면 제안한 기법은 사용자의

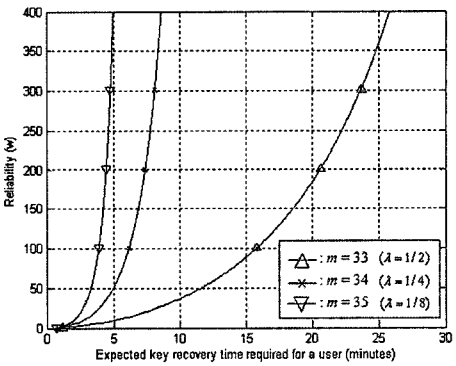


그림 6 다른 키복구 시간에 대한 안정성 레벨($N=2^{10}$, $n=64$)

계산 능력 및 적은 힌트 메시지를 추가로 활용함으로써 안정적이고 확장성있는 키분배를 가능하게 하였다. 제안한 기법은 사용자의 계산 능력, 안정성의 레벨, 통신 비용 등의 시스템 요구사항을 고려해서 다양한 매개변수를 선택함으로써 공공의 순수 브로드캐스트 환경에서의 접근제어시스템에 대한 확장성 있고 유연한 해결책이 될 수 있다.

참 고 문 헌

[1] A. Perrig, J. D. Tygar, *Secure Broadcast Communication in Wired and Wireless networks*, Springer-Verlag, 2002.

[2] A. Fiat and M. Naor, "Broadcast Encryption," *Proc. CRYPTO 1993, Lecture Notes in Computer Science*, vol.773, pp.480-491, 1993.

[3] D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," *Proc. CRYPTO 2001, Lecture Notes in Computer Science*, vol.2139, pp.41-62, 2001.

[4] C. K. Wong, M. G. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," *ACM SIGCOMM*, pp.68-79, 1998.

[5] D. A. McGrew and A. T. Sherman, "Key Establishment in Large Dynamic Groups Using One-way Function Trees," Tech. Rep. 0755, TIS Labs at Network Associates, Inc., Glenwood, Md.

[6] R. Canetti, J. Garay, G. Itkis, D. Miccianancio, M. Naor, and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," *Proceedings of IEEE INFOCOM 1999*, pp.708-716.

[7] A. Perrig, D. Song, and J. D. Tygar, "ELK, a New Protocol for Efficient Large-Group Key Distribution," *Proceedings of IEEE Symposium on Security and Privacy*, pp.247-262, 2001.

[8] C. Blundo, Luiz A. Frota Mattos, and D. R. Stinson, "Generalized Beimel-Chor Schemes for Broadcast Encryption and Interactive Key Distribution," *Theoretical Computer Science*, vol.200,

no.1-2, pp.313-334, 1998.
[9] D. R. Stinson and Tran van Trung, "Some New Results on Key Distribution Patterns and Broadcast Encryption," *Designs, Codes and Cryptography*, vol.14, no.3, pp.261-279, 1998.
[10] S. Rafaeli, D. Hutchison, "A Survey of Key Management for Secure Group Communication," *ACM Computing Surveys*, vol.35, no.3, pp.309-329, 2003.
[11] M. Steiner, G. Tsudik, and M. Waidner, "Cliques: A New Approach to Group Key Agreement," *Proc. International Conference on Distributed Computing Systems*, pp.380-387, 1998.
[12] T. Hardjono and L. R. Dondeti, Multicast and Group Security, first ed., Artech House, 2003.
[13] D. Halevy and A. Shamir, "The LCD Broadcast Encryption Scheme," *Proc. CRYTO 2002, Lecture Notes in Computer Science*, vol.2442, pp.47-60, 2002.
[14] M. J. Mihaljevic, "Reconfigurable Key Management for Broadcast Encryption," *IEEE Communications Letters*, vol.8, no.7, pp.440-442, 2004.
[15] C. Blundo, P. D'Arco, and A. D. Santis, "On Self-Healing Key Distributions Schemes," *IEEE Transactions on Information Theory*, vol.52, no.12, pp.5455-5467, 2006.
[16] http://www.cryptopp.com/benchmarks.html, Crypto++ 5.5 Benchmarks.
[17] T. Krovetz, "Message Authentication on 64-bit Architectures," Selected Areas of Cryptography, Springer, 2006.
[18] M. Abdalla, Y. Shavitt, and A. Wool, "Key Management for Restricted Multicast Using Broadcast Encryption," *IEEE/ACM Transactions on Networking*, vol.8, no.4, pp.443-454, 2000.

부 록

정리 1 증명: 그룹 멤버가 아닌 사용자 u_t는 키갱신 메시지를 복호화하는데에 사용되는 어떠한 키들 뿐 아니라 힌트 메시지를 계산하기 위해 의사난수생성 함수에 사용되는 어떠한 키들도 모른다. u_t가 수신한 키갱신 메시지의 $\forall D^i \in \mathfrak{D}$에 대해서 D^i안의 모든 갱신된 KEK는 그들 각각의 자식 KEK로 암호화 되며, 그것들은 u_t가 알 수 없다. 게다가, 힌트 메시지의 $\forall B^i \in \mathfrak{B}$에 대해서, B^i는 키트리에서 갱신된 각각의 키에 대해서 오직 절반의 정보만을 포함하고 있다. u_t가 B^i를 이용해서 갱신된 키들 중 임의의 키를 복구하기 위해서는 최소한 키의 $n/2$ 비트 정보를 알고 있어야 한다. 그러나 u_t는 키를 전수조사하는데에 필요한 그 정보를 알 수 없기 때문에 u_t는 전송되는 $\mathfrak{D}$와 $\mathfrak{B}$를 가지고서도 sk_t^i의 어떠한 키도 알아낼 수 없다. 그러므로 그룹키를 계산하기 위해서는 $O(2^n)$의 가능성에 대한 전수조사를 수행하는 수 밖에 하는데 이것은 계산적으로 불가능하다. □

정리 2 증명: 우리는 i 세션에 새로 가입한 멤버u_t가 GK^{i-1}를 알 수 없다는 것을 증명한다. GK^{i-1}은 $i-1$ 세션에 새로운 멤버가 가입할 경우 GK^{i-2}로부터 유도되거나 기존 멤버가 탈퇴할 경우 K_2^{i-1}과 K_3^{i-1}로부터 계산 된다. 사용자 u_t는 i 세션에 그룹에 가입할 경우 sk_t^i를 안전하게 받는다. 그러나 sk_t^i는 GK^{i-2}를 포함하고 있는 sk_t^{i-2}의 어떤 키도 유도해 낼 수 없고, 힌트를 계산해서 GK^{i-1}를 유도해 내는데에 사용되는 K_2^{i-1}과 K_3^{i-1}를 포함하는 sk_t^{i-1}의 어떤 키도 유도해 낼 수 없다. 또 다른 공격 시나리오는 자신의 sk_t^i 안의 GK^i를 가지고 $GK^{i-1} = PRF^{-1}(GK^i)$를 계산하는 것이다. 그러나 이것은 의사난수생성함수의 단방향성으로 인해 계산적으로 불가능하다. 따라서 i 세션에 새로 가입한 멤버는 계산적으로 불가능한 $O(2^n)$의 연산을 수행하지 않고서는 이전 세션의 그룹키 GK^{i-1}를 유도해 낼 수 없다. □

정리 3 증명: 우리는 i 세션에 그룹에서 탈퇴한 멤버 u_t가 GK^{i+1}을 알아낼 수 없음을 증명한다. u_t가 그룹을 탈퇴하면 sk_t^i의 모든 패스키 K_j^i는 K_{2j}^{i+1}과 K_{2j+1}^{i+1}을 이용해 갱신된다. u_t가 트리에서 위치하고 있던 가지 쪽의 값은 sk_t^i와는 독립적으로 상향식 패스키 갱신 절차에 따라 갱신되고, 반대 쪽의 값 또한 PRF 함수에 의해서 갱신된다. u_t가 탈퇴 이전에 자신의 패스키에 대한 값들을 알고 있을지라도 그 값들은 탈퇴 이후에 PRF에 의해 모두 갱신되게 된다. 따라서 u_t는 GK^{i+1}를 생성하는데 필요한 sk_t^{i+1}의 어떠한 정보도 갖지 못하게 된다. 그룹키를 찾기 위해 할 수 있는 유일한 방법은 힌트 메시지 안의 키검증 정보를 이용해서 $O(2^n)$개의 가능성에 대한 전수조사를 수행하는 것이지만 이것은 계산적으로 불가능하다. □

허 준 범
정보과학회논문지 : 정보통신
제 36 권 제 4 호 참조

윤 현 수
정보과학회논문지 : 정보통신
제 36 권 제 4 호 참조