트리 기반 그룹키 인증 및 합의 프로토콜

이상원, 김진, 김광조

국제정보보호기술연구소

한국정보통신대학원대학교 공학부

Tree-based Authenticated Group Key Agreement Protocol

Sang-won Lee, Zeen Kim, and Kwangjo Kim

International Research center for Information Security (IRIS)

School of Engineering, Information and Communication Univ. (ICU)

요 약

안전하고 안정적인 그룹통신은 최근 그룹 및 그룹 구성원간의 협조가 필요한 응용 분야가 발전하면서 점차 그 필요성이 대두되고 있다. 이 중 가장 중요한 문제는 그룹내의 키 관리 문제이다. 중앙에 의존하는 키 관리 방식의 경우 대용량의 멀티캐스트 그룹에 어울리는 반면 중앙 센터 없이 구성원간의 협조에 의하여 이루어지는 그룹의 경우 분산 키 관리 방법, 즉 그룹키 합의 방법이 필요하다. 기존의 그룹 키 합의 방법들은 계산량의 효율성에 치중한 연구만을 해왔다. 한 가지 예외로 STR 프로토콜[4]은 디피-헬만 프로토콜을 키 트리에 응용하고 키 트리가 한쪽으로 치우친 구조를 가지고 있어 통신량을 최적화하고 있다. 하지만 계산량에 있어서 그룹 멤버의 변경 시 현재 그룹 구성원의 수에 비례한 계산량이 필요하다. 본 논문에서는 pairing을 응용하여 STR 키 합의 방식에 계산량에 있어서 효율성을 제공하고 통신 효율성을 유지하며 그룹키를 인증할 수 있는 방식을 제시한다.

I. 서론

최근 컴퓨팅 환경에서는 급속한 통신기술의 발달과 함께 신뢰할 수 있는 통신이 중요시되고 있다. 또한 전자우편과 파일공유와 같은 중앙 집중적인 서비스는 근래에는 다중의 서버와 네트워크를 통해 제공되는 분산시스템으로 바뀌고 있다. 이러한 분산, 협동적인 새로운 환경에서는 신뢰할 수 있는 안전한 통신을 필요로 한다.

구성원의 멤버쉽이 자주 바뀌고, 모든 구성원이 동일한 권리와 의무를 가지며 또한 작은 규모이며 어떠한 구성원이든 수신자나 송신자가 될 수 있는 그룹을 DPG(Dynamic Peer Group)라 부른다. 우리는 이러한 DPG 환경에서의 보안 요구사항 중에서 그룹키 관리에 주목한다.

지금까지 DPG상의 그룹키 관리 방법은 대부분 계산 오버헤드를 줄이는 방향으로 연구되어 왔으며, 하드웨어의 발달과 함께 많은 발전이 있었다. 하지만 이에 비해 통신 지연에 대한 연구는 그에 따라갈 만큼 발전하지 못했다. 네트워크 기기나 설비는 상당히 빨라지고 싸졌으며 결국 어디서나 통신에 접속할 수 있는 환경이 되었다. 이는 결국 엄청난 네트워크 대역폭을 요구하게 되었다. 특히 최근 P2P 응용 서비스의 발전은 네트워크 대역폭의 50% 이상을 차지하고 있다. 이로 인해 네트워크 정체는 크게 줄지 않는 상황이다. 이러한 점에서 통신 오버헤드를 줄이는 것이 오히려 암호 프로토콜을 좀더 효율적으로 만들 것이다.

본 논문은 통신량 면에서 최적화 되어 있는 Y. Kim 등에 의해 제안된 STR 프로토콜[4]을 바탕으로 pairing을 사용하여 트리 기반의 인증이 가능한 그룹키 합의 프로토콜을 제안한다. 본 논문의 구성은 다음과 같다. 본문의 1장에서는 관련연구

를 살펴보고, 2장에서는 그룹키 합의 방법을 설명한다. 3장에서는 성능에 대한 분석을 하고 마지막으로 결론을 맺는다.

II. 본문

1. 관련연구

1) 그룹키 관리

DPG 환경에서 그룹키를 관리하는 방법은 크게 그룹키 분배와 그룹키 합의로 나눌 수 있다. 중앙 그룹키 분배(Centralized Group Key Distribution) 방식은 하나의 서버가 각 구성원의 키를 생성하여 이를 배포하는 방식이다. 키 서버는 실제 키 분배를 위한 안전한 양자간 통신을 위해 각 그룹 구성원과 장기 공유키(long-term shared key)를 유지한다. 이러한 방법의 한 가지 유형은 고정된 TTP(Trusted Third Party)를 키 서버로서 사용하는 것이다. 이러한 접근은 항상 서버가 가용해야 한다는 것과 네트워크 분할 이벤트에서 계속적인 운용을 지원하기 위해 모든 가능한 그룹의 부분집합에 TTP가 존재해야 한다는 두 가지 문제점을 갖는다.

이러한 중앙 그룹키 분배 방식의 변형으로, 키를 생성하고 분배하는 키 서버를 동적으로 선택할 수 있는 방식인 비 중앙 그룹키 분배(Decentralized Group Key Distribution)방식이 있다. 이 방식은 키 서버를 선택함으로써 어떠한 그룹분할 상황에서도 계속적으로 운영될 수 있기 때문에 좀더 견고하여, 결국 다대다 그룹에 더 적용가능하다. 하지만 중앙 그룹키 분배방식과 마찬가지로, 키 서버는 반드시 그룹키 배분을 위한 모든 그룹 구성원과의 장기간의 안전한 쌍방향 채널을 가져야 한다는 단점을 가진다. 결국 매번 새로운 키 서버가 선택되어 그룹키 분배 역할을 하게 되어 쌍방향의 안전한 채널을 만드는데 심각한 비용이 초래된다. 또 다른 단점으로는 안전한(예로, 암호학적으로 강한) 키를 생성하는 한 개체에 대한 의존성이다.

위의 접근에 대조적으로, 그룹키 합의(Group Key Agreement)방식은 각 그룹 구성원이 공통의 그룹키(이는 모든 구성원들의 기여의 함수로서 계산된다.)에 동등한 몫을 기여하는 것을 요구한다. 따라서 신용의 집중과 한 지점에서의 오류의 문제를 피할 수 있다. 또한 몇몇의 기여 그룹키 관리 방법은 그룹 구성원들 간의 안전한 쌍방향 채널의 수립을 요구하지 않는다[4,5].

2) 타원곡선을 이용한 키 합의

먼저 겹선형성에 대하여 설명하고, 키 합의에 필요한 두 가지 문제에 대하여 알아본다.

소수 위수 q의 덧셈 군을 G_1이라 하고 동일한 위수 q의 곱셈 군을 G_2라 한다. $G_1 \times G_1$에서 G_2로의 효과적으로 계산 가능한 겹선형의 사상 $\hat{e}$의 확장을 가정한다. 전형적으로 G_1는 유한체 상의 타원곡선 위의 점 군의 부분 군이 될 것이다. G_2는 관련된 유한체의 곱셈 군의 부분 군이 될 것이고 $\hat{e}$는 타원곡선 상의 Weil이나 Tate pairing으로부터 파생될 것이다. 우리는 또한 G_1에 속하며 $\hat{e}(P,P) \neq 1_{G_2}$를 만족하는 P가 알려져 있다고 가정한다. 겹선형인 $\hat{e}$에 의해, 모든 $Q, P \in G_1$과 $a, b \in Z_q^*$에 대해

$$\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$$

이다.

겹선형 디피-헬만 (BDH: Bilinear Diffie-Hellman Problem) 문제는 P, aP, bP, cP이 주어졌을 때 $\hat{e}(P,P)^{abc}$을 계산하는 문제이다. 삼자간의 키합의는 이러한 겹선형 디피-헬만 문제의 어려움에 기반한다 [2].

타원곡선 상에서 양자간 키 교환은 타원곡선 위의 디피헬만(ECDH: Elliptic Curve Diffie-Hellman) 문제에 기반하는데, 미 문제는 P, aP, bP가 주어졌을 때 abP를 계산하는 문제이다.

2. 제안 프로토콜

1) 표기

우리는 그룹키를 합의하고자 하는 n명의 구성원 $\{1,2,\ldots,n\}$을 고려한다. 이들 구성원은 각자의 비밀키 $r_1, r_2, \ldots, r_n \in Z_q^*$을 가진다. 구성원 $\{1,2,\ldots,n\}$에 대한 부분집합으로 C를 정의하며, 프로토콜에서 중요한 역할을 하게 되는 스폰서를 $S(C)$로 정의한다. $S(C)$는 일반적으로 가장 최근에 가입된 구성원이다.

타원곡선 상의 임의의 점 P를 정의하고 $H: G_2 \rightarrow Z_q^*$를 선택한다. ID 기반의 인증을 사용하기 위해 키 생성 센터(KGC)는 임의의 비밀키 $s \in Z_q^*$를 선택하고 $P_{pub} = [s]P$를 계산한다. KGC는 $\{P, P_{pub}\}$를 배포하고 s는 메인키로서 안전하게 보관한다. ID_i를 가진 사용자는 장기 공개키

$Q_i = H(ID_i)$를 가지며 이를 KGC에 보냄으로써 장기 비밀키 $S_i = sQ_i$를 얻게된다.

2) 기본개념

그림 1은 제안하는 그룹키 방법의 트리 구조이다. 삼자간의 키 합의 결과를 다음 라운드에서 다시 삼자간의 키 합의 요소로 사용하는 방식으로 최종적으로 그룹키가 생성된다. 결국 트리의 높이만큼의 라운드를 요구한다.

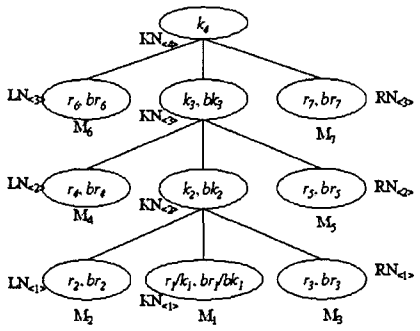

그림 1 : 트리구조

그림 1에서 루트노드에 연결되는 그룹키는 다음과 같은 형태를 갖는다:

$$k_4 = H_1(\hat{e}(P,P)^{r_6 r_7 H_1(\hat{e}(P,P)^{r_4 r_5 H_1(\hat{e}(P,P)^{r_1 r_2 r_3})})})$$

기본적인 그룹키 프로토콜은 다음과 같다. 우리는 모든 구성원들이 키-트리의 구조와 트리 내에서의 위치를 알고 있다고 가정한다. 각각의 구성원은 자신의 비밀키를 알고 또한 모든 다른 구성원의 은닉키와 은닉 비밀키도 안다. 최초 구성원 M_1, M_2 그리고 M_3는 $KN_{<2>}$에 연결된 그룹키를 계산할 수 있다. M_1은 다음을 계산할 수 있다:

$k_1 = r_1$

$k_2 = H_1(e(r_2P, r_3P)^{r_1}) = H_1(e(P,P)^{r_1 r_2 r_3})$

$k_3 = H_1(e(br_4, br_5)^{k_2}) = H_1(e(P,P)^{k_2 r_4 r_5})$

...

$k_l = H_1(e(P,P)^{k_{n-1} r_{2n-2} r_{2n-1}})$

다음으로 M_1은 모든 은닉키를 담은 트리를 브로드캐스트한다. 이 메시지를 받음으로써 모든 구성원들은 그룹키 k_l을 계산할 수 있다.

3) 그룹키 인증 및 합의 프로토콜

우리는 양자간의 키합의 함수인 TwoParty와 삼자간의 키합의 함수인 ThreeParty, 그리고 이와 함께 이 두 함수를 사용하여 만들어지는 그룹키 합의 함수인 KeyAgreement를 소개한다.

양자간의 키합의 함수는 N.P. SMART에 의해 제안되어진 방법을 사용한다[6].

Procedure TwoParty(C_1, C_2, r_1, r_2)
  $S(C_1)$ sends
    $T_{S(C_1)}(=[r_1]P)$ to all members of C_2;
  $S(C_2)$ sends
    $T_{S(C_2)}(=[r_2]P)$ to all members of C_1;
  each member of C_1 computes
    $k_{C_1} = \hat{e}([r_1]W_{C_2}, P_{KGC}) \cdot \hat{e}(w_{C_1}, T_{S(C_2)})$;
  each member of C_2 computes
    $k_{C_2} = \hat{e}([r_2]W_{C_1}, P_{KGC}) \cdot \hat{e}(w_{C_2}, T_{S(C_1)})$;
  return $H(k_{C_i})$;
end TwoParty

삼자간의 키합의 함수는 Nalla와 Reddy에 의해 제안되어진 프로토콜 ID-AK-3를 사용한다[7].

Procedure ThreeParty(C_1, C_2, C_3, r_1, r_2, r_3)
  for i = 1 to 3 **do**
    Let j, k = {1,2,3}\{i};
    $S(C_i)$ sends
      ($[r_i]P$, $[r_i]W_j$) to all members of C_k;
      ($[r_i]P$, $[r_i]W_k$) to all members of C_j;
  end do
  for i = 1 to 3 **do**
    Let j, k = {1,2,3}\{i};
    each member of C_i computes
    $k_{C_i} = \hat{e}([r_i](W_{C_i} + W_{C_k}, P_{KGC})$
      $\cdot \hat{e}(w_{C_i}, ([r_j]P + [r_k]P))$
      $\cdot \hat{e}([r_j]W_{C_i}, P_{KGC}) \cdot \hat{e}([r_k]W_{C_i}, P_{KGC})$
  end do
  return $H(k_{C_i})$;
end ThreeParty

그룹키 합의 함수는 상위의 두 함수를 사용하여 구성된다. 트리 높이만큼의 라운드를 요구한다.

```
Procedure KeyAgreement(n, C, X)
    h= ⌊ n/2 ⌋ ;
    m=n-2h;
    K={M1};
    k1=r1;
    for i= 1 to h+1 do
       KEY = call ThreeParty( M2i, K, M2i+1, r2i,
k_i, r2i+1);
       K=K∪M2i∪M2i+1;
       ki=KEY;
       if (i == h+1) & (m == 1) then
          KEY = call TwoParty( M2i, K, r2i, ki);
       end if
    end do
end KeyAgreement
```

4) 분석

인증을 사용하지 않는 경우는 라운드 수만큼의 pairing 연산과 포인트 곱셈을 요구한다. 즉 트리의 높이만큼의 연산이 이루어진다. 인증이 이루어지는 경우는 ID-AK-3가 4번의 pairing 연산과 포인트 곱셈을 요구하기 때문에 $4h$ 만큼의 pairing 연산과 포인트 곱셈을 요구한다.

제안된 프로토콜의 안전성을 기반하는 양자간 그리고 삼자간의 키합의 방법의 안전성에 의존한다. 따라서 제안된 프로토콜 또한 사용자 ID 기반의 인증을 제공하며 전방위 안전성과 후방위 안전성 및 키 독립성을 제공한다.

III. 결론

제안된 프로토콜은 pairing을 사용한 삼자간의 키 합의 프로토콜을 트리 구조를 이용하여 그룹으로 확장시킨 것이다. 또한 그룹키 인증이 가능하며 STR과 마찬가지로 통신에 대하여 최적화 되어 있는 프로토콜이다. 제안된 프로토콜은 최적화되어 있는 통신량과 적은 계산량을 보인다. 하지만 pairing 기반의 암호시스템의 성능은 pairing 연산에 의존적이므로 pairing의 고속화가 중요하다. 아직까지 pairing 연산의 부담이 있으나 하드웨어의 발달과 함께 이러한 연산에 대한 고속화 연구가 계속되고 있으며 지수 연산과 같은 계산에 대한 비용 또한 싸지고 있다. 결국 통신 지연에 대한 비용이 프로토콜에 대한 실행시간을 결정하는데 있어 계산비용을 지배하게 된다. 따라서 제안프로토콜은 고-지연 광대역 네트워크에서 효율적인 그룹키 인증 및 합의 프로토콜이다.

참고문헌

[1] S. Al-Riyami and K. Paterson, "Authenticated three party key agreement protocols from pairings," Cryptology ePrint Archive, Report 2002/035, available at http://eprint.iacr.org/2002/035/.

[2] D. Boneh and M. Franklin. "Identity-based encryption from the Weil pairing," Advances in Cryptology-Crypto 2001, LNCS 2139, pp.213-229, Springer-Verlag, 2001. http://www.crypto.stanford.edu/~dabo/abstracts/ibe.html

[3] A. Joux, "A one round protocol for tripartite Diffie-Hellman," In W. Bosma, editor, Proceedings of Algorothmic Number Theory Symposium - ANTS IV, volume 1838 of LNCS, pages 385-394. Springer-verlag, 2000

[4] Y. Kim, A. Perrig and G. Tsudik, "Communication-Efficient Group Key Agreement," IFIP SEC 2001, Jun. 2001.

[5] Y. Kim, A. Perrig, G. Tsudik, "Tree-based Group Diffie-Hellman Protocol," In Submission.

[6] N.P. Smart, "An identity based authenticated key agreement protocol based on the weil pairing," Election. Lett., Vol.38, No.13, pp.630-632, 2002

[7] Divya Nalla, K.C.Reddy, "ID-based tripartite Authenticated Key Agreement Protocols from pairings," Cryptology ePrint Archive, Report 2003/004, available at http://eprint.iacr.org/2003/004/.