Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.6, Dec. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.6.1281

사용자 폐기를 지원하는 책임 기관 ID 기반 암호*

최 수 리,† 이 동 훈‡
고려대학교 정보보호대학원

Accountable Authority Revocable Identity-Based Encryption*

Suri Choi,† Dong Hoon Lee‡
Graduate School of Information Security, Korea University

요 약

2001년 Boneh와 Franklin이 제안한 ID 기반 암호는 기존 공개키 기반 구조(PKI)와 달리 사용자의 ID를 공개키로 사용하기 때문에 공개키 검증을 위한 인증서가 필요하지 않다. 하지만 ID 기반 암호는 키 생성 기관(PKG)이 사용자의 비밀키를 직접 발급하기 때문에 키 위탁 문제가 발생한다. 또한, 한 번 발급받은 비밀키는 유효성이 지속되기 때문에 키 유출 등으로 인한 비밀키 폐기를 효율적으로 진행하기 어려운 문제가 있다.

본 논문에서는 키 위탁 문제를 완화하는 책임 기관 ID 기반 암호(A-IBE)와 사용자 폐기를 지원하는 ID 기반 암호(RIBE)를 기반으로 두 가지 문제를 모두 해결하는 사용자 폐기를 지원하는 책임 기관 ID 기반 암호(A-RIBE)를 제안한다. 또한 A-RIBE에 적합한 안전성 모델을 새롭게 정의하고, 기반하는 A-IBE와 RIBE에 따른 A-RIBE의 설계원리와 그 장·단점을 분석한다.

ABSTRACT

In 2001, Boneh and Franklin proposed Identity-Based Encryption(IBE) that does not require a certificate like Public Key Infrastructure(PKI) by using user's Identity as a public key. However, IBE has a key escrow problem because the Private Key Generator(PKG), who is a trusted authority, generates a secret key of every user. Also, it does not support efficient revocation when the user's secret key is exposed or the system needs to revoke the user. Therefore, in order to use IBE as PKI that currently used, it is necessary to solve the key escrow problem and the revocation problem.

In this paper, to solve those two problems, we suggest Accountable Authority Revocable IBE(A-RIBE) based on Accountable Authority IBE that mitigates the key escrow problem and Revocable IBE that solves the revocation problem. Also, we define the security model suitable foe A-RIBE, and analyze the principle of designing A-RIBE according to based A-IBE and RIBE and their advantage and disadvantage.

Keywords: Identity-Based Encryption, Revocable IBE, Accountable Authority IBE, PKI

I. 서 론

1.1 개요

공개키 기반 구조(Public Key Infrastructure, PKI)에서 송신자는 공개되어 있는 수신자의 공개키를 사용하여 평문을 암호화 해 수신자에게 전송하며 수신자는 자신의 개인키를 사용하여 암호문을 복호화 해 평문을 확인한다. 이때, 수신자의 공개키는 단순

Received(09. 12. 2017), Modified(11. 30. 2017), Accepted(12. 04. 2017)
* 본 논문은 2017년도 한국정보보호학회 하계학술대회에 발표한 우수논문을 개선 및 확장한 것임
* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2015-0-00320, 계층적 식별자를 가진 인터넷 개체의 공개키 인증 구조 연구)
† 주저자, sol2-@naver.com
‡ 교신저자, donghlee@korea.ac.kr(Corresponding author)

한 난수 형태이기 때문에 송신자는 자신이 암호화에 사용하는 공개키가 수신자의 것인지 확인할 수 있는 방법이 필요하다. 만약 이러한 방법을 사용하지 않는다면 공격자가 수신자의 공개키를 자신의 공개키로 위장하더라도 송신자는 구분할 수 없고 송신자가 보내는 암호문의 내용을 공격자가 모두 복호화 해 볼 수 있게 된다. 따라서 공개키 기반 구조에서는 수신자의 공개키와 수신자를 연결하는 인증서를 사용한다. 하지만 공개키 기반 구조의 인증서는 안전한 사용을 위해 저장과 폐기, 분배 등의 관리 문제가 발생한다. 이러한 인증서 문제를 해결하기 위하여 ID 기반 암호(Identity-Based Encryption, IBE)가 제안되었다.

1984년 Shamir는 공개키 기반 구조의 인증서 문제를 해결하기 위해 ID 기반 암호 시스템을 제안하였고[18] 2001년 Boneh와 Franklin은 처음으로 ID 기반 암호 기법을 설계하였다[3]. ID 기반 암호 기법은 공개키 기반 구조에서 난수이던 수신자의 공개키를 메일 주소, 전화 번호, IP 주소 등의 식별 가능한 문자로 대체하여 인증서 없이도 공개키와 수신자를 연결 가능하다. ID 기반 암호 시스템은 신뢰 기관인 PKG(Private Key Generator)가 일괄적으로 생성한 공개 파라미터와 각 수신자의 ID가 공개키로 사용되며 따라서 이때 수신자의 비밀키는 PKG가 마스터키와 수신자의 ID를 사용하여 일방적으로 생성한 뒤 안전한 채널을 통해 수신자에게 전달한다. 그러나 PKG가 시스템상의 모든 사용자의 비밀키를 발급해 주기 때문에 PKG가 모든 사용자의 비밀키를 알게 되는 키 위탁(key escrow) 문제가 발생한다. 또한 공개키 기반 구조에서 인증서를 폐기하고 갱신하는 것처럼 사용자에게 발급된 비밀키가 노출되어 비밀키를 갱신해야 하거나 사용자를 시스템에서 폐기해야 할 경우 폐기 문제(revocation problem)가 발생하게 된다. 따라서 ID 기반 암호 시스템을 현재 사용되고 있는 공개키 기반 구조처럼 사용하기 위해서는 키 위탁 문제와 사용자 폐기 문제를 해결하는 것이 중요하다. 책임 기관 ID 기반 암호 시스템(Accountable Authority Identity -Based Encryption, A-IBE)[6]과 사용자 폐기를 지원하는 ID 기반 암호 시스템(Revocable Identity-Based Encryption, RIBE)[1]은 각각 키 위탁 문제와 폐기 문제를 해결한 대표적인 기술이다.

A-IBE는 PKG가 일방적으로 비밀키를 생성하는 ID 기반 암호 시스템과 다르게 PKG와 사용자가 모두 참여하는 프로토콜을 통해 비밀키를 생성하며 PKG는 사용자가 최종적으로 어떤 비밀키를 발급받았는지 알 수 없다. PKG는 마스터키를 알고 있기 때문에 정당한 비밀키를 생성할 수 있지만 사용자가 선택한 난수를 모르기 때문에 사용자의 비밀키와 동일한 비밀키를 생성하는 것은 불가능 하다. 또한 비밀키의 생성자를 추적할 수 있는 알고리즘이 존재하여 악의적인 의도의 PKG가 마스터키로 비밀키를 생성하여 배포하였을 경우 배포된 비밀키의 생성자를 추적하여 PKG를 고소할 수 있다.

RIBE에서는 PKG가 비밀키를 생성하여 사용자에게 전달한 후 폐기되지 않은 사용자에게만 업데이트키를 발급한다. 사용자는 발급받은 비밀키와 업데이트키를 결합해 복호화키를 생성하고 복호화키를 사용하여 암호문을 복호화한다. PKG는 업데이트키의 유효기간을 설정하여 시간에 따른 업데이트키를 발급하며 폐기된 사용자에게는 업데이트키를 발급하지 않는 방법으로 사용자 폐기를 제공한다.

본 논문에서는 ID 기반 암호의 두 가지 문제점을 함께 해결하여 실제 사용하고 있는 공개키 기반 구조처럼 사용 가능한 사용자 폐기를 지원하는 책임 기관 ID 기반 암호(Accountable Authority Revocable Identity-Based Encryption, A-RIBE)를 제안한다.

1.2 기여도

본 논문에서는 ID 기반 암호를 실제 환경에 적용하기 위해 복호화키 추적을 제공하며 사용자 폐기가 가능한 새로운 프리미티브인 A-RIBE를 제안한다. 제안하는 프리미티브는 ID 기반 암호의 주요 문제인 키 위탁 문제와 사용자, 비밀키 폐기 문제를 각각 A-IBE, RIBE를 적용하여 해결하였다.

A-RIBE는 암호 기법의 안전성, 추적 알고리즘의 악의적인 사용자와 악의적인 PKG에 대한 안전성을 모두 증명해야 한다. 본 논문에서는 A-RIBE의 적합한 안전성 모델을 제시하고 설계한 기법의 안전성을 증명한다. 암호 기법은 IND-RID-CPA 모델에서 기반 RIBE 기법의 안전성에 기반하여 증명하며, 악의적인 PKG에 대한 안전성은 약한 Black box 모델에서, 악의적인 사용자에 대한 안전성은 adaptive-ID 모델에서 기반 A-IBE의 안전성에 기반하여 증명한다.

A-RIBE는 기반하는 A-IBE의 특징과 RIBE의 특징에 따라 다양하게 결합하여 기법을 생성하는 것이 가능하다. 본 논문에서는 A-RIBE를 설계하는 원리를 소개하고, 적합한 환경에 사용할 수 있도록 설계 원리에 따른 다양한 A-RIBE를 비교·분석한다.

1.3 관련연구

Accountable Authority IBE IBE의 키 위탁 문제를 완화하기 위해 Goyal은 처음으로 A-IBE의 개념을 제시하고 Gentry IBE[4]와 Waters IBE[19]에 기반한 두 개의 기법을 제시하였다[6]. Gentry IBE에 기반한 첫 번째 기법은 White box 모델에서 추적이 가능하며 Waters IBE에 기반한 두 번째 기법은 약한 Black box 모델에서 추적이 가능한 기법이다. Goyal 등은 [6]의 연구에 이어 Black box 모델에서의 완전 안전성을 증명 가능한 A-IBE 기법을 제안하였다[7]. 하지만 비밀키의 계산량과 암호문 크기가 그룹 원소의 보안 상수에 비례하여 증가하며 악의적인 사용자에 대한 안전성이 selective 모델에서 증명되었다. Libert와 Vergnaud는 Goyal의 첫 번째 기법을 보완하여 일정한 암호문 크기를 제공하는 A-IBE 기법을 제안하였다[11]. Lai 등은 비밀키의 분실과 사용자의 비협조 등을 문제로 들며 처음으로 공개 추적성을 제공하는 A-IBE 기법을 제안하였다[10]. 또한 Kiayias와 Tang은 IBE 기법을 A-IBE 기법으로 변형 가능한 일반적인 변형 방법(generic transform)을 제안하였다[9]. Kiayias와 Tang은 단순한 기본 형태의 A-IBE와 공개 추적성, ID 재사용 등의 기능을 추가로 제공 가능한 A-IBE 기법까지 추가로 제안하였다.

Revocable IBE Boneh와 Franklin이 제안한 IBE 기법은 사용자 ID에 비밀키 유효시간정보를 추가하여 키 폐기 문제를 해결하였으나, 매 시간마다 PKG로부터 비밀키를 새로 발급받아야 하는 단점이 있다[3]. 이러한 문제를 해결하기 위해 Boldyreva 등은 Sahai와 Waters의 Fuzzy IBE[15]와 Naor 등의 트리 기반 폐기 시스템[13]을 결합한 RIBE 기법을 제안하였다[1]. 또한 Libert와 Vergnaud는 능동 안전성(Adaptive Security)을 만족하는 IBE[19]를 이용하여 능동 안전성을 만족하는 RIBE 기법을 제안하였다[12]. 이후, Seo와 Emura는 복호화 키 노출을 고려한 새로운 안전성 모델을 제안하였고 안전성을 증명하였다[17].

본 논문의 구성은 다음과 같다. Ⅱ장에서는 배경지식을 다루고, Ⅲ장에서는 A-RIBE의 알고리즘과 안전성 모델을 정의한다. Ⅳ장에서는 설계한 A-RIBE기법을 설명하고, Ⅴ장에서 안전성을 증명한다. Ⅵ장에서 타 기법들과 비교·분석하며 마지막으로, Ⅶ장에서 결론을 맺는다.

Ⅱ. 배경지식

2.1 겹선형 그룹(Bilinear Groups)

G과 G_T가 위수를 소수 p로 갖는 순환 군(group)이라고 하자. 군 G과 G_T에서 모두 이산대수 문제(Discrete Logarithm Problem)가 어렵다고 가정하자. 겹선형 함수(bilinear map)는 다음과 같은 성질을 갖는 $G\times G$에서 군 G_T 위로 맵핑되는 함수 $e : G\times G\rightarrow G_T$이다.

1. 겹선형성(Bilinearity) : 임의의 $g_1, g_2 \in G$와 $a, b \in Z_p$에 대해 $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$가 성립한다.
2. 비소실성(Non-degeneracy) : $e(g,g) \neq 1$을 만족하는 $g \in G$가 존재한다.
3. 계산 가능성(Computability) : 임의의 $g_1, g_2 \in G$에 대해서 $e(g_1, g_2)$를 계산하는 효율적인 알고리즘이 존재한다.

또한 e가 겹선형 함수일 때 (p, G, G_T, e, g)를 겹선형 그룹 시스템이라고 한다.

2.2 복잡도 가정(Complexity Assumption)

결정적 불완전 q-ABDHE 가정 겹선형 그룹 시스템 (p, G, G_T, e, g)에서의 결정적 불완전 q-ABDHE 가정은 임의의 $g' \in G$, $\alpha, z \in Z_p$에 대해 $\left(D=(g', g'^{(\alpha^{q+2})}, g, g^{\alpha}, g^{(\alpha^2)}, ..., g^{(\alpha^q)}), Z\right)$가 주어졌을 때, $Z = e(g,g')^{(\alpha^{q+1})}$와 $Z = e(g,g')^{z}$를 의미있는 확률로 구분 가능한 PPT 알고리즘 A가 존재하지 않는다는 것이다. A의 이점(advantage)은 다음과 같이 정의 된다.

$$Adv_A = |\Pr[A(D, Z=e(g,g')^{(\alpha^{q+1})})]=1 - \Pr[A(D, Z=e(g,g')^{z})]=1|$$

수정된 DDH-1 가정 겹겹선형 그룹 시스템 (p, G, G_T, e, g)에서의 수정된 DDH-1 가정은 임의의 $a, b, z \in Z_p$에 대해 $(g^a, e(g,g)^b, T=e(g,g)^{ab})$과 $(g^a, e(g,g)^b, T=e(g,g)^z)$를 의미있는 이점으로 구분 가능한 PPT 알고리즘 A가 존재하지 않는다는 것이다. A의 이점은 다음과 같이 정의 된다.

$$Adv_A = |\Pr[A(g^a, e(g,g)^b, T=e(g,g)^{ab})=1] - \Pr[A(g^a, e(g,g)^b, T=e(g,g)^z)=1]|$$

2.3 영지식 프로토콜과 ZK-PoK 프로토콜 (Zero-Knowledge Protocol and Proof of Knowledge Protocol)

영지식 프로토콜(Zero-Knowledge Protocol)은 증명하려는 내용을 드러내지 않고 내용이 참이라는 것을 증명하는 증명 방식이다[5]. 겹선형 그룹 시스템 (p, G, G_T, e, g)에 대해 영지식 프로토콜 $ZK\{(a, h_2): A=e(g,g)^a \wedge e(h_1, h_2)=A \bullet B\}$은 공통 입력 $((p, G, G_T, e, g), h_1, A, B)$에 대해 $A=e(g,g)^a$와 $e(h_1, h_2)=A \bullet B$를 만족하는 a와 h_2가 존재한다는 것이다[8]. 또한 이산 로그를 검증하는 ZK-PoK 프로토콜은 증명자에게 지수에 올라간 난수가 무엇인지 드러내지 않고 해당 난수가 올라가 있다는 것을 증명한다[16].

2.4 SE의 RIBE[17]

SE의 RIBE는 키 업데이트 과정의 효율성을 위해 이산 트리 기반으로 폐기 알고리즘을 진행하며 이를 위해 KUNode 알고리즘을 정의한다. KUNode 알고리즘은 이산 트리를 기반으로 하여 특정 노드에 위치하는 사용자가 폐기 되었을 때 해당 노드를 제외한 나머지 노드에만 업데이트키를 발급할 수 있도록 설계된 알고리즘이며 자세한 알고리즘의 정의는 [1]을 참고한다.

SE의 RIBE는 다음과 같은 7개의 알고리즘으로 이루어진다. F_{wat}와 F_{BB}는 각각 Waters IBE[19]와 Boneh-Boyen의 IBE[2] 기법에서 사용된 함수로 $ID=(b_1, b_2, ..., b_n) \in \{0,1\}^n$ 일 때 다음과 같이 정의된다.

$$F_{wat}(ID) = u' \prod_{i=1}^{n} u_i^{b_i} \ , \ F_{BB}(T) = v' v^T$$

- ***Setup*** (λ, N) : PKG는 보안 상수 λ를 이용하여 겹선형 그룹 시스템 (p, G, G_T, e, g)을 생성하고, $g, g_2, u_1, ..., u_n, u', v, v' \in G$와 $\alpha \in Z_p$를 임의로 선택해 공개 파라미터 $PP=\{g, g_1=g^\alpha, g_2, u_1, ..., u_n, u', v, v'\}$, 마스터키 $MSK=g_2^\alpha$를 생성한다. 이후 잎 노드(leaf node)의 개수가 최대 사용자 수 N인 이진트리 BT를 생성하여 상태정보 $st=BT$를 설정한다. 마지막으로 공집합인 폐기 리스트 RL를 생성한다.
- ***PKG*** (PP, ID, MSK, st) : PKG는 BT에서 할당되지 않은 잎 노드 η을 선택하여 사용자 ID를 η에 저장한다. 이후 루트에서부터 η까지의 경로에 있는 모든 노드 $\theta \in Path(\eta)$에 대해 $(g_\theta, \tilde{g}_\theta)$가 정의되어 있는지 확인하고, 만약 정의되어 있지 않다면 $(g_\theta, \tilde{g}_\theta = g_2 / g_\theta)$를 생성하여 노드 θ에 저장한다. 비밀키 SK_{ID}는 모든 $\theta \in Path(\eta)$에 대해 $r_\theta \in Z_p$를 임의로 선택하여 $SK_{ID} = \{\theta, D_{\theta,0} = g_\theta^\alpha F_{wat}(ID)^{r_\theta}, D_{\theta,1} = g^{r_\theta}\}_{\theta \in Path(\eta)}$와 같이 생성하여 사용자에게 전송한다.
- ***KeyUp*** (PP, MSK, T, RL, st) : PKG는 폐기 시점 T의 모든 노드 $\theta \in KUNode(BT, RL, T)$에 대해 $(g_\theta, \tilde{g}_\theta)$가 정의되어 있는지 확인하고, 정의되어 있지 않다면 $(g_\theta, \tilde{g}_\theta = g_2 / g_\theta)$를 생성하여 노드 θ에 저장한다. 이후 모든 $\theta \in KUNode(BT, RL, T)$에 대해 난수 $s_\theta \in Z_p$를 선택하여 업데이트키 $UK_T = \{\theta, \tilde{D}_{\theta,0} = \tilde{g}_\theta^\alpha F_{BB}(T)^{s_\theta}, \tilde{D}_{\theta,1} = g^{s_\theta}\}_{\theta \in KUNode(RL, BT, T)}$를 계산한다.
- ***DKG*** (PP, SK_{ID}, UK_T) : 사용자는 비밀키 $SK_{ID} = \{\theta, D_{\theta,0}, D_{\theta,1}\}_{\theta \in I}$와 업데이트키 $UK_T = \{\theta, \tilde{D}_{\theta,0}, \tilde{D}_{\theta,1}\}_{\theta \in J}$에 대해 $|I \cap J| = 0$ 이면 $\perp$를 출력하고, $|I \cap J| \neq 0$ 이면 노드 $\theta \in I \cap J$와

임의 값 $r, s \in Z_p$를 이용하여 복호화키 $DK_{ID,T}$
$= (D_{\theta,0}\widetilde{D}_{\theta,0}F_{wat}(ID)^r F_{BB}(T)^s, D_{\theta,1}g^r, \widetilde{D}_{\theta,1}g^s)$
$= (g_2^{\alpha}F_{wat}(ID)^{r+r_\theta}F_{BB}(T)^{s+s_\theta}, g^{r+r_\theta}, g^{s+s_\theta})$
$= (D_1, D_2, D_3)$를 생성한다.
이때 $I = Path(\eta)$ 이며 $J = KUNode(RL, BT, T)$를 의미한다.

- ***Enc*** (PP, ID, T, M) : 임의로 $t \in Z_p$를 선택하여 암호문 $CT = \langle C_0 = M(g_1, g_2)^t, C_1 = g^{-t}, C_2 = F_{wat}(ID)^t, C_3 = F_{BB}(T)^t \rangle$를 계산한다.
- ***Dec*** $(PP, CT, DK_{ID,T})$: 암호문 CT, 복호화키 $DK_{ID,T}$를 사용하여 다음과 같이 복호화 한다.
  $M = C_0 \times \prod_{i=1}^{3} e(C_i, D_i)$ 암호화 시점과 복호화 키의 T가 일치한다면 메시지 M을 출력하고, 그렇지 않다면 $\perp$를 출력한다.
- ***Revoke*** (PP, ID, T, RL, st) : 폐기된 사용자의 아이디 ID와 폐기 시점 T, 폐기 리스트 RL을 입력받아 폐기 리스트를 업데이트 한다.

III. 정의와 안전성 모델

3.1 White box 추적가능성과 Black box 추적가능성(White box traceability and Black box traceability)

A-IBE의 추적 가능성은 크게 두 가지 개념으로 나눌 수 있다. White box 추적가능성은 악의적인 PKG나 사용자가 비밀키를 그대로 유출했을 경우를 의미하며 추적 알고리즘에서는 유출된 비밀키의 그룹 번호(Family number)를 통해 생성자를 추적한다. Black box 추적가능성은 비밀키 그대로가 아닌 내부 알고리즘을 알 수 없지만 비밀키가 포함되어 암호문에 대한 복호화를 제공하는 복호화 박스 형태로 유출된 경우를 의미한다. 추적 알고리즘에서는 유출된 복호화 박스에 어떤 비밀키가 포함되어 있는지를 통해 생성자를 추적한다.

3.2 사용자 폐기를 지원하는 책임 기관 ID 기반 암호 (Accountable Authority Revocable Identity-Based Encryption, A-RIBE)

A-RIBE는 다음과 같은 8개의 알고리즘으로 구성 된다.

- ***Setup***$(\lambda, N) \rightarrow PP, MSK, RL = \varnothing, BT$: 보안 상수 λ과 시스템의 사용자 수 N을 입력 받아 공개 파라미터 PP와 마스터키 MSK, 공집합 상태인 폐기 리스트 RL과 키 업데이트에 사용할 BT를 출력한다.
- ***SKeyGen protocol*** $(PP, ID / MSK, BT) \rightarrow SK_{ID}$: 이 알고리즘은 사용자와 PKG 사이의 상호 프로토콜로 사용자와 PKG는 공통으로 PP, 사용자의 아이디 ID를 입력받으며 PKG는 추가로 MSK와 BT를 입력받는다. 프로토콜을 통하여 사용자는 비밀키 SK_{ID}를 출력 받으며 PKG는 비밀키 정보를 알 수 없다.
- ***UKeyGen*** $(PP, MSK, T, RL, BT) \rightarrow UK_T$: PKG는 PP, MSK, RL, BT 그리고 키가 업데이트 되는 시점인 T를 입력으로 받아 사용자의 비밀키를 주기적으로 업데이트 가능한 업데이트키 UK_T를 생성한다. 업데이트키는 BT에서 폐기된 사용자를 제외한 노드들에게만 발급된다.
- ***DKeyGen*** $(PP, SK_{ID}, UK_T) \rightarrow DK_{ID,T}$: 사용자는 PKG로부터 발급받은 업데이트키 UK_T와 자신의 비밀키 SK_{ID}와 PP를 입력으로 받아 특정 시점의 복호화키인 $DK_{ID,T}$를 생성한다.
- ***Encrypt***$(PP, ID, T, M) \rightarrow C$: 공개 파라미터 PP, 사용자의 아이디 ID, 암호화 시점 T 그리고 메시지 M을 입력받아 암호문 C를 출력한다.
- ***Decrypt***$(PP, C, DK_{ID,T}) \rightarrow M$: 암호문 C, 공개 파라미터 PP, 복호화키 $DK_{ID,T}$를 입력받는다. 암호화 시점과 복호화키의 T가 일치한다면 메시지 M을 출력하고, 그렇지 않다면 $\perp$를 출력한다.
- ***Trace*** $(DK_{ID,T} / PP, ID, DecBox_{ID,T}) \rightarrow n_F / User \text{ or } PKG$: 이 알고리즘은 비밀키의 생성자를 추적하는 알고리즘이다. White box로 사용자의 복호화키 $DK_{ID,T}$ 키를 입력받아 그룹 번호를 출력하거나 Black box로 $DecBox_{ID,T}$를 입력받아 사용자 또는 PKG를 출력한다. 이 알고리즘은 판사(Judge)가 수행한다.
- ***Revoke*** $(ID, T, RL) \rightarrow RL$: 폐기된 사용자의 아이디 ID와 폐기 시점 T, 폐기 리스트 RL을 입력받아 폐기 리스트를 업데이트 한다.

공개 추적성을 제공하는 A-RIBE (A-RIBE with Public Traceability) 본 논문에서는 A-RIBE 기법에 추가로 공개 추적성을 제공하는 기법을 제안한다. 공개 추적성은 사용자의 복호화키가 아닌 공개된 공개 추적키를 Trace 알고리즘의 입력으로 한다. 사용자의 복호화키가 추적 단계에서 항상 필요한 기존의 기법의 단점을 보완하여 사용자가 복호화키를 분실하였거나 추적에 협조적이지 않은 경우의 상황에도 복호화키의 생성자를 추적할 수 있는 기능이다. 공개 추적성을 제공하는 A-RIBE 알고리즘은 기존의 A-RIBE 알고리즘과 다음과 같은 차이가 있다.

- ***SKeyGen protocol*** $(PP, ID/MSK, BT) \rightarrow SK_{ID}/t_{ID}$: SKeyGen 프로토콜은 사용자에게 비밀키가 생성되며 PKG에게는 공개 추적키인 t_{ID}가 생성된다.
- ***Trace*** $(t_{ID}/PP, ID, DecBox_{ID,T}) \rightarrow User$ or PKG : 사용자의 복호화키인 $DK_{ID,T}$ 대신 공개 추적키인 t_{ID}을 입력으로 하여 복호화 박스의 생성자를 추적한다.
- ***Judge protocol*** $(PP, ID, t_{ID}/DK_{ID,T}) \rightarrow \perp$ or $Accept$: 사용자는 추적 알고리즘에 사용된 공개 추적키가 자신의 것이라는 것을 판사에게 증명한다. 공통으로 공개 파라미터 PP, 사용자의 ID, 공개 추적키 t_{ID}가 입력되며 사용자의 복호화키 $DK_{ID,T}$ 는 사용자만 입력받는다. 판사는 공개 추적키의 정당성을 판단한다.

3.3 A-RIBE의 Black box 추적가능성에 대한 안전성 모델

A-RIBE는 암·복호화 기법에 대한 안전성 증명과 추적 알고리즘에 대해 각각 PKG와 사용자가 악의적인 상황을 고려하여 총 3개의 게임을 통해 안전성을 증명한다. PKG가 복호화 오라클(Decryption Oracle)에 접근 가능한 모델을 완전 안전성 모델(full Black box model)이라 하며 접근 불가능한 모델을 약한 안전성 모델(weak Black box model)이라 한다. 본 논문에서는 약한 안전성 모델에 대한 게임을 정의한다.

▶ IND-RID-CPA 게임

A-RIBE의 IND-RID-CPA 게임은 기존 RIBE와 매우 유사하다.

- ***Setup*** : 챌린저는 ***Setup***알고리즘을 실행하여 PP, MSK, RL, BT를 얻고 PP를 공격자에게 준다.
- ***Oracle Query*** : 공격자는 챌린저에게 각각 비밀키, 업데이트키, 폐기된 ID에 대한 질의를 할 수 있다. 자세한 내용은 다음과 같다.
  - ✓ SKey Query : ID에 대해 비밀키 SK_{ID}를 생성하여 답한다.
  - ✓ UKey Query : 시간 T에 대해 업데이트키 UK_T를 생성하여 답한다.
  - ✓ Revoke : ID와 시간 T에 대해 업데이트 된 폐기 목록 RL을 답한다.
  - ✓ DKey Query : ID와 시간 T에 대해 복호화키 $DK_{ID,T}$를 생성하여 답한다.
- ***Challenge*** : 챌린저는 임의의 두 평문 M_0, M_1을 챌린지 ID인 ID^*, 챌린지 시점 T^*로 암호화한 두 챌린지 암호문 C_0^*, C_1^*을 생성한 뒤 둘 중 하나를 선택하여 C_b^* , $b \in \{0,1\}$을 공격자에게 준다.
- ***Guess*** : 공격자는 챌린지 암호문이 어떠한 평문으로 암호화 되었는지 추측하여 $b' \in \{0,1\}$을 출력한다.

이때 $b = b'$이라면 공격자가 승리한다.

▶ Dishonest PKG 게임

Dishonest PKG 게임은 PKG가 악의적인 상황에 대한 안전성 모델이다.

- ***Setup*** : 악의적인 PKG처럼 행동하는 공격자는 보안 상수 λ를 선택하고 ***Setup***알고리즘을 실행하여 PP와 MSK를 얻고 PP와 공격하고자 하는 ID^*를 챌린저에게 준다.
- ***Key Generation*** : 공격자와 챌린저는 A-RIBE 기법의 키 생성 프로토콜을 시행하여 ID^*에 대한 비밀키를 생성한다.
- ***Key Update*** : 악의적인 PKG처럼 행동하는 공격자는 사용자가 복호화키를 생성할 수 있도록 시간에 맞는 업데이트키를 생성하여 챌린저에게 준다.
- ***Create Decryption box*** : 공격자는 ID^*에 대한 복호화 박스를 만들어서 출력한다.

만약 이 복호화 박스를 입력으로 하는 추적 알고리즘이 '사용자'를 출력한다면 공격자가 게임에서 승

리한다.

▶ Dishonest User 게임

Dishonest User 게임은 여러 명의 악의적인 사용자들이 공모한 상황에 대한 안전성 모델이다. 능동적인(adaptive) 공격자일 경우 게임의 모든 시점에 비밀키에 대해 질의가능하다.

- ***Setup*** : 챌린저는 보안 상수 λ를 선택하고 ***Setup***알고리즘을 실행하여 PP와 MSK를 얻고 PP를 공모한 악의적인 사용자 집단처럼 행동하는 공격자에게 준다.
- ***Key Query*** : 공격자와 챌린저는 A-IBE 기법의 키 생성 프로토콜을 시행하여 $ID_1,...,ID_q$에 대한 비밀키를 생성한다.
- ***Key Update*** : 챌린저는 시간에 맞는 업데이트키를 생성하여 공격자에게 준다.
- ***Create Decryption box*** : 공격자는 ID^*에 대한 복호화 박스를 만들어서 출력한다.

만약 이 복호화 박스를 입력으로 하는 추적 알고리즘이 'PKG'를 출력한다면 공격자가 게임에서 승리한다.

IV. 제안 기법

- ***Setup*** : 주어진 보안 상수 λ와 시스템의 사용자 수 N에 대해 PKG는 겹선형 그룹 시스템 (p, G, G_T, e, g)을 생성한 뒤, $g, g_2, u', u_1, ..., u_n, v' , v \in G$ 와 $\alpha, \beta, \gamma \in Z_p$를 임의로 선택한다. 공개 파라미터와 마스터 비밀키는 다음과 같다. $PP = \{g, g_1 = g^{\alpha}, g_2, g_3 = g^{\gamma}, h = g^{\beta}, u', u_1, ..., u_n, v', v\}$, $MSK = \{\alpha, \beta, \gamma\}$. 또한 공집합 상태인 폐기 리스트 RL과 이산 트리인 $st = BT$를 출력한다. 이때 BT의 잎(leaves)수는 사용자 수인 N과 같다.
- ***SKeyGen protocol*** : 이 알고리즘은 사용자와 PKG 사이의 상호 프로토콜로 사용자와 PKG는 공통으로 PP, 사용자의 아이디 ID를 입력받으며 PKG는 추가로 MSK와 BT를 입력받는다. 자세한 프로토콜은 다음과 같다.

1. PKG는 아직 할당되지 않은 잎 η을 BT에서 선택한 후 사용자의 ID를 노드 η에 저장한다. $\theta \in Path(\eta)$에 대해 다음과 같이 연산한다. g_θ가 저장되어 있다면 사용하며 저장되어 있지 않다면 $g_\theta \in G$를 임의로 선택한 후 노드 θ에 $(g_\theta, \tilde{g}_\theta = g_2 / g_\theta)$ 를 저장한다. $a_\theta \in Z_p$를 임의로 선택한 후 $SK_1 = \{\theta, SK_1', SK_1''\} = \left\{\theta, g_\theta^{\gamma} F_{wat}(ID)^{a_\theta}, g^{a_\theta}\right\}_{\theta \in Path(\eta)}$를 계산하여 사용자에게 전송한다.
2. 사용자는 $r \in Z_p$를 임의로 선택하여 $R = g^r$ 을 계산한 뒤 PKG에게 전송한다. PKG는 ZK-PoK를 통해 g에 대한 R의 이산 로그가 r임을 검증한다.
3. PKG는 ZK-PoK를 체크하여 정당하지 않다면 프로토콜을 중단한다. 만약 ZK-PoK이 정당하다면 $r' \in Z_p$를 임의로 선택하여 $R' = R^{\beta} = h^r$, $h' = (R' \cdot g^{-r'})^{1/(\alpha - ID)}$ 를 계산한 뒤 사용자에게 $(R', (r', h'))$을 전송한다.
4. 사용자는 다음의 연산이 동일한지 체크한다.
$$\begin{cases} e(g, R') = e(R, h) \\ e(g_1 \cdot g^{-ID}, h') = e(g, R' g^{-r'}) \end{cases}$$
만약 연산이 일치하지 않는다면 사용자는 프로토콜을 중단한다.
5. 사용자는 $r_{ID} = \frac{r'}{r}$, $h_{ID} = h'^{\frac{1}{r}} = (h g^{-r_{ID}})^{\frac{1}{\alpha - ID}}$ 를 계산하여 $SK_2 = (r_{ID}, h_{ID})$ 로 설정한다.
사용자의 비밀키는 다음과 같다.
$SK_{ID} = (SK_1, SK_2)$
6. 사용자는 $R_{ID} = e(g, g)^{r_{ID}}$를 계산하여 PKG에게 전송한 뒤 상호 영지식 증명 $ZK\{(r_{ID}, h_{ID}) : R_{ID} = e(g, g)^{r_{ID}} \wedge e(g_1 \cdot g^{-ID}, h_{ID}) = e(g, h) \cdot R_{ID}^{-1}\}$을 통해 PKG에게 R_{ID}가 정당함을 증명한다. 이때 효율적인 영지식 증명 기법을 사용하여 본 기법의 효율성을 증대할 수 있다. R_{ID}가 정당함이 증명되면 PKG는 공개 추적키인 $t_{ID} = (ID, R_{ID})$를 공개 추적키 리스트인 TK에 추가한다.

- ***UKeyGen*** : PKG는 $st = BT$를 분석하고 각 노드 $\theta \in KUNode(BT, RL, T)$에 대해 $b_\theta \in Z_p$를 임의로 선택한 후 노드에 저장된 $\tilde{g}_\theta$으로 사용자의 업데이트키를 계산한다. 사용자의 업데이트키는 다음과 같다.
$UK_T = \{\theta, UK', UK''\}$
$= \left\{\theta, \tilde{g}_\theta^{\gamma} F_{BB}(T)^{b_\theta}, g^{b_\theta}\right\}_{\theta \in KUNode(RL, BT, T)}$

- ***DKeyGen*** : 사용자는 PKG로부터 발급받은 업데이트키 $UK_T=\{\theta, UK', UK''\}_{\theta\in J}$와 자신의 비밀키 $SK_{ID}=\{SK_1, SK_2\}_{\theta\in I}$ 를 분석하여 만약 $I\cap J=\varnothing$ 라면 $\perp$를 출력하고 그렇지 않다면 $\theta\in I\cap J$ 를 선택하고 $a,b\in Z_p$를 임의로 선택하여 복호화키 $DK_{ID,T}$를 다음과 같이 생성한다.

  $DK_1=$

  $(SK_1' UK' F_{wat}(ID)^a F_{BB}(T)^b, SK_1'' g^a, UK'' g^b)$

  $=(g_2^{\gamma} F_{wat}(ID)^{a+a_\theta} F_{BB}(T)^{b+b_\theta}, g^{a+a_\theta}, g^{b+b_\theta})$

  $=(d_1,d_2,d_3)$

  $DK_{ID,T}=(DK_1, DK_2=SK_2)$
- ***Encrypt*** : 메시지 공간에서 임의의 m_1을 선택하여 $m_1\oplus m_2=m$ 이 되도록 설정한다. 임의로 $t,s\in Z_p$를 뽑은 후, 각 메시지에 대해 암호문을 다음과 같이 계산한다.

  $C_1=\langle m_1 \cdot e(g_3,g_2)^t, g^t, F_{wat}(ID)^t, F_{BB}(T)^t\rangle$

  $=\langle C_{1,0}, C_{1,1}, C_{1,2}, C_{1,3}\rangle$

  $C_2=\langle g_1^s g^{-sID}, e(g,g)^s, m_2 \cdot e(g,h)^{-s}\rangle$

  $=\langle C_{2,0}, C_{2,1}, C_{2,2}\rangle$

  $C=\langle C_1, C_2\rangle$
- ***Decrypt*** : 암호문 C, 복호화키 $DK_{ID,T}$를 사용하여 다음과 같이 복호화 한다.

  $$m_1=C_{1,0}\times\frac{e(C_{1,2},d_2)e(C_{1,3},d_3)}{e(C_{1,1},d_1)}$$

  $m_2=e(C_{2,0},h_{ID}) \cdot C_{2,1}^{r_{ID}} \cdot C_{2,2}$

  $m=m_1\oplus m_2$

  암호화 시점과 복호화키의 T가 일치한다면 메시지 m을 출력하고, 그렇지 않다면 $\perp$를 출력한다.
- ***Trace*** : 주어진 공개 파라미터 PP와 사용자의 ID, 공개 추적키 t_{ID} 그리고 ϵ-확률 복호화 박스 $DecBox_{ID,T}$에 대해 다음과 같은 과정으로 복호화 박스의 생성자를 추적한다.

1. 카운트를 $ctr=0$으로 설정하고 다음의 실험을 $8\lambda/\epsilon$번 반복한다.

   - 서로 다른 $s,s'\in Z_p$를 임의로 선택한다.

   임의의 메시지 $m\in G_T$를 선택하고 다음과 같이 변형된 형태의 암호문을 생성한다.

   $C_{2,0}=g_1^s g^{-sID}$ $C_{2,1}=e(g,g)^{s'}$

   $C_{2,2}=m_2 \cdot e(g,h)^{-s} \cdot R_{ID}^{(s-s')}$

   - ϵ-확률 복호화 박스 $DecBox_{ID,T}$에 변형된 형태의 암호문을 입력하면 메시지 m'을 출력한다. 만약 $m=m'$ 이라면 ctr을 증가시킨다.
2. 만약 $ctr=0$ 이라면 PKG를 출력하고 $ctr\neq 0$ 이라면 사용자를 출력한다.

- ***Judge*** : 사용자는 비밀키 생성 프로토콜의 마지막 단계에서 PKG에게 R_{ID}가 정당함을 증명했던 것과 같은 방법으로 상호 영지식 증명을 통해 t_{ID}가 자신의 공개 추적키라는 것을 증명한다.
- ***Revoke*** : 폐기된 사용자의 아이디 ID와 폐기 시점 T, 폐기 리스트 RL을 입력받아 폐기 리스트를 업데이트 한다.

V. 안전성 증명

5.1 IND-RID-CPA 안전성

정리 1. 제안한 A-RIBE는 결정적 불완전 q-ABDHE 가정과 수정된 DDH-1 가정이 성립하고 SE의 RIBE 기법[17]이 IND-ID-CPA 공격자에게 안전할 때 IND-RID-CPA 모델에서 안전하다.

증명. 만약 제안된 기법에 대한 IND-RID-CPA공격자 A가 존재한다면 SE의 RIBE 기법을 IND-RID-CPA 모델에서 공격에 성공하는 공격자 B를 설계가능하다. 해당 증명은 제안 기법을 SE의 RIBE의 IND-RID-CPA 공격자로 리덕션하여 증명한다.

- ***Setup*** : B는 입력으로 SE의 RIBE의 공개 파라미터 PP_{RIBE}를 받는다. B는 마스터키 α,β를 뽑은 후 $g_1=g^{\alpha}, h=g^{\beta}$를 계산하여 A-RIBE 기법의 공개 파라미터 $PP=\{g_1,g_2,PP_{RIBE}\}$를 공격자 A에게 준다.
- ***Oracle Query*** : B는 A의 질의에 대해 SE의 RIBE의 오라클을 사용하여 답을 한다. 자세한 질의는 다음과 같다.

✓ SKey Query : A의 질의에 대해 B는 RIBE의 PKG(·) 오라클을 사용하여 SK_1을 생성하며

자신이 뽑은 마스터키를 사용하여 SK_2를 생성하여 질의에 답한다.

✓ UKey Query : A의 질의에 대해 B는 RIBE의 KeyUp(·) 오라클을 사용하여 UK_T를 생성하여 질의에 답한다.

✓ DKey Query : A의 질의에 대해 B는 SKey Query와 UKey Query를 통해 얻은 SK_{ID}와 UK_T를 사용하여 $DK_{ID,T}$를 생성하여 질의에 답한다.

- ***Challenge*** : A는 B에게 타겟 ID^*와 T^*를 그리고 두 메시지 M_0^*, M_1^*를 보낸다. B는 메시지를 각각 $M_0^* = m_{0,1}^* \oplus m_{0,2}^*$ $M_1^* = m_{1,1}^* \oplus m_{1,2}^*$로 나눈 뒤 SE의 RIBE 공격자에게 $ID^*, T^*, m_{0,1}^*, m_{1,1}^*$를 전송하여 SE의 RIBE의 챌린지 암호문 $C_1^* = (C_{1,0}^*, C_{1,1}^*, C_{1,2}^*, C_{1,3}^*)$를 받아온다. B는 RIBE의 챌린지 암호문 C_1^*이 $m_{0,1}^*, m_{1,1}^*$중 어떤 평문으로 암호화 하였는지 추측하여 해당하는 평문의 나머지 부분인 $m_{b,2}^*$를 암호화하여 C_2^*를 생성한다. B의 추측이 1/2의 확률로 맞다고 가정한다.
  B는 A에게 챌린지 암호문으로 $C^* = (C_1^*, C_2^*)$를 준다.
- ***Guess*** : 공격자는 챌린지 암호문이 어떠한 평문으로 암호화 되었는지 추측하여 $b' \in \{0,1\}$을 출력한다. □

5.2 Dishonest PKG 안전성

정리 2. 제안한 A-RIBE 기법은 LDZW'13 기법 [10]이 Dishonest PKG 게임에서 안전하다면 Dishonest PKG 게임에서 안전하다.

증명. 만약 제안된 A-RIBE 기법에 대해 Dishonest PKG 게임에서 승리하는 공격자 A가 존재한다면 LDZW'13 기법의 Dishonest PKG 게임에서 승리하는 공격자 B를 설계 가능하다.

- ***Setup*** : A는 보안 상수 λ를 선택하고 Setup 알고리즘을 실행하여 A-RIBE의 공개 파라미터 PP와 마스터키 MSK를 얻는다. A는 공격할 ID인 ID^*와 PP를 B에게 전달한다. B는 PP의 일부분인 (g, g_1, h)를 LDZW'13 기법의 공개 파라미터 PK로 설정하여 ID^*와 함께 챌린저에게 전달한다.
- ***Key Query*** : B는 A와의 ID^*에 대한 키 생성 프로토콜을 통해 SK_{ID^*}를 얻는다. B는 A와의 키 생성 프로토콜 과정에서 얻은 $(R', (r', h'))$과 챌린저에게 전송하여 LDZW'13의 키 생성 프로토콜을 완료한다.
- ***Key Update*** : A는 B에게 시간에 맞는 업데이트키를 생성하여 준다.
- ***Create Decryption box*** : A는 ID^*에 대한 복호화 박스를 만들어서 출력한다. B는 이 복호화 박스에 DK_1에 대한 정보를 제거하여 출력한다.

DK_1은 B가 LDZW'13 기법의 Dishonest PKG 게임에서 승리하는데 사용되지 않으며 A가 출력한 복호화 박스의 DK_2는 항상 A-RIBE 기법의 Dishonest PKG 게임에서 승리하는 복호화 박스이므로 LDZW'13 기법의 Dishonest PKG 게임에서 승리한다. □

5.3 Dishonest User 안전성

정리 3. 제안한 A-RIBE 기법은 LDZW'13 기법 [10]이 Dishonest User 게임에서 안전하다면 Dishonest User 게임에서 안전하다.

증명. 만약 제안된 A-RIBE 기법에 대해 Dishonest User 게임에서 승리하는 공격자 A가 존재한다면 LDZW'13 기법의 Dishonest User 게임에서 승리하는 공격자 B를 설계 가능하다.

- ***Setup*** : B는 LDZW'13 기법의 공개 파라미터 PK를 입력으로 받는다. 마스터키 γ를 선택하고 $g_2, u', u_1, \ldots, u_n, v', v$ 를 임의로 선택하여 공개 파라미터 $PP = \{PK_{LDZW}, \quad g_2, g_3 = g^{\gamma}, u', u_1, \ldots, u_n, v', v\}$ 를 생성한 뒤 A 에게 준다.
- ***Key Query*** : B는 A와의 $ID_1, \ldots, ID_q$에 대한 키 생성 프로토콜에서 SK_1을 자신이 뽑은 마스터키로 생성하여 전달하고 나머지 단계는 LDZW'13 의 키 생성 프로토콜 오라클을 통하

여 생성한다.

- ***Key Update*** : B는 시간에 맞는 업데이트키를 생성하여 A에게 준다.
- ***Create Decryption box*** : A는 ID^*에 대한 복호화 박스를 만들어서 출력한다. B는 이 복호화 박스에 DK_1에 대한 정보를 제거하여 출력한다.

DK_1은 B가 LDZW'13 기법의 Dishonest User 게임에서 승리하는데 사용되지 않으며 A가 출력한 복호화 박스의 DK_2는 항상 A-RIBE 기법의 Dishonest User 게임에서 승리하는 복호화 박스이므로 LDZW'13 기법의 Dishonest User 게임에서 승리한다. □

VI. 비교와 분석

본 절에서는 다양한 A-IBE 기법과 RIBE 기법을 통해 생성 가능한 A-RIBE에 대해 살펴보고 각각의 기법에 대해 비교 분석한다.

- 메시지를 분리하는 방법

Goyal의 Waters IBE와 Fuzzy IBE를 기반으로 하는 두 번째 A-IBE 기법[6]은 하나의 메시지 m을 XOR 연산을 활용하여 $m = m_1 \oplus m_2$를 만족하는 두 개의 메시지 m_1, m_2로 분리하여 각각의 메시지를 기반 IBE를 사용하여 따로 암호화하는 방법을 사용한다. 이때 두 개의 IBE 기법 중 하나의 비밀키만이 A-IBE의 추적을 위한 비밀키로 활용된다. 이러한 메시지 분리 방법을 사용하여 A-RIBE를 설계가능하다. 메시지를 m_1, m_2로 분리한 뒤 기존의 A-IBE와 RIBE 기법을 사용하여 m_1은 A-IBE 기법으로 m_2는 RIBE 기법으로 암호화하여 암호문을 생성한다. 이때 두 개의 메시지 중 하나의 메시지를 고정된 값으로 사용한다면 고정된 메시지를 암호화하는 암호의 기능을 온전히 제공하지 못한다. 예를 들어 RIBE 부분의 메시지를 고정하여 사용한다면 사용자는 m_2를 알고 있기 때문에 시스템에서 폐기된 후에도 폐기 시점이 지난 복호화키를 가지고 완벽하게 메시지를 복호화 할 수 있으며 따라서 사용자 폐기 기능을 제공하지 못하는 암호 시스템이 된다. 따라서 암호화 과정에서 메시지를 분리할 때는 항상 새로운 값으로 분리하여야 한다. 복호화키는 A-IBE의 비밀키와 RIBE의 복호화키를 합친 형태가 되며 A-IBE의 키로 생성자를 추적하며 RIBE의 비밀키를 업데이트 하는 방법으로 폐기를 제공가능하다. [Table.1]는 메시지를 분리하는 방법을 사용했을 때 생성되는 A-RIBE의 키와 암호문 크기, 안전성 모델과 제공 가능한 기능을 나타낸다.

Table.1 key and ciphertext size, security and provided skills of A-RIBE using message separation skill

secret key size		the size of A-IBE's secret key + the size of RIBE's secret key
update key size		the size of RIBE's update key
ciphertext size		the size of A-IBE's cipher text + the size of RIBE's cipher text
security model	security of encryption scheme	same as RIBE and A-RIBE encryption scheme
	Dishonest PKG	same as A-IBE
	Dishonest User	same as A-IBE
public traceability		same as A-IBE
ID reuse		same as A-IBE

메시지를 분리하는 방법은 A-IBE와 RIBE를 간단히 결합하여 A-RIBE를 설계할 수 있고 비밀키와 암호문의 크기가 각 기법의 비밀키와 암호문을 합쳐 놓은 것만큼만 길어진다는 장점이 존재하지만 두 기법의 공개 파라미터가 겹치지 않거나 다른 가정을 기반으로 할 경우 공개 파라미터의 길이가 늘어나며 기반 A-IBE가 제공하지 않는 기능을 제공할 수 없는 단점이 존재한다. 따라서 메시지를 분리하여 A-RIBE를 설계하는 방법은 추가적으로 기능을 제공하는 것이 필요하지 않으며 비밀키의 저장 공간이 한정적이고 통신하는 암호문의 길이가 상대적으로 작아야하는 환경에 적합하다.

- Kiayias와 Tang의 일반적 변형 방법을 활용

Kiayias와 Tang이 제안한 IBE를 A-IBE 기법으로 변형하는 방법[9]은 비밀키에 생성자를 추적할 수 있는 정보를 담기 위해 ID를 비트 단위로 분리하

여 사용자가 선택한 비트 스트링을 결합하고 OT 프로토콜[14]를 사용하여 비밀키를 발급 받는 형태이다. Kiayias와 Tang의 일반적 변형 방법을 IBE 기법이 아닌 RIBE 기법에 적용하면 A-RIBE의 설계가 가능하다. PKG와 사용자는 OT 프로토콜을 통해 비밀키를 생성하며 PKG는 시간에 따른 업데이트키를 생성하여 사용자에게 전송한다. 이때 비밀키는 ID의 길이 만큼 생성하여 발급하며 업데이트키는 ID 정보를 포함되지 않으므로 매 시간마다 생성하면 된다. [Table.2]는 Kiayias와 Tang의 일반적 변형 방법을 사용했을 때 생성되는 A-RIBE의 키와 암호문 크기, 안전성 모델과 제공 가능한 기능을 나타낸다.

Table. 2 key and ciphertext size, security and provided skills of A-RIBE using KT's method

secret key size		relative to ID length
update key size		the size of RIBE's update key
ciphertext size		relative to ID length
security model	security of encryption scheme	same as RIBE encryption scheme
	Dishonest PKG	weak black box
	Dishonest User	adaptive
public traceability		always provide
ID reuse		always provide

Kiayias와 Tang의 일반적 변형 방법은 모든 RIBE를 A-RIBE로 변형 가능하고 공개 추적성과 ID 재사용을 항상 제공 가능한 장점이 존재하지만 ID의 길이만큼 비밀키와 복호화키의 개수, 암호문의 개수, 그리고 키를 생성하는 과정에서 OT 프로토콜 진행 횟수와 연산량이 늘어난다는 단점이 존재한다. 따라서 Kiayias와 Tang의 일반적 변형 방법을 사용하여 A-RIBE를 설계하는 방법은 다양한 기능을 제공해야 하며 비밀키와 복호화키를 충분히 저장 가능 하고 여러 번의 통신이 가능한 환경에 적합하다.

VII. 결 론

본 논문에서는 ID 기반 암호의 키 위탁 문제를 완화하는 복호화키 추적 기능과 사용자를 폐기하고 복호화키를 업데이트 할 수 있는 암호 프리미티브인 A-RIBE와 그 안전성 모델을 제안하였다. 또한 결합 가능한 A-IBE와 RIBE의 종류에 따른 장·단점에 대해 분석하고 어떤 환경에 적합한 지 제안하였다. 향후에, 완전 안전성 모델에서 증명 가능한 A-RIBE 기법을 설계하고 증명하는 연구와 더욱 효율적인 기법을 설계하는 연구가 필요하다.

References

[1] A. Boldyreva, V. Goyal and V. Kumar, "Identity-based encryption with efficient revocation", Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 417-426, Oct. 2008.

[2] D. Boneh and X. Boyen, "Efficient selective-id identity based encryption without random oracles", Advances in Cryptology, EUROCRYPT'04, LNCS 3027, pp. 223 - 238, 2004.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," Advances in Cryptology, CRYPTO'01, LNCS 2139, pp. 213-229, 2001.

[4] C. Gentry, "Practical identity-based encryption without random oracles," Advances in Cryptology, EUROCRYPT'06, LNCS 4004, pp. 445-464, 2006.

[5] O. Goldreich, The Foundations of Cryptography, Basic Techniques, vol. 1, Cambridge University Press, 2001.

[6] V. Goyal, "Reducing trust in the PKG in identity based cryptosystems," Advances in Cryptology, CRYPTO'07, LNCS 4622, pp. 430-447, 2007.

[7] V. Goyal, S. Lu, A. Sahai and B. Waters, "Black-box accountable authority identity-based encryption," Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 427-436, Oct. 2008.

[8] J. Groth and A. Sahai, "Efficient

Non-interactive Proof Systems for Bilinear Groups", Advances in Cryptology, EUROCRYPT'08, LNCS 4965, pp. 415 - 432, 2008.

[9] A. Kiayias and G. Tang, "Making Any Identity-Based Encryption Accountable, Efficiently," European Symposium on Research in Computer Security, pp. 326-346, Sep. 2015.

[10] J. Lai, R.H. Deng, Y. Zhao and J. Weng, "Accountable authority identity-based encryption with public traceability," Topics in Cryptology-CT-RSA'13, LNCS 7779, pp. 326-342, 2013.

[11] B. Libert and D. Vergnaud, "Towards black-box accountable authority IBE with short ciphertexts and private keys", Proc. of the PKC'09, LNCS 5443, pp. 235 - 255, 2009.

[12] B. Libert and D. Vergnaud, "Adaptive-id secure revocable identity-based encryption", Topics in Cryptology-CT-RSA'09, LNCS 5473, pp. 1-15, 2009.

[13] D. Naor, M. Naor and J. Lotspiech, "Revocation and tracing schemes for stateless receivers", Advances in Cryptology, CRYPTO'01, LNCS 2139, pp. 41-62, 2001.

[14] M. Naor and B. Pinkas, "Efficient oblivious transfer protocols", Proceedings of the 12th annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, pp. 448 - 457, Jan. 2001.

[15] A. Sahai and B. Waters, "Fuzzy identity based encryption", Advances in Cryptology, EUROCRYPT'05, LNCS 3494, pp. 457-473, 2005.

[16] C.P. Schnorr, "Efficient Identification and Signatures for Smart Cards", Advances in Cryptology, CRYPTO'89, LNCS 435, pp. 239 - 252, 1990.

[17] J.H. Seo and K. Emura, "Revocable identity-based encryption revisited: Security model and construction", Proc. of the PKC'13, LNCS 7778, pp. 216-234, 2013.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes", Advances in Cryptology, CRYPTO'84, LNCS 196, pp. 47-53, 1985.

[19] B. Waters, "Efficient identity-based encryption without random oracles", Advances in Cryptology, EUROCRYPT'05, LNCS 3494, pp. 114-127, 2005.

〈저자소개〉

최 수 리 (Suri Choi) 학생회원
2016년 2월: 고려대학교 수학과 졸업
2016년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 공개키 암호, 암호 이론, 암호 프로토콜

이 동 훈 (Dong Hoon Lee) 종신회원
1983년 8월: 고려대학교 경제학사 졸업
1987년 12월: Oklahoma University 전산학과 석사 졸업
1992년 5월: Oklahoma University 전산학과 박사 졸업
1993년 3월~1997년 2월: 고려대학교 전산학과 조교수
1997년 3월~2001년 2월: 고려대학교 전산학과 부교수
2001년 3월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 암호프로토콜, 암호이론, USN이론, 키 교환, 익명성 연구, PET 기술