Bethencourt등의 Ciphertext Policy 속성기반 암호화에서 효율적인 속성값 철회 기법1)

전윤구, 이훈정, 오희국
한양대학교 컴퓨터공학과

imyg@naver.com, hjlee@infosec.hanyang.ac.kr, hkoh@hanyang.ac.kr

Efficient Revocation Scheme for Bethencourt's Ciphertext-Policy Attribute Based Encryption

Yun-Koo Jeon, Hoon-Jung Lee, Hee-Kuk Oh
Division of Computer Science and Engineering, Hanyang University

요 약

본 논문에서는 Bethencourt등의 CP-ABE에서 효율적인 속성값 철회 기법에 대해 알아본다. 기존에 제안된 속성값 철회 기법은 대부분 KP-ABE에 대한 것이며, CP-ABE에서 속성값 철회는 철회를 위한 메시지 크기가 철회자에 비례해 커지고 NOT연산을 필요로 한다는 측면에서 효율적이지 못하다. 이에 대해 Bethencourt등의 CP-ABE와 기존의 속성값 철회 기법에 대해 알아본 후 Bethencourt등의 CP-ABE에서 효율적인 속성값 철회 기법에 대해 제시하고자 한다.

1. 서론

속성기반 암호화는 신원기반 암호화 방식의 개념을 확장한 것으로 Sahai등이 최초로 개념을 제안했다. 속성기반 암호화는 사용자가 가지는 속성값과 암호문에 포함되는 속성값을 비교에 메시지 복호화에 충분한 속성값이 있을 경우 메시지를 복호화 할 수 있는 방법을 말한다.

속성기반 암호화는 암호 정책에 따라 Key-Policy 속성기반 암호화(KP-ABE)와 Ciphertext-Policy 속성기반 암호화(CP-ABE)로 분류된다. KP-ABE는 각각의 사용자들에게 발급되는 개인키에 Access Tree가, 암호문에는 속성값이 들어가는 방식이다. 개인키를 발급받은 사용자는 암호문에 포함된 속성값이 개인키의 Access Tree를 만족할 경우 메시지를 복호화 하게 된다[1]. CP-ABE는 사용자들의 개인키에 각각의 사용자들이 가지는 속성값이, 암호문에는 Access Tree가 포함된다. 사용자들은 자신들이 가지는 속성값들이 암호문의 Access Tree를 만족할 경우 메시지를 복호화 할 수 있게 된다[2].

속성기반 암호화는 사용자가 서비스 탈퇴등으로 자신이 가지는 속성값을 철회해야 하는 경우 속성값을 철회한 사용자가 더 이상 서비스를 받을 수 없도록 하는 방법이 필요하다. 기존에 제안된 속성값 철회 방법들은 Access Tree에 속성값 철회자의 목록을 포함하는 방법등이 제안되었다. 속성값 철회자 목록을 사용하는 방법의 경우 Access Tree에서 NOT을 지원하기 위한 추가적인 연산과 철회자가 계속 늘어날 경우 철회자 목록이 비례해 증가, 철회를 위한 메시지가 커지는 문제를 가진다.

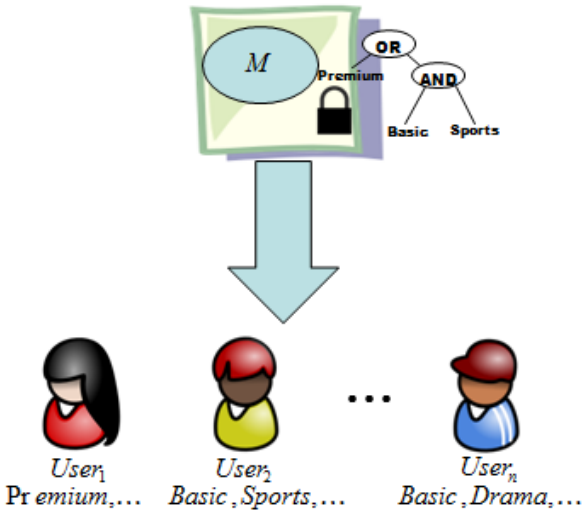
본 논문에서는 CP-ABE에서 효율적인 속성값 철회 기법을 제안한다. 2장에서 Bethencourt등의 CP-ABE에 대해 알아본 후 3장에서 기존의 속성값 철회 기법에 대해 설명하며 4장에서 제안하는 기법에 대해 설명한 후 5장에서 결론을 통해 논문을 마무리하고 향후 연구 문제에 대해 제시하고자 한다.

2. Bethencourt등의 CP-ABE

Bethencourt등이 제안한 CP-ABE[2]는 key-policy 속성기반 암호화에서 암호자가 암호문에 대한 사용자의 접근 제어 권한을 가지지 못하는 문제를 해결한다. CP-ABE의 암호문에는 복호화에 필요한 속성값에 대한 tree-access구조와 암호화된 데이터로 구성되며, 사용의 개인키는 사용자들이 가지는 속성에 대해 키 발급자가 발급한 키로 구성 된다. 사용자가 가지는 속성값이 암호문에 포함된 tree-access구조에 맞을 경우 암호문을 복호화 할 수 있는 것이다. Bethencourt등의 CP-ABE의 모습은 (그림 1)에서 나타내고 있다. (그림 1)에서 각각의 사용자들은 'Premium', 'Basic', 'Drama'등의 속성을 가지고 있으며 메시지는 $Premium \vee (Basic \wedge Sports)$의 Access Tree로 암호화 돼있다. 이때 Use_1은 'Premium'속성을 가

1) "본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음"(NIPA-2010-C1090-1011 - 0010).

1) "이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임"(No. 2010-0000438).



(그림 1) Bethencourt등의 CP-ABE

지고 있어 Access Tree를 만족하며 메시지를 복호화 할 수 있다. 또한 $User_2$도 Access Tree를 만족해 메시지를 복호화 할 수 있다. 하지만 $User_n$은 Access Tree를 만족하지 못해 메시지를 복호화 하지 못한다.

Bethencourt등의 CP-ABE는 기능적으로 Setup, Key Generation, Encryption, Decryption의 4가지의 과정으로 분류할 수 있다.

Setup에서는 아무런 입력이 필요 없으며 공개키 PK와 마스터 키 MK를 생성한다. PK는 데이터를 암호화 하는 과정에서 사용되며, MK는 사용자 개인키를 생성하는데 사용된다.

Key Generation(MK, S)은 MK와 사용자 각자가 가지는 속성값의 집합 S를 입력으로 가지며, 사용자의 속성에 맞는 개인키 SK를 생성한다.

Encryption(PK, M, A)은 PK와 암호화 하고자 하는 메시지 M, 암호문에 대한 접근 제한 구조인 A를 입력으로 가지며, A에 기반한 암호문 CT를 생성한다.

Decryption(PK, CT, SK)은 PK와 CT, SK를 입력으로 가지며 CT가 참조하는 A와 SK를 비교해 CT문을 복호화 해 메시지 M을 구한다.

3. 기존의 속성값 철회 기법

속성기반 암호화에서 속성값 철회를 위한 기법은 대부분 KP-ABE에서 속성값 철회자의 목록을 사용한 기법들이다. CP-ABE에서 속성값 철회 기법은 속성값에 유효기간을 두는 방법등이 있다. 기존에 제안된 철회 기법들은 다음과 같다.

3.1 Bethencourt등의 기법

Bethencourt등[2]은 CP-ABE기법을 제안 하면서 속성값 철회의 방법에 대해 언급 하였다. 이들의 속성값 철회 기법은 개인키와 암호문의 속성값에 유효기간을 두는 방법이다. 사용자 각각의 개인키에 포함된 속성값의 유효기간과 암호문 속성값의 시간값을 비교해 유효기간이 지났을 경우 사용자는 더 이상 속성값을 사용하지 못하도록 하는 방식이다. 이 방법은 속성값의 유효기간을 갱신하는 기법이 제안되지 않았으며, 유효기간 중 속성값을 철회하는 기법 역시 고려되지 않고 있다.

3.2 Ostrovsky등의 기법

Ostrovsky등[3]은 KP-ABE에서 NOT연산을 지원하는 기법을 제안하였다. NOT연산은 속성값 철회 리스트를 사용하는 방법에서 핵심적인 역할을 하는 요소이다. NOT 연산을 지원하는데 있어 고려해야할 부분은 암호문에 사용되는 속성값 이외의 임의의값을 넣어도 NOT연산을 통과할 수 있냐는 것이다. Ostrovsky등은 암호문에 포함되는 속성값의 수가 항상 d라고 가정해 d차 다항식을 생성한다. NOT속성을 가지고 있지 않은 사용자는 d+1개의 다항식에 대한 값을 얻어 보간법을 통해 비밀값을 구할 수 있으나, NOT속성을 가진 사용자는 d개의 값만 얻어 비밀값을 구하지 못하는 방식이다. 암호문의 속성값을 d로 고정시키는 방법으로 그 활용성이 크기 못하다.

3.3 Attrapadung등의 기법

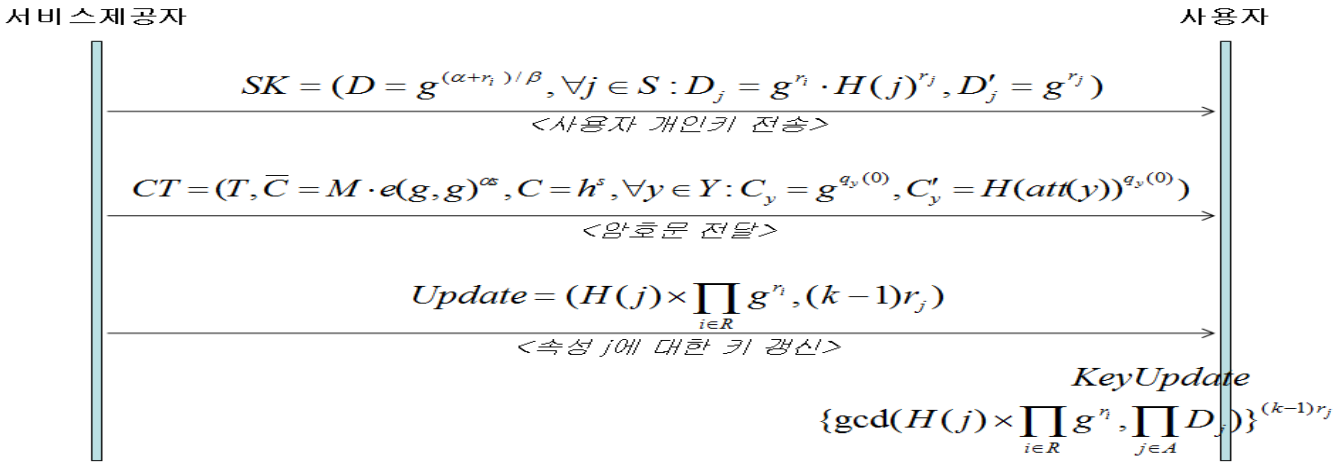
Attrapadung등[4]은 KP-ABE에서 Direct Revocation 기법과 Indirect Revocation기법을 제안했다. Direct Revocation은 서비스 철회자 목록을 사용해 직접적으로 키를 갱신하는 기법이다. Indirect Revocation은 인증서버를 두고 사용자들을 트리로 구성해 철회를 한 사용자외의 다른 사용자들의 키를 갱신해 주는 방법이다. 이 방법을 사용할 경우 사용자를 관리하기 위한 트리를 구성해 비밀값을 공유하는 방법이 요구된다.

4. 제안하는 기법

지금까지 속성기반 암호화에서 속성값 철회와 관련한 논문은 대부분 KP-ABE를 위한 것이다. 또한 철회리스트 사용을 위한 NOT연산이 효율적이지 못하다. 본 장에서는 Bethencourt등의 CP-ABE에서의 효율적인 속성값 철회 기법을 제안한다. 제안하는 기법은 속성값 자체에 대한 철회를 지원하며 전체적인 모습은 (그림 2)와 같다.

<표 1> 표기법

표기법	내 용
SK	개인키
CT	암호문
S	사용자 속성값 집합
j	속성값 $j \in S$
i	사용자 i
r_j	속성값 j에 대한 임의의 값 $r_j \in Z_p$
r_i	사용자 i에 대한 임의의 값 $r_i \in Z_p$
Y	Access Tree leaf노드 집합
y	Access Tree leaf노드
att(y)	y노드의 속성값
H(j)	해시 함수 $H: \{0,1\}^* \rightarrow G_o$
k	키 갱신을 위한 임의의 값



(그림 2) 제안하는 속성값 철회 기법

[정의] 논문에서 사용한 표기법은 <표 1>에서 나타내고 있으며 제안하는 방법에서 사용하는 bilinear map과 해시 함수의 정의는 다음과 같다. G_0는 위수가 p, 생성자가 g인 bilinear group 이며 e는 $e:G_0\times G_0\rightarrow G_1$인 bilinear map이다. 해시 함수 $H:\{0,1\}^*\rightarrow G_o$ 을 만족한다.

[사용자 개인키 전송] 이 과정은 Bethencourt등의 CP-ABE에서의 과정과 동일하다. 서비스 제공자는 각각의 사용자가 가지는 속성값에 대한 키를 사용에게 전송한다. 사용자가 가지는 속성값과 관련한 값은 D_j와 D_j'에서 포함하고 있으며 α는 사용자 각각마다 다르게 주어지는 임의의 값이다.

[암호문 전송] 이 과정역시 Bethencourt등의 CP-ABE에서의 과정과 동일하다. 서비스 제공자는 암호화된 메시지와 leaf노드가 속성값으로 구성된 Access Tree를 사용자에게 전송한다. Access Tree의 leaf노드가 가지는 속성값 정보는 C_y와 C_y'에서 포함하고 있다.

[속성키 갱신 메시지 전송] 키 갱신 과정에서는 암호문과 사용자 개인키에서 공통적으로 가지는 속성값 정보 $H(j)$를 이용한다. 키 갱신을 위한 메시지는 $(H(j)\times\prod_{i=R}g^{r_i},(k-1)r_j)$이다. $H(j)$는 갱신의 대상이 되는 속성값 j의 해시값, $\prod_{i=R}g^{r_i}$는 속성값 j를 철회한 사용자의 목록, $(k-1)r_j$는 사용자가 속성값 j의 키를 갱신하기 위해 필요한 값이다.

[속성키 갱신] 키 갱신 메시지를 받은 사용자는 자신의 개인키에서 속성값 정보를 포함하는 D_j와 키 갱신 메시지를 이용해 다음의 과정을 진행한다.

$$KeyUpdate=\gcd(H(j)\times\prod_{i\in R}g^{r_i},\prod_{i\in A}D_j)^{(k-1)r_j}$$

KeyUpdate과정 진행한 후 철회자는 $\prod_{i=R}g^{r_i}$에 포함돼 $(g^{r_i}\cdot H(j))^{(k-1)r_j}=(g^{r_i+n})^{(k-1)r_j}$의 갱신값을 가지게 되며, 비철회자는 $(H(j))^{(k-1)r_j}=(g^n)^{(k-1)r_j}$의 갱신값을 가지게 된다. 비철회자는 계산된 갱신값을 통해 D_j값을 갱신한다. $newD_j=(g^n)^{(k-1)r_j}\times D_j=g^{r_i}\times g^{nkr_j}$

[메시지 복호화] 갱신된 속성값을 이용한 암호문은 기존의 암호문의 C'에서 $C_y'=H(att(y))^{k\cdot q_y(0)}$으로 변경된다. 비철회자의 경우 $newD_j$를 이용해 메시지를 복호화 할 수 있으나 철회자의 경우 $newD_j$를 구하지 못해 메시지를 복호화 하지 못한다. 비철회자가 Access Tree의 leaf노드를 복호화 하는 과정은 다음과 같다.

$$\begin{aligned}DecyptNode(CT,SK,y)&=\frac{e(D_j,C_y)}{e(D_j',C_y')}\\&=\frac{e(g^{r_i}\cdot g^{nkr_j},g^{q_y(0)})}{e(g^{r_j},g^{knq_y(0)})}\\&=\frac{e(g,g)^{r_iq_y(0)+nkr_jq_y(0)}}{e(g,g)^{r_jknq_y(0)}}\\&=e(g,g)^{r_iq_y(0)}\end{aligned}$$

메시지를 복호화 하는 이후의 과정은 Bethencourt등의 CP-ABE에서 Access Tree의 internal노드에서의 복호화 과정과 동일하다.

5. 결론

본 논문에서는 Bethencourt등의 CP-ABE에서의 속성값 철회기법을 제안했다. 제안하는 방법은 사용자에게 발급되는 개인키와 암호문이 공통적으로 가지는 속성값 정보를 이용해 속성값 자체에 대한 갱신을 한다. 철회자는 올바른 갱신값을 얻지 못해 갱신에 실패하며 비철회자는 올바른 갱신값을 얻어 속성값 갱신에 성공하게 된다. 제안하는 방법을 Bethencourt등이 제안한 CP-ABE에 적용할 경우 access tree를 재구성 할 필요가 없어 효율적으로 속성값 철회 기능 제공이 가능해진다. 향후 연구해야할 방향은 속성값에 NOT을 사용하는 방법과의 정량적인 연산량 분석과 속성값 갱신을 좀 더 효율적으로 하기 위해 여러

속성값에 대한 갱신을 한 번에 진행하는 방법에 대한 연구가 필요하다.

참고문헌

[1] A.Sahai, B.Waters, "Fuzzy Identity-Based Encryption," Lecture Notes in Computer Science, 2005.

[2] J.Bethencourt, A.Sahai, and B.Waters, "Cipher text-Policy Attribute-Based Encryption," IEEE Symposium on Security and Privacy, 2007.

[3] R.Ostrovsky, A.Shai, and B.Waters, "Attrobute-Based Encryption with Non-Monotonic Access Structures," 14th ACM conference on Computer and communications security, 2007.

[4] N.Attrapadung, H.Imai, "Attribute-Based Encryption Supporting Direct/Indirect Revocation Modes," LNCS Cryptography and Coding, 2009.