다중 코어를 가지는 CBT에서의 그룹키 관리

조태남*0 김상희* 은상아* 이상호* 채기준* 박원주** 나재훈**
*이화여자대학교 컴퓨터학과, **ETRI 정보보호기술연구본부
{tncho, kshee0, ivory, shlee, kjchae}@ewha.ac.kr, {jwpark, jhnah}@etri.re.kr

Group key management using CBT with multi-core

Tae-Nam Cho*0 Sang-Hee Kim* Sang-A Eun* Sang-Ho Lee* Kijoon Chae* Won-Ju Park** Jae-Hoon Na**
*Dept. of Computer Science and Engineering, Ewha Womans University
**Network Security Department, ETRI

요 약

원격화상회의나 소프트웨어 배포 등 멀티캐스팅의 보안을 위해서 사용하는 그룹키는 그룹의 규모가 클 경우에도 멤버쉽 변화에 대하여 효율적으로 갱신이 이루어져야 한다. 본 논문에서는 DEP 구조를 멀티캐스트 프로토콜인 CBT에 적용한 2계층 관리 구조를 제안함으로써 키갱신 메시지 전송의 암·복호화 횟수를 제한하였으며, 서브 그룹 관리자를 그룹 통신으로부터 배제할 수 있는 효율적인 키관리 프로토콜을 제시하였다.

1. 서 론

멀티캐스트와 같은 통신 기술의 발달로 원격회의나 실시간 정보 서비스, 유료 영상 서비스와 같은 그룹통신 응용의 개발이 성장하고 있다. 이들 응용에서 전송되는 데이터는 비밀 정보이거나 유료 정보로서 정당한 수신자에게만 접근이 허용되는 경우가 많다. 이러한 접근제어의 수단으로서 그룹의 멤버들만이 그룹키를 공유하고 그룹키를 이용한 암호화 통신을 한다. 멤버의 가입과 탈퇴가 발생할 경우에는 가입 멤버가 가입 이전의 키를 알 수 없고, 탈퇴한 멤버는 탈퇴 이후의 키를 알 수 없도록 하는 BS(Backward Secrecy)와 FS(Forward Secrecy)[1,2]를 만족시키기 위하여 그룹키를 갱신해야 하는데, 그룹의 규모가 커지고 빈번한 멤버쉽 변동이 있을 경우에는 키갱신으로 인한 과부하로 그룹통신의 성능을 저하시키게 된다.

그룹키 관리 프로토콜은 다양한 특성을 가지므로 응용의 특성 및 보안 요구사항에 적절한 구조와 키관리 프로토콜을 사용하는 것이 바람직하다. 본 논문에서는 확장성과 효율성을 위하여 다중 코어를 가지는 CBT(Core Based Tree)에 기반한 그룹키 관리 구조를 설계하였다. CBT의 각 노드들의 역할과 특성을 분석하여 그들에 적합한 프로토콜을 수행하는 키관리 구조 및 프로토콜을 제안한다.

2. 관련 연구

2.1 CBT 프로토콜[3,4]

CBT 프로토콜은 특히 수신자가 넓은 지역에 분포하면서 밀집되어 있지 않은 경우에 적합한 멀티캐스트 프로토콜이다. 이 프로토콜은 송신자들과 상관없이 코어(core)라고 불리는 특정 노드를 중심으로 그룹에 유일한 배달트리인 CBT가 형성되며, 수신자의 요구에 의해 트리가 확장되는 특성을 가진다. 코어는 CBT의 루트로서 데이터 전송의 중심이 된다. 코어는 하나의 주코어(primary core)와 여러 개의 부코어(secondary cores)들로 구성될 수 있는데, 부코어는 일반적으로 정적으로 선정된다. 멀티캐스트 효율성을 극대화시키기 위한 부코어 선정 방법들이 여러 가지 연구되어 있으며, 다양한 상황에서 CBT가 효율적인 것으로 분석되었다[5].

CBT는 새로운 멤버의 가입 절차를 그룹키 관리에 이용하기에 적합한 특성을 가지고 있지만, 별도의 그룹키 관리자 없이 CBT에 기반한 [4]는 BS와 FS를 만족하지 못하며, 이를 개선한 프로토콜 [6]은 BS는 만족하지만, FS는 만족하지 못한다.

2.2 DEP(Dual Encryption Protocol)[7]

DEP에서는 하나의 그룹 관리자 GM이 있고, 그룹을 서브그룹 S_i들로 분할하여 서브그룹 관리자 SGM_i이 이를 관리하도록 함으로써 멤버쉽 변동의 영향을 지역화 하였다. 또한 SGM들을 그룹통신으로부터 배제하기 위하여 GM과 멤버들 사이에 공유하는 별도의 키 KEK_j(Traffic Encryption Key)를 도입하였다. 이들의 키공유 관계는 그림 1과 같다. GM이 각 SGM_i에게 그룹키 TEK(Key Encryption Key)를 $\{\{TEK\}_{KEK_i}\}_{LS_i}$와 같이 암호화하여 보내면, 각 SGM_i들은 이를 복호화한 후, $\{\{TEK\}_{KEK_i}\}_{LS_i}$로 재암호화하여 S_i에게 보낸다. 그림 1에서 SGM_4와 같이 서브 그룹 관리자가 다른 서브 그룹의 일원일 수 있는데, S_4의 멤버들은 TEK 수신을 위해 세 번의 암·복호화 과정을 거쳐야만 한다. 즉, 계층구조의 깊이에 비례하여 암복호화 비용도 증가한다. 키의 변경시에는 관리자가 새로운 키를 생성하여 대상자에게 일대일로 암호화하여 송신하므로, 멤버의 수가 n이라 할 때 $O(n)$의 복잡도를 가진다. 새로운 멤버가 등록할 때에는 가입 요청 메시지에 응답하는 임의의 서브 그룹 관리자의 서브 그룹에 가입하게 되며, 두 개 이상의 서브 그룹에 이중 가입할 수 있기 때문에 관리자가 모든 멤버들을 관리하면서 가입 허가를 해야 한다. 이것은 등록 절차를 복잡하게 할 뿐 아니라 탈퇴한 멤버들의 리스트를 유지해야 하는 단점

이 있다.

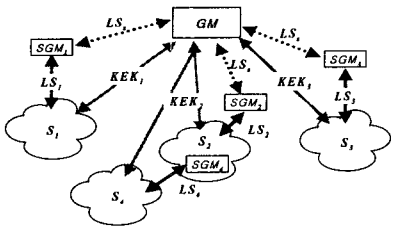

그림 1 DEP의 키공유 관계

2.3 HKTM(Height-balanced Key Tree based Management)[8]

이것은 하나의 코어를 가지는 CBT를 이용한 키관리 기법으로서 멤버의 가입과 탈퇴에 대한 효율성을 제공하기 위한 것이다. 관리자가 (2,4)-트리로 구성된 키트리를 이용함으로써 멤버의 가입과 탈퇴시의 키갱신을 $O(\log n)$에 보장한다. 관리자는 CBT의 코어로서, 멤버의 가입 메시지의 경로를 따라 수집한 정보를 토대로 하여 RT(Reduced Tree)를 유지하고, 이를 키트리에 반영한다. 이것은 키트리에 멤버들의 네트웍 구성 정보를 반영한 것으로써, 네트웍 장애로 일부 멤버가 탈퇴되거나 장애 복구로 재가입될 때에도 $O(\log n)$에 키갱신을 할 수 있도록 보장해 준다.

2.4 VersaKey[9]

이 방법에서는 각 멤버들이 자기 ID의 각 비트에 대응되는 키를 소유한다. 예로, 그림 2에서 ID=0100_2인 멤버는 TEK와 KEK3.0, KEK2.1, KEK1.0, KEK0.0을 소유한다. 관리자는 키 테이블을 소유하므로 $2\lceil \log n \rceil + 1$개의 키만 저장하면 된다. 멤버가 탈퇴할 경우에는, 새로운 키를 그 멤버가 소유하지 않은 키들로 암호화하여 전송한다. 이 방법에서는 멤버의 ID가 미리 결정되어야 하기 때문에 멤버가 고정되어 있거나 미리 예상할 수 있을 때 적합하다. 또한 ID가 서로 보수인 2명의 멤버가 결탁할 경우에는 모든 키가 노출되는 위험이 있다.

	TEK	
ID Bit #0	KEK0.0	KEK0.1
ID Bit #1	KEK1.0	KEK1.1
ID Bit #2	KEK2.0	KEK2.1
ID Bit #3	KEK3.0	KEK3.1
	value=0	value=1

그림 2 VersaKey의 키 테이블

3. 다중 코어를 가지는 CBT 기반의 키관리 구조 제안

멤버쉽의 변동이 빈번한 경우에는 이로 인한 그룹키 갱신이 잦게 되어 효율성을 저하시킨다. 그러므로 확장성을 제공하면서 멤버쉽 변동으로 인한 키갱신의 범위를 줄이는 것이 효율적이다.

이러한 효율성 제공을 위한 방안으로서 DEP가 제안되었다. 이 장에서는 DEP의 단점을 보완하기 위하여 하나의 주코어와 여러 개의 부코어를 가지는 CBT에 변형하여 적용하고, 이에 따른 서브 그룹 관리자 및 멤버들의 특성을 분석하여 HKTM과 VersaKey를 접목시킨 키관리 구조 및 프로토콜을 제안한다.

3.1 관리 구조

제안하는 관리 구조에서는 주코어를 관리자 GM으로 설정하고, 각 부코어들을 서브그룹 관리자 SGM_i($1 \le i \le$부코어의 수)로 설정한다. GM은 TEK, KEK_{ij}($1 \le i \le$서브그룹의 수, j=1,2)와 LS_s의 관리 및 멀티캐스트 데이터의 전송을 담당한다. SGM_i들은 서브 그룹 S_i의 멤버쉽 관리와 서브그룹키 LS_i의 관리를 담당한다. 이들은 멤버 가입을 하지 않는 한, 멀티캐스트 데이터에 대한 수신 권한을 가지고 있지 않는다. SGM이 아닌 모든 멤버는 CBT 프로토콜을 통하여 가장 가까운 하나의 서브 그룹 관리자 SGM_i을 찾아 S_i에만 가입되며, KEK_{i1}, KEK_{i2}와 LS_i를 소유한다(단, SGM인 멤버는 KEK_{i1}만을 소유한다). 이 구조는 DEP의 깊이를 2로 고정시킴으로써 키관리 메시지의 암복호화 비용을 줄이며, SGM들만 멤버쉽 관리를 하도록 함으로써 관리자가 GM이 멤버 리스트를 유지할 필요없이 SGM들의 리스트만 유지하도록 한다. 그림 2는 CBT에서의 서브 그룹의 구성 예를 나타내고, 그림 3은 이에 대응되는 서브 그룹키의 공유 관계를 나타낸다.

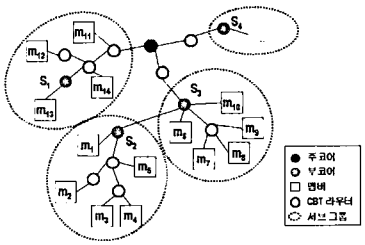

그림 2 CBT에서의 서브 그룹

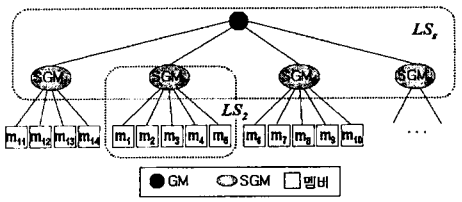

그림 3 서브 그룹키 공유 관계

3.2 키관리 프로토콜

멤버쉽 변동이 빈번한 대규모 그룹에서, 이 때마다 그룹의 모든 멤버가 키를 갱신한다거나 DEP에서와 같이 일대일로 새로운 키를 분배한다면 매우 비효율적일 것이다. 제안하는 구조에서는 SGM_i가 HKTM을 이용하여 LS_i를 관리한다. 즉, LS_i는 서브그룹 S_i를 관리하기 위한 키트리의 루트이며, S_i의 각 멤버는 $O(\log|S_i|)$만큼의 키를 소유한다.

SGM들은 GM의 정책에 의해 사전에 지정된 노드들로서 서브 그룹 관리자 역할에 대한 빈번한 가입·탈퇴는 없지만, 네트웍 장애 등으로 인한 변동은 고려해야 한다. 한편 SGM들은 그룹 통신 데이터에 접근할 수 있는 노드들이 아

니기 때문에 서로 결탁할 가능성이 매우 낮다. 그러므로 GM의 정책에 따라 사전에 SGM들의 수를 결정하여 이에 따라 ID를 할당하고, GM이 VersaKey 방식을 이용하여 관리함으로써 네트웍 장애와 복구에 대비하여 일부 서브그룹 관리자를 배제한 키갱신을 수행할 수 있도록 한다.

멀티캐스트 데이터를 암호화하는 그룹키 TEK는 GM이 $\{\{TEK\}_{KEK_{i1}}\}_{LS_i}$로 각 SGM_i들에게 전송하고, 각 SGM_i는 $\{\{TEK\}_{KEK_{i1}}\}_{LS_i}$로 바꾸어 멤버들에게 전송한다.

3.3 가입 및 탈퇴 프로토콜

일반 멤버가 가입하거나 탈퇴할 때는 새로운 멤버가 소속될 서브 그룹이나 탈퇴하는 멤버가 소속되어 있던 서브 그룹키 LS_i만 갱신한다. TEK는 LS_i와 KEK_{i1}를 모두 알아야 얻을 수 있으므로 FS와 BS는 보장된다. 이것은 키갱신의 범위를 하나의 서브 그룹으로 국한시키고, $O(\log|S_i|)$개의 메시지만으로 갱신할 수 있도록 한다. 한편, SGM들은 그룹 형성시에 관리자에 의해 지정되는 노드들이기 때문에 임의의 멤버가 SGM이 될 수는 없지만, 기존의 SGM이 멤버로서 가입·탈퇴는 가능하다. 멤버였던 어떤 SGM_i가 탈퇴하고 서브 그룹 관리자의 역할만 하고자 할 경우에는 SGM_i가 알고 있던 KEK_{i1}을 변경해 주어야 한다.

SGM이나 일반 멤버들의 가입과 탈퇴에 따른 LS_i와 KEK_{i1}의 갱신 프로토콜에서 사용되는 표기는 다음과 같다.

$join-req$	멤버 가입 신청 메시지
$id_x, cert_x, addr_x$	x의 식별자, 공개키 인증서, 주소
nc_x	x가 생성한 난수
U_x, R_x	x의 공개키와 개인키
$A \rightarrow B: msg$	A가 B에게 msg를 송신
LS_{ix}	LS_i를 루트로 하는 HKT_i에서 멤버 x가 소유해야 할 서브 그룹키 집합
LS_i^*, LS_{ix}^*, KEK	갱신된 LS_i, LS_{ix}, KEK_{i1}
$\{msg\}_{key}$	msg를 key로 암호화한 암호문
$\{msg\}^{key}$	msg와 msg를 key로 서명한 서명문
$\{LS_{ix}^*\}_{LS_{ix}}$	HTKM의 방식으로 LS_{ix}를 이용하여 LS_{ix}^*를 암호화한 메시지

■ SGM이 아닌 멤버 m이 S_i에 가입

$m \rightarrow SGM_i: \{join-req, id_m, nc_m, addr_m\}^{R_m}, cert_m$

$SGM_i \rightarrow m: \{\{LS^*_{im}\}_{U_m}, nc_m, nc_{SGM_i}, addr_{GM}\}^{R_{SGM_i}}, cert_{SGM}, cert_{GM}$

$SGM_i \rightarrow m'$ (모든 $m' \in S_i$): $\{LS^*_{im'}\}_{LS_{im'}}$

$SGM_i \rightarrow GM: \{\{join-req, id_m, nc_m, addr_m\}^{R_m}, nc_{SGM_i}\}^{R_{SGM_i}}, cert_m, cert_{SGM_i}$

$GM \rightarrow m: \{\{id_m, nc_m, nc_{SGM_i}, KEK_{i1}, KEK_{i2}\}_{U_m}\}^{R_{GM}}$

■ $SGM_i(=m)$이 S_i에 멤버로 가입

$SGM_i \rightarrow GM: \{\{join-req, id_m, nc_m, addr_m\}^{R_m}\}, cert_m$

$GM \rightarrow m'$ (모든 $m' \in S_i$): $\{KEK_{i1}^*\}_{KEK_{i2}}$

$GM \rightarrow m: \{\{id_m, nc_m, KEK_{i1}^*\}_{U_m}\}^{R_{GM}}, cert_{GM}$

■ SGM이 아닌 S_i의 멤버 m이 탈퇴

$SGM_i \rightarrow m'$ (모든 $m' \in S_i$ & $m' \neq m$): $\{LS^*_{im'}\}_{LS_{im'}}$

■ SGM_i가 멤버 탈퇴

$GM \rightarrow SGM_i: \{\{KEK^*_{i1}\}_{KEK_{i2}}\}_{LS_i}$

$SGM_i \rightarrow S_i: \{\{KEK^*_{i1}\}_{KEK_{i2}}\}_{LS_i}$

KEK_{i2}를 갱신하는 이벤트는 없으므로 정책에 따라 주기적으로 $\{KEK^*_{i2}\}_{KEK_{i2}}$ 와 같이 키를 갱신할 수 있다.

4. 결론

동적 그룹에서 멤버쉽 변동으로 인한 그룹키 갱신의 범위를 줄이는 것은 대규모 그룹으로의 확장에 매우 중요한 요소이다. 본 논문에서는 서브 그룹 구조를 가지면서 서브 그룹 관리자의 멀티캐스트 데이터 접근을 통제할 수 있는 DEP와 다중 코어를 가지는 CBT를 접목하였다. 이에 따라 각 관리자의 역할을 재정의하고 각 서브 그룹별 특성을 분석하여 적절한 프로토콜을 적용함으로써, 키갱신의 효율성을 증가시키고 그룹 관리자가 유지해야 하는 정보의 양을 줄였다. 특히, 서브 그룹마다 두 개의 KEK를 유지하도록 함으로써 갱신 비용이 가장 컸던 KEK의 갱신을 매우 간단하고 효율적으로 수행할 수 있게 되었다.

5. 참고문헌

[1] C. K. Wong, M. Gouda, S. S. Lam, "Secure Group Communications using Key Graphs," Proc. of ACM SIGCOMM, 1988.

[2] A. Perrig, D. Song and J. D. Tygar, "ELK, a New Protocol for Efficient Large-Group Key Distribution," 2001 IEEE Symposium on Security and Privacy, 2001.

[3] A. Ballardie, "Core Based Trees(CBT) Multicast Routing Architecture," IETF RFC 2201, 1997.

[4] A. Ballardie, "Scalable Multicast Key Distribution," IETF RFC 1949, 1996.

[5] K. L. Calvert, E. W. Zegura, M. J. Donahoo, "Core Selection Methods for Multicast Routing," IEEE IC3N, 1995.

[6] 김봉한, 이재광, "그룹 통신을 위한 멀티캐스트 키 분배 프로토콜 설계 및 검증," 정보보호학회 논문지, 제 10권, 제 2호, 2000.

[7] L. R. Dondeti, S. Mukherjee, A. Samal, "Scalable Secure one-to-many Group Communication using Dual Encryption," Proc. of IEEE International Symposium on Computer Communication, 1999.

[8] 조태남, 이상호, "(2,4)-트리를 이용한 그룹키 관리," 정보보호학회 논문지, 제 11권, 제 4호, 2001.

[9] M. Waldvogel, G .Caronni, D. Sun, N. Weiler and B. Plattner, "The VersaKey Framework: Versatile Group Key Management," IEEE Journal on Selected Areas in Communications, 1999.