속성 기반의 이자간 키 교환 방법*

*이광수 **이동훈
고려대학교, 정보경영공학전문대학원
{guspin, donghlee}@korea.ac.kr

Attribute-Based Two-Party Key Agreement

*Kwangsu Lee **Dong-Hoon Lee
Graduate School of Information Management and Security, Korea Univ.

요약

속성 기반의 키 교환 방법은 사용자의 아이디가 다수의 속성들로 표현되고, 사용자가 지정한 정책을 만족하는 속성들을 소유한 두 사용자간에 안전한 키 교환이 가능한 방법이다. 속성 기반 키 교환은 기존의 아이디 기반 키 교환 (Identity-Based Key Agreement) 방법과 달리 키 교환을 수행할 상대방을 미리 결정할 필요가 없다. 즉, 사용자는 단지 키 교환을 맺고자 하는 상대 사용자가 어떠한 속성을 가지기를 지정한 정책을 지정하고 이 정책을 만족하는 임의의 사용자와 키 교환을 수행하게 된다. 속성 기반 키 교환은 사용자를 자신이 수행하는 역할들의 리스트로 기술하여 접근 통제를 가능하게 하는 역할 기반 접근 통제 (Role-Based Access Control) 시스템에 적용이 가능하다.

1. 서론

키 교환 프로토콜은 안전한 통신 및 인증 등의 작업을 수행하기 위한 가장 기본적인 요소다. 키 교환 프로토콜은 초기 Diffie-Hellman의 논문 [7] 이후 이자간 키 교환, 그룹 키 교환, 패스워드 기반 키 교환 등 다양한 환경에 적용 가능한 많은 연구가 진행 되었다 [3,5,9,1].

기존 키 교환 프로토콜의 제약점은 키 교환을 수행하기 전에 미리 키 교환을 수행할 상대방이 누구인지 정확이 알고 있어야 한다는 것이다. 일반적으로 키 교환을 수행하고자 하는 상대방은 소수이기 때문에 그 리스트를 저장하고 그에 대한 정보를 가지고 있으면 누구와 키 교환을 수행하면 어떤 작업을 할 수 있는지 정확히 알 수 있기 때문에 대부분의 환경에서는 이런 가정이 적절하다. 하지만 일상생활에서 우리가 일을 처리하는 방식을 살펴보면 이와는 다소 다르다는 것을 알 수 있다. 한 예로 병원에서 진료를 받는 경우 환자는 정확히 어떤 의사와 접촉해야 할지 알 수 없고 단지 의사 직책을 가진 사람과 필요한 진료를 받을 수 있는 것이다. 이를 키 교환 시나리오로 다시 고려하면 정확인 누구와 키 교환을 수행해야 할지 모르고 단지 특정 권한을 가지고 있는 사람과 키 교환을 수행하는 것으로 충분한 것이다.

권한 또는 역할을 이용하는 대표적인 시스템은 역할 기반 접근 통제 (Role-Based Access Control) 시스템이다 [13]. 역할 기반 접근 통제 시스템에서는 모든 참여자를 그가 가지고 있는 역할의 리스트로 표현한다. 한 예로 직장의 모든 직원을 {이름, 부서, 직책, 성별, 나이} 리스트로 표현하는 것이다. 이와 같이 모든 직원을 역할의 리스트로 구분하는 경우 직원에 대한 접근 통제 시스템을 구현하는 것이 용이하다. 이와 같이 시스템이 역할 기반으로 주체를 표현하는 경우 이들 간에 안전한 통신을 위해서 역할 기반에 적합한 메시지 암호화, 전자 서명 그리고 키 교환 방법 등이 필요하게 된다.

본 논문에서는 이와 같이 사용자의 아이디가 속성 (또는 역할이나 권한)들의 리스트로 표현되는 시스템에 적합한 속성 기반 이자간 키 교환 기법을 제시한다. 제시된 속성 기반 키 교환 방법은 최초의 방법으로 수동적 공격자에 대해서 안전성을 제공한다. 또한 키 교환 상대방에 대한 추가적인 어떠한 정보도 제공하지 않는 익명성을 제공한다.

암호 시스템 설계시 사용자의 아이디가 속성으로 표현되는 경우를 고려한 것은 Sahai와 Waters가 제시한 퍼지 아이디 기반 암호화 기법이 최초이다 [12]. 그 후 Goyal 등은 좀 더 일반화된 속성 기반 암호화 시스템을 제시하였다 [8]. 속성 기반 암호화 시스템은 기존 아이디 기반 암호화 시스템 [14]을 일반화 시킨 형태이다. 그 후 속성 기반 암호화에 대한 다양한 연구가 진행되고 있다 [11,4,6,10].

본 논문의 구성은 다음과 같다. 먼저 2절에서는 논문의 이해에 필요한 배경 지식을 기술하고 3절에서는 속성 기반 키 교환을 정의한다. 그런 뒤 4절에서는 속성 기반 키 교환 기법을 제시하고 안전성을 보인다. 마지막으로 5절에서는 결론을 맺도록 한다.

2. 배경 지식

이 절에서는 본 논문을 이해하기 위해 필요한 배경 지식으로 접근 구조, bilinear 함수, DBDH 가정, 유사 랜덤 함수 그리고 Lagrange 보간법을 설명한다.

가. 접근 구조

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-(C1090-0801-0025))

접근 구조(Access Structure)는 다음과 같이 정의된다 [2]. 먼저 $\{P_1, P_2, ..., P_n\}$를 속성들의 집합이라고 하자. 이때 모임 $A \subseteq 2^{\{P_1, P_2, ..., P_n\}}$에 대하여 모든 집합 B, C에 대하여 만일 $B \in A$이고 $B \subseteq C$이면 $C \in A$ 만족하는 경우 모임 A는 단조라고 한다. 접근 구조 (또는 단조 접근 구조)는 $\{P_1, P_2, ..., P_n\}$ 집합의 공집합이 아닌 부분 집합의 모임으로 정의된다. 즉, $A \subseteq 2^{\{P_1, P_2, ..., P_n\}} \setminus \{\varnothing\}$.

나. Bilinear 함수

먼저 G와 G_T는 소수 위수 p를 갖는 곱셈 순환군 (multiplicative cyclic group)이다. 그리고 g는 G의 생성원이고 e는 bilinear 함수로 $e : G \times G \rightarrow G_T$로 정의되고 다음의 성질을 가지는 계산 가능한 함수이다.

- Bilinearity: 모든 $u, v \in G$와 $a, b \in Z_p^*$에 대해, $e(u^a, v^b) = e(u, v)^{ab}$가 성립한다.
- Non-degeneracy: 모든 $u, v \in G$에 대해, $e(u, v) \neq 1$이 성립한다.

만일 G에서의 그룹 연산과 bilinear 함수 e의 연산을 효율적인 계산 가능한 경우 G를 bilinear 그룹이라고 한다.

다. DBDH 가정

먼저 그룹 G는 소수 p를 윗수로 가진 순환군이라 하자. 이때 Bilinear 그룹 G에서 DBDH (Decisional Bilinear Diffie-Hellman) 가정은 어떠한 확률적인 다항시간 알고리즘 A도 $(g, g^a, g^b, g^c, e(g,g)^{abc})$ 리스트와 $(g, g^a, g^b, g^c, e(g,g)^z)$ 리스트를 의미 있는 확률로 구분할 수 없다는 것이다.

라. 유사 랜덤 함수

유사 랜덤 함수(Pseudo-Random Function)은 함수의 집합 F로 정의되며 어떠한 확률적인 다항시간 알고리즘 A도 랜덤하게 선택한 키 K를 이용하는 $F_K()$ 함수와 모든 함수의 집합에서 랜덤하게 선택된 랜덤 함수 $R()$ 값을 의미 있는 확률로 구분할 수 없는 성질을 지닌다.

마. Lagrange 보간법

Lagrange 보간법은 주어진 $k+1$개의 좌표 정보 $(1, y_1), (2, y_2), ..., (k+1, y_{k+1})$를 이용하여 유일한 k차 다항식을 다음과 같은 식으로 계산 할 수 있다는 것이다.

$$q(x) = \sum_{1 \le i \le k+1} y_i \Delta_{i,N}(x)$$

이때 $N = \{1, 2, ..., k+1\}$이고 Lagrange 상수는 $\Delta_{i,N}(0) = \prod_{j \in N, j \neq i} \frac{x-j}{i-j}$로 정의된다.

3. 속성 기반의 이자간 키 교환 정의

속성 기반 이자간 키 교환 기법은 두 개의 알고리즘과 하나의 프로토콜로 이루어진다. 자세한 설명은 다음과 같다.

- *Setup*: 이 알고리즘은 보안 파라미터를 입력으로 받아서 공개 파라미터 PP와 마스터 키 MK 출력한다.
- *KeyGen*: 사용자 키 생성 알고리즘은 사용자의 비밀키를 기술하는 속성 집합, 마스터 키 MK 그리고 공개 파라미터 PP를 입력으로 받아서 사용자의 비밀키 UK를 출력한다.
- *KeyAgree*: 키 교환 프로토콜은 접근 구조를 입력으로 받아서 이들 접근 구조를 만족하는 두 사용자간의 키 교환을 수행하고 그 결과로서 세션키 SK를 출력한다.

가. 안전성 정의

속성 기반 이자간 키 교환은 두 가지 안전성을 만족해야 한다. 첫 번째는 일반적인 키 교환의 안전성과 동일한 의미론적 안전성 (Semantic Security)으로 공격자는 세션 키에 대한 어떠한 정보도 얻을 수 없어야 한다는 것이다. 두 번째는 공격자는 프로토콜에 참여한 사용자가 누군지 구분할 수 없어야 한다는 익명성(Anonymity) 조건을 만족해야 한다.

4. 속성 기반의 이자간 키 교환 기법

본 장에서는 속성 기반 이자간 키 교환 기법을 제안한다. 먼저 접근 권한을 접근 트리로 구현하는 방법을 알아보고 속성 기반 이자간 키 교환 기법을 기술한다.

가. 접근 트리

접근 권한을 구현하는 하나의 방법은 접근 트리(Access Tree)를 이용하는 것이다. 접근 트리 T는 접근 권한 A를 표현하는 트리로 각각의 내부 노드는 입계 값과 자식 노드를 가지는 입계치 게이트이다. 만일 노드 x의 자식 수가 최대 num_x인 경우 입계치 값 k_x는 $1 \le k_x \le \mathrm{num}_x$ 값을 가진다. 그리고 각각의 리프 노드는 속성 값과 입계치 값이 1인 노드이다.

접근 트리를 이용하기 위해서 아래와 같은 함수들을 정희하자. 먼저 *parent(x)* 함수는 노드 x의 부모 노드를 지정한다. *att(x)* 함수는 노드 x가 리프 노드인 경우 속성 값을 지정한다. *index(x)* 함수는 노드 x가 부모노드의 자식 노드 중에서 몇 번째 순서인지 지정한다.

만일 트리 T가 루트 노드 r을 가지는 접근 트리인 경우, T_x는 노드 x를 루트로 하는 T의 서브-트리이다. 만일 속성 집합 S가 접근 트리 T_x를 만족하는 경우, $T_x(S) = 1$ 로 표기한다. 그리고 $T_x(S)$ 계산은 다음과 같다. 먼저 노드 x가 내부 노드인 경우, x의 모든 자식 노드 x'에 대하여 $T_{x'}(S)$ 값을 계산한다. 만일 k_x 보다 많거나 같은 수의 자식에 대해서 1 값이 성립하면 $T_x(S)$ 값은 1이다. 만일 노드 x가 리프 노드인 경우 노드 x의 속성이 S의 원소이면 $T_x(S)$ 값은 1이다.

접근 트리에 다항식 값을 할당하는 알고리즘 *AssignPoly*(T, a)은 접근 트리 T와 임의의 값 a를 입력으로 받아서 루트 노드 r부터 리프 노트까지 다항식의 값을 할당하기 위해서 다음과 같은 과정을 수행한다. 먼저 모든 접근 트리에 있는 모드 노드 x에 대하여 다항식의 차수 d_x 값을 입계치 값 k_x 보다 하나 작은 값으로 설정한다. 즉, $d_x = k_x - 1$. 그 다음 루트 노드 r부터 시작해서, 루트 노드의 경우

다항식 값으로 $q_r(0)=a$ 값을 할당하고 d_r 개의 랜덥 값을 추가로 선택하여 다항식 q_r을 설정한다. 다른 노드 x의 경우 먼저 $q_x(0)=q_{parent(x)}(\mathrm{index}(x))$ 값으로 할당하고 d_x 개의 랜덤 값을 추가로 선택하여 다항식 q_x를 설정한다. 마지막으로 접근 트리의 모든 리프 노드 x에 대하여 $q_x(0)$ 값을 출력한다.

다항식 재구성 알고리즘 *AssemblePoly*$(T,\{Y_z\}_{z\in L})$ 함수는 접근 트리 T, 접근 트리 리프 노드 z에 대한 Y_z 값들의 집합을 입력으로 받아서 리프 노드에서 루트 노드까지의 다항식을 재구성하기 위해서 다음과 같은 과정을 수행한다. 먼저 접근 트리 T의 루트 노드를 x라고 하고 입력으로 받은 Y_z 값은 $\perp$ 또는 $X^{q_z(0)}$ 값으로 표현된다고 하자. 만일 노드 x 가 리프 노드인 경우 알고리즘은 Y_x 값을 리턴한다. 그렇지 않은 경우 노드 x의 모든 자식 노드 y에 대하여 *AssemblePoly*$(T_y,\{Y_z\}_{z\in L_y})$ 함수를 호출하고 그 결과를 Y_y 값에 저장한다. S_x 집합은 임의의 k_x 개의 자식 노드 y들의 모임에서 Y_y 값이 $\perp$ 이 아닌 것들의 집합이라고 하자. 이때 Y_x 값은 Lagrange 보간법을 이용하여 다음과 같이 계산된다. (먼저 i = *index(y)*이고 S'_x = *{index(y):*$y\in S_x$*}*로 표기하자)

$$\begin{aligned}Y_x &= \prod_{y\in S_x} Y_y^{\Delta_{i,S'_x}(0)} = \prod_{y\in S_x}(X^{q_y(0)})^{\Delta_{i,S'_x}(0)}\\ &= \prod_{y\in S_x}(X^{q_{\mathrm{parent}(y)}(\mathrm{index}(y))})^{\Delta_{i,S'_x}(0)}\\ &= \prod_{y\in S_x} X^{q_x(i)\cdot\Delta_{i,S'_x}(0)}\\ &= X^{q_x(0)}.\end{aligned}$$

나. 스킴

속성 기반 이자간 키 교환 스킴은 다음과 같이 기술된다.

- ***Setup***(1^k): 알고리즘은 먼저 전체 가능한 속성의 집합 $U=\{1,2,\ldots,|U|\}$ 을 정의한다. 그 다음 랜덤 생성원 $g\in G$을 선택하고 랜덤 난수 $\alpha\in Z_p^*$를 선택한다. 마지막으로 모든 속성 $i\in U$에 대하여 랜덤 난수 $t_i\in Z_p^*$선택한다. 이때 공개 파라미터 PP와 마스터 키 MK는 다음과 같다.

$$PP=(g,\ g_1=g^{\alpha},\ g_2=g^{\beta},\ u_1=g^{t_i},\ldots,u_{|U|}=g^{t_{|U|}})$$
$$MK=(\alpha,\ \beta,\ t_1,\ \ldots,t_{|U|}).$$

- ***KeyGen(S, MK, PP)***: 사용자 속성 S에 대한 비밀키를 생성하기 위해서 알고리즘은 먼저 랜덤 난수 $v\in Z_p^*$ 선택하고 비밀키를 다음과 같이 생성한다.

$$UK=(S,\ \{g^{(\alpha+v)/t_i}\}_{i\in S},\ g^{v/\beta}).$$

- ***KeyAgree***: 사용자 A의 비밀 키 UK_A는 속성 집합 S_A를 포함하고 있고 이 사용자는 접근 트리 T_B를 만족하는 상대방과 키 교환을 수행하려고 한다고 가정하자. 먼저 사용자 A는 자신의 속성 집합을 만족하는 T_A를 선택한다. 그다음 랜덤 난수 $s_a\in Z_p^*$ 선택하고 *AssignPoly*(T_B,s_a) 함수를 실행하여 모든 리프 노드 집합 L_B 원소에 $q_x(0)$ 값을 할당한다. 사용자 A는 아래의 메시지를 생성하여 다른 사용자들에게 전송한다.

$$TM_1=(T_A,\ T_B,\ \{u_i^{q_x(0)}\}_{x\in L_B},\ g_2^{s_a})$$

만일 임의의 사용자 B의 속성 집합 S_B가 접근 트리 T_B를 만족하는 경우, 사용자 B는 랜덤 난수 $s_b\in Z_p^*$ 선택하고 *AssignPoly*(T_A,s_b) 함수를 실행하여 모든 리프 노드 집합 L_A 원소에 $q'_x(0)$ 값을 할당한다. 사용자 B는 아래의 메시지를 생성하여 사용자 A에게 전달한다.

$$TM_2=(T_A,\ \{u_i^{q'_x(0)}\}_{x\in L_A},\ g_2^{s_b})$$

세션 키를 계산하기 위해서 두 사용자 A, B는 먼저 공유 키 계산 알고리즘을 이용하여 공유 키 K 값을 계산한다. 그런 뒤 유사 랜덤 함수(PRF)를 이용하여 다음과 같이 세션 키 SK를 계산한다.

$$SK=PRF_K(TM_1\|TM_2)$$

공유 키 계산 알고리즘 *ComputeKey*(TM,UK,s')은 입력으로 상대방에게서 받은 메시지 $TM=(T',T,\{E_x\}_{x\in L},E')$ (또는 $TM=(T,\{E_x\}_{x\in L},E')$) 과 사용자 비밀키 $UK=(S,\{D_i\}_{i\in S},D')$ 그리고 자신이 사용했던 랜덤 난수 s'을 입력으로 받아 다음과 같이 동작한다. 먼저 접근 트리 T에 있는 모든 리프 노드 x에 대하여 속성 값이 S 집합에 속하는 경우 Y_x 값을 아래와 같이 계산하고 아닌 경우 $\perp$ 값을 설정한다.

$$\begin{aligned}Y_x &= e(E_x,D_i)=e(g^{t_i\cdot q_x(0)},g^{(\alpha+v)/t_i})\\ &= e(g,g)^{(\alpha+v)\cdot q_x(0)}.\end{aligned}$$

그런 다음 *AssemblePoly*$(T,\{Y_z\}_{z\in L})$ 함수를 실행하여 다항식을 재구성하여 그 결과를 Y_r 값으로 저장하다. 만일 이 값이 $\perp$ 이면 종료한다. 그렇지 않은 경우 입력으로 받는 s' 값을 이용하여 아래와 같은 계산을 수행하여 공유 키를 계산한다.

$$\begin{aligned}K&=(Y_r\cdot e(E',D')^{-1})^{s'}\\ &=(e(g,g)^{(\alpha+v)s}\cdot e(g,g)^{-vs})^{s'}\\ &=e(g_1,g)^{s\cdot s'}.\end{aligned}$$

사용자 A의 경우 *ComputeKey*(TM_2,UK_A,s_a) 함수를 호출하고 사용자 B의 경우 *ComputeKey*(TM_1,UK_B,s_b) 함수를 호출하여 서로 동일한 공유 키 $K=e(g_1,g)^{s_a\cdot s_b}$ 를 계산할 수 있다.

다. 효율성 분석

제시된 키 교환 방법의 효율성을 분석하면 다음과 같다. 어떤 연산 $oper$을 수행하는데 걸리는 시간 비용을 t_{oper}라는 값으로 표현하는 경우 키 생성, 키 교환에 필요한 시간 비용은 다음과 같이 표현된다.

$$\begin{aligned}t_{keygen} &\simeq O(|S|)\cdot t_{\exp}\\ t_{keyagree} &\simeq O(|L|)\cdot t_{pair}+O(|L|)\cdot t_{\exp}\end{aligned}$$

이때 t_{keygen}은 키 생성에 필요한 시간 비용, $t_{keyagree}$는 키 교환에 필요한 시간 비용, t_{pair}는 bilinear 함수 연산에 필요한 시간 비용 그리고 $t_{\exp}$는 지수 연산에 필요한 시간 비용이다.

라. 안전성 분석

본 논문에서 제시된 키 교환 방법은 수동적 공격자에 대하여 의미론적 안정성을 제공한다. 증명의 아이디어는 속성 기반 이자간 키 교환 방법의 결과로 생성되는 세션 키의 형태가 DBDH 문제에서 구분해야 하는 값과 같은 형태이기 때문에 가능하다. 즉, 공격자가 선택한 사용자 A, B가 공유한 세션 키 값으로 DBDH 문제에서 주어진 값 $e(g,g)^{abc}$ 또는 $e(g,g)^{z}$ 값을 이용하여 공격자에게 제시하면 공격자가 올바른 값을 구분하는 경우 이를 이용하여 DBDH 문제를 풀 수 있다. 또한 공격자에게 추가적으로 주어져야 하는 다른 TM_i 메시지들 역시 DBDH 값들을 이용하여 충분히 시뮬레이션 하는 것이 가능하다.

프로토콜에 참여하는 사용자 A는 자신이 선택한 접근 트리 T를 만족하는 두 상대방 사용자 B, C가 존재하는 경우 사용자 A는 자신이 두 사용자 B, C 중에서 어떠한 사용자와 키 교환을 수행하고 있는지 알 수 없기 때문에 익명성을 제공한다. 증명의 아이디어는 두 상대방 사용자 B, C가 사용자 A가 선택한 접근 트리 T를 만족하는 경우 사용자 B와 사용자 C가 생성하는 TM_2 메시지 그리고 세션 키 값은 통계적으로 동일한 값이기 때문이다.

5. 결론

본 논문에서는 속성 기반의 이자간 키 교환 방법을 제시하였다. 제시된 방법은 최초의 속성 기반 이자간 키 교환 방법으로 수동적 공격자에 대하여 의미론적 안전성과 익명성을 제공한다. 앞으로의 연구 방향은 제시된 키 교환 스킴을 능동적 공격자에 안전하도록 설계하는 것과 시스템의 마스터 키가 노출 되어도 기존에 프로토콜을 수행한 사용자들 간의 세션 키 정보가 노출되지 않도록 설계하는 것이다.

참고문헌

[1] M. Abdalla, E. Bresson, O. Chevassut, and D. Pointcheval. Password-based group key exchange in a constant number of rounds. In PKC'06, 2006.

[2] A. Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[3] M. Bellare and P. Rogaway. Entity authentication and key distribution. In CRYPTO'93, 1993.

[4] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In Proceedings of the IEEE Symposium on Security and Privacy, 2007.

[5] E. Bresson, O. Chevassut, D. Pointcheval, and J.J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. In ACM conference on Computer and Communications Security (ACM CCS), 2001.

[6] M. Chase. Multi-authority attribute-based encryption. In The Fourth Theory of Cryptography Conference (TCC), 2007.

[7] W. Diffie and M. Hellman, New Directions in Cryptography, IEEE Trans. Inform. Theory, vol. IT-22, pp. 472-492, 1976.

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute based encryption for fine-grained access control of encrypted data. In ACM conference on Computer and Communications Security (ACM CCS), 2006.

[9] J. Katz and M. Yung. Scalable Protocols for Authenticated Group Key Exchange. In CRYPTO'03, 2003.

[10] R. Ostrovsky, A. Sahai and B. Waters. Attribute-based encryption with non-monotonic access structures. In ACM conference on Computer and Communications Security (ACM CCS), 2007.

[11] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure attribute-based systems. In ACM conference on Computer and Communications Security (ACM CCS), 2006.

[12] A. Sahai and B. Waters. Fuzzy identity-based encryption. In EUROCRYPT'05, 2005.

[13] R. Sandhu, E. Coyne, H. Feinstein, C. Youman. Role-based access control models. IEEE Computer, 1996.

[14] A. Shamir, Identity-based Cryptosystems and Signature Schemes, In CRYPTO'84, 1984.