Journal of the KIMST, Vol. 17, No. 3, pp. 358-363, 2014
ISSN 1598-9127
DOI http://dx.doi.org/10.9766/KIMST.2014.17.3.358

Research Paper 정보기술 부문

이진트리 기반의 속성기반 암호전송 알고리즘

이문식[*,1] · 김홍태[1] · 홍정대[2]

[1] 공군사관학교
[2] 국군기무사령부

Two Attribute-based Broadcast Encryption Algorithms based on the Binary Tree

Moon Sik Lee[*,1] · HongTae Kim[1] · Jeoung Dae Hong[2]

[1] *Faculty, Korea Air Force Academy*
[2] *Defense Security Command, Korea*

(Received 7 November 2013 / Revised 13 March 2014 / Accepted 11 April 2014)

ABSTRACT

In this paper, we present two constructions of the attribute-based broadcast encryption(ABBE) algorithm. Attribute-based encryption(ABE) algorithm enables an access control mechanism over encrypted data by specifying access policies among private keys and ciphertexts. ABBE algorithm can be used to construct ABE algorithm with revocation mechanism. Revocation has a useful property that revocation can be done without affecting any non-revoked uers. The main difference between our algorithm and the classical ones derived from the complete subtree paradigm which is apt for military hierarchy. Our algorithm improve the efficiency from the previously best ABBE algorithm, in particular, our algorithm allows one to select or revoke users by sending ciphertext of constant size with respect to the number of attributes and by storing logarithm secret key size of the number of users. Therefore, our algorithm can be an option to applications where computation cost is a top priority and can be applied to military technologies in the near future.

Key Words : Cryptography, Attribute-based Encryption, Broadcast Encryption, Binary Tree

1. 서 론

속성기반 암호 알고리즘(Attribute-based Encryption Algorithm : 이하 ABE)은 여러 가지 속성(Attributes)을 사용하여 접근을 제어할 수 있는 암호 알고리즘이다. 즉 비밀자료를 속성(나이, 성별, 부서 등등)에 따라 암호화를 하고 복호화는 이들 속성을 만족하는 사용자만이 할 수 있는 알고리즘이다. ABE 개념은 Sahai, Waters[1]에 의해 처음 소개되었으며, 이를 Goyal 등[4]에

* Corresponding author, E-mail: kafa0443@gmail.com

의해 사용자 개인키를 AND, OR로 구성된 접근구조로 표현하여 더욱 확장하였다. ABE 알고리즘에 암호 전송(Broadcast Encryption) 알고리즘 기능을 추가한 것을 속성기반 암호전송 알고리즘(Attribute-based Broadcast Encryption : 이하 ABBE)이라 한다. ABBE는 다수의 사용자에게 암호문을 효율적으로 전송할 수 있고 상황에 따라 특정 사용자를 제외시킬 수 있는 제외 기능이 포함되어 있는 ABE 알고리즘이다.

ABBE 관련 대표적인 연구는 Attrapadung, Imai 논문[2]으로 Linear Secret Sharing Schemes을 이용해서 설계했다. ABBE 알고리즘은 비밀 자료를 다양한 속성으로 암호화하고 비밀키의 속성들이 일정한 조건을 만족하면 복호화 할 수 있는 것으로 미래 우리군의 전장환경 변화에 필수적으로 개발해야하는 알고리즘이다. 우리군 장비에 ABBE 알고리즘을 적용한다면 복호화 키가 있더라도, 자료의 속성에 해당되는 속성키가 없으면 복호화 할 수 없어 보안성을 상당히 높일 수 있고 정당한 키를 가진 관계자라도 속성과 무관하다면 자료의 접근을 제어할 수 있는 장점이 있다. 특히, 우리군의 조직은 지휘본부에서 작전부대원에 이르는 계층적(hierarchy) 구조를 갖고 있고, 각 계층별 비밀 접근에 대한 한계가 정해져 있다. 예를 들어, 군에서는 계급에 따라 취급할 수 있는 비밀 등급이 정해져 있고, 근무 조직에 따라서 취급할 수 있는 비밀 종류가 구분되어 있다. 하지만 비밀 취급인가만 되어있다면 어느 비밀이든 관계없이 확인할 수 있는 고전적 시스템이다. 미래 우리군의 전장이 크게 확대된다면 인공위성을 통한 자료전송은 필수적이며 위의 고전적 자료접근 시스템은 적합하지 않다.

따라서 자료의 속성에 따른 접근을 제어할 수 있는 ABBE 알고리즘이 절대적으로 필요하다고 할 수 있다. 이를 위해 본 논문에서는 군 환경에 적합한 효율적인 ABBE 알고리즘을 제안하고자 한다.

본 논문에서는 두개의 ABBE 알고리즘을 제안한다. 두 개의 알고리즘 모두 군의 계층적 구조에 적합한 이진트리 구조를 기본으로 알고리즘을 설계하였다. 첫 번째 알고리즘은 이진트리 구조에서의 효율적 전송방법인 complete subtree[2] 방법과 사용자의 속성, 암호문의 속성 사이의 관계를 라그랑주 보간법을 사용하여 구체화하는 알고리즘이고, 두 번째 알고리즘은 라그랑주 보간법의 연산이 없이 겹선형 사상의 성질을 활용하여 속성을 제어하는 알고리즘으로 첫 번째 알고리즘의 효율성을 개선한 것이다.

2. 배경지식

2.1 겹선형 사상

먼저 G_1와 G_2는 소수 위수 p를 갖는 곱셈 순환군이다. 그리고 g는 G_1의 생성원이고, e는 겹선형 사상으로 $e : G_1 \times G_1 \rightarrow G_2$로 정의되는 다음의 성질을 가지는 함수이다.

- Bilinearity : 모든 $u, v \in G_1$과 $a, b \in Z_p^*$에 대해서 $e(u^a,\ v^b) = e(u,\ v)^{ab}$가 성립한다.
- Non-degeneracy : $e(g,\ g) \neq 1$이 성립한다.

2.2 라그랑주 보간법

차수가 z인 다항식 $f(x)$은 $z+1$개의 $(j_0, f(j_0))$, ..., $(j_z, f(j_z))$점이 주어진다면 만들 수 있는 방법으로 p를 소수라고 하고 차수가 z인 다항식 $f(z)$를 Z_p 위에서 정의된 다항식이라 하자. 그러면, 다항식 $f(x)$는 다음과 같이 구해진다.

$$f(x) = \sum_{t=0}^{z} (f(j_t) \cdot \triangle_{i,S}(x))$$

여기서 $\triangle_{i,S}(x) = \prod_{\substack{j \neq i \\ j \in S}}^{z} \frac{x-j}{i-j}$은 라그랑주 계수이며, $S = \{j_0, \cdots, j_z\}$ 이다.

2.3 ABBE 알고리즘 구조(Structure)

ABBE 알고리즘은 초기화, 키 생성, 암호화, 복호화의 구조로 구성되어 있으며, 세부 구조는 다음과 같다.

2.3.1 초기화(Set-up)

보안 파라미터를 입력받아 시스템 속성집합 *W*, 공개키 *PK*와 마스터키 *MK*를 생성한다.

2.3.2 키 생성(Key-Gen)

시스템 속성집합 *W*, 공개키 *PK*, 마스터키 *MK*를 입력받아 사용자의 속성집합 w, 개인키 *SK*를 생성한다.

2.3.3 암호화(Enc)

수신자의 속성집합 w', 메시지 *M*, 수신자 집합, 공개키 *PK*를 입력받아 암호문 *CT*을 생성한다.

2.3.4 복호화(Dec)

암호문 CT를 입력받고 사용자와 수신자의 속성이 일정 관계를 만족하면 메시지 M을 출력한다.

3. 제안 ABBE 알고리즘

본 논문에서는 두 개의 효율적인 ABBE 알고리즘을 제안한다. 첫 번째 알고리즘과 두 번째 알고리즘의 큰 차이는 효율성(size)과 속성집합이 유계인 것과 아닌 것이다. 먼저 첫 번째 알고리즘은 다음과 같다.

3.1 알고리즘 I

3.1.1 초기화(Set-up)

시스템 매니저는 총 사용자 n명의 집합 $U = \{u_1, u_2, ..., u_n\}$에 대해서 $n = 2^d$인 이진트리를 만든다. 루트 노드에는 $\varnothing \in \{0,1\}^0$ 라벨을 대응시키고, 각각의 내부 노드에는 트리의 깊이(depth)에 따라 왼쪽 자식노드에는 $s \in \{0,1\}^j$에 대하여 $s\|0 \in \{0,1\}^{j+1}$ 라벨을 대응시키고, 오른쪽 자식노드에는 $s\|1 \in \{0,1\}^{j+1}$ 라벨을 대응시킨다. 각각의 사용자는 잎 노드에 $s \in \{0,1\}^d$의 라벨을 대응시킨다. 그리고 사용자마다 l 개 이상의 속성을 갖도록 구성한다. 그리고 모든 노드에는 노드키 $L_s \in Z_p$를 임의로 선택하여 부여한다. 예를 들면 $n = 2^2$이라면 Fig. 1과 같다. 또한 시스템 매니저는 암호화에 필요한 공개키 PK와 마스터키 MK를 다음과 같이 생성하여 공개한다.

$$PK = \left[\{H(i)\}_{i \in W}, \ \left\{g^{L_s}\right\}_{s \in \{0,1\}^d}, \ \ e(g, g)^{\alpha} \right]$$
$$MK = \{\alpha\}, \ \alpha \in Z_p$$

여기서 $|g^{L_s}| = 2n-1$이고 $W = \{w_1, w_2, ..., w_l, ...\}$는 속성집합이며 $|W| \geq l$이다. $H : \{0,1\}^* \rightarrow G_1$이다.

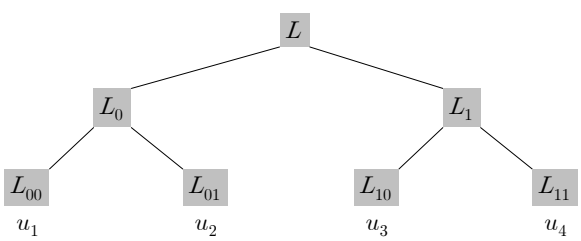

Fig. 1. $n = 2^2$ binary tree and node key L_s, $s \in \{0,1\}^2$

3.1.2 키 생성(Key-Gen)

시스템 매니저는 사용자들이 속해있는 잎 노드에서 루트 노드까지 상위노드 키를 랜덤하게 만들고 임의의 $r \in Z_p$에서 선택하여 다음과 같이 사용자 u의 개인키 SK_u를 만든다.

$$SK_u = \left[w, \left\{ D_{1,i} = g^{q(i)/L_s} H(i)^{r/L_s} \right\}_{i \in w}, \ D_2 = g^r \right]$$

여기서, $|D_{1,i}| = (\log n + 1) \cdot |w|$이고 $q(x)$는 Z_p상의 $l-1$차 다항식으로 $q(0) = \alpha$을 만족한다. w는 사용자 속성을 의미하고 $w = \{w_i, \cdots, w_j\}$, $|w| \geq l$이다. Fig. 1의 경우, 사용자 u_2는 $\left\{g^{q(i)/L}, g^{q(i)/L_0}, g^{q(i)/L_{01}}\right\}$의 노드키 성분을 갖는다.

3.1.3 암호화(Enc)

지휘본부는 보내고자 하는 암호문의 속성 w'과 전송하고자 하는 사용자 집합 $\tilde{U} = U \setminus R$에 대해서 다음과 같이 암호문을 생성하여 전송한다. R은 제외하는 사용자 집합을 의미한다.

먼저 임의의 $t \in Z_p$를 선택하고 $\tilde{U}$를 포함하는 집합의 노드키 L_s를 선택하여 다음을 생성한다.

$$CT = \left[\begin{array}{l} w', \left\{C_{1,i} = H(i)^t\right\}_{i \in w'}, \left\{C_2 = g^{tL_s}\right\} \\ \\ M \cdot e(g,g)^{\alpha t} \end{array} \right]$$

$w' = \left\{w_i', \cdots, w_k'\right\} \subset W$ 이고, $|w'| \geq l$ 이다.

3.1.4 복호화(Dec)

만일 사용자 $u \in \tilde{U}$ 이라면, 먼저 $|w \cap w'| \geq l$ 조건을 만족하는 속성집합 $\tilde{W}$를 자신의 속성집합에서 선택하고 다음과 같이 복호화 한다.

$$M = M \cdot e(g,g)^{\alpha t} \cdot \frac{e\left(D_2, \prod_{i \in \tilde{W}} (C_{1,i})^{\Delta_{i,s}(0)}\right)}{e\left(C_2, \prod_{i \in \tilde{W}} (D_{1,i})^{\Delta_{i,s}(0)}\right)}$$

위식의 세부계산은 다음과 같고, 먼저 등식 오른쪽을 계산하면

$$\prod_{i\in\widetilde{W}}\left(\frac{e(g^r,\ H(i)^t)}{e(g^{tL_s},\ g^{q(i)/L_s})\cdot e(g^{tL_s},\ H(i)^{r/L_s})}\right)^{\Delta_{i,S}(0)}$$
$$=\prod_{i\in\widetilde{W}}\frac{1}{e(g,\ g)^{tq(i)\Delta_{i,S}(0)}}=\frac{1}{e(g,\ g)^{\alpha t}}$$

이므로 M을 구할 수 있다. 마지막 등식은 라그랑주 보간법을 사용하여 α를 계산한 것이다.

만일 사용자 $u\not\in\widetilde{U}$이라면, u는 개인키 SK_u 속에 암호문의 $C_2 = g^{tL_s}$에 해당되는 노드키 성분이 없기 때문에 복호화를 할 수 없다.

제안하는 알고리즘의 장점은 시스템에서 제외하고자 하는 사용자 집합 R과 사용자들이 보유하고 있는 속성 w에 따라서 복호화를 제어할 수 있다는 것이다. 즉 사용자 $u\in\widetilde{U}$하더라도 암호문의 w' 조건과 $|w\cap w'|\geq l$을 만족하지 못하면 복호화를 할 수 없는 알고리즘이다. 제안하는 알고리즘의 효율성은 다음과 같다. 복호화에 필요한 연산량은 l번의 지수승 연산과 2번의 겹선형 사상 연산이 필요하다.

Table 1. The size of principal parameters of algorithm I

	공개키	비밀키	암호문
제안 알고리즘	$O(n+l)$	$O(\log n\ \cdot l)$	$O(r\log n/r+l)$

여기서 r은 제외되는 사용자 수이다. $|R|=r$

암호문의 크기는 제외되지 않는 사용자를 포함한 집합의 크기 $r\log n/r$과 속성의 개수 l을 포함한 크기이다. 본 알고리즘의 단점은 사용자의 개인키가 해킹등으로 노출되었을 때 해당 사용자의 키를 추적하는 것이 불가능하다는 것이다. 즉, 지휘본부가 잎 노드키를 제외한 내부 노드키들로 암호문을 생성하였을 경우, 내부 노드키를 공통으로 가지고 있는 사용자가 있으므로 어느 사용자의 키가 노출되었는지 알 수 없다는 것이다. 이를 개선하고자 사용자 정보 ID를 개인키에 추가하여 개인키가 노출 또는 악의적인 사용자들 사이에 개인키를 공유하여도 복호화 할 수 없는 알고리즘을 다음에 제안하고자 한다.

3.2 알고리즘 II

3.2.1 초기화(Set-up)

시스템 매니저는 총 사용자 n명의 집합 $U=\{u_1, u_2, \ldots, u_n\}$에 대해서 알고리즘 I과 같이 $n=2^d$인 이진트리를 만든다. 또한 시스템 매니저는 암호화에 필요한 공개키 PK와 마스터 키 MK를 다음과 같이 생성하여 공개한다.

$$PK=\left[g^{\alpha}, W=\{w_1,\cdots,w_l\}, \left\{e(g,\ g)^{\alpha L_s}\right\}\right]$$
$$MK=\{\alpha\},\ \alpha\in Z_p$$

여기서 W는 속성집합이고 속성 $w_i\in G_1$, $|W|=l$이다.

3.2.2 키 생성(Key-Gen)

시스템 매니저는 사용자들이 속해있는 잎 노드에서 루트 노드까지 상위노드 키를 랜덤하게 만들고 다음과 같이 사용자 u_i의 개인키 SK_i를 만든다.

$$SK_i=\left[w=\left\{w_i^{ID_i},\cdots,w_j^{ID_i}\right\},\ \left\{D_{1,i}=g^{L_s}w^{ID_i}\right\},\ D_2=g^{ID_i}\right]$$

여기서 $|L_s|=\log n+1$이므로 $|D_{1,i}|=\log n+1$이다.

3.2.3 암호화(Enc)

지휘본부는 보내고자 하는 암호문의 속성 w'과 전송하고자 하는 사용자 집합 $\widetilde{U}=U\setminus R$에 대해서 다음과 같이 암호문을 생성하여 전송한다.

먼저 임의의 $t\in Z_p$를 선택하고 $\widetilde{U}$를 포함하는 노드키 L_s를 선택하여 다음 암호문을 생성하여 전송한다.

$$CT=$$
$$\left[w',\ C_1=g^{\alpha t},\ C_2=(w\,w_i\cdots w_j)^{\alpha t},\left\{M\cdot e(g,g)^{\alpha t L_s}\right\}\right]$$

3.2.4 복호화(Dec)

만일 사용자 $u\in\widetilde{U}$ 이라면, 먼저 암호문 조건을 만족하는 속성집합을 선택하고 개인키를 이용해 $\widetilde{D_{1,i}}=g^{L_s}w^{ID_i}\prod_{i\in w'}(w_i^{ID_i}\cdots w_j^{ID_i})$를 계산한다. 그리고 다음과 같이 복호화 한다.

$$M=M\cdot e(g,g)^{\alpha t L_s}\cdot\frac{e(D_2,\ C_2)}{e(\widetilde{D_{1,i}},\ C_1)}$$
$$=M\cdot e(g,g)^{\alpha t L_s}\cdot\frac{e\left((ww_1\cdots w_x)^{\alpha t},\ g^{ID_i}\right)}{e\left(g^{L_s}(w\,w_1\cdots w_x)^{ID_i},\ g^{\alpha t}\right)}$$

위식의 세부계산은 다음과 같고, 먼저 등식 오른쪽을 계산하면

$$\frac{e\left((ww_1\cdots w_x)^{\alpha t},\ g^{ID_i}\right)}{e\left(g^{L_s},\ g^{\alpha t}\right)\cdot e((ww_1\cdots w_x)^{ID_i},\ g^{\alpha t})}$$

$$= \frac{e\left((ww_1\cdots w_x),\ g\right)^{\alpha t\, ID_i}}{e(g,\ g)^{L_s\alpha t}\cdot e((ww_1\cdots w_x),\ g)^{ID_i\alpha t}}$$

$$= \frac{1}{e(g,\ g)^{L_s\alpha t}}$$

이므로 M을 구할 수 있다.

여기서 만일 사용자 $u \notin \tilde{U}$이라면, u는 개인키 SK에 암호문의 $M\cdot e(g,\ g)^{\alpha t L_s}$에서 L_s에 해당되는 노드키를 가지고 있지 않기 때문에 복호화를 할 수 없다. 또한 사용자 속성과 암호문 속성이 $w \not\supseteq w'$이라면 복호화 속성에 해당되는 속성이 없으므로 복호화 할 수 없는 알고리즘이다. 알고리즘 Ⅰ과 마찬가지로 복호화 연산량은 같다.

Table 2. The size of principal parameters of algorithm Ⅱ

	공개키	비밀키	암호문
제안 알고리즘	$O(n+l)$	$O(\log n+l)$	$O(r\log n/r)$

여기서 r은 제외되는 사용자 수이다. $|R| = r$

알고리즘 Ⅱ의 핵심은 개인키의 $g^{L_s}w^{ID_i}$에서 w^{ID_i}를 분리할 수 없다는 것이다. 만일 분리된다면 $e(g^{L_s},\ g^{\alpha t})$ = $e(g,\ g)^{L_s\alpha t}$이므로 속성과 관계없이 모든 암호문을 복호화할 수 있으므로 알고리즘이 안전하지 않다고 할 수 있다.

3.2.5 효율성

본 논문에서 제안하는 알고리즘과 기존에 제시된 대표적인 논문의 효율성(size)은 다음과 같다.

제안하는 알고리즘과 ABBE의 대표적인 논문 [2]을 비교하면 비밀키의 크기가 매우 적다는 것을 알 수 있으며, 암호문의 크기 또한 큰 차이를 보이지 않는다는 것을 알 수 있다.

Table 3. The comparison of efficiency

	공개키	비밀키	암호문
[3]	$O(n+l)$	$O(n+l)$	$O(r+l)$
제안 Ⅰ	$O(n+l)$	$O(\log n\cdot l)$	$O(r\log n/r+l)$
제안 Ⅱ	$O(n+l)$	$O(\log n+l)$	$O(r\log n/r)$

(l : 총 속성 개수, n : 총 사용자 수,
r : 제외되는 사용자 수)

알고리즘 Ⅱ를 사용하여 우리군의 병력을 고려하여 제안하는 알고리즘을 사용하여 개인키를 저장할 인원은 대략 13만명($\approx 2^{17}$)이며, 속성집합을 최대 32가지(2^5)으로 고려한다면, 공개키의 크기는 16Mbyte이고 개인 비밀키는 대략 8kbyte, 암호문의 크기는 r = 1로 가정한다면 8kbyte 이므로 알고리즘을 활용하는데 있어서 쉽게 시스템 구현이 가능하다.

3.2.6 안전성

본 논문의 알고리즘은 다음 가정의 안전성에 기초를 둔다.

- Decisional Bilinear Diffie-Hellman Assumption : Decisional BDH

  임의의 $a,b,c,z \in Z_p$에 대하여 $(g^a,g^b,g^c,e(g,g)^{abc})$와 $(g^a,g^b,g^c,e(g,g)^z)$를 다항식 시간에 무시하지 못할 확률로 구별할 수 있는 공격자는 없다는 가정.

- Decisional Modified Bilinear Diffie-Hellman Assumption : Decisional MBDH

  임의의 $a,b,c,z \in Z_p$에 대하여 $(g^a,g^b,g^c,e(g,g)^{ab/c})$와 $(g^a,g^b,g^c,e(g,g)^z)$를 다항식 시간에 무시하지 못할 확률로 구별할 수 있는 공격자는 없다는 가정.

본 논문에서 제안한 알고리즘의 안전성은 위의 가정에 기초를 두며, 다음 정리로서 안전성을 증명할 수 있다.

정리 : 만일 공격자가 알고리즘 Ⅰ(Ⅱ)의 안전성을 깰 수 있다면, 시뮬레이터는 Decisional MBDH(Decisinal BDH) 가정을 무시하지 못할 확률로 깰 수 있는 모델을 만들 수 있다.

4. 결 론

본 논문에서는 효율적인 속성 기반의 접근제어 암호 전송 알고리즘(ABBE)을 제안하였다. 본 논문은 이진 트리를 기반으로 설계하여 군의 계층적인 구조에 적합하며, 효율적인 전송 및 제외기능을 포함하고 있어 필요시 불필요한 인원의 자료에 대한 접근을 차단할 수 있고, 자료의 속성에 따라 복호화를 제어할 수 있는 알고리즘이다. 제안하는 알고리즘은 기존의 대표적인 논문보다 비밀키의 크기가 사용자 수에 비해 크게 개선한 논문이며, 복호화에 필요한 연산량 또한 두배로 줄어 복호화가 빠르다는 장점을 가지고 있다.

군의 작전환경을 고려하면 신속한 복호화가 큰 척도로 작용할 수 있으므로 제안 알고리즘의 활용성은 높다고 할 수 있다. 기존의 연구에서 속성에 대해 AND, OR, NOT 과 같은 속성제어 방식을 사용하지만 키 속성만으로도 충분히 안전한 전송과 속성을 제어할 수 있는 알고리즘을 제안하였다.

References

[1] A. Sahai and B. Waters, “Fuzzy Identity-based Encryption,” Eurocrypt 2005, LNCS 3494, pp. 457-473.

[2] D. Naor, M. Naor and Lotspiech, “Revocation and Tracing Schemes for Stateless Receivers,” Crypto 2001, LNCS 2139, pp. 41-62.

[3] N. Attrapadung and H. Imai, “Conjunctive Broadcast and Attribute-based Encryption”, Pairing 2009, LNCS 5671, pp. 248-265.

[4] V. Goyal, O. Pandy, A. Sahai and B. Waters, “Attribute-based Encryption for Fine-grained Access Control of Encrypted Data,” ACM CCS 2006, pp. 89-98.