논문 2009-46CI-1-7

CCA 안전성을 제공하는 ID기반 프락시 재암호화 기법

(ID-Based Proxy Re-encryption Scheme with Chosen-Ciphertext Security)

구 우 권*, 황 정 연**, 김 형 중***, 이 동 훈**

(Woo Kwon Koo, Jung Yeon Hwang, Hyoung-Joong Kim, and Dong Hoon Lee)

요 약

ID기반 재암호화 기법(ID-based proxy re-encryption scheme)은 사용자 간의 복호 능력 위임을 가능하게 하며 분산 데이터 저장, DRM, 이메일 전달 시스템 등의 다양한 분산 암호시스템을 위해 활발히 연구되고 있다. 최근 재암호화키 생성의 비상호성(Non-interactivity)을 제공하는 기법이 Green과 Ateniese에 의해 제안되었다. 이 기법은 선택 암호문 공격에 대한 안전성을 제공하기 위해 설계되었다. 본 논문에서는 Green-Ateniese ID기반 재암호화 기법이 근본적으로 사용자 키 노출 공격에 취약함을 보이고 선택 암호문 공격에 대한 안전성이 보장되지 않음을 증명한다. 그리고 이러한 보안 취약점을 해결하는 새로운 두 가지 ID기반 재암호화 기법들을 제안한다. 제안 기법들이 랜덤 오라클 모델(Random Oracle Model)에서 단순 평문 공격과 선택 암호문 공격에 대해 각각 안전함을 증명한다. 선택 암호문 공격에 안전한 제안 기법을 구성하기위해, 본 논문에서는 최초로 짧은 서명에 기반한 자가 인증 기법을 고안하여 적용한다. 제안 기법의 중요한 특징은 재암호화 후 암호문의 구조가 유지되는 것이다. 따라서 이전 기법들과는 대조적으로 암호문 확장이 발생되지 않는다. 또한 재암호화의 횟수에 제한이 없어서 연속적인 암호문 변환이 가능하여 다중의 사용자를 위한 복호 능력 위임을 구현할 수 있다.

Abstract

A proxy re-encryption scheme allows Alice to temporarily delegate the decryption rights to Bob via a proxy. Alice gives the proxy a re-encryption key so that the proxy can convert a ciphertext for Alice into the ciphertext for Bob. Recently, ID-based proxy re-encryption schemes are receiving considerable attention for a variety of applications such as distributed storage, DRM, and email-forwarding system. And a non-interactive identity-based proxy re-encryption scheme was proposed for achieving CCA-security by Green and Ateniese. In the paper, we show that the identity-based proxy re-encryption scheme is unfortunately vulnerable to a collusion attack. The collusion of a proxy and a malicious user enables two parties to derive other honest users' private keys and thereby decrypt ciphertexts intended for only the honest user. To solve this problem, we propose two ID-based proxy re-encryption scheme schemes, which are proved secure under CPA and CCA in the random oracle model. For achieving CCA-security, we present self-authentication tag based on short signature. Important features of proposed scheme is that ciphertext structure is preserved after the ciphertext is re-encrypted. Therefore it does not lead to ciphertext expansion. And there is no limitation on the number of re-encryption.

Keywords: Identity-Based Proxy Re-encryption, Identity-Based Encryption, CCA-Security, Collusion Attack

I. 서 론

1. 배경

ID기반 (프락시) 재암호화 기법(ID-based proxy re-encryption scheme)은 프락시(proxy server)를 통해서 위임자(delegator)의 공개 아이디(Identity)로 암호화된 암호문을 대리자(delegatee)의 비밀키로 복호화 할

* 학생회원, ** 정회원, *** 평생회원.
고려대학교 정보경영공학전문대학원
(Graduate School of Information Management & Security, Korea University)
※ "이 연구에 참여한 연구자(의 일부)는 '2단계 BK21 사업'의 지원비를 받았음"
접수일자: 2008년12월10일, 수정완료일: 2009년1월12일

수 있도록 암호문을 변환시키는 기법이다. ID기반 재암호화 시스템에서 위임자는 자신의 비밀키를 노출시키지 않고 일시적으로 자신의 암호문에 대한 복호화 권리를 프락시를 통하여 대리자에게 분산시킬 수 있다. 이처럼 재암호화 기법은 DRM(Digital Rights Management), 이메일 전달 시스템, 분산 데이터 저장에서의 접근 제어 등 여러 응용 분야에 원천 기술로 사용될 수 있으며 최근 활발하게 연구되고 있다[2, 8, 11]. 한 예로, 여러 응용 분야 중에 분산된 네트워크 저장소에 대한 접근 제어(access control) 시스템을 고려할 수 있다. 이러한 접근 제어 시스템에서 위임자 A는 컨텐츠 제공자로서 자신의 공개 아이디로 컨텐츠를 암호화하여 네트워크 저장소에 저장하고 프락시는 시스템에 대한 접근 제어자로서 A가 대리자 B의 자신의 컨텐츠로의 접근을 허용할 때, 저장된 A의 암호문을 B의 비밀키로 복호화 할 수 있는 암호문으로 변환시키는 역할을 수행한다. 프락시는 위임자의 비밀키 노출 없이 암호문을 변환할 수 있으므로 접근 제어 서버에 대한 신뢰도를 줄일 수 있는 이점을 지닌다.

다양한 활용성에도 불구하고 유용한 속성들과 강한 안전성을 동시에 만족하는 ID기반 재암호화 기법을 구성하는 것은 단순하지 않다. 특히 완전한 신뢰가 주어지지 않는 프락시를 고려해야 하므로 문제는 더욱 더 복잡해진다. 기본적으로, 프락시가 암호문을 변환하는 재암호화 과정을 수행할 때 프락시는 위임자와 대리자의 비밀키를 알 수 없어야 하고 단독으로는 암호문에 대한 평문의 정보를 알 수 없어야 한다. 또한 프락시와 위임자 또는 프락시와 대리자의 사이의 공모를 통하여 시스템의 보안 취약점이 드러나지 않아야 한다. ID기반 재암호화 기법의 구조상, 프락시와 대리자의 사이의 공모는 위임자의 아이디로 암호화된 암호문 (또는 암호문 중 일부)을 (주어진 특정한 조건 내에서) 항상 복호화 할 수 있다. 하지만 프락시와 대리자의 공모를 통해 위임자의 비밀키에 대한 유용한 정보가 노출된다면 이는 심각한 보안 위협이 된다. 사용자의 비밀키는 다양한 형태로 중요하고 사용자-민감한(user-sensitive) 보안 업무들을 수행하는 데 이용되기 때문이다.

최근에는 재암호화 과정의 단방향성(Unidirectionality)과 재암호화키 생성의 비상호성(Non-interactivity)을 제공하는 ID기반 재암호화 기법이 Green과 Ateniese에 의해 제안되었다[10]. 이 기법은 선택 암호문 공격(Chosen Ciphertext Attack, CCA)에 대한 안전성을 제공하기 위해 설계되었으며 재암호화 횟수에 대해 제한을 두고 있다. 즉, 재암호화된 암호문을 다시 재암호화 할 수 없는 일회성을 지닌다. Green과 Ateniese의 접근 방식은 비상호성 문제를 해결하는 좋은 형태이나 (본 논문에서 보이는 바와 같이) 사용자 비밀키 노출과 관련한 심각한 보안 취약성을 가지고 있다. 이러한 보안 취약성은 난수를 대리자의 아이디로 암호화하고 그 난수를 이용하여 위임자의 비밀키를 숨기는 방식으로 재암호화키를 생성하는 구조에 기인한다. (알려진 대부분의 비상호성을 갖는 재암호화 기법들은 유사한 구조를 갖는다.) 하지만, 재암호화키 생성을 위한 비상호성 제공을 위해 이러한 구조는 필연적인 요구사항으로 보이며 따라서 키 노출 공격에 대한 안전성과 비상호성은 양립할 수 없는 것처럼 간주된다.

2. 논문의 결과

본 논문에서는 (앞에서 언급한 바와 같이) Green-Ateniese 기법에 대한 암호분석을 하고, [10]에서의 주장과는 다르게 Green-Ateniese 기법이 선택 암호문 공격에 대한 안전성을 충족하지 않음을 형식적으로 증명한다. 현재 선택 암호문 공격에 안전한 ID기반 재암호화 기법의 설계는 어려운 것으로 인식되고 있다. 이는 CCA 안전성은 암호문에 대한 변형 불가능성(Non-malleability)을 필요로 하지만 재암호화 기법는 재암호화 과정에서 변형성을 요구하므로 개념적인 면에서 상호적으로 배치되기 때문이다. 본 논문에서는 최초로 CCA 안전성을 제공하는 새로운 ID기반 재암호화 기법을 제안한다. 이를 위해 선택 평문 공격에 대하여 안전한 ID기반 재암호화 기법을 제안한다. 그리고 짧은 서명에 기반한 자가 인증 기법을 고안하고 이를 앞의 기법에 결합하여 선택 암호문 공격에 대하여 안전 기법을 제안한다. 두 기법들은 모두 변형된 결정적 Bilinear Diffie-Hellman(DBDH) 문제의 가정 하에서 랜덤 오라클 모델에서 고려되었다.

제안 기법들에서는 프락시와 사용자의 공모시에도 위임자 또는 대리자의 비밀키가 노출되지 않는다. 제안 기법들은 이전에 알려진 방법과는 다르게 새로운 재암호화키 생성 구조에 기반한다. 특히 사용자와 프락시 사이의 공모 공격을 방지하지 위해 재암호화키를 생성할 때 독립적인 마스터키를 이용하여 위임자의 비밀키와 대리자의 임시 암호화키를 결합한다. 이러한 설계원리를 통하여, 제안 기법은 기존 연구와 비교하여 다양

한 추가적인 이점을 제공할 수 있다. 제안 기법의 중요한 특징 중 하나는 재암호화 후 암호문의 구조가 유지되는 것이다. 따라서 이전 기법들과는 대조적으로 암호문 확장이 발생되지 않는다. 또한 재암호화의 횟수에 제한 없이 연속적인 암호문 변환이 가능하여 다중의 사용자를 위한 복호 능력 위임을 구현할 수 있다.

3 관련 연구들

[11]에서 소개된 복호화 권리를 위임하는 암호화 기법 이후로 [3]에서 처음으로 재암호화 기법이 제안되었다. 그러나 BBS 기법에서는 악의적인 프락시가 A에서 B로 암호문을 변환시키는 재암호화 키를 받게 되면 B에서 A로 암호문을 변환시키는 재암호화 키를 쉽게 도출할 수 있는 양방향성(Bidirectional)의 성질을 가진다. 양방향성은 프락시에게 추가적인 재암호화 권리를 제공하므로 바람직하지 않다. 최근에는 양방향성 대신 단방향성(Unidirectionality)의 성질을 지니는 재암호화 기법들이 주요하게 연구되고 있다[10~11]. 참고로 단일방향성을 제공하는 기법이 존재하면 양방향성을 제공하는 기법을 쉽게 구현할 수 있다. 한편, 재암호화된 암호문을 다시 재암호화 할 수 있는, 즉 재사용성(Multiple use capability)을 제공하는 기법도 연구되고 있다[10]. 효율성을 증진시키기 위한 연구도 활발히 수행되고 있다. 재암호화 시, 재암호화된 암호문의 길이는 변형되기 이전의 암호문에 비하여 증가하지 않아야 한다. CCA-안전성을 제공하는 최근 ID기반 재암호화 기법들은 재암호화시 암호문의 길이가 선형적으로 증가한다[10~11]. 최근 몇 가지 관련 기법들이 제안되고 있으나 강한 안전성을 보장하고 위의 나열한 성질들을 모두 보장하는 기법은 현재 알려지지 않고 있다.

4 논문의 구성

이 후 논문의 구성은 다음과 같다. Ⅱ장에서는 제안한 기법을 위한 배경지식을 설명한다. Ⅲ장에서는 ID기반 재암호화 기법에 대한 형식적인 모델을 기술한다. Ⅳ장에서는 기존의 재암호화 기법의 공모 공격에 대한 취약점을 분석한다. Ⅴ장에서는 ID기반 재암호화 기법들을 제안하고 CPA 및 CCA 관점에서 안전성을 증명한다. 또한 제안 기법의 유용한 특성들에 대해서 설명한다. Ⅵ장에서는 결론을 내린다.

Ⅱ. 배경 지식(Preliminaries)

본 장에서는 제안 기법의 구성 및 안전성 증명에 필요한 겹선형 함수(bilinear map) 및 이와 관련된 복잡도(complextity assumption) 가정들을 살펴본다.

겹선형 함수(Bilinear Maps). G_1과 G_2가 위수를 소수 q로 갖는 순환 군(group)이라고 하자. 군 G_1과 G_2에서 모두 이산대수문제(Discrete Logarithm Problem)가 어렵다고 가정하자. 겹선형 함수(bilinear map)은 다음과 같은 성질을 갖는 $G_1 \times G_1$에서 군 G_2 위로 맵핑되는 함수 $e : G_1 \times G_1 \rightarrow G_2$이다:

(1) 겹선형성 (Bilinearlity): 임의의 군 원소 $g \in G_1$와 $a, b \in Z_q^*$에 대하여 $e(g^a, g^b) = e(g,g)^{ab}$을 만족한다.

(2) 비소실성 (Non-degeneracy): $e(g, g) \neq 1$을 만족시키는 $g \in G_1$가 존재한다.

(3) 계산 가능성 (Computability): 임의의 $g_1, g_2 \in G_1$에 대해서 $e(g_1, g_2)$를 계산하는 효율적인 알고리즘이 존재한다.

계산적 Diffie-Hellman 문제 및 가정 (Computational Diffie-Hellman Assumption, CDH). 주어진 G_1의 생성원 g에 대해 g^a, g^b가 주어졌을 때 CDH 가정이란 g^{ab}을 의미 있는 확률로 효율적으로 계산할 수 있는 알고리즘 A이 존재하지 않음을 말한다. 알고리즘 A의 이점(advantage)는 다음과 같은 확률 값으로 정의된다.

$$\Pr\left[g^{ab} \leftarrow A(G_1, g, g^a, g^b) \mid g \leftarrow_R G_1; a, b \leftarrow_R Z_q^*\right].$$

겹선형 Diffie-Hellman 문제 및 가정 (Bilinear Diffie-Hellman Assumption, BDH). 주어진 G_1의 생성원 g에 대해 g^a, g^b, g^c가 주어졌을 때 BDH 가정이란 $e(g,g)^{abc}$을 의미 있는 확률로 효율적으로 계산할 수 있는 알고리즘 A이 존재하지 않음을 말한다. 알고리즘 A의 이점(advantage)는 다음과 같은 확률 값으로 정의된다.

$$\Pr\left[e(g,g)^{abc} \leftarrow A(G_1, g, g^a, g^b, g^c) \mid g \leftarrow_R G_1; a, b, c \leftarrow_R Z_q^*\right]$$

수정된 결정적 겹선형 Diffie-Hellman 문제 및 가

정 (Modified Decision Bilinear Diffie-Hellman Assumption, MDBDH). 주어진 G_1의 생성원 g에 대해 $g^a, g^b, g^c, g^{ad}, g^{acd}, g^{d^{-1}}, T$가 주어졌을 때 MDBDH 가정이란 $T=e(g,g)^{abc}$인지 T가 임의의 난수인지를 의미 있는 확률로 판단할 수 있는 알고리즘 A이 존재하지 않음을 말한다. 알고리즘 A의 이점(advantage)는 다음과 같은 확률 값으로 정의된다.

$$\Pr\left[1 \leftarrow A(G_1, g, g^a, g^b, g^c, g^{ad}, g^{acd}, g^{d^{-1}}, e(g,g)^{abc}) \mid g \leftarrow_R G_1; a,b,c \leftarrow_R Z_q^*\right]$$

$$-\Pr\left[1 \leftarrow A(G_1, g, g^a, g^b, g^c, g^{ad}, g^{acd}, g^{d^{-1}}, T) \mid g \leftarrow_R G_1; a,b,c \leftarrow_R Z_q^*: T \leftarrow_R G_2\right]$$

Ⅲ. 형식적 모델(Model)

본 장에서는 ID기반 재암호화 기법의 형식적 정의와 안전성 모델에 대해 설명한다. 보다 자세한 설명은 논문 [10]을 참조한다.

1. ID기반 재암호화 기법의 정의

ID기반 재암호화 기법(Identity-Based Re-encryption Scheme, IBRS)은 다음과 같은 6개의 다항식 시간(polynomial-time) 알고리즘들로 구성된다.

- $Setup(1^\lambda)$: 셋업(Setup) 알고리즘은 보안 상수 κ을 입력으로 받고 마스터 비밀키 MK와 공개 상수(parameter) PP을 출력한다.
- $KeyGen(MK, PP, ID)$: 키 생성 알고리즘은 마스터 비밀키 MK, 공개 상수 PP와 ID을 입력 받고 비밀키 SK_{ID}을 출력한다.
- $Encryption(M, PP, ID)$: 암호화 알고리즘은 메시지 M, 공개 상수 PP, identity로 ID을 입력으로 받고 메시지 M에 대한 암호문 C_{ID}을 출력한다.
- $RKGen(PP, SK_{ID_1}, ID_1, ID_2, \beta)$: 재암호화키 생성 알고리즘은 공개 상수 PP, ID_1의 비밀키 SK_{ID_1}, identity $\{ID_1, ID_2\}$와 마스터 키 β를 입력 받고 재암호화키 $RK_{ID_1 \rightarrow ID_2}$을 출력한다.
- $Re\text{-}Encryption(PP, C_{ID_1}, RK_{ID_1 \rightarrow ID_2})$: 재암호화 알고리즘은 공개 상수 PP, ID_1의 공개키로 암호화된 암호문 C_{ID_1}와 재암호화키 $RK_{ID_1 \rightarrow ID_2}$를 입력 받고 ID_2의 공개키로 암호화된 암호문 $C_{ID_2}=$을 출력한다.
- $Decryption(PP, C, SK_{ID})$: 복호화 알고리즘은 공개 상수 PP, 암호문 C와 비밀키 SK_{ID}을 입력 받고 암호문에 대한 평문 M을 출력한다.

2. ID기반 재암호화 기법의 안전성 모델

ID기반 재암호화 기법의 안전성을 정의하기 위해 챌린저(challenger) C와 공격자(adversary) 알고리즘 F 사이의 상호적으로 수행되는 다음의 게임을 고려한다.

- $Setup(1^\lambda)$: 챌린저 C는 κ를 선택하고 마스터 비밀키 MK와 공개 상수 PP을 얻기 위해 $Setup$ 알고리즘을 실행한다. 챌린저 C는 공개 상수 PP을 공격자 F에게 준다.
- $Query\ phase\ 1$: 공격자 F는 챌린저 C에게 다음과 같은 질의를 선택적으로 한다.
  - $Extract(ID)$: 공격자가 ID에 대한 비밀키를 요청할 때, C는 비밀키 SK_{ID}을 생성하여 준다.
  - $RKExtract(ID_1, ID_2)$: 공격자가 ID_1의 암호문에서 ID_2의 암호문으로 변경하는 재암호화 키를 요청할 때, 챌린저 C는 재암호화 키 $RK_{ID_1 \rightarrow ID_2}$를 생성하여 준다.
  - $Re\text{-}Encryption(ID_1, ID_2, C_{ID_1})$: 공격자가 ID_1의 암호문에서 ID_2의 암호문으로 변경하는 재암호화를 요청할 때, 챌린저 C는 C_{ID_2}를 생성하여 준다.
  - $Decryption(ID, C_{ID})$: 공격자가 암호문 C_{ID}에 대한 평문을 요청할 때, 챌린저 C는 평문 M을 생성하여 준다.
- $Challenge$: 공격자는 챌린저 C에게 챌린지 메시지 (ID^*, m_0, m_1)를 준다. 만약 다음의 경우가 이전에 발생하지 않았다면, 챌린저 C는 $b \in \{0,1\}$을 선택하고 $C^* = Encrypt(u, ID^*, M_b)$를 생성하여 공격자에게 준다.
  - $Extract(ID^*)$가 $Query\ phase\ 1$에서 일어난 경우

- 모든 ID'에 대하여 $RKExtract(ID^*, ID')$와 $Extract(ID')$가 *Query phase* 1에서 발생했을 경우

- *Query phase* 2 : A는 *Query phase* 1에서와 같은 질의를 다음의 경우를 제외하고 계속하여 요청할 수 있다.

- $Extract(ID^*)$;
- $RKExtract(ID^*, ID')$, $Extract(ID')$;
- $RKExtract(ID^*, ID')$, $Decryption(ID', C_{ID'})$;
- $Re\text{-}Encryption(ID^*, ID', C^*)$, $Extract(ID')$;
- $Decryption(ID^*, C_{ID^*})$;
- $Decryption(ID', C_{ID'})$,

$$Re\text{-}Encryption(ID^*, ID', C^*);$$

- *Guess* : 공격자 F는 $b' \in \{0,1\}$를 출력하고 $b' = b$인 경우 F가 위의 게임에서 이긴다.

정의 1. 어떠한 다항 함수 시간에 동작되는 공격자 F에 대하여도 F가 위에서 정의된 게임에서 이길 성공 확률이 무시할(negligible)만 하다면 주어진 ID기반 재암호화 기법은 IND-prID-ATK 관점에서 안전하고 정의한다. 여기서 ATK는 공격유형에 따라 CPA(Chosen Plaintext Attack) 또는 CCA(Chosen Ciphertext Attack)로 정해진다.

Ⅳ. Green-Ateniese ID기반 재암호화 기법의 분석

최근 Green과 Ateniese는 재암호화 단방향성(Unidirectionality) 문제와 재암호화 키 생성의 비상호성(Non-interactivity) 문제를 해결하는 ID기반 재암호화 기법을 제안하였다[10]. 제안 기법은 선택 암호문 공격에 안전한, CCA-안전성(Adaptive Chosen Ciphertext Security)을 제공하기 위해 설계되었다. 이러한 안전성을 위해 제안 기법은 재암호화 횟수에 대해 제한을 두어 재암호화된 암호문을 다시 재암호화 할 수 없는 일회성(Single use)의 성질을 지닌다. 하지만 본 장에서 보이는 바와 같이 그들의 접근 방식에는 중요한 보안 취약점이 존재한다. 다음에서는 Green-Ateniese 기법을 설명하고 공모 공격의 키 사용자 비밀키 노출에 대한 취약점에 대해 분석한다.

1. Green-Ateniese ID기반 재암호화 기법[10]

• $Setup(1^\kappa)$. 보안 상수 $\kappa \in Z^+$를 입력으로 받고 다음과 같이 작동한다.

겹선형 함수와 관련된 순환군(cyclic group)들 (G_1, G_2, e)을 생성한다. 이 때 G_1과 G_2는 소수 q을 위수로 갖으며, $e : G_1 \times G_1 \rightarrow G_2$는 어드미서블 겹선형 함수(admissible bilinear map)이다. 다음으로, G_1에서 임의의 생성원 g와 암호학적인 해쉬 함수들 $H_i : \{0,1\}^* \rightarrow G_1^*$ (i=1,2,3), $H_4 : G_2^* \times \{0,1\}^n \rightarrow Z_q^*$, $H_5 : G_2 \rightarrow \{0,1\}^n$ 을 선택한다. 임의의 난수 $s \in Z_q^*$를 선택하고 g^s를 계산한다. 그리고 공개 상수 PP와 마스터 비밀키인 MK를 다음과 같이 설정한다:

$PP = (G_1, G_2, e, q, g, g^s, H_1, H_2, H_3, H_4, H_5)$,
$MK = s$.

• $Extract(MK, PP, ID)$. 마스터 비밀키 MK, 공개 상수 PP와 $ID \in \{0,1\}^*$을 입력 받는다. ID에 대하여 비밀 복호화 키 $SK_{ID} = H_1(ID)^s$을 생성한다.

• $Encryption(M, PP, ID)$. 메시지 M, 공개 상수 PP와 ID을 입력 받는다. 임의의 $\sigma \in G_2$를 선택하고 $r = H_4(\sigma, M)$을 계산한다. $c' = (A, B, C) = (g^r, \sigma \cdot e(g^s, PK_{ID})^r, M \oplus H_5(\sigma))$을 계산한다. 그리고 $h = H_3(ID \| c') \in G_1$를 생성한 후 $S = h^r$를 계산한다. 마지막으로 암호문 $C_{ID} = \langle S, A, B, C \rangle$을 출력한다.

• $RKGen(PP, SK_{ID_1}, ID_1, ID_2)$. 공개 상수 PP, 위임자 아이디 ID_1, 대리자 아이디 ID_2, 비밀키 SK_{ID_1}를 입력 받고 $RK_{ID_1 \rightarrow ID_2}$를 다음과 같이 생성한다. 임의의 난수 $N \in \{0,1^n\}$를 선택하고 $K = e(SK_{ID_1}, H_1(ID_2))$을 계산한다. 재암호화키 $RK_{ID_1 \rightarrow ID_2} = \langle N, H_2(K \| ID_1 \| ID_2 \| N) \cdot SK_{ID_1} \rangle$을 계산한다.

• $Re\text{-}Encryption(PP, C_{ID_1}, RK_{ID_1 \rightarrow ID_2})$. 공개 상수 PP, 암호문 $C_{ID_1} = (S, A, B, C)$, 재암호화키 $RK_{ID_1 \rightarrow ID_2} = \langle N, H_2(K \| ID_1 \| ID_2 \| N) \cdot SK_{ID_1} \rangle$을 입력 받고 ID_2에 대한 암호문 C_{ID_2}을 다음과 같이 생성한다.

- $h = H_2(ID \| \langle A, B, C \rangle)$을 생성한 후

$e(g,S)=e(h,A)$인지 검증한다. 만약 등식이 성립하지 않는다면 $\perp$을 반환하고 등식이 성립한다면 다음의 재암호화 과정을 수행한다.

- 임의의 $t\in Z_q^*$을 선택하고 다음을 계산한다.

$$B'=B\cdot e(g^t,S)\cdot e(A,R\cdot h^t)^{-1}$$

- 재암호화 된 암호문 $C_{ID_2}=(A,B',C,ID_1,N)$을 생성한다.

• $Decryption(PP, C_{ID}, SK_{ID})$. 공개 상수 PP, 암호문 C_{ID}와 비밀키 SK_{ID}을 입력 받는다. 암호문 C_{ID}의 형태 (또는 레벨)에 따라 다음의 두 가지 방식 중 한 방법을 따른다.

• Level-1 복호화. ($Encryption$ 알고리즘에 의해 처음 생성된) 암호문 $C_{ID}=\langle S,A,B,C\rangle$의 복호화 과정은 다음과 같다.

- $h=H_2(ID\|\langle A,B,C\rangle)$을 계산한다. 임의의 $t\in Z_q^*$을 선택하고 $\sigma'=B/\dfrac{e(A,SK_{ID}\cdot h^t)}{e(g^t,S)}$을 계산한다.

- $M'=C\oplus H_5(\sigma')$을 계산하여 메시지를 복호화한 후, $r'=H_4(\sigma',M')$을 생성한다.

- $S=h^{r'}$와 $A=g^{r'}$가 등호가 성립하는지 확인한다. 만약 등호가 성립하면 M'을 반환하고 그렇지 않다면 $\perp$을 반환한다.

• Level-2 복호화. 재암호화된(Second-Level) 암호문 $C_{ID}=\langle A,B,C,ID_{src},N\rangle$의 복호화 과정은 다음과 같다.

- $K=e(H_1(ID_{src}),SK_{ID})$을 계산한 후 $\sigma'=B\cdot e(A,H_2(K\|ID_{src}\|ID\|N))$을 계산한다.

- $M'=C\oplus H_5(\sigma')$을 계산하여 메시지를 복호화한 후, $r'=H_4(\sigma',M')$을 생성한다.

- $A=g^{r'}$가 등호가 성립하는지 확인한다. 만약 등호가 성립하면 M'을 반환하고 그렇지 않다면 $\perp$을 반환한다.

2. 공모 공격에 대한 보안 취약성

위 재암호화 기법은 결정적 BDH(Bilinear Diffie-Hellman) 가정하에서 CCA 안전성을 가짐이 증명되었다. 보다 정확히 말해서, [10]의 IND-prID-CCA의 형식적 정의를 충족함이 증명되었다. 하지만 본 절에서는 앞의 정의에서 유효한 키 노출 공격을 제시하고 증명이 유효하지 않음을 보인다.

Green-Ateniese 재암호화 기법에 대한 키 복구 공격

일반성을 잃지 않고, 프락시(proxy), 위임자 ID_1, 대리자 ID_2를 고려한다. 위임자 ID_2는 비밀키 생성과정을 거쳐 $SK_{ID_2}=H_1(ID_2)^s$을 얻는다. 프락시(proxy)는 재암호화키 생성과정을 통하여 재암호화 키 $RK_{ID_1\to ID_2}=\langle N, H_2(K\|ID_1\|ID_2\|N)\cdot SK_{ID_1}\rangle$를 얻는다. 여기서 N은 $\{0,1\}^{n(k)}$에서 임의로 선택한 수이고 $SK_{ID_1}=H_1(ID_1)^s$, $K=e(H_1(ID_1),H_1(ID_2))^s$이다.

악의적인 프락시와 대리자 ID_2는 공모를 통하여 위임자 ID_1의 비밀키 SK_{ID_1}를 다음과 같이 계산한다.

(1)먼저 SK_{ID_2}를 이용하여 $K=e(H_1(ID_1),SK_{ID_2})$
$=e(H_1(ID_1),H_1(ID_2))^s$를 계산한다.

(2) 등록된 재암호화 키 $RK_{ID_1\to ID_2}$와 위에서 계산된 (위임자와 대리자 사이에 공유된 키) K를 이용하여 SK_{ID_1}을 다음과 같이 계산한다.

$$SK_{ID_1}=(H_2(K\|ID_1\|ID_2\|N)\cdot SK_{ID_1})\cdot H_2(K\|ID_1\|ID_2\|N)^{-1}$$

ID_1의 비밀키 SK_{ID_1}를 얻음으로써 공격자는 ID_1의 공개키(아이디)로 암호화된 모든 암호문들을 쉽게 복호화 할 수 있다. 프락시와 대리자의 공모를 통해서도 ID_1의 암호문 중 일부를 복호화 할 수 있지만 ID_1의 비밀키의 노출은 이와 같은 복호화 권리 이상의 매우 심각한 보안 취약점을 나타낸다.

한 예로, 공격자는 프락시와 대리자의 공모를 통해서는 허가되지 않은 복호화 능력을 가지게 된다. 이를 명확히 보이기 위하여, $C_{ID^*\to ID_1}$를 ID^*의 공개키로 암호화된 암호문을 ID_1의 암호문으로 재암호화한 (Second-level) 암호문이라 가정하자. $C_{ID^*\to ID_1}$를 다음과 같이 나타내자.

$$C_{ID^*\to ID_1}=(A,B,C,ID_1,N^*)$$
$$=(g^r,\sigma\cdot e(g^r,W)^{-1},M\oplus H_5(\sigma),ID_1,N^*)$$

이때 $W = H_2(e(H_1(ID^*), H_1(ID_1))^s \| ID^* \| ID_1 \| N^*)$, $\sigma \in G_2$, $r = H_4(\sigma, M)$이다. 기법에서는 재암호화가 한번만 허용되므로, 프락시와 대리자의 공모를 통해 재암호화된 암호문를 복호화할 수 없다. 하지만 ID_1의 비밀키 SK_{ID_1}을 얻은 공격자는 재암호화된 암호문 $C_{ID^* \to ID_1}$를 다음의 과정을 통하여 쉽게 복호화 할 수 있다.

- $K^* = e(SK_{ID_1}, H_1(ID^*)) = e(H_1(ID_1), H_1(ID^*))^s$ 을 계산한다.
- $W = H_2(K^* \| ID^* \| ID_1 \| N^*)$을 계산한다.
- $M = C \oplus H_5(B \cdot e(A, W))$
  $= M \oplus H_5(\sigma) \oplus H_5(\sigma \cdot e(g^r, W)^{-1} \cdot e(g^r, W))$

을 계산한다.

부가적으로, 공격자는 사용자 ID_1의 비밀키 SK_{ID_1}을 가지고 있으므로 재암호화 키 $RK_{ID_i \to ID_1}$을 이용하여 다른 정직한 사용자들의 비밀키 SK_{ID_i}을 계산해 낼 수 있다. 따라서 공격자는 그 후에 계속해서 재암호화한(Second-level) 암호문 $C_{ID_i \to ID_1}$에 대하여 유사한 공격을 시도할 수 있다.

위에서 제시한 공격은 Green-Ateniese 재암호화 기법의 안전성 증명을 위해 고려된 [10]의 형식적 보안 모델에서 유효함을 쉽게 보일 수 있다. 이는 (재암호화 횟수의 제약으로) 재암호화 알고리즘은 재암호화한(Second-level) 암호문 상에서 더 이상 동작하지 않으므로 첼린지 암호문에 대한 어떠한 첼린지 유도 암호문(Challenge Derivatives)*을 생성할 수 없기 때문이다.

V. 강한 안전성을 갖는 ID기반 재암호화 기법들

Green과 Ateniese의 접근 방식은 (재암호화 키 생성의) 비상호성(Non-interactivity) 문제를 해결하는 좋은 형태이지만 위에서 설명한 바와 같이 사용자 비밀키 노출과 관련한 심각한 보안 취약성을 가지고 있다. 사실상, 이러한 보안 취약성은 난수를 대리자의 아이디로 암호화하고 그 난수를 이용하여 위임자의 비밀키를 감추는 방식으로 재암호화키를 생성하는 구조에 기인한다. (비상호성을 갖는 대부분의 재암호화 기법들은 유사한 구조를 갖는다.) 하지만 비상호성(Non-interactivity) 문제를 해결하기 위해, 이러한 방식은 필연적으로 요구되는 것으로 보이며 키 노출 공격에 안전하며 비상호성을 갖는 ID기반 재암호화 기법의 구성은 새로운 접근 방식이 요구된다.

본 장에서는 공모공격에 완전한 보안성을 갖는 두 가지 ID기반 재암호화 기법들을 제안한다. 제안 기법들은 비밀키 노출 약점을 극복하기 위해 독립적인 마스터 비밀값을 이용하여 재암호화 키를 생성한다. 첫 번째 제안 기법 IDPRE-1은 CPA-안전성을 가지며 두 번째 제안 기법 IDPRE-2은 보다 강한 CCA-안전성을 갖는다. 현재까지 알려진 기법들과는 대조적으로, 두 제안 기법들의 큰 장점은 재암호화 횟수의 제한이 없으며, 재암호화 과정에서 암호문 확장이 발생하지 않는다는 것이다.

1. CPA-안전한 ID기반 재암호화 기법(IDPRE-1)

IDPRE-1은 다음과 같이 셋업, 비밀키 생성, 암호화, 재암호화키 생성, 재암호화, 복호화의 6개의 알고리즘들로 구성된다.

• $Setup(1^\kappa)$. 셋업 알고리즘은 보안 상수 $\kappa \in Z^+$를 입력으로 받고 다음과 같이 작동한다.

- 적절한 크기의 소수 q를 생성하고 (G_1, G_2, e)을 생성한다. 이 때 G_1과 G_2는 소수 q을 위수로 갖는 순환군들이고, $e : G_1 \times G_1 \to G_2$는 어드미서블 겹선형함수(admissible bilinear map)이다. G_1의 임의의 생성원 g와 암호학적인 해쉬 함수 $H_1 : \{0,1\}^* \to G_1^*$을 선택한다.
- 임의의 난수 $\alpha, \beta \in Z_q^*$를 선택하고 $g_0 = g^\alpha$와 $g_1 = g^{\alpha\beta}$를 계산한다. 공개 상수 PP와 마스터 비밀키인 MK를 다음과 같이 설정한다.

$$PP = (G_1, G_2, e, q, g, g_0, g_1, H_1), \quad MK = (\alpha, \beta).$$

• $Extract(MK, PP, ID)$. 비밀키 생성 알고리즘은 마스터 비밀키 MK, 공개 상수 PP와 $ID \in \{0,1\}^*$을 입력 받는다. $PK_{ID} = H_1(ID)$를 계산하고 ID에 대응하는 비밀키 $SK_{ID} = PK_{ID}^\alpha$을 출력한다.

• $Encryption(M, PP, ID)$. 암호화 알고리즘은 메

* 논문 [10]에서 정의한 ID기반 재암호화 기법의 보안 모델에서는 프락시를 통한 단순한 복호능력을 배제하기 위하여 첼린지 암호문의 유도 암호문들을 정의하고 이에 대한 공격자의 질의를 제한하고 있다.

시지 M, 공개 상수 PP와 ID을 입력 받는다. 임의의 난수 $r \in Z_q^*$을 선택한 후 메시지 M에 대한 암호문을 다음과 같이 계산한다.

$$C_{ID} = (A, B, C) = (g^r,\ g_1^r,\ M \cdot e(g_0, PK_{ID})^r).$$

• $RKGen(PP, SK_{ID_1}, ID_1, ID_2, \beta)$. 공개 상수 PP, 위임자 아이디 ID_1, 대리자 아이디 ID_2, 비밀키 SK_{ID_1}, 그리고 $RKGC$ 마스터키 β로부터 암호화 키 $RK_{ID_1 \to ID_2} = (RK_1, RK_2)$을 다음과 같이 생성한다. 위임자 ID_1과 $RKGC$ 사이의 통신은 인증 채널을 가정한다. 알려진 ID기반 서명 기법을 이용하여 인증 채널은 쉽게 구현될 수 있다.

- ID_1은 임의로 난수 $\delta_1 \in Z_q^*$를 선택하고 $\gamma = g^{-\delta_1}$를 계산하고 $RKGC$에 (ID_1, ID_2, γ)를 보낸다.

- $RKGC$는 (ID_1, ID_2, γ)를 받은 후 γ이 순환군 G_1의 원소인지 확인한다. 만일 확인이 유효하면 난수 $\delta_2 \in Z_q^*$ 를 선택하고 $PK_{ID_2} = H_1(ID_2)$를 계산한 후 $RK_1 = (\gamma \cdot g^{-\delta_2} \cdot PK_{ID_2})^{\beta^{-1}} = (g^{-(\delta_1+\delta_2)} \cdot PK_{ID_2})^{\beta^{-1}}$와 $\theta = g^{\alpha\delta_2}$를 생성한다. (RK_1, θ)을 ID_1에게 전송한다.

- ID_1은 (RK_1, θ)를 받은 후, $RK_2 = \theta \cdot g^{\alpha\delta_2} \cdot SK_{ID_1}^{-1} = g^{\alpha(\delta_1+\delta_2)} \cdot SK_{ID_1}^{-1}$을 생성한다.

- 재암호화키 $RK_{ID_1 \to ID_2} = (RK_1, RK_2) = (g^{-\delta\beta^{-1}} \cdot PK_{ID_2}^{\beta^{-1}},\ g^{\alpha\delta} \cdot SK_{ID_1}^{-1})$을 출력한다. 여기서 $\delta = \delta_1 + \delta_2$이다.

• $Re\text{-}Encryption(PP, C_{ID_1}, RK_{ID_1 \to ID_2})$. 재암호화 알고리즘은 공개 상수 PP, 암호문 $C_{ID_1} = (A, B, C)$, 재암호화키 $RK_{ID_1 \to ID_2} = (RK_1, RK_2)$를 입력 받는다. 임의의 난수 $t \in Z_p^*$를 선택하고 ID_2에 대한 암호문 $C_{ID_2} = (\tilde{A}, \tilde{B}, \tilde{C})$을 다음과 같이 계산한다.

$$\begin{aligned}
&\tilde{A} = A \cdot g^t = g^{r+t},\ \tilde{B} = B \cdot g_1^t = g^{\alpha\beta(r+t)},\\
&\tilde{C} = C \cdot e(B, RK_1) \cdot e(A, RK_2) \cdot e(g_0, PK_{ID_2})^t\\
&\quad = M \cdot e(g^\alpha, PK_{ID_1})^r \cdot e(g^{\alpha\beta\gamma}, g^{-\delta\beta^{-1}} \cdot PK_{ID_2}^{\beta^{-1}}) \cdot\\
&\qquad e(g^r, g^{\alpha\delta} \cdot SK_{ID_1}^{-1}) \cdot e(g^\alpha, PK_{ID_2})^t\\
&\quad = M \cdot e(g^\alpha, PK_{ID_1})^r \cdot e(g^{\alpha\gamma}, g^{-\delta}) \cdot e(g^{\alpha\gamma}, PK_{ID_2})\\
&\qquad \cdot e(g^r, g^{\alpha\delta}) \cdot e(g^r, SK_{ID_1}^{-1}) \cdot e(g^\alpha, PK_{ID_2})^t\\
&\quad = M \cdot e(g^\alpha, PK_{ID_2})^r \cdot e(g^\alpha, PK_{ID_2})^t\\
&\quad = M \cdot e(g^\alpha, PK_{ID_2})^{r+t}
\end{aligned}$$

• $Decryption(PP, C_{ID}, SK_{ID})$. 복호화 알고리즘은 공개 상수 PP, 암호문 $C_{ID} = (A, B, C)$와 비밀키 SK_{ID}을 입력 받는다. 다음과 같이 메시지를 복호한다.

$$M = C \cdot e(A, SK_{ID}^{-1}).$$

Green-Ateniese 기법과는 다르게 위 제안 기법에서 생성된 First Level 암호문(최초 암호문)과 Second Level 암호문(재암호화된 암호문)은 구조가 동일하다. 따라서 복호화 시 별도의 구분이 필요하지 않으며 암호문 확장이 발생하지 않는다. 또한 ID에 상관없이 여러 번의 재암호화를 수행할 수 있다.

위의 제안 기법은 정확성을 가짐을 다음과 같이 쉽게 보일 수 있다.

- 재암호화 과정. 주어진 암호문 $C_{ID_1} = (A, B, C) = (g^r,\ g^{\alpha\beta r},\ M \cdot e(g^\alpha, PK_{ID_1})^r)$와 재암호화키 $RK_{ID_1 \to ID_2} = (g^{-\delta\beta^{-1}} \cdot PK_{ID_2}^{\beta^{-1}},\ g^{\alpha\delta} \cdot SK_{ID_1}^{-1})$에 대하여 $C_{ID_2} = (g^{r+t}, g^{\alpha\beta(r+t)}, M \cdot e(g^\alpha, PK_{ID_2})^{r+t})$을 다음의 과정을 통하여 생성할 수 있다.

$$\begin{aligned}
&\tilde{A} = A \cdot g^t = g^{r+t},\ \tilde{B} = B \cdot g_1^t = g^{\alpha\beta(r+t)}\\
&\tilde{C} = C \cdot e(B, RK_1) \cdot e(A, RK_2) \cdot e(g_0, PK_{ID_2})^t\\
&\quad = M \cdot e(g^\alpha, PK_{ID_1})^r \cdot e(g^{\alpha\beta\gamma}, g^{-\delta\beta^{-1}} \cdot PK_{ID_2}^{\beta^{-1}}) \cdot\\
&\qquad e(g^r, g^{\alpha\delta} \cdot SK_{ID_1}^{-1}) \cdot e(g^\alpha, PK_{ID_2})^t\\
&\quad = M \cdot e(g^\alpha, PK_{ID_1})^r \cdot e(g^{\alpha\gamma}, g^{-\delta}) \cdot e(g^{\alpha\gamma}, PK_{ID_2})\\
&\qquad \cdot e(g^r, g^{\alpha\delta}) \cdot e(g^r, SK_{ID_1}^{-1}) \cdot e(g^\alpha, PK_{ID_2})^t\\
&\quad = M \cdot e(g^\alpha, PK_{ID_2})^r \cdot e(g^\alpha, PK_{ID_2})^t\\
&\quad = M \cdot e(g^\alpha, PK_{ID_2})^{r+t}
\end{aligned}$$

- 복호화 과정. 주어진 암호문 $C_{ID} = (A, B, C) = (g^r, g^{\alpha\beta r}, M \cdot e(g^\alpha, PK_{ID})^r)$을 다음의 계산을 통해 메시지 M을 얻을 수 있다.

$$\begin{aligned}
M &= C \cdot e(A, SK_{ID}^{-1})\\
&= M \cdot e(g^\alpha, PK_{ID})^r / e(g^r, PK_{ID}^\alpha)
\end{aligned}$$

제안한 ID기반 재암호화 기법 IDPRE-1는 랜덤 오라클(Random Oracle) 모델에서 MDBDH가정 하에서 IND-prID-CPA 관점에서 안전함을 쉽게 증명할 수 있다. 여기서는 생략한다.

2. CCA-안전한 ID기반 재암호화 기법 (IDPRE-2)

논문 [10]에서 주지한 바와 같이, 단순히 Fujisaki-Okamoto 변환 방법[4, 9, 12]을 CPA 안전한 재암호화 기법에 적용시킴으로써 CCA 안전한 재암호화 기법을 구

성할 수 없다. 이것은 재암호화 과정을 통해 공격자가 자신의 의도한 암호문을 재암호화해 적합하게 자신의 공격에 이용할 수 있기 때문이다. 본 장에서는 CCA 안전한 재암호화 기법을 구성하기 새로운 방법을 제시한다. 핵심 아이디어는 짧은 서명에 기반한 자가 인증 기법에 기반하며 이 기법과 Fujisaki-Okamoto 변환 방법을 이전 장에서 제안한 IDPRE-1 기법에 적용하여 CCA 안전성을 갖는 효율적인 ID기반 재암호화 기법 IDPRE-2를 설계한다. 제안된 기법에서 재암호화된 암호문은 본래의 암호문과 동일한 구조를 유지하며 암호문의 유효성도 전이가 된다.

IDPRE-2은 다음과 같이 셋업, 비밀키 생성, 암호화, 재암호화키 생성, 재암호화, 복호화의 6개의 알고리즘들로 구성된다.

• $Setup(1^{\kappa})$. 셋업 알고리즘은 보안 상수 $\kappa \in Z^{+}$를 입력으로 받고 다음과 같이 작동한다.

- 적절한 크기의 소수 q를 생성하고 (G_1, G_2, e)을 생성한다. 이 때 G_1과 G_2는 소수 q을 위수로 갖는 순환군들이고, $e: G_1 \times G_1 \rightarrow G_2$는 어드미서블 겹선형함수(admissible bilinear map)이다. G_1의 임의의 생성원 g와 암호학적인 해쉬 함수 $H_1: \{0,1\}^{*} \rightarrow G_1$, $H_3: G_2 \times \{0,1\}^{n} \rightarrow Z_q^{*}$, $H_4: G_2 \rightarrow \{0,1\}^{n}$, $H_5: G_1 \rightarrow G_1$를 선택한다.

- 임의의 난수 $\alpha, \beta \in Z_q^{*}$를 선택하고 $g_0 = g^{\alpha}$와 $g_1 = g^{\alpha\beta}$를 계산한다. 공개 상수 PP와 마스터 비밀키인 MK를 다음과 같이 설정한다.

$$PP = (G_1, G_2, e, q, g, g_0, g_1, H_1, H_3, H_4, H_5),$$
$$MK = (\alpha, \beta).$$

• $Extract(MK, PP, ID)$. 비밀키 생성 알고리즘은 마스터 비밀키 MK, 공개 상수 PP와 $ID \in \{0,1\}^{*}$을 입력 받는다. $PK_{ID} = H_1(ID)$를 계산하고 ID에 대응하는 비밀키 $SK_{ID} = PK_{ID}^{\alpha}$을 출력한다.

• $Encryption(M, PP, ID)$. 암호화 알고리즘은 메시지 M, 공개 상수 PP와 ID을 입력 받는다. 메시지 M에 대한 암호문을 다음과 같이 계산한다. 임의의 $\sigma \in G_2$를 선택하고 $r = H_3(\sigma, M)$과

$$C_{ID} = (A, B, C, D, E)$$
$$= (g^{r}, g^{\alpha\beta\gamma}, \sigma \cdot e(g^{\alpha}, PK_{ID})^{r}, M \oplus H_4(\sigma), H_5(g^{r})^{r})$$

을 계산한다.

• $RKGen(PP, SK_{ID_1}, ID_1, ID_2, \beta)$. 공개 상수 PP, 위임자 아이디 ID_1, 대리자 아이디 ID_2, 비밀키 SK_{ID_1}, 그리고 $RKGC$ 마스터 키 β로부터 암호화 키 $RK_{ID_1 \rightarrow ID_2} = (RK_1, RK_2)$을 다음과 같이 생성한다. 위임자 ID_1과 $RKGC$ 사이의 통신은 인증 채널을 가정한다. 알려진 ID기반 서명 기법을 이용하여 인증 채널은 쉽게 구현될 수 있다.

- ID_1은 임의로 난수 $\delta_1 \in Z_q^{*}$를 선택하고 $\gamma = g^{-\delta_1}$를 계산하고 $RKGC$에 (ID_1, ID_2, γ)를 보낸다.

- $RKGC$는 (ID_1, ID_2, γ)를 받은 후 γ이 순환군 G_1의 원소인지 확인한다. 만일 확인이 유효하면 임의의 난수 $\delta_2 \in Z_q^{*}$ 를 선택하고 $PK_{ID_2} = H_1(ID_2)$를 계산한다. $RK_1 = (\gamma \cdot g^{-\delta_2} \cdot PK_{ID_2})^{\beta^{-1}} = (g^{-(\delta_1+\delta_2)} \cdot PK_{ID_2})^{\beta^{-1}}$와 $\theta = g^{\alpha\delta_2}$를 생성한다. (RK_1, θ)을 ID_1에게 전송한다.

- ID_1은 (RK_1, θ)를 받은 후, $RK_2 = \theta \cdot g^{\alpha\delta_2} \cdot SK_{ID_1}^{-1} = g^{\alpha(\delta_1+\delta_2)} \cdot SK_{ID_1}^{-1}$을 생성한다.

- 재암호화키 $RK_{ID_1 \rightarrow ID_2} = (RK_1, RK_2) = (g^{-\delta\beta^{-1}} \cdot PK_{ID_2}^{\beta^{-1}}, g^{\alpha\delta} \cdot SK_{ID_1}^{-1})$을 출력한다. 여기서 $\delta = \delta_1 + \delta_2$이다.

• $Re\text{-}Encryption(PP, C_{ID_1}, RK_{ID_1 \rightarrow ID_2})$. 재암호화 알고리즘은 공개 상수 PP, 암호문과 재암호화 키 $C_{ID_1} = (A, B, C, D, E)$, $RK_{ID_1 \rightarrow ID_2} = (RK_1, RK_2)$를 입력 받는다. 프락시는 다음과 같은 유효성 체크를 한다: $e(H_5(g^{r})^{r}, g) = e(H_5(A), A)$. 등식이 성립하면 ID_2에 대한 암호문 $C_{ID_2} = (\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}, \tilde{E})$을 다음과 같이 계산한다. 그렇지 않으면, $\perp$을 반환한다.

$$\tilde{A} = A, \ \tilde{B} = B,$$
$$\tilde{C} = C \cdot e(B, RK_1) \cdot e(A, RK_2) = \sigma \cdot e(g^{\alpha}, PK_{ID_2})^{r}$$
$$\tilde{D} = D. \ \tilde{E} = E.$$

• $Decryption(PP, C_{ID}, SK_{ID})$. 복호화 알고리즘은 공개 상수 PP, 암호문 $C_{ID} = (A, B, C, D, E)$와 비밀키 SK_{ID}을 입력 받는다. 다음이 성립하는지 검증

한다: $e(H_5(g^r)^r, g) = e(H_5(A), A)$. 등식이 성립하면 다음과 같이 메시지를 복호화한다.

- 먼저 $\sigma = C \bullet e(A, SK_{ID}^{-1})$를 계산한다.
- $M = D \oplus H_4(\sigma)$ 를 계산한다.
- $r' = H_3(\sigma, M)$ 계산한다. $A = g^{r'}$ 을 계산한 후 등호가 성립하면 메시지 M을 출력한다. 성립하지 않으면 $\perp$을 반환한다.

위의 제안 기법은 정확성을 가짐을 다음과 같이 쉽게 보일 수 있다.

- **재암호화** 과정. 주어진 암호문과 재암호화키 $C_{ID_1} = (g^r, g^{\alpha\beta\gamma}, \sigma \bullet e(g^{\alpha}, PK_{ID_1})^r, M \oplus H_4(\sigma), H_5(g^r)^r)$, $RK_{ID_1 \to ID_2} = (g^{-\delta\beta^{-1}} \bullet PK_{ID_2}^{\beta^{-1}}, g^{\alpha\delta} \bullet SK_{ID_1}^{-1})$에 대하여 올바른 재암호화된 암호문 $\widetilde{C_{ID_2}} = (\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}, \widetilde{E}) = (g^r, g^{\alpha\beta\gamma}, \sigma \bullet e(g^{\alpha}, PK_{ID_2})^r, M \oplus H_4(\sigma), H_5(g^r)^r)$을 다음의 과정을 통하여 생성할 수 있다.

$$\begin{aligned}\widetilde{C} &= C \bullet e(B, RK_1) \bullet e(A, RK_2) \\ &= \sigma \bullet e(g^{\alpha}, PK_{ID_1})^r \bullet e(g^{\alpha\beta\gamma}, g^{-\delta\beta^{-1}} \bullet PK_{ID_2}^{\beta^{-1}}) \\ &\quad \bullet e(g^r, g^{\alpha\delta} \bullet SK_{ID_1}^{-1}) \\ &= \sigma \bullet e(g^{\alpha}, PK_{ID_1})^r \bullet e(g^{\alpha\gamma}, g^{-\delta}) \bullet e(g^{\alpha\gamma}, PK_{ID_2}) \\ &\quad \bullet e(g^r, g^{\alpha\delta}) \bullet e(g^r, SK_{ID_1}^{-1}) \\ &= \sigma \bullet e(g^{\alpha}, PK_{ID_2})^r\end{aligned}$$

- **복호화** 과정. 주어진 암호문 $C_{ID} = (g^r, g^{\alpha\beta\gamma}, \sigma \bullet e(g^{\alpha}, PK_{ID})^r, M \oplus H_4(\sigma), H_5(g^r)^r)$을 다음의 계산을 통해 메시지 M을 얻을 수 있다:

$$\sigma = C \bullet e(A, SK_{ID}^{-1}) = \sigma \bullet \frac{e(g^{\alpha}, PK_{ID})^r}{e(g^r, PK_{ID}^{\alpha})}$$

$$M = D \oplus H_4(\sigma) = M \oplus H_4(\sigma) \oplus H_4(\sigma).$$

3. 안전성 증명

정리. 제안한 ID기반 재암호화 기법 IDPRE-2는 랜덤 오라클(Random Oracle) 모델에서 MDBDH 가정 하에서 IND-prID-CCA 관점에서 안전하다.

증명. 제안 기법 IDPRE-2의 IND-prID-CCA 안전성을 의미있는(non-negligible) 확률로 깰 수 있는 알고리즘 F가 존재한다고 가정하자. 그러면 F를 이용하여 $MDBDH$ 문제를 효율적으로 해결할 수 있는 알고리즘 B가 존재함을 보일 것이다. B는 임의로 선택된 $a, b, c, d \in Z_q^*$와 $g \in G_1$에 대한 $MDBDH$ 문제 인스턴스(instance) $(G_1, q, g, g^a, g^b, g^c, g^{d^{-1}}, g^{ad}, g^{acd}, T)$을 입력 받는다. B의 목적은 $T = e(g,g)^{abc}$임을 판단하는 것이다. F의 질의에 대하여 충돌과 모순이 없는 응답을 하기 위하여 B는 초기에는 비어 있는 리스트 $L_{H_1} = \{\langle ID_i, h, z_i, \alpha_i \rangle\}$, $L_{H_3} = \{\langle \sigma, M, r \rangle\}$, $L_{H_4} = \{\langle \sigma, y \rangle\}$, $L_{H_5} = \{\langle \pi, v \rangle\}$을 관리한다. B는 F를 하위 루틴으로 실행하고 F의 공격 환경을 다음과 같이 시뮬레이션 한다. 여기서는 해쉬 함수의 충돌쌍이 발생할 확률은 무시할 만(negligible) 하므로 고려하지 않는다.

- *셋업(Setup).* 공개 상수를 생성하기 위해 B는 $g_0 = g^{\alpha} = g^a$, $g_1 = g^{\alpha\beta} = g^{ad}$로 설정하고 IDPRE-2의 공개 상수 $(G_1, G_2, e, q, g, g_0, g_1, H_1, H_3, H_4, H_5)$을 F에게 준다.
- *질의(Queries).* B는 F의 오라클 질의들의 응답을 다음과 같이 시뮬레이션 한다.

$-H_1 - query$: F가 ID에 대한 H_1 질의를 요청할 때, 만약 리스트 L_{H_1}에 $\langle ID_i, h, z_i, \alpha_i \rangle$가 포함되어 있다면, B는 h를 반환한다. 그렇지 않으면 B는 임의의 코인 $\alpha \in \{0,1\}$와 $z_i \in Z_q^*$을 선택한 후 $\alpha = 0$이면 $h = (g^b)^{z_i}$을 계산하고 $\alpha = 1$이면 $h = g^{z_i}$을 계산하여 h를 반환한 후 L_{H_1}에 $\langle ID_i, h, z_i, \alpha_i \rangle$를 추가한다.

$-H_3 - query$: F가 σ, M에 대한 H_3 질의를 요청할 때, 만약 리스트 L_{H_3}에 $\langle \sigma, M, r \rangle$가 포함되어 있다면, B는 r를 반환한다. 그렇지 않으면 B는 임의의 $r \in Z_q^*$을 선택하여 r을 반환한 후 L_{H_3}에 $\langle \sigma, M, r \rangle$를 추가한다.

$-H_4 - query$: F가 σ에 대한 H_4 질의를 요청할 때, 만약 리스트 L_{H_4}에 $\langle \sigma, y \rangle$가 포함되어 있다면, B는 y를 반환한다. 그렇지 않으면 B는 임의의 $y \in Z_q^*$을 선택하여 y을 반환한 후 L_{H_4}에 $\langle \sigma, y \rangle$를 추가한다.

$-H_5 - query$: F가 π에 대한 H_5 질의를 요청할 때, 만약 리스트 L_{H_5}에 $\langle \pi, v \rangle$가 포함되어 있다면, B는 u를 반환한다. 그렇지 않으면 B는 임의의 $v \in Z_q^*$을 선택하여 $u = g_0^v$을 반환한 후 L_{H_5}에 $\langle \pi, v \rangle$를 추가한다.

$-Extract(ID_i)$: F는 ID_i 대한 비밀키를 요청할

때 B는 먼저 ID_i에 대한 해쉬값을 얻기 위하여 위의 $H_1-query$을 위한 목록 L_{H_1}을 참조한 후 아이디에 대응하는 순서쌍 $\langle ID_i, h, z_i, \alpha_i\rangle$을 찾는다. (여기서는 $Extract(ID_i)$ 질의를 던지기 전에 ID_i에 대한 $H_1-query$가 이미 던져졌다고 가정한다) 만일 $\alpha_i=0$이라면, B는 시뮬레이션을 중단한다. 그렇지 않다면, 즉 $\alpha_i=1$이면, B는 $(g^a)^{z_i}$를 F에게 반환한다.

-$RKExtract(ID_i, ID_j)$: 공격자 F가 ID_1의 암호문에서 ID_2의 암호문으로 변경하는 재암호화 키를 요청할 때, B는 ID_i와 ID_j에 대한 해쉬값을 얻기 위하여 위의 $H_1-query$을 위한 목록 L_{H_1}을 참조한 후 아이디에 대응하는 순서쌍 $\langle ID_i, h, z_i, \alpha_i\rangle$와 $\langle ID_j, h, z_j, \alpha_j\rangle$을 찾는다. B는 첼린지 ID을 위한 부가적인 지수값 $w\in Z_q^*$을 선택하고 저장한다.

(1) $Case\,1$: $\alpha_i=1$ 이고 $\alpha_j=1$인 경우, B는 임의의 $k\in Z_q^*$를 선택하고 $RK_1=(g^{d^{-1}})^{-k}\cdot(g^{d^{-1}})^{z_j}$, $RK_2=(g^a)^k\cdot(g^{-a})^{z_i}$를 생성하여 F에게 반환한다.

(2) $Case\,2$: $\alpha_i=1$ 이고 $\alpha_j=0$인 경우, B는 임의의 $k\in Z_q^*$를 선택하고 $RK_1=(g^{d^{-1}})^{-k}\cdot(g^{d^{-1}})^{\omega}$, $RK_2=(g^a)^k\cdot(g^{-a})^{z_i}$를 생성하여 F에게 반환한다.

(3) $Case\,3$: $\alpha_i=0$ 이고 $\alpha_j=1$인 경우, B는 임의의 $k\in Z_q^*$를 선택하고 $RK_1=(g^{d^{-1}})^{-k}\cdot(g^{d^{-1}})^{z_j}$, $RK_2=(g^a)^k\cdot g^{\omega}$를 생성하여 F에게 반환한다.

참고로 (2), (3)의 경우는 복호화 질의과정에서 B에 의해서 일관성있게 통제될 것이다.

-$Re\text{-}Encryption(ID_i, ID_j, C_{ID_i})$: 공격자 F가 ID_i의 암호문에서 ID_j의 암호문으로 변경하는 재암호화를 요청할 때, B는 주어진 암호문 $C_{ID_i}=(A, B, C, D, E)$에 대하여 다음과 같이 유효성을 체크한다: $e(H_5(g^r)^r, g)=e(H_5(A), A)$. 만일 등식이 성립하지 않으면 $\perp$을 반환한다. 등식이 성립하면 목록 L_{H_5}를 참조하여 대응하는 값 $\langle\pi, v\rangle$을 찾고 $V=E^{v^{-1}}=g_0^r$을 계산한 후 목록 L_{H_1}에서 대응하는 값 $\langle ID_i, h, z_i, \alpha_i\rangle$을 찾고 다음과 같이 $\tilde{C}$을 계산한다.

$$\tilde{C}=C\cdot e(V, H_1(ID_i)^{-1}\cdot H_1(ID_j)).$$

재암호화된 암호문 $C_{ID_j}=(A, B, \tilde{C}, D, E)$을 F에게 반환한다.

• $Decryption(ID_i, C_{ID_i})$: 공격자 F가 ID_i암호문 C_{ID_i}에 대한 복호화를 요청할 때, B는 암호문에 대한 유효성 검증을 다음과 같이 수행한다. $e(H_5(g^r)^r, g)=e(H_5(A), A)$이 성립하는지 확인한다. 성립하지 않으면 $\perp$을 반환한다. 등식이 성립하면 목록 L_{H_1}에서 B는 ID_i에 대응하는 값 $\langle ID_i, h, z_i, \alpha_i\rangle$을 얻는다.

- 만일 $\alpha_i=1$이면 다음을 수행한다.

(1) ID_i에 대한 사용자 비밀키 $(g^a)^{z_i}$를 알고 있으므로 정상적인 복호화 방식을 따른다. 즉, 먼저 $\sigma=C\cdot e(A, (g^a)^{z_i})$을 계산하고 $M=D\oplus H_4(\sigma)$을 계산한 후, $r'=H_3(\sigma, M)$을 계산한다. $A=g^{r'}$인지 확인한 후 등호가 성립하면 메시지 M을 반환한다.

(2) $A=g^{r'}$이 성립하지 않으면 $\sigma'=C\cdot\left\{e(E^{w^{-1}}, g^{bz})\cdot e(A, g_0^{z_i}g^{\omega})\right\}^{-1}$을 계산하고 $M=D\oplus H_4(\sigma')$을 계산한 후, $r'=H_3(\sigma', M)$을 계산한다. 여기서 z는 첼린지 ID의 해쉬값에 대응하는 지수이다. $A=g^{r'}$인지 확인한 후 등호가 성립하면 메시지 M을 반환한다. 그렇지 않으면 $\perp$을 반환한다. (이 경우는 프락시키가 첼린지 ID에 대응하는 임시 비밀키와 연관된 경우를 고려한 것이다.)

- 만일 $\alpha_i=0$이면 다음을 수행한다.

(1) $L_{H_3}=\langle\sigma_i, M_i, r_i\rangle$와 $L_{H_4}=\langle\sigma_i, y_i\rangle$ 목록 중에 동일한 σ_i를 포함하고 있는 L_{H_3}과 L_{H_4}목록 쌍들을 찾아서 짝지어 묶는다. 위에서 같이 묶인 L_{H_3}와 L_{H_4}의 모든 리스트 쌍들에 대하여 $M_i\oplus y_i$를 계산하여 D와 일치하는 쌍 $\langle\sigma_i, M_i, r_i\rangle$와 $\langle\sigma_i, y_i\rangle$을 찾는다.

$Case\,1$: $C\cdot\sigma^{-1}=e(A, g_0^{\omega})$이 성립하는지 확인한다. 만일 성립하면 $A=g^{r_i}$인지 확인한 후 등호가 성립하면 메시지 M을 반환한다. 성립하지 않으면 $\perp$을 반환한다.

$Case\,2$: $C\cdot\sigma^{-1}=e(A, g_0^{\omega})$의 등호가 성립하지 않으면 $\perp$을 반환한다. (이 경우는 프락시키가 첼린지 ID에 대응하는 임시 비밀키와 연관된 경우를 고려한 것이다.)

(2) 동일한 σ_i를 포함하고 있는 L_{H_3}과 L_{H_4}목록 쌍

이 존재하지 않는 경우, 목록 L_{H_5}를 참조하여 대응하는 값 $\langle \pi, v \rangle$을 찾고 $V = E^{v^{-1}} = g_0^r$을 계산한다. $\sigma = C \bullet e(V, H_1(ID_i)^{-1})$와 $M = D \oplus H_4(\sigma)$을 계산한 후, $r' = H_3(\sigma, M)$을 계산한다. $A = g^{r'}$인지 확인한 후 등호가 성립하면 시뮬레이션을 중단한다. 성립하지 않으면 $\perp$을 반환한다.

- *Challenge*: 공격자 A는 선택한 아이디 ID^*와 챌린지 메시지들의 쌍 (ID^*, m_0, m_1)를 B에게 준다. B는 ID^*에 대해 목록 L_H을 참조한 후 대응하는 해쉬값을 얻는다. $\langle ID^*, h, z, \alpha \rangle$를 ID^*의 $H-query$값이라고 하자. 만약 $\alpha = 1$이면 B는 시뮬레이션을 중단하고 $\alpha = 0$이라면 $b \in \{0,1\}$를 선택한 후 ID^*에 대한 챌린지 암호문 $C_{ID^*} = \{g^c, g^{acd}, M_b \bullet T^z\}$을 생성하여 A에게 반환한다.
- *Guess*: 공격자 B는 자신의 추측 비트값 b'을 출력한다.

A는 출력된 비트값 b'을 b와 비교한다. 만일 $b' = b$인 경우 A는 자신의 추측 비트값으로 b'를 출력한다. 만일 $b' \neq b$인 경우는 A는 임의의 비트 값을 출력한다.

다음을 사항을 관찰해 보자. 만일, T가 임의의 난수값이면 공격자에게 챌린지 암호문은 난수값으로 보이므로 공격자의 이점을 이용할 수 없다. 만일 $T=e(g,g)^{abc}$이면 위의 시뮬레이션은 "중단"이 발생되지 않는 한, 완전한 공격 시뮬레이션을 제공한다. 즉, 제안 기법 IDPRE-2의 실행환경과 동일하다. 위에서 "중단"이 발생할 사건은 두가지 경우이다. 첫 번째 공격자가 선택할 챌린지 ID를 시뮬레이터가 맞히지 못하였을 경우 발생한다. 이 경우는 알려진 기법[4]에 의해 1/e로 바운드되어진다.(여기서 e는 자연 상수이다.) 두 번째 경우는 복호화 질의에서 해쉬값을 참조하지 않고 공격자가 정당한 암호문 생성하여 질의하는 경우이다. 이 경우는 Fujisaki-Okamoto 변환기법의 안전성을 깨는 경우에 해당된다. Fujisaki-Okamoto 변환기법은 논문[9,14]에서 안전하다고 증명되었다. 따라서 두 번째 사건이 발생될 확률은 무시할만 하다. 위의 사항을 조합하면 다음과 같은 결과를 얻을 수 있다. 의미있는 확률로 제안 기법 IND-prID-CCA의 안전성을 깰 수 있는 알고리즘이 존재한다면, $MDBDH$ 문제를 해결할 수 있는 효율적인 알고리즘이 존재한다. □

4. 제안기법이 제공하는 유용한 특성

2장에서 정의된 보안모델 하에서 고려된 위의 안전성 증명과정은 부가적으로 제안 기법 *IDPRE-2*가 공모 공격을 통한 사용자 키 노출에 안전함을 의미하며 재암호화 단방향성(Unidirectionality)을 제공함을 쉽게 알 수 있다. 제안 기법은 또한 재암호화키의 비이행성(Non-Transitivity) 특성을 제공한다. 비이행성은 주어진 재암호화키들을 이용하여 새로운 재암호화키를 생성할 수 없어야 됨을 의미한다[2]. 제안 기법에서는 재암호화키 생성시 마다 독립적인 난수가 이용되므로 이에 대한 공격을 효과적으로 방지할 수 있다.

제안 기법에서는 재암호화된 암호문이 재암호화 하기 전의 암호문의 형태와 완벽하게 일치하므로 재암호화된 암호문에 대해서도 계속하여 재암화호화가 가능한 다이용성(Multiple use capacity)을 만족한다. 또한 재암호시에 암호문의 길이가 항상 일정하므로 재암호문에 대해서도 수신자의 입장에서 매우 효율적으로 복호화 할 수 있으며 전송 비용 측면에서도 매우 효율적이다

VI. 결 론

본 논문에서는 최근 제안된 Green-Ateniese ID기반 재암호화 기법이 사용자 키 노출 공격에 취약함을 보였다. 그리고 이를 해결하는 새로운 두 가지 기법들을 제안하였다. 첫 번째 기법은 CPA-안전성을 가지며 두 번째 기법은 CCA-안전성을 가진다. 제안기법들은 재암호화 시 암호문 구조를 보존하여 암호문 확장이 발생하지 않으며 재암호화의 횟수에 대한 제한이 없어 다양한 응용환경에 매우 유용하게 활용될 수 있다. 향후에는 랜덤 오라클을 없이 안전성이 증명되는 기법의 설계가 필요하다. 또한 재암호화 키 생성 시 비상호성(Non-interactivity)을 가지며 사용자 키 노출 공격에 강한 ID 기반 재암호화 기법의 구성은 흥미로운 연구과제가 될 것이다.

참 고 문 헌

[1] G. Ateniese, K. Fu, M. Green and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage", In: *NDSS* (2005).
[2] G. Ateniese, K, Fu, M. Green, S. Hohenberger, "Improved proxy re-encryption schemes with

applications to secure distributed storage", *ACM TISSEC 9(1),* 1 - .30 (2006).

[3] M. Blaze, G. Bleumer, M. Strauss, "Divertible protocols and atomic proxy cryptography", *In: Proceedings of Eurocrypt '98.* Volume 1403. (1998) 127 - 44

[4] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing", *In: Kilian, J. (ed.) CRYPTO 2001. LNCS,* vol. 2139, pp. 213 - .229. Springer, Heidelberg (2001).

[5] D. Boneh, E.J. Goh, T. Matsuo, "Proposal for P1363.3 Proxy Re-encryption" (http://grouper.ieee.org/groups/1363/IBC/submissions/NTTDataProposalfor-P1363.3-2006-09-01.pdf)

[6] R. Canetti, S. Halevi, J. Katz., "Chosen-Ciphertext Security from Identity-Based Encryption", *In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207 - .222. Springer, Heidelberg* (2004).

[7] R. Canetti, S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption", *In: ACM CCS* 2007, pp. 185 - .194. New York (2007).

[8] Y. Dodis, A. Ivan, "Proxy cryptography revisited", *In: Proceedings of the Tenth Network and Distributed System Security Symposium.* (2003).

[9] E. Fujisaki, T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes" *In: Proceedings of Crypto '99. Volume 1666 of Lecture Notes in Computer Science., Springer* (1999) 537 - 54

[10] M. Green, G. Ateniese, "Identity-Based Proxy Re-encryption", In: Katz, J., Yung, M. (eds.) *ACNS 2007. LNCS, vol. 4521, pp. 288 - .306. Springer, Heidelberg* (2007).

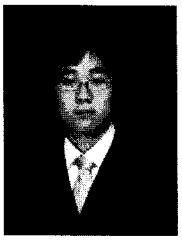
[11] B. Libert, D. Vergnaud,, "Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption", *In: PKC 2008. LNCS, vol. 4939, pp. 360 - .379. Springer, Heidelberg* (2008).

[12] M. Mambo, E. Okamoto,: Proxy Cryptosystems, "Delegation of the Power to Decrypt Ciphertexts", *IEICE Trans. Fund. Elect. Communications and CS,* E80-A/1, 54 - .63 (1997).

[13] T. Matsuo, "Proxy Re-encryption Systems for Identity-Based Encryption", *In: Pairing 2007. LNCS, vol. 4575, pp. 247 - .367. Springer, Heidelberg*

[14] P. Yang, T. Kitagawa, G. Hanaoka, R. Zhang, K. Matsurra, H. Imai, "Applying Fugisaki-Okamoto to Identity-Based Encryption," *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC-16). LNCS, vol. 3857, pp. 183 - 192. Springer, Heidelberg* (2006).

저 자 소 개



구 우 권(학생회원)
2006년 고려대학교 수학과 이학사 졸업.
2008년 고려대학교 정보경영 공학과 공학석사 졸업.
2008년~현재 고려대학교 정보경영공학과 박사과정.
<주관심분야 : 암호프로토콜, 정보보호이론>

김 형 중(평생회원)
1978년 서울대학교 제어계측 공학과 공학사.
1986년 서울대학교 제어계측 공학과 공학석사.
1989년 서울대학교 제어계측 공학과 공학박사.
1990년~2006년 강원대학교 교수.
2006년~현재 고려대학교 정보경영전문대학원 교수.
2008년~현재 대한전자공학회 컴퓨터소사이어티 회장.
<주관심분야 : Parallel Computing, Image Hashing, Data Compression, Steganography>

황 정 연(정회원)
1999년 고려대학교 수학과 이학사 졸업.
2003년 고려대학교 정보경영 공학과 공학석사 졸업.
2006년 고려대학교 정보경영 공학과 공학박사 졸업
2008년~현재 고려대 정보경영공학대학원 BK21 유비쿼터스 정보보호 사업단 연구교수
<주관심분야 : 암호프로토콜, 정보보호이론>

이 동 훈(정회원)
1983년 고려대학교 경제학과 경제학사 졸업.
1987년 Oklahoma University 전산학 석사 졸업.
1992년 Oklahoma University 전산학 박사 졸업
1993년~1997년 고려대학교 전산학과 조교수.
1997년~2001년 고려대학교 전산학과 부교수
2001년~현재 고려대학교 정보경영공학전문 대학원 교수.
<주관심분야 : 암호프로토콜, 암호이론, USN, 키 교환, 익명성 연구, PET 기술>