제36회 한국정보처리학회 추계학술발표대회 논문집 제18권 제2호 (2011. 11)

모바일 네트워크 환경에서 효율적인 키 갱신메시지 전달을 이용한 그룹키 분배 기법

안병욱, 김강석, 예홍진
아주대학교 지식정보보안학과
e-mail : coriahn@ajou.ac.kr, kangskim@ajou.ac.kr, hjyeh@ajou.ac.kr

A GroupKey Distribution Mechanism using Efficient Key Update Message in Mobile Network Environments.

Byeonguk Ahn, Kangseok Kim, Hongjin Yeh
Dept. of Knowledge Information Security, Graduate School Of Ajou University.

요 약

모바일 기기와 무선통신의 발달로 모바일 기기를 대상으로 하는 다양한 종류의 멀티캐스트 네트워크 기반 멀티미디어 서비스가 증가하고 있다. 이러한 멀티캐스트 통신 환경에서 보안성을 제공하기 위해 사용되는 그룹키를 전달하고 관리하는 것이 중요한 문제가 되고 있다. 또한 연구도 꾸준히 진행 되고 있으며, 본 논문은 기존연구(LKH)를 바탕으로 상대적으로 대역폭이 적은 모바일 환경에서 키-인덱스를 이용하여 키를 갱신하고 동기화하는 메시지의 양을 줄이는 방법을 제시한다.

1. 서론

스마트폰을 비롯하여 태블릿, 노트북 등 모바일 기기가 활성화 되고, YouTube, 다음TV팟, aFreeca 등 멀티미디어의 소비가 활발히 이루어지고 있다. 이들 중 유료로 서비스되는 컨텐츠나, 허가된 사용자만이 참여하는 화상회의의 경우 보안은 중요한 문제가 된다.

기존의 데이터 전송방법은 서버와 클라이언트 간에 1:1로 암호화해서 전송 하는 방법이 사용되었으나, 사용자가 많아질수록 서버와 네트워크에 부하가 커지게 된다[7].

멀티캐스트 통신을 이용함으로써 서버와 네트워크에 대한 부하를 해결 하였다. 하지만, 동일한 데이터를 여러 경로로 동시에 전송하고, 정해진 대역의 주소를 사용하는 특성상 데이터에 대한 접근과 수집이 용이하고, 보안에 취약하다. 이에 따라 안전한 데이터 전송을 위해 암호화가 요구되고, 송수신자가 공유하게 되는 키를 그룹키라 하고, 이에 대한 연구가 활발히 이루어지고 있다.

안정적인 보안성을 제공하기위해 그룹키는 그룹 멤버의 변화가 있을 때마다 변경되어야 한다. 일반적으로 그룹키 관리 기법의 안전성 평가는 아래와 같다[6].

Backward secrecy: 멤버가 추가되었을 때 새로운 멤버는 참여하기 이전의 데이터를 복호화 할 수 없다.
Forward secrecy: 탈퇴한 멤버는 탈퇴 이후의 데이터를 복호화 할 수 없다.

그룹이 커지고, 멤버수가 증가하게 되면 키를 갱신하는 메시지를 전송하기 위한 메시지가 증가하고, 네트워크 소모가 많아지게 된다.

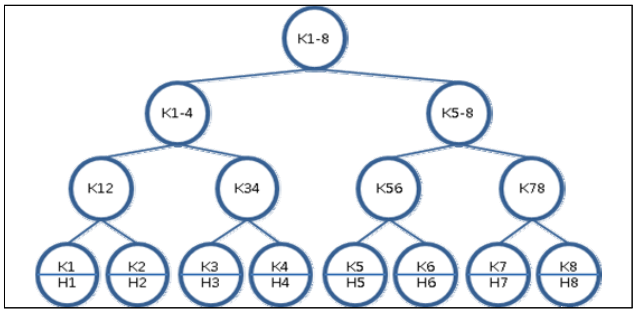
본 논문은 2장에서 기존의 키 갱신 알고리즘을 알아보고, 3장에서 유선 네트워크에 비해 대역폭이 낮은 무선 네트워크를 위해 전송되는 키 갱신 메시지의 크기를 줄이기 위한 개선방안을 제시한다. 4장에서 기존 알고리즘과 비교하여 효율성을 검토하고, 5장에서 결론을 맺는다.

2. 그룹키 관리 기법의 기존 연구

그룹키 관리를 위해서 키를 갱신하는 경우 논리적 키 계층(LKH : Logical Key Heterarchy)[1]가 효율적이기 때문에 이를 기반으로 한 연구들이 다양하게 있다[5]. LKH를 기반으로 One-way Function을 이용하여 멤버 추가에 대한 키 갱신 메시지를 줄인 LKH+[2]와 OFT(One-way Function Tree)[3], PRG(Pseudo-Random-Generator)를 이용하는 방식[4] 등이 있다.

LKH는 (그림 1)과 같이 이진 형태의 키트리를 형성 하고, 그룹 멤버는 각 키트리의 말단 노트와 대응되고, 트리상의 모든 노드는 개개의 키를 가지게 된다. 호스트는 자신으로부터 루트노드 사이에 경로키를 가지게 된다. 예를 들어 호스트 H1은 {K1, K12, K1-4, K1-8} 4개의키를 가지게 된다. 이들 중 K12와 K1-4의 경우 그룹키를 암호화해서 전달하기위한 키 암호화키(KEK : Key Encryption Key)[6]이 된다.

그룹 멤버가 참여 혹은 탈퇴 하는 경우 보안을 유지하기 위해 그룹키를 비롯하여 해당 멤버에서 루트 노드에 이르는 경로상의 키를 변경하고, 변경된 키들에 영향을 받는 노드에 변경된 키를 전달한다.



(그림 1) 8개의 호스트를 가지는 이진 키 트리

구성원이 추가 탈퇴할 때의 LKH와 LKH+의 키 변화와 키 갱신과정은 다음과 같다[6].

● LHK

(그림 1)에서 H6에 대응하는 멤버에 변화가 생겼을 때.

① 경로상의 키{K1-8, K5-8, K56}를 {K′1-8, K′5-8, K′56}로 변경
② H5에 K′56를 K5로 암호화해서 전달
③ H5에 K′5-8를 K′56로 암호화해서 전달
④ H7, H8에 K′5-8를 K7-8로 암호화해서 전달
⑤ H5, H7, H8에 K′1-8를 K′5-8로 암호화해서 전달
⑥ H1, H2, H3, H4에 K′1-8를 K1-4로 암호화해서 전달

❖ N : 멤버의 수, K : 키의 길이
❖ ②③④⑤⑥에서 키 전송이 일어난다.
❖ $(2(\log_2 N)-1)K = (2d-1)K$

● LHK+

기존의 정보를 가지고 One-Way HASH 함수를 통해 키를 갱신하기 때문에 멤버 탈퇴에는 적용할 수 없다.
그림 1에서 H6에 대응하는 멤버가 추가 될 때.

① H6의 인덱스(I-H6)를 각 멤버에 전달
② H6에 K′1-8과 K′5-8를 K6로 암호화해서 각각 전달
③ H6에 K′56을 K6로 암호화해서 전달
④ H5에서 h(K56)을 이용해 K′56을 갱신
⑤ H5, H7, H8 에서 h(K5-8)를 이용해 K′5-8를 갱신
⑥ H1, H2, H3, H4, H5, H7, H8에서 h(K1-8)를 이용해 K′1-8를 갱신

❖ ①에서 추가된 멤버의 위치 값(I-H6)전송이 일어난다.
❖ ②③에서 키 전송이 3회 일어난다.
❖ **$(\log_2 N)K+I = dK+I$**

3. 제안방식(LKH+Ki)

기존의 알고리즘에서는 키를 암호화해서 직접 전달하는 방법을 사용하여, 키를 전달하는 횟수를 줄이는데 초점이 맞춰져 있으나, 본 논문에서 제안하는 방식(LKH+Ki)은 키의 인덱스 값을 전달하여 키 갱신에 필요한 메시지의 크기를 줄이는데 중점하고 있다.

키 관리서버(GKMS: Group Key Management System)로부터 그룹키를 One-Way HASH함수를 이용하여 8배수의 HASH 값을 생성한다. 이를 다시 8bit 단위로 잘라 블록을 생성하고, 키 길이단위로 잘라서 8개의 키 블록 세트를 생성 한다. 이로 인해 8*(K/8)의 배열이 생성한다. 그리고 GKMS로부터 전달 받은 키-인덱스 값으로 단위블록을 조합하여 키를 생성하고 서버와 동기화 하게 된다.

1개의 단위 블록을 키-인덱스로 표현함으로써 단위블록의 크기 8bit를 키-인덱스의 크기 3bit로 줄어들어 전송되는 전체 데이터의 크기를 줄 일수 있다.

● 예시

```
01000101 01010010 10101010 … 10101010 K
10101010 01010101 01010101 … 01010101 h(K)
00101101 10101011 10100111 … 10010010 h(h(K))
01000101 00100100 01001001 … 01010100 h(h(h(K)))
01111011 01010110 10010110 … 10101010 h(h(h(h(K))))
11101001 00010010 10101010 … 01010110 h(h(h(h(h(K)))))
11101101 10001010 10101110 … 10001010 h(h(h(h(h(h(K))))))
01010111 11010000 11100011 … 11110001 h(h(h(h(h(h(h(K)))))))

키-인덱스 : 100 010 110 … 011

Key : 01111011 10101011 10101110 … 01010100
```

(그림 2) 키 생성 방식

(그림 2)는 클라이언트는 GKMS로부터 전달 받은 그룹키(K : 01000101 01010010 10101010 … 10101010)로부터 One-Way HASH 함수를 이용하여 8배수의 HASH 값을 생성하고, 키-인덱스(100 010 110 … 011)를 이용하여 해당위치에 있는 블록을 조합하여 새로운 키(01111011 10101011 10101110 … 01010100)를 생성한다. 서버에서도 동일한 방법으로 키를 생성하면 서버와 키를 동기화 한다.

LKH+Ki에서의 구성원이 추가되고 탈퇴할 때의 키 변화와 갱신과정은 다음과 같다.

● 멤버 추가

① H6의 인덱스(I-H6)를 각 멤버에 전달
② H6에 그룹키(K'1-8)를 K6으로 암호화해서 전달
③ 각 멤버에서 그룹키(K'1-8)이용하여 One-Way HASH 함수로 8배수 HASH 값 생성
④ H6에 Ki′5-8, Ki′56을 K6로 암호화해서 전달
⑤ H5에서 h(Ki56)을 이용해 Ki′56을 갱신
⑥ H5, H7, H8 에서 h(Ki5-8)를 이용해 Ki′5-8를 갱신
⑦ H1, H2, H3, H4, H5, H7, H8에서 h(Ki1-8)를 이용해 Ki′1-8를 갱신

❖ Ki(Key-Index): 8배수의 HASH 값에서 키블록의 좌표
❖ ①에서 추가된 멤버의 위치 값(I-H6)이 전달, ②에서 키 전송, ④에서 키-인덱스의 전송
❖ $K+(\log_2 N-1)Ki+I = K+(d-1)Ki+I$

● 멤버 탈퇴

① 경로상의 키-인덱스{Ki1-8, Ki5-8, Ki56} 를 {Ki′1-8, Ki′5-8, Ki′56}로 변경

② H5에 Ki′56, Ki′5-8를 K56로 암호화해서 전달

③ H7, H8에 Ki5-8를 K7-8로 암호화해서 전달

④ H5, H7, H8에 Ki′1-8를 K′5-8로 암호화해서 전달

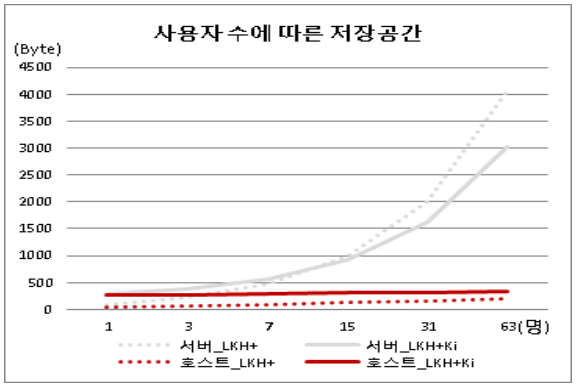
⑤ H1, H2, H3, H4에 K′1-8를 K1-4로 암호화해서 전달

❖ ②③④⑤에서 키-인덱스의 전송

❖ $(2\log_2 N-1)Ki = (2d-1)Ki$

4. 효율성 분석

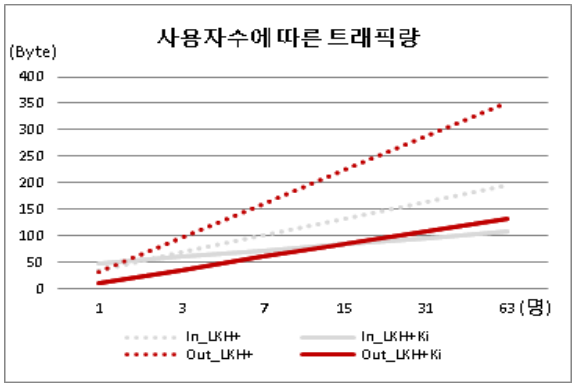
2장에서 소개된 기존 알고리즘 LKH+와 이 논문에서 제안하는 LKH+Ki에서 사용되는 저장 공간과 네트워크 트래픽 량을 비교한다.



(그림 3) 사용자 증가에 따른 저장 공간 비교

	키관리 서버	클라이언트
LKH	(2d-1)K	dK
LKH+Ki	K(KS+N)+(2d-1)Ki	K*KS +dKi

(표 1) 서버와 클라이언트의 저장공간



(그림 4) 사용자 증가에 따른 네트워크 추이

	구성원 추가	구성원 탈퇴
LKH	dK+I	(2d-1)K
LKH+Ki	K+dKi+I	(2d-1)Ki

(표 2) 구성원 변화에 따른 키 갱신 메시지 크기

위의 표에서는 아래의 기호와 값을 사용한다.

N : 사용자의 수(EA)
K : 키 의 크기(256 bits)
d : 트리의 깊이($d=\log_2 N$)
I : 호스트-인덱스 사이즈(32bits)
KS : Key-Set 개수(8EA)
Ki : 키-인덱스 사이즈(96bits)
($Ki=\log_2 KS/KB*K$)
KB : Key-Block 사이즈(8bits)

LKH+Ki에서 8배의 HASH 값을 저장하게 되어 (그림 3)과 같이 초기 메모리의 소비가 큰 편이나, 사용자의 수가 증가하게 되면, 작은 크기의 키-인덱스를 저장하는 LKH+Ki와 기존 시스템이 큰 차이가 없거나 좀더 줄어든 것을 볼 수 있다. 더불어 (그림 4)에서는 키 갱신에 사용되는 메시지의 크기가 기존 시스템에 비해 확연하게 줄어 든 것을 알 수 있다.

5. 결론

본 논문에서 제시한 LKH+Ki는 기존 LKH+와 비교하여 그룹 관리자와 그룹 구성원의 키 저장 공간은 크게 증가되지 않으면서, 키 갱신 메시지의 크기를 줄임으로 인해 무선 네트워크에서 키 갱신과 GKMS와 키를 동기화 속도를 향상시킬 수 있을 것으로 기대 된다.

하지만, 클라이언트에서 8배의 HASH값을 생성하고, 키를 조합하기위한 연산이 늘어난다는 단점과 안정성에 관한 연구가 더 필요하다.

참고문헌

[1] H. Harney, E. Harder, "Logical key hierarchy protocol", IETF Internet draft, 1999.

[2] M. Waldvogel, G. Caronni, "The VersaKey Framework: Versatile Group Key Management", IEEE Journal on Selected Areas in Communications, 1999

[3] T. Sherman, David A, "Key establishment in large dynamic groups using one-way function trees," IEEE Transactions on Software Engineering, 2003

[4] R. Canetti, J. Garay, G, Itkis, D. Micciancio, M. Naor, B Pinkas, "Multicast security: a taxonomy and some efficient constructions," IEEE INFOCOM 99, pp. 708-716, 1999.

[5] 권정옥, 황정연, 김현정, 이동훈, 임종인, "일방향 함수와 XOR을 이용한 효율적인 그룹키 관리 프로토콜: ELKH", 정보보호학회 논문지, 제 12권 제 6호, 2002.

[6] 신승재, 허준범, 이한진, 윤현수, "클러스터화된 무선 네트워크에서 전송량을 고려한 효율적인 멀티캐스트 키 관리 기법", 정보과학회논문지, 정보통신 제 36권 제 5호, 2009.

[7] 임효준, 김종권, "계층 비디오 멀티캐스트를 위한 효율적인 키 분배 방법", 정보과학회논문지, 정보통신 제 27권 제 4호, 2000.