情 報 保 護 學 會 誌
第19卷 第4號, 2009. 8

ID 기반의 그룹 키 교환 기법에 대한 연구 동향

최 재 탁*, 이 동 훈*

요 약

키 교환은 이미 오래전부터 많이 연구되어 왔으나 끊임없이 변화하는 통신 환경과 그에 따른 새로운 위협에 따라 안전한 키 교환이 필요하게 되었다. 특히 키 교환을 수행하는 개체의 비밀키가 노출되었을 경우에도 안전한 키 교환 프로토콜이 필요하다. 또한 그룹의 구성원 수와 상관없이 상수 라운드를 가지는 효율적인 키 교환 프로토콜이 요구된다.

본 논문에서는 ID 기반의 그룹 키 교환 기법에 대한 기술 동향을 분석하고, 그룹 키 교환 환경에서의 다양한 공격 환경 및 공격자에 따른 안전성 모델에 대한 안전성을 분석한다.

I. 서 론

그룹 키 교환(group key agreement) 프로토콜은 두 명 혹은 그 이상의 사용자들이 안전하지 않는 공개된 통신망에서 안전하게 공통된 키를 생성하는 기법이다. 키 교환 프로토콜을 수행하여 생성된 공통된 키를 세션 키라고 한다. 이렇게 생성된 세션키는 암호화/복호화, MAC, 인증과 디지털 서명 등의 키로 사용되어 진다.

인증 그룹 키 교환(authenticated group key agreement) 프로토콜이란 인증 기법이 포함되어 있는 그룹 키 교환 프로토콜을 말한다. 인증이란 키 교환에 참여하는 사용자가 의도한 사용자인지에 대한 신원을 확인하는 것을 말한다.

ID 기반의 공개키 방식의 기본 개념은 Adi Shamir[1]가 처음으로 제안하였다. ID 기반의 암호 시스템에서 공개키로 비밀 키를 가지고 있는 사용자를 쉽게 확인할 수 있는 이메일 주소와 같은 공개 정보로 구성된 공개 확인자(identity, ID)를 사용하고 있다. Bilinear pairing을 이용한 Boneh 와 Franklin[2]의 ID 기반의 암호 시스템에 관한 연구 이래로 ID 기반의 그룹 키 교환에 관한 연구가 진행되어 왔다. ID 기반의 인증된 그룹 키 교환 프로토콜은 K. C. Reddy[3]가 N. P. Smart[4]의 양자간 키 교환 방식을 이용하여 트리 기반의 그룹 키 교환 기법을 제안하였다. 그 후, R. Barua[5]는 A. Joax[6]의 3자간 키 교환 방식을 이용한 트리 기반의 그룹 키 교환 기법을 제안하였다. 하지만 위의 두 가지 방식은 트리의 깊이의 수만큼 키 교환이 필요하다는 단점이 있다. 그래서 Reddy와 Barua의 기법은 많은 사용자를 가지는 그룹 키 교환 방식에는 적용하기에는 비효율적이다. 고정된 라운드를 가지는 ID 기반의 인증된 그룹 키 교환 프로토콜이 Choi et al.[7]와 Du et al.[8]에 의해서 제안 되었다. Choi와 Du의 기법은 2 라운드를 필요로 한다. 그러나 Zhang과 Chen[9]은 이 두 기법에 대한 위장공격의 취약점을 발견하였다. 그 후, Du et al.[10]는 자신의 기법에 대한 위장공격에 안전하게 보완한 그룹 키 교환 기법을 제안하였다.

[11]에서 Y. Shi는 1라운드를 가지는 ID 기반의 인증된 그룹 키 교환 기법을 제안하였다. 최근에 L. Zhou[12]는 두 개의 그룹 키 교환 기법을 제시하였는데, 하나는 1라운드가 필요하고 다른 하나는 2라운드가 필요하다. 2라운드가 필요한 기법은 1라운드가 필요한 기법보다는 전송 효율이 더 좋은 기법이다.

본 논문에서는 ID 기반의 그룹 키 교환 기법에 대한 기술 동향을 분석하고, 그룹 키 교환 환경에서의 다양한 공격 환경 및 공격자에 따른 안전성 모델에 대한 안전성을 분석한다.

* 고려대학교 정보경영공학전문대학원 (algedi@cist.korea.ac.kr, donghlee@korea.ac.kr)

Ⅱ. 정 의

2.1 The Bilinear Map

G_1을 위수가 q인 덧셈 연산군 이라 하고, G_2를 위수 q를 갖는 곱셈 연산군 이라고 하자. 이때 G_1, G_2에서의 이산 대수 문제(discrete logarithm problem)는 어렵다고 가정한다.

Bliniear Diffie-Hellman(BDH) 파라미터 생성자를 확률적 다항식 시간 연산 알고리즘 이라고 하자. 다항식 시간 안에 BDH 파라미터 생성자는 위수가 소수 q를 가지는 두 그룹 G_1, G_2와 다음 성질을 만족하는 bilinear map $e: G_1 \times G_1 \rightarrow G_2$를 생성한다.

- Blinear: 모든 $P, Q \in G_1$와 $a, b \in Z_q^*$에 대해서 $e(aP, bQ) = e(P, Q)^{ab}$
- Non-degenerate: 모든 $Q \in G_1$에 대해서 $e(P, Q) = 1$를 만족하면, $P = 0$이다.
- Computable: 모든 $P, Q \in G_1$에 대하여 다항식 시간 안에 $e(P, Q)$를 계산할 수 있는 효율적인 알고리즘이 존재한다.

2.2 안전성 성질

본 절에서는 그룹 키 교환에 관한 기본적인 안전성 성질들에 관하여 알아본다.

키 교환 프로토콜의 가장 기본적인 안전성 요구사항은 키 기밀성 (key secrecy)이다. 유한한 계산 능력을 지닌 공격자는 정직한 사용자 간의 통신을 도청하거나, 공격자가 프로토콜에 참여함으로써 정직한 사용자에게 메시지를 전송 할 수 있다. 키 기밀성은 이러한 공격자가 세션 키에 대한 어떠한 정보도 얻을 수 없어야 한다는 것이다. 키 교환 프로토콜에 요구되는 다른 안전성은 전방향 안전성 (forward secrecy)과 기지 키 공격에 대한 안정성 (known-key secrecy)이다.

전방향 안전성은 사용자의 롱텀 비밀 키(long-term key)를 알고 있는 어떠한 공격자라도 정직한 구성원 간에 성공적으로 확립된 이전의 세션 키에 대한 어떠한 정보도 없을 수 없어야 함을 의미한다.

기지키에 대한 안전성은 여러 세션에서 얻은 세션 키들을 이용해도 노출되지 않은 세션들의 세션 키들의 키 기밀성에는 영향을 주지 않아야 함을 의미한다.

이 밖에 비밀키 사용 위장에 대한 안전성(security against key compromise impersonation)과 파트너 혼돈 공격에 대한 안전성(security against unkown impersonation)이 존재한다. A의 비밀키가 노출된다면, 공격자는 A의 개인키를 가지고 B에게 A인척 위장할 수 있다. 비밀키 사용 위장에 대한 안전성이란 A의 롱텀키인 비밀키가 노출되더라도 A의 비밀키를 가진 공격자가 A에게 B인척 위장할 수 없어야 한다는 것을 말한다. 파트너 혼돈 공격에 대한 안전성이란 A와 B가 똑같은 세션키를 계산했으면 A는 현재 B와 키를 교환하고 있다고 인식해야 하며, B또한 A와 키 교환하고 있다고 인식해야 한다는 것을 말한다.

Ⅲ. ID 기반의 그룹 키 교환 프로토콜

본 절에서는 ID 기반의 그룹 키 교환 기법의 연구동향을 살펴본다.

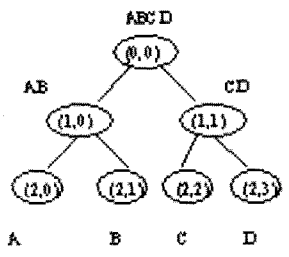
3.1 Reddy와 Nalla의 기법

이 기법은 Smart[4]의 ID 기반의 양자간 키 교환 기법을 일방향 함수 트리를 이용하여 처음으로 ID 기반의 그룹 키 교환 기법을 설계하였다.

Setup. KGC가 임의의 상수 s와 $P \in G$를 선택하고, $P_{KGC} = sP$를 계산한다. 그리고 (P, P_{KGC})를 공개값으로 하여 공개한다.

KeyGne. 사용자의 ID에 대하여, 공개키 $Q_{ID} = H(ID)$와 개인키 $S_{ID} = sQ_{ID}$를 계산하여 사용자에게 전송한다.

Protocol. [그림 1]은 키 트리(key tree)의 한 예이다.



〔그림 1〕 키 트리의 예

이 트리에 대한 그룹 키의 계산은 다음과 같다.

먼저, A와 B는 Smart의 양자간 ID 기반의 인증된 키 교환 기법을 이용하여 노드(1, 0)에 해당하는 공통된 키를 계산한다. C와 D도 같은 방법으로 노드(1, 1)에 해당하는 공통된 키를 계산한다. [그림 2]는 키 교환 과정을 나타낸 것이다.

A	B	C	D
The public key Q_A is sent to the KGC to get the private key $S_A=[s]Q_A$	The public key Q_B is sent to the KGC to get the private key $S_B=[s]Q_B$	The public key Q_C is sent to the KGC to get the private key $S_C=[s]Q_C$	The public key Q_A is sent to the KGC to get the private key $S_D=[s]Q_D$
selects a random value $a\in Z_q^*$	selects a random value $b\in Z_q^*$	selects a random value $c\in Z_q^*$	selects a random value $d\in Z_q^*$
Compute $T_A=[a]P$	compute $T_B=[b]P$	compute $T_C=[c]P$	compute $T_D=[d]P$
Exchange T_A and T_B $T_A=[a]P$ ⟶ ⟵ $T_B=[b]P$		$T_C=[c]P$ ⟶ ⟵ $T_D=[d]P$	
$\hat{e}([a]Q_B,P_{KGC})\cdot\hat{e}(S_A,T_B)$ $=k_{AB}$	$\hat{e}([b]Q_A,P_{KGC})\cdot\hat{e}(S_B,T_A)$ $=k_{BA}$	$\hat{e}([c]Q_D,P_{KGC})\cdot\hat{e}(S_C,T_D)$ $=k_{CD}$	$\hat{e}([d]Q_C,P_{KGC})\cdot\hat{e}(S_D,T_C)$ $=k_{DC}$

〔그림 2〕 중간 노드의 키

k_{AB}와 k_{CD}는 암호화 키로 직접 사용할 수 없다. 대신에 그룹 비밀 값으로부터 생성된 특별한 목적을 가지는 하위키(sub-key)를 생성하여 사용한다. 사용자 A와 B는 노드(1, 0)에 해당하는 하위키는 k_{AB}를 일방향 함수 $f_t: F_{q^k}^* \rightarrow Z_q^*$에 적용하여 k_{AB}'로 한다. 같은 방법으로 C와 D는 노드(2,0)에 해당하는 하위키 k_{CD}'를 생성한다. 이 두 값을 다음 상위 레벨의 키를 계한하는데 사용한다. 중간 노드들은 바로 상위 레벨의 키를 계산하기 위해서 자신의 공개키와 개인키가 필요하다. 하지만 중간 노드들은 ID가 존재하지 않기 때문에 하위 레벨의 두 노드의 공개키를 더해서 자신의 공개키로 생성하고, KGC로부터 개인키를 얻는다. 즉, 노드(1, 0)은 자신의 공개키로 $Q_{AB}=Q_{\langle 1,0\rangle}=H'(Q_{\langle 2,0\rangle}+Q_{\langle 2,1\rangle})$로 한다. KGC는 $Q_{\langle 1,0\rangle}$의 개인키 $S_{AB}=S_{\langle 1,0\rangle}=sQ_{\langle 1,0\rangle}$를 계산한다.

사용자 A와 B는 [그림 3]과 같이 노드(1, 0)의 ID와

AB	CD
The public key $Q_{AB}=Q_{BA}=H'(Q_A+Q_B)$ is sent to the KGC to get the private key $S_{AB}=[s]Q_{AB}$	The public key $Q_{CD}=Q_{DC}=H'(Q_C+Q_D)$ is sent to the KGC to get the private key $S_{CD}=[s]Q_{CD}$
Compute $k_{AB}'=f_t(k_{AB})$	Compute $k_{CD}'=f_t(k_{CD})$
Compute $T_{AB}=[k_{AB}']P$	Compute $T_{CD}=[k_{CD}']P$
Exchange T_{AB} and T_{CD} $T_{AB}=[k_{AB}']P$ ⟶ ⟵ $T_{CD}=[k_{CD}']P$	
$\hat{e}([k_{AB}']Q_{CD},P_{KGC})\cdot\hat{e}(S_{AB},T_{CD})$	$\hat{e}([k_{CD}']Q_{AB},P_{KGC})\cdot\hat{e}(S_{CD},T_{AB})$

〔그림 3〕 상위 노드의 키

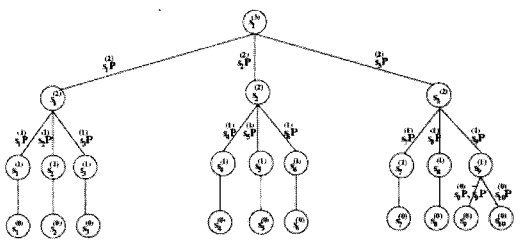
노드(1, 1)의 공개키 Q_{CD}, 그리고 T_{AB}를 이용하여 노드 (0, 0)의 키를 계산할 수 있다. 사용자 C와 D도 같은 방법으로 노드(0, 0)의 키를 계산한다.

위와 같은 방법으로 n명의 사용자에 대한 그룹 키를 계산할 수 있다. 트리의 하위 노드에 위치하는 그룹 사용자들은 상위 노드까지에 이르는 노드들의 비밀키값과 형제노드들의 공개키를 알고 있으면 공통된 그룹 키를 생성할 수 있다.

3.2 Barua, Dutta와, Sarkar의 기법

Barua등은 Joux[6]의 3자간 1라운드 키 교환 프로토콜을 이용하여 Reddy등의 기법과 같은 트리 구조를 이용하여 그룹 키 교환 기법을 설계하였다. 설계 방법은 Diffie-Hellman 프로토콜과 셋으로 이루어지는 트리구조로 간단하지만, 이에 대한 안전성을 증명하는 것은 간단하지 않다. 이 논문에서는 수동적 공격자를 전제로 하여 안전성에 대한 증명을 제시하였다는 것이 중요한 결과이다.

3자간 키 교환 프로토콜을 CombineThree라 하고, 2자간 키 교환 프로토콜을 CombineTwo라고 할 때 CombineThree와 CombineTwo로 구성되는 n명의 그룹 키 교환 프로토콜을 KeyAgreement라고 한다. 다음 [그림 4]는 10명의 사용자에 대한 KeyAgreement의 실행 예이다.



〔그림 4〕 사용자의 수가 10명일 때 키 교환의 실행 예제

위의 그림에서 세 명의 사용자가 하위 그룹을 이룰 때에는 CombineThree 알고리즘을 사용하여 공통된 비밀 값을 생성하고, 두 명의 사용자가 하위 그룹을 이룰 때에는 CombineTwo 알고리즘을 사용하여 공통 비밀 값을 생성한다.

다음 그림들은 CombineThree와 CombineTwo 프로토콜을 설명한 그림이다. 인증되지 않은 3자간 키 교환 프로토콜은 Joux의 기법이며, 인증된 3자간 키 교환 프로토콜은 Zhang등에 의해서 제시된 기법이다. 아래 그림에서 사각형 내부의 부분은 인증된 프로토콜을 위한 부분으로 인증이 필요하지 않는 환경에서는 생략할 수 있는 부분이다.

procedure CombineThree($U[1,2,3], s[1,2,3]$)
$i = 1$ to 3 **do**
  $Rep(U_i)$ computes $P_i = s_i P$
  and $T_{Rep(U_i)} = \hat{H}(P_i) S_{Rep(U_i)} + s_i P_i$;
  Let $\{j,k\} = \{1,2,3\} \backslash \{i\}$;
  $Rep(U_i)$ sends P_i, $T_{Rep(U_i)}$ to all members of both U_j, U_k;
end do

$i = 1$ to 3 **do**
  Let $\{j,k\} = \{1,2,3\} \backslash \{i\}$;
each member of U_i
verifies : $e(T_{Rep(U_j)} + T_{Rep(U_k)}, P) = e(\hat{H}(P_j) Q_{Rep(U_j)} + \hat{H}(P_k) Q_{Rep(U_k)}, P_{pub})$
$e(P_j, P_j) e(P_k, P_k)$ and
computes $H(e(P_j, P_k)^{s_i})$;
end do

end CombineThree

〔그림 5〕 CombineThree 알고리즘

procedure CombineTwo($U[1,2], s[1,2]$)
$i = 1$ to 2 **do**
  $Rep(U_i)$ computes $P_i = s_i P$
  and $T_{Rep(U_i)} = \hat{H}(P_i) S_{Rep(U_i)} + s_i P_i$;
end do
$Rep(U_1)$ generates $\overline{s} \in_R Z_q^*$ and sends $\overline{s}P$
and $\overline{T}_{Rep(U_1)} = \hat{H}(\overline{s}P) S_{Rep(U_1)} + \overline{s}^2 P$ to the rest of the users;
each member of U_1, U_2 except $Rep(U_1)$ verifies :
$e(\overline{T}_{Rep(U_1)}, P) = e(\hat{H}(\overline{s}P) Q_{Rep(U_1)}, P_{pub}) e(\overline{s}P, \overline{s}P)$;
$Rep(U_1)$ sends P_1, $T_{Rep(U_1)}$ to all members of U_2;
$Rep(U_2)$ sends P_2, $T_{Rep(U_2)}$ to all members of U_1;
each member of U_1
verifies : $e(T_{Rep(U_2)}, P) = e(\hat{H}(P_2) Q_{Rep(U_2)}, P_{pub}) e(P_2, P_2)$ and
computes $H(e(P_2, \overline{s}P)^{s_1})$;
each member of U_2
verifies : $e(T_{Rep(U_1)}, P) = e(\hat{H}(P_1) Q_{Rep(U_1)}, P_{pub}) e(P_1, P_1)$ and
computes $H(e(P_1, \overline{s}P)^{s_2})$;

end CombineTwo

〔그림 6〕 CombineTwo 알고리즘

procedure KeyAgreement($m, U[i+1, \ldots, i+m]$)
  if $(m = 1)$ **then**
    $KEY = s[i+1]$;
  end if
  if $(m = 2)$ **then**
    call CombineTwo($U[i+1, i+2], s[i+1, i+2]$);
    Let KEY be the agreed key between user sets U_{i+1}, U_{i+2};
  end if
  $n_0 = 0$; $n_1 = \lfloor \frac{m}{3} \rfloor$; $n_3 = \lceil \frac{m}{3} \rceil$; $n_2 = m - n_1 - n_3$;
  $j = 1$ to 3 **do**
    call KeyAgreement($n_j, U[i + n_{j-1} + 1, \ldots, i + n_{j-1} + n_j]$);
    $\hat{U}_j = U[i + n_{j-1} + 1, \ldots, i + n_{j-1} + n_j]$; $\hat{s}_j = KEY$; $n_j = n_{j-1} + n_j$;
  end do;
  call CombineThree($\hat{U}[1,2,3], \hat{s}[1,2,3]$);
  Let KEY be the agreed key among user sets $\hat{U}_1, \hat{U}_2, \hat{U}_3$;
end KeyAgreement

〔그림 7〕 KeyAgreement 알고리즘

CombineThree와 CombineTwo 알고리즘은 Key Agreement에서 하위 알고리즘으로 사용된다. 그룹 키 생성과정은 Reddy와 Nally의 방법과 동일하므로 구체적인 방법에 대한 기술은 생략하기로 한다. 다음 [그림 7]은 KeyAgreement 알고리즘을 나타낸다.

3.3 Choi, Hwang과, Lee의 기법

앞에서 살펴본 트리 기반의 그룹 키 교환 기법들은 트리의 깊이의 수만큼 키 교환이 필요하다는 단점이 있다. 그래서 많은 사용자를 가지는 그룹에 적용하기에는 비효율적이다. 이 논문에서는 참가자 그룹에 대해 확장 효과적인(scalable) 2라운드를 가지는 그룹 키 교환 기법을 제시하였다. 특히, Pairing 계산에서 단지 $O(n)$의 복잡도를 가진다.

제시된 ID 기반의 인증된 그룹 키 교환 기법은 다음과 같다.

BDH 파라미터 생성 알고리즘. BDH 파라미터 생성 알고리즘 IG_{BDH}는 보안 상수 1^k를 입력 값으로 받아 위수 q인 두 군 G_1, G_2와 admissible bilinear map e를 다항식 시간 안에 출력하는 알고리즘이다.

Setup. 먼저 KGC는 BDH 파라미터 생성기를 실행한다. 임의의 $s \in Z_p$를 선택하여 G_1의 생성자 P를 이용해서 $P_{pub} = sP$를 계산한다. 그리고 KGC는 s를 마스터 비밀 키로 하고, 시스템 파라미터를 $params = \{e, G_1, G_2, q, P, P_{pub}, H, H_1\}$을 공개 값으로 하여 공개한다.

Extract. 사용자가 자신의 ID를 이용한 키 쌍을 얻기 원할 때, KGC는 $Q_{ID} = H_1(ID)$을 공개키로 $S_{ID} = sQ_{ID}$를 비밀키로 계산하여 (Q_{ID}, S_{ID})를 사용자에게 돌려준다.

$U_1, U_2, \ldots, U_n$을 그룹 키를 생성하려는 사용자들의 집합 이라고 하자. $ID_1, ID_2, \ldots ID_n$은 각 사용자의 확인자 (ID)이다. 모든 참가자들은 순번이 순환이 되도록 정해져 있다. 즉, 사용자 인덱스는 법(modular) n(=사용자의 수)의 규칙을 따르며, $U_i (1 \le i \le n)$는 그룹의 i번째 사용자를 나타내며 롱텀 공개키와 비밀키는 $[ID_i, S_i = sQ_i]$이다.

Round 1. 각 사용자 U_i는 $a_i \in Z_q^*$값을 임의로 선택하고 $P_i = a_i P$, $h_i = H(P_i)$와 $T_i = a_i P_{pub} + h_i S_i$를 계산한다. U_i는 비밀값으로 하고 $\langle P_i, T_i \rangle$를 브로드캐스트 한다.

Round 2. 각 사용자 U_i는 $\langle P_{i-1}, T_{i-1} \rangle$, $\langle P_{i+1}, T_{i+1} \rangle$와 $\langle P_{i+2}, T_{i+2} \rangle$을 다른 사용자들로부터 받아 다음과 같이 메시지에 대한 검증을 한다.

$$e\left(\sum_{k\in\{-1,1,2\}} T_{i+k}, P\right) = e\left(\sum_{k\in\{-1,1,2\}} (P_{i+k} + h_{i+k} Q_{i+k}), P_{pub}\right)$$

만약 위와 같은 식이 성립하면 U_i는 $D_i = e(a_i(P_{i+2} - P_{i-1}), P_{i+1})$를 계산하여 다른 사용자들에게 전송한다. 그 외의 경우는 프로토콜을 중지한다.

Key Computation. 각 사용자 U_i는 다음과 같이 그룹 키를 계산한다.

$$K_i = e(a_i P_{i-1}, P_{i+1})^n D_i^{n-1} D_{i+1}^{n-2} \cdots D_{i-2}$$

정당한 모든 사용자들은 공통된 그룹 키 $K = e(P,P)^{a_1 a_2 + a_2 a_3 + \cdots + a_{n-1} a_1}$를 생성한다. 라운드 2에서 각 사용자들은 메시지 인증과정을 수행한다. 이 인증과정은 사용자들이 받은 각각의 메시지에 대한 개별적인 인증은 제공하지 않는다. 그러나 위의 프로토콜에서 메시지 인증의 목적은 받은 메시지들에 대한 정당성만은 요한다.

3.4 Shi, Chen와 Li의 기법

이 논문에서는 이전에 존재하는 기법보다 효율적인 ID 기반의 그룹 키 교환 기법을 제시하였다. 제안하는 기법은 1라운드가 필요하고, 서명 검증 없이 인증을 제공한다. 하지만 이 기법은 사용자들이 그룹 관리자에 의해서 생성된 공개키를 가지고 있어야 하고, 다른 사용자의 공개키를 사용하기 위해서 검증과정이 필요하다. 이러한 과정은 인증서 기반의 PKI의 개념에 더 가깝다.

Setup. 주어진 보안상수 1^k에 대해서, KGC는 시스템 파라미터를 다음과 같이 생성한다.

$\langle q, G_1, G_2, e, P, P_{pub}, P_{pub}', H_1 \rangle$

G_1과 G_2는 위수가 q인 두 개의 군이다. P는 G_1의 생성자이다. s_1과 s_2는 KGC의 비밀키로 Z_q^*에서 랜덤하게 선택된 값이고, $P_{pub} = s_1 P$와 $P_{pub}' = s_2 P$는 공개키이다. e는 $e: G_1 \times G_1 \rightarrow G_2$인 bilinear map이고 $H_1 : \{0,1\}^* \rightarrow Z_q^*$은 암호학적인 해쉬 함수 이다.

Key Extraction. 사용자 U_i가 자신의 ID_i를 KGC에게 보내면, KGC는 $I_i = H_1(ID_i)$를 계산한다. 그리고 $Q_i = (I_i s_1 + s_2)P$를 사용자의 공개키로 한다. 다른 사용자들은 이 공개키를 $I_i P_{pub} + P_{pub}'$으로 계산할 수 있다. KGC는 U_i의 개인키로 $S_i = (I_i s_1 + s_2)^{-1} P$를 계산하여 사용자 U_i에게 안전하게 전송한다.

$U_1, U_2, ..., U_n$을 그룹 키를 생성하려는 사용자들 이라고 하고, $ID_i (1 \le i \le n)$을 사용자들의 식별자라고 하자. U_i는 KGC로부터 받은 자신의 공개키 Q_i와 개인키 S_i를 가지고 있다. 키쌍 (Q_i, S_i)를 롱텀 개인키(long-term private key)라고 한다.

Round 1. 각 사용자 $U_i (1 \le i \le n)$는 임의의 정수 $a \in Z_q^*$를 선택하고 자신의 임시 개인키(ephemeral private key)로 한다. 그리고 U_i는 자신의 공개키를 제외한 다른 사용자의 공개키를 이용하여 $T_{i,j} = a_i Q_j$ $(1 \le j \le n, j \ne i)$를 계산한 후 U_j에게 전송한다.

Key Computation. U_i가 다른 사용자들로부터 $T_{1,i}, T_{2,i}, ..., T_{i-1,i}, T_{i+1,i}, ..., T_{n,j}$를 받으면, 세션키를 다음과 같이 계산한다.

$$K_i = e(T_{1,i} + T_{2,i} + \cdots + T_{i-1,i} + aQ_i + T_{i+1,i} + \cdots + T_{n,j}, S_i)$$

위의 식은 모든 사용자들이 동일한 키를 계산할 수 있게 됨을 다음과 같이 확인할 수 있다.

$$\begin{aligned} K_i &= e(T_{1,i} + T_{2,i} + \cdots + T_{i-1,i} + aQ_i + T_{i+1,i} + \cdots + T_{n,j}, S_i) \\ &= e(Q, S_i)^{a_1 + a_2 + \cdots + a_n} \\ &= e(P,P) \end{aligned}$$

3.5 Zhou, Susilo와 Mu의 기법

4절에서 소개한 논문에서 쓰이는 기법은 변형된 ID-PKI의 그룹 키 교환 방식이다. 이 논문에서는 ID기반의 1라운드를 가지는 효율적인 인증된 그룹 키 교환 프로토콜을 제안하였다. 그리고 이 프로토콜을 변형하여 전송효율이 더 좋은 2라둔드를 가지는 ID 기반 그룹 키 교환 기법도 제안하였다.

3.5.1 1라운드를 가지는 ID기반의 인증된 그룹 키 교환 프로토콜(O-AGKA)

그룹 관리자(Group Administrator: GA)는 암호학적 해쉬 함수 $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : G_2 \rightarrow \{0,1\}^n$와 $H_3 : \{0,1\}^n \rightarrow \{0,1\}^n$를 선택한다.

Setup. GA는 BDH 파라미터 생성자를 이용하여 소수 q, 위수가 q인 두 개의 군 G_1, G_2와 bilinear map $e : G_1 \times G_1 \rightarrow G_2$를 생성한다. GA는 임의의 생성자 $P \in G_1$와 임의의 난수 $s \in Z_q^*$를 선택한 후 $P_{pub} = sP$를 계산한다. 그리고, GA는 s를 마스터 비밀키로 하고 공개 시스템 파라미터 $params = \{e, G_1, G_2, q, P, P_{pub}, H_1, H_2, H_3\}$를 공개한다.

Extract. 확인자(identity)가 ID_i인 사용자 U_i가 비밀키를 얻기 위해서, GA는 $Q_i = H_1(ID_i)$를 계산한 후 사용자의 롱텀 비밀키로 $S_i = sQ_i$를 계산하여 사용자에게 안전한 채널로 전송한다.

$U_1, ..., U_n$을 그룹 세션키를 생성하기를 원하는 사용자들의 집합이라고 하자. 그룹 키 교환 프로토콜은 다음과 같다.

Round 1. 각 사용자 U_i는 먼저 임의의 난수 $\delta_i \leftarrow G_2$와 $r_i, k_i \leftarrow \{0,1\}^n$을 선택한 후, $P_i^j = H_2(e(S_i, Q_j) \cdot \delta_j) \oplus r_i$, $1 \le j \le n, j \ne i$를 계산한다. 그리고 U_i는 D_i를 다음과 같이 계산하여 다른 사용자들에게 전송한다.

$D_i = \langle \delta_i, P_i^1, ..., P_i^{i-1}, P_i^{i+1}, ..., P_i^n, H_3(r_i) \oplus k_i, \mathfrak{I} \rangle$

$\mathfrak{I}$는 P_i^j가 어느 사용자와 연관되어 있는지에 관한 정보를 포함하는 분류표시이다.

Key Computation. $D_j = \langle R_j, P_j^1, ..., P_j^n, V_j, \mathfrak{I} \rangle$이라 하자. 다른 사용자들로부터 D_j를 전송 받은 후에, 사용자 U_i는 $\mathfrak{I}$의 정보를 검색하여 자신의 P_j^i를 찾는다. 그리고 다음을 계산한다.

$k_j' = H_3\big(H_2(e(Q_j, S_i) \cdot R_j) \oplus P_j^i\big) \oplus V_j$

각 사용자 U_i는 공통된 세션키를 다음과 같이 계산할 수 있다.

$K = K_i = k_1' \oplus \cdots \oplus k_{i-1}' \oplus k_i \oplus \cdots \oplus k_n'$

3.5.2 2라운드를 가지는 ID기반의 인증된 그룹 키 교환 프로토콜(T-AGKA)

T-AGKA는 O-AGKA를 변형한 것으로 O-AGKA보다 전송량이 효율적인 기법이다.

Setup. O-AGKA 기법과 동일하다. 추가적으로 T-AGKA는 새로운 세 개의 암호학적 해쉬 함수 $H_4 : G_2 \rightarrow \{0,1\}^n$, $H_5 : \{0,1\}^n \rightarrow Z_q^*$, $H_6 : G_1 \rightarrow \{0,1\}^n$을 생성한다.

Extract. O-AGKA 기법과 동일하다.

$U_1, ..., U_n$을 그룹 세션키를 생성하기를 원하는 사용자들의 집합이라고 하면. 그룹 키 교환 프로토콜은 다음과 같다.

Round 1. 개시자(initiator) U_1은 먼저 임의의 난수 $\delta \leftarrow G_2$, $r \leftarrow \{0,1\}^n$과 $k_1 \leftarrow Z_p^*$를 선택한다. 그리고 U_1은 $D_1 = \langle R, P_2, ..., P_n, V, W, \mathfrak{I} \rangle$을 다음과 같이 계산하여 다른 사용자들에게 전송한다.

$$D_1 = \langle \delta, r \oplus H_4(e(S_1, Q_2) \cdot \delta), ..., r \oplus H_4(e(S_1, Q_n) \cdot \delta), H_5(r) \cdot k_1 P, k_1, P_{pub}, \mathfrak{I} \rangle$$

$\mathfrak{I}$은 P_i가 어느 사용자와 연관되어 있는지에 관한 정보를 포함하는 분류 표시이다.

Round 2. 각 응답자 $U_i (2 \le i \le n)$가 U_1으로부터 D_1을 받으면, $\mathfrak{I}$의 정보를 이용하여 자신의 P_i를 찾은 후에 $r' = H_4(e(Q_1, S_i) \cdot R) \oplus P_i$를 계산한다. 만약에 메시지 D_1이 정당하다면 $r' = r$임을 알 수 있다. 그리고 U_i는 임의의 난수 $k_i \leftarrow Z_p^*$를 선택한 후 $D_i = \langle H_5(r) \cdot k_i P, k_i P_{pub} \rangle$를 계산하여 다른 모든 사용자들에게 전송한다.

Key Computation. $D_j = \langle X_j, Y_j \rangle$라 하자. 다른 사용자들로부터 D_j를 전송 받은 후에, 개시자 U_1과 사용자 U_i는 $z_1 = H_5(r)^{-1} \cdot V$ 과 $z_j = H_5(r)^{-1} \cdot X_j$ $(2 \le j \le n)$를 계산한다. 그리고, 각 사용자 U_i는 계산된 z_j를 저장하고 z_j가 올바른 값인지를 다음의 식을 이용하여 확인한다.

$$e\Big(P, \sum_{j=1}^{n} Y_j\Big) \stackrel{?}{=} e\Big(P_{pub}, \sum_{j=1}^{n} z_j\Big)$$

〔표 1〕 ID기반 그룹 키 교환 프로토콜의 비교

프로토콜	계산량			안전성	
	라운드 수	연산량 (각 사용자)	전송 메시지 수 (각 사용자)	제공 안전성	가정
Reddy 외	$\log n$	$\log n$×곱셈 + $2n\log n$×페어링	$O(\log n)$	–	–
Barua 외	$\lceil \log_3 n \rceil$	$< \frac{5}{2}(n-1)$× 스칼라곱	$< \frac{5}{2}(n-1)$	FS & KK	스탠다드
Choi 외	2	$O(1)$×페어링	$O(n)$	FS & KK	랜덤 오라클
Shi 외	1	$2(n-1)$×곱셈+1×페어링	$n-1$	–	–
Zhou 외	1	n^2×페어링	$n(n+2)$	KK	랜덤 오라클
	2	$4n$×페어링	$3n$	KK	랜덤 오라클

위의 식을 계산하여 z_j가 모두 올바른 값으로 검증되면, U_i는 다른 그룹 사용자들이 정당한 사용자라는 것을 확인할 수 있게 된다. 따라서 모든 사용자 U_i는 공통된 세션키를 다음과 같이 계산할 수 있다.

$K = K_i = H_6(z_1) \oplus \cdots \oplus H_6(z_n)$

Ⅳ. 분 석

본 장에서는 앞에서 소개한 기법들에 대해서 비교 분석해보기로 한다.

[표 1]은 3장에서 소개한 기법들에 대한 비교 분석한 표이다. 여기서 사용하는 기호들의 의미는 다음과 같다.

- n 그룹 사용자의 수
- FS 전방향 안전성(forward secrecy)
- KK 기지 키 공격에 대한 안정성 (known-key secrecy)

Ⅴ. 결 론

본 논문에서는 ID 기반의 그룹 키 교환 프로토콜의 기술 동향을 분석하였다. 그리고 그룹 키 교환 환경에서의 다양한 공격 환경 및 공격자에 따른 안전성 모델에 대한 안전성을 분석하였다. 이를 통해 키 교환 프로토콜 분야를 연구하는 관련자에게도 동향을 분석하고 이해하는데 도움이 될 것이다.

참고문헌

[1] A. Shamir, "Identity Based Cryptosystems and Signature Schemes", Advances in Cryptology-CRYPTO'84, Springer-Verlag, LNCS 196, pages 47-53, 1985

[2] D. Boneh and M. Franklin, "Identity Based Encryption from the Weil Pairing", Advances in Cryptology-CRYPTO 2001, Springer-Verlg, LNCS 2139, pages 213-229, 2001.

[3] K. C. Reddy, and D. Nalla. "Identity Based Authenticated Group Key Agreement Protocol", In Proceeding of INDOCRYPT 2002, LNCS 2551, pages 215-233, 2002.

[4] N. P. Smart, "An Identity Baed Authenticated Key Agreement Protocol Based on the Weil Pairing", Cryptology ePrint Archive, Report 2001/111, 2001, http://eprint.iacr.org/.

[5] R. Barua, R. Dutta, and P. Sarkar. "Extending Joux's Protocol to Multi Party Key Agreement", In Proceeding of INDOCRYPT 2003, LNCS 2904, pages 205-217, 2003.

[6] A. Joux. "A One Round Protocol for Tripartite Diffie-Hellman", In Proceeding of ANTS IV, LNCS 1838, pages 385-394, 2000.

[7] K. Y. Choi, J. Y. Hwang and D. H. Lee. "Efficient ID-Based Group Key Agreement with Bilinear Maps", 2004 International Workshop on Practice and Theory in Public Key Cryptography (PKC '04), LNCS 2947, pages 130-144. 2004.

[8] X. Du. Y. Wang, J. Ge, and Y. Wang. "ID-

Based Authenticated Two Round Multi-Party Key Agreement", Cryptology ePrint Archive, Report 2003/247, 2003.

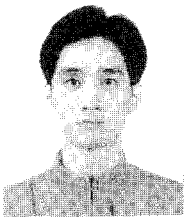
[9] F. Zhang, and X. Chen. "Attack on an ID-based AuthenticatedGroup Key Agreement Scheme from PKC 2004", Information Processing Letters, Vol. 91, pages 191-193, 2004.

[10] X. Du. Y. Wang, J. Ge, and Y. Wang. "An Improved ID-Based Authenticated Group Key Agreement Scheme", Cryptology ePrint Archive, Report 2003/260, 2003.

[11] Y. Shi, G. Chen, and J. Li. "Id-Based One Round Authenticated Group Key Agreement Protocol with Bilinear Pairing", International Conference on Information Technology: Coding and Computing (ITCC '05), Vol. I, pages 757-761, 2005.

[12] L. Zhou, W. Susilo, and Y. Mu. "Efficient ID-Based Authenticated Group Key Agreement from Bilinear Pairing", In Proceeding of MSN '06, LNCS 4325, pages 521-532, 2006.

〈著者紹介〉



최 재 탁 (Jae Tark Choi)
정회원

2002년: 충북대학교 수학과 학사
2005년: KAIST 수학과 석사
2005~현재: 고려대학교 정보경영공학전문대학원 박사과정
<관심분야> 암호이론, 암호프로토콜, 키 교환

이 동 훈 (Dong Hoon Lee)

1984년: 고려대학교 경제학 학사
1987년: University of Oklahoma 전산학과 석사
1992년: University of Oklahoma 전산학과 박사
1993년~1997년: 고려대학교 전산학과 조교수
1997년~2001년: 고려대학교 전산학과 부교수
2001년~현재 : 고려대학교 정보보호대학원 교수
<관심분야> 암호이론, 암호프로토콜, USN 이론, 키 교환, 익명성 연구, PET 기술