다자간 원격 회의시스템에서 보안을 위한 효율적인 그룹 키교환 방식

○
이 인 영, 곽 재 창
서경대학교 컴퓨터과학과

Efficient Group-Key Distribution Method For Security in Remote Conference System

In-Young Lee, Jae-Chang Kwak
Dept. of Computer Science, Seokyeong Univ.

요 약

m-to-m 그룹 통신상의 보안은 기존의 m-to-1 통신에서는 고려되지 않았던 보안 문제가 발생한다. 이중에서도 통신데이터정보의 비밀을 보장하기 위한 암호화에 사용되어지는 키교환을 어떻게 할 것인가에 대하여 해결해야한다. m-to-m 통신상의 데이터 암호화는 통신에 참여하는 모든 멤버가 동일한 키를 가져야 한다. 따라서 본 논문에서는 Diffie-Hellman 알고리즘을 이용한 그룹상의 키교환 방식을 살펴본 후 효율성면에서 개선한 그룹 키교환 방식을 제안한다.

1. 서론

기업과 같은 단체에서 의사 교환의 수단으로서 사용되는 회의 시스템은 인터넷의 발달로 전세계 어디에서나 사용 가능하게 되었으며 보안이 요구된다. 클라이언트와 서버간의 통신은 m-to-1 통신으로 여러 사용자가 하나의 서버에 접속하여 각각 서로 다른 정보를 교환하며 데이터의 비밀을 보장하기 위하여 서로 다른 키를 사용하여 암호화 통신을 한다. 하지만 회의 시스템과 같이 m-to-m 통신을 하는 시스템은 여러 사용자가 하나의 서버를 사용하여 정보를 공유하고 서버는 회의에 관여할 수 없다. 따라서 회의에 참여하는 모든 멤버는 동일한 키를 사용하여 암호화 통신을 하고 회의 시스템을 담당하는 서버는 그 키를 알 수 없어야 한다.[5] 이와 같은 그룹 통신에서는 그룹 키교환 방식이 필요하며 대표적인 키교환 알고리즘인 Diffie-Hellman 알고리즘은 그룹 구성원이 n명일 경우 키 동의를 위하여 각 멤버가 n-1번의 정보 교환이 필요하므로 많은 시간이 요구된다. 본 논문에서는 이를 해결하기 위하여 Hughes의 방식을 이용한다. 2장에서는 Diffie-Hellman의 그룹 키교환 방식의 비효율적인 면에 대해서 설명하고 3장에서는 Hughes의 키 교환 방식에 대해서 살펴본 후에 4장에서 Hughes의 방식을 이용하여 Diffie-Hellman 방식을 효율성 면에서 개선한 그룹 키교환 방식에 대해서 설명한다.

2. Diffie-Hellman Algorithm

Diffie-Hellman Algorithm[1][3]은 1-to-1 키교환을 기본으로 하고 있으며 A, B는 각각 개인값 A_k, B_k를 가지며 g, n 값을 공유한 후 A는 식(1)을 연산한 값을 B에게 전달하고, B는 식(2)를 연산한 값을 A에게 전달한다.

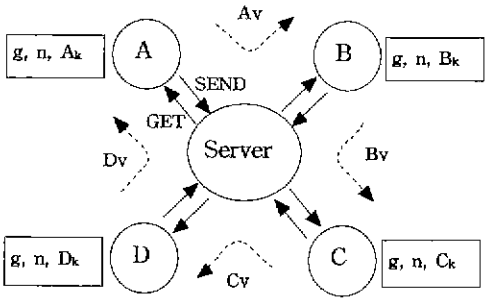
$$A_v = g^{Ak} \bmod n \quad \text{--- (1)}$$
$$B_v = g^{Bk} \bmod n \quad \text{--- (2)}$$

A와 B는 (1),(2)식의 계산 값인 A_v, B_v를 교환한 후에 아래의 (3),(4)연산을 수행하여 키값인 k, k'를 구한다. k와 k'는 동일한 연산을 수행한 결과이므로 같은 값을 갖는다. : $(k = k' = g^{AkBk} \bmod n)$

$$A : k = B_v^{Ak} \bmod n \quad \text{--- (3)}$$

B : $k' = A_v^{Bk} \bmod n$ --- (4)

위의 Diffie-Hellman Algorithm을 이용한 그룹 키교환은 아래와 같은 방식으로 수행하며 서로의 정보교환은 서버를 통해 이루어진다.



A : ① SEND : $A_v = g^{Ak} \bmod n$
GET : $D_v = g^{Dk} \bmod n$
② SEND : $A_v = D_v^{Ak} \bmod n$
GET : $D_v = C_v^{Dk} \bmod n$ ($C_v = g^{Ck} \bmod n$)
③ SEND : $A_v = D_v^{Ak} \bmod n$
GET : $D_v = C_v^{Dk} \bmod n$
($C_v = B_v^{Ck} \bmod n$, $B_v = g^{Bk} \bmod n$)

A의 키값 계산 :
$k = Dv^{Ak} \bmod n$
$= g^{BkCkDkAk} \bmod n$

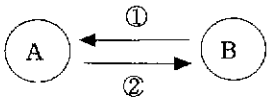
B : ① SEND : $B_v = g^{Bk} \bmod n$
GET : $A_v = g^{Ak} \bmod n$
② SEND : $B_v = A_v^{Bk} \bmod n$
GET : $A_v = D_v^{Ak} \bmod n$ ($D_v = g^{Dk} \bmod n$)
③ SEND : $B_v = A_v^{Ak} \bmod n$
GET : $A_v = D_v^{Ak} \bmod n$
($D_v = C_v^{Dk} \bmod n$, $C_v = g^{ck} \bmod n$)

B의 키값 계산 :
$k = Av^{Bk} \bmod n$
$= g^{CkDkAkBk} \bmod n$

위의 ①②③의 과정은 멤버 A와 B가 수행하는 과정이며 멤버 C, D도 이와 비슷한 과정을 수행하여 $k = g^{AkBkCkDk} \bmod n$ 이라는 동일한 연산으로 같은 키값을 갖는다. 위와 같은 방식에서 N명의 키교환을 위해서는 각각의 멤버들이 N-1번의 통신이 필요하므로 많은 시간이 소요된다. 이와 같은 비효율적인 면을 개선하기 위하여 3장의 Hughes방식을 이용하였으며 Hughes방식을 이용한 그룹 키교환 방식에 대해 4장에서 설명한다.

3. Hughes의 키교환 방식

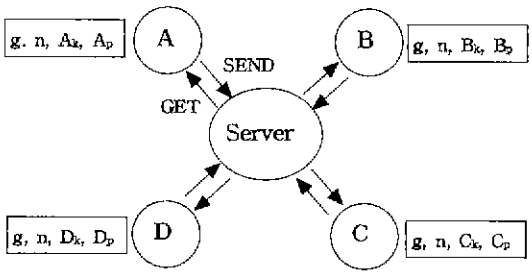
Hughes의 키교환 방식[1][2]은 한쪽이 먼저 키를 소유한다는 것이 Diffie-Hellman 방식과 다르다. A는 먼저 키값을 계산한 후 B가 전송한 Bv에 Ak로 연산하여 다시 B에게 전송하면 B는 식(5)와 같은 연산으로 키값을 계산한다.



A : $k = g^{Ak} \bmod n$
① $B_v = g^{Bk} \bmod n$
② $A_v = B_v^{Ak} \bmod n$
B : $k' = A_v^{-Bk} \bmod n$ --(5)

4. 그룹 키교환 방식

그룹의 멤버가 A, B, C, D 4명으로 구성되어져 있을 때 각각의 멤버는 Diffie-Hellman 방식과 같이 g, n 값을 공유하고 개인값을 가진다. 그러나 차이점은 2개의 개인값 Ak, Ap을 가지며 하나의 값 Ap는 Hughes의 방식을 이용하기 위하여 운반자(Carrier)로 쓰인다. 이를 이용하여 키를 교환하는 과정은 다음과 같고 Diffie-Hellman Algorithm을 이용한 그룹 키교환 방식에서와 같이 중앙에 서버를 두어 정보를 교환한다.



A : ① SEND : $A_v = g^{Ap} \bmod n$
GET : $B_v = g^{Bp} \bmod n$
$C_v = g^{Cp} \bmod n$
$D_v = g^{Dp} \bmod n$
② SEND : $AB_v = B_v^{Ak} \bmod n$
$AC_v = C_v^{Ak} \bmod n$
$AD_v = D_v^{Ak} \bmod n$
GET : $BA_v = A_v^{Bk} \bmod n$ ($A_v = g^{Ap} \bmod n$)
$CA_v = A_v^{Ck} \bmod n$ ($A_v = g^{Ap} \bmod n$)
$DA_v = A_v^{Dk} \bmod n$ ($A_v = g^{Ap} \bmod n$)

B : ① SEND : $B_v = g^{Bp} \bmod n$
GET : $A_v = g^{Ap} \bmod n$
$C_v = g^{Cp} \bmod n$
$D_v = g^{Dp} \bmod n$
② SEND : $BA_v = A_v^{Bk} \bmod n$
$BC_v = C_v^{Bk} \bmod n$
$BD_v = D_v^{Bk} \bmod n$
GET : $AB_v = B_v^{Ak} \bmod n \ (B_v = g^{Bp} \bmod n)$
$CB_v = B_v^{Ck} \bmod n \ (B_v = g^{Bp} \bmod n)$
$DB_v = B_v^{Dk} \bmod n \ (B_v = g^{Bp} \bmod n)$

위의 과정은 멤버 A와 B가 수행하는 과정이고 C와 D도 마찬가지로 비슷한 과정을 수행한다. 각각의 멤버는 아래의 식(6),(7),(8),(9)과 같은 연산을 수행하여 키값을 계산한다.

$$A : k = g^{Ak} \bmod n * BA_v^{-Ap} \bmod n * CA_v^{-Ap} \bmod n * DA_v^{-Ap} \bmod n \quad \text{--- (6)}$$

$$B : k' = g^{Bk} \bmod n * AB_v^{-Bp} \bmod n * CB_v^{-Bp} \bmod n * DB_v^{-Bp} \bmod n \quad \text{--- (7)}$$

$$C : k'' = g^{Ck} \bmod n * AC_v^{-Cp} \bmod n * BC_v^{-Cp} \bmod n * DC_v^{-Cp} \bmod n \quad \text{--- (8)}$$

$$D : k''' = g^{Dk} \bmod n * AD_v^{-Dp} \bmod n * BD_v^{-Dp} \bmod n * CD_v^{-Dp} \bmod n \quad \text{--- (9)}$$

k, k', k'', k'''는 식(10)와 같이 계산되며 서로 같은 값을 가지게된다.

$$\text{키값} = g^{Ak} * g^{Bk} * g^{Ck} * g^{Dk} \bmod n = g^{Ak+Bk+Ck+Dk} \bmod n \quad \text{--- (10)}$$

위의 키교환 방식에서 중앙 서버가 키값을 계산할 수 없도록 하기 위하여 Hughes의 방식을 이용하여 운반자로 A_p, B_p, C_p, D_p 값을 사용함으로써 그룹의 각 멤버에게 자신의 개인값을 전달한다. 본 논문의 그룹 키교환 방식이 가지는 장점은 Diffie-Hellman 방식을 이용할 경우 멤버가 N명일때 각각의 멤버가 N-1번의 통신을 해야만 키를 계산할 수 있는 반면에 멤버의 수가 얼마이든지 서버와 2번의 통신으로 키값을 계산할 수 있기 때문에 통신의 비용을 줄일 수 있다. 멤버가 N명 일 때 두 방식의 키 값 계산과 통신 빈도수의 비교는 <표 1>에 정리하였다.

(멤버가 N명일 경우)

Diffie-Hellman 그룹 키교환 방식	
Key 계산	$k = g^{A1kA2k \ldots ANk} \bmod n$
통신 빈도수	N*(N-1)
본 논문의 그룹 키교환 방식	
Key 계산	$k = g^{A1k + A2k + \cdots + ANk} \bmod n$
통신 빈도수	N*2

<표 1>

5. 결론

본 논문의 그룹 키교환 방식은 그룹의 멤버가 많을 경우에 효율적으로 키를 교환할 수 있는 방법으로 2장에서의 Diffie-Hellman 알고리즘을 이용한 키교환 방식과 같이 많은 통신 과정을 거치지 않아도 되는 장점을 가지고 있다. 또한 중앙 서버가 키를 알지 못하므로 그룹 통신의 비밀성이 보장되며 Man-in-the-middle Attack[4]에도 안전하다. 앞으로 그룹의 어느 한 멤버가 키를 생성시킨 후 공개키 암호화 방식을 이용하여 키를 교환하는 방법과 본 논문에서 제안한 방법간의 효율성 및 안전성을 비교 평가할 예정이다.

6. 참고 문헌

[1] Bruce Schneier, "Applied Cryptography", John Wiley & Sons, Inc. 1996, pp. 513-515.

[2] E. Hughes, "An Encrypted Key Transmission Protocol," CRYPTO '94, Aug 1994.

[3] W. Diffie and M.E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, Nov 1976, pp. 644-654.

[4] Bruce Schneier, "Applied Cryptography", John Wiley & Sons, Inc. 1996, pp. 48-49.

[5] I. Ingemarsson, D.T. Tang, and C.K. Wong, " A Conference Key Distribution System," IEEE Transactions on Information Theory, Sep 1982, pp. 714-720.