논문 2009-46CI-1-3

불연계성을 갖는 다중 공개키 암호 시스템

(Multiple and Unlinkable Public Key Encryption without Certificates)

박 소 영*, 이 상 호**

(Soyoung Park and Sang-Ho Lee)

요 약

본 논문에서는 서로 다른 그룹 및 응용 서비스에서 다수의 아이디 기반 공개키를 사용하되, 하나의 복호키를 이용하여 각 공개키로 암호화된 암호문을 모두 복호화 할 수 있는 불연계성을 갖는 다중 아이디 기반 공개키 암호 시스템을 새롭게 제안한다. 공개키는 서로 불연계성을 갖기 때문에, 공격자가 알려진 공개키를 이용하여 사용자 정보나 행동 패턴을 수집하거나 추적할 수 없으므로, 사용자 프라이버시가 보장되고, 인증서를 필요로 하지 않을 뿐만 아니라, 아이디 기반 암호 스킴이 갖는 key escrow문제도 해결하였다. 반면에, 다수의 공개키에 대해서 하나의 복호키가 사용되므로, 복호키의 안전성을 제공하기 위해 복호키 갱신 프로토콜도 함께 제공한다. 마지막으로, 제안한 암호 시스템이 랜덤 오라클 모델에서 선택적 암호문 공격(adaptively chosen-ciphertext attack)에 대해 안전함을 증명한다.

Abstract

We newly propose a multiple and unlinkable identity-based public key encryption scheme which allows the use of a various number of identity-based public keys in different groups or applications while keeping a single decryption key so that the decryption key can decrypt every ciphertexts encrypted with those public keys. Also our scheme removes the use of certificates as well as the key escrow problem so it is functional and practical. Since our public keys are unlinkable, the user's privacy can be protected from attackers who collect and trace the user information and behavior using the known public keys. Furthermore, we suggest a decryption key renewal protocol to strengthen the security of the single decryption key. Finally, we prove the security of our scheme against the adaptive chosen-ciphertext attack under the random oracle model.

Keywords : Public-Key Encryption, Identity-Based Encryption, Personal Privacy, Multiple Identities, Unlinkability

I. 서 론

인터넷 사용자 및 인터넷 기반 서비스의 저변 확대와 유비쿼터스 컴퓨팅 환경의 도래로, 점점 더 많은 인간 활동이 다양한 형태의 통신 장비를 이용한 유무선 네트워크에 의존함에 따라, 자각하지도 못하는 사이에 엄청난 양의 개인 정보가 네트워크상에 넘쳐나고 있다. 특히, 유비쿼터스 컴퓨팅 환경은 공격자로 하여금 더 많은 민감한 개인 정보를 보다 쉽게 획득하고 수집할 수 있는 환경을 제공함에 따라, 이러한 공격자들로부터 개인의 프라이버시를 보호할 수 있는 보다 강력하고 융통성 있는 보안 장치가 필요하다. 본 논문에서는 이종 네트워크 및 다양한 서비스들이 서로 유기적으로 공존하는 유비쿼터스 컴퓨팅 환경에서 개인 프라이버시 및 데이터 기밀성을 제공할 수 있는 새로운 형태의 공개키 암호 시스템을 제안한다.

1976년[4]과 1978년[9] 처음으로 제안된 공개키 암호 시스템(PKE: Public Key Encryption)은 네트워크상에서 누구와 통신을 하더라도 단 한 쌍의 공개키-개인키

* 정회원, School of Electrical Engineering and Computer Science, University of Central Florida
** 정회원, 이화여자대학교 컴퓨터공학과
(Dept. of Computer Science and Engineering, Ewha Womans University)
※ 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임. (KRF-2006-214-D00154)
접수일자: 2008년12월10일, 수정완료일: 2009년1월12일

만 있으면 비밀 통신이 가능하므로 키 관리를 혁신적으로 개선하였다. 하지만, 공개키는 기본적으로 공개 네트워크를 통해 상대에게 전달되거나 또는 공개 디렉토리에 저장되므로, 공개키 자체의 무결성을 보장할 수 있는 방안이 요구되고, 이를 위해 인증기관을 통한 공개키 인증서의 사용이 필수적이다[7]. 하지만, 인증서의 사용은 인증서 보관, 분배, 파기 등을 포함하는 인증서 관리를 위한 부가적인 연산이 많이 필요하여, 결과적으로 PKE 스킴의 발전에 가장 큰 장애요인이 되고 있다.

2001년 제안된 아이디 기반 암호 스킴(IBE: Identity-Based Encryption)[3]은 이름, IP 주소, 전화 번호, 이메일 주소 등 일반적인 아이디 정보로부터 공개키를 추출하여 사용함으로써, 기존 공개키의 인증서 문제를 효과적으로 해결하였다. 하지만, 공개키에 상응하는 복호키는 사용자가 직접 생성하지 못하고 제 3의 신뢰기관인 키 생성 기관(KGC: Key Generation Center)이 생성하므로, KGC는 모든 사용자의 복호키를 알게 되는 키 예탁 (Key escrow) 문제를 본질적으로 포함한다.

IBE 스킴의 키 예탁 문제를 해결함과 동시에, 공개키의 인증서 문제를 해결하기 위해 전통적 PKE 시스템과 IBE 시스템의 장점만을 취한 인증서 기반 암호 시스템(CBE: Certificate-Based Encryption)[6]과 인증서가 필요 없는 공개키 암호 시스템(CL-PKE: Certificateless Public Key Encryption)[1] 등이 제안되었다. 이들은 아이디와 사용자가 생성한 공개 정보를 함께 공개키로 사용하고, 상응하는 복호키는 사용자와 KGC가 함께 생성함으로써, 인증서 문제 및 키 예탁 문제를 동시에 해결한다. 아이디 기반 공개키는 공개키 생성, 분배 및 인증에 따른 공개키 관리가 매우 용이하다는 점에서 큰 장점을 갖는다. 하지만, 지금까지 제안된 PKE 및 IBE 시스템은 한 쌍의 공개키-비밀키의 사용을 기본적으로 가정하고 있다.

하지만, 실제로 많은 온라인, 오프라인 서비스에서 서로 다른 형태의 아이디의 사용을 요구하고 있으며, 이메일 주소, 전화번호, 웹 아이디, 주민 번호 등 다양한 형태의 아이디들을 서로 다른 용도로 사용하고 있다. 예를 들어, Alice는 전자 정부관련 서비스를 이용하기 위해 주민번호를 아이디로 사용하는 반면, 인터넷 뱅킹을 위해서는 직장 이메일 주소를 아이디로 사용할 수 있다. 전자 서비스가 서로 다른 형태의 아이디를 요구하는 것도 있지만, 사용자 입장에서 주민번호와 같은 민감한 개인 정보를 신뢰할 수 없는 서비스의 아이디로 사용하는 것을 꺼릴 수도 있다. 따라서 하나의 보편적 아이디의 사용은 유비쿼터스 컴퓨팅 환경에는 적합하지 않으며, 공격자가 알려진 아이디를 통해 사용자 정보 또는 행동 패턴을 쉽게 수집하고 추적할 수 있으므로 심각한 개인 프라이버시 문제를 야기할 수 있다.

따라서 본 논문에서는, 사용자가 서로 다른 다수의 아이디를 사용할 때, 이 아이디들로부터 각각의 서로 다른 공개키를 추출하여 안전한 통신을 가능하도록 하는 새로운 다중 아이디 기반 공개키 암호 시스템을 제안한다. 기존의 PKE 및 IBE 시스템은 공개키의 수가 늘어나면, 상대적으로 상응하는 개인키 및 복호키의 수도 증가하므로, 안전하게 유지해야 할 비밀 값이 많아져서 공개키 암호 시스템의 장점인 키 관리의 효율성을 잃어버리게 된다. 따라서 비밀 값의 개수가 증가하지 않으면서 다수의 공개키를 사용할 수 있는 새로운 접근 방법이 필요하다.

사용자 프라이버시 보호를 위해 슈도님 시스템[8, 13]이 제안되었는데, 다수의 랜덤 슈도님이 사용자 아이디 및 공개키로 사용된다는 점에서는 본 논문의 목표와 일치한다. 하지만 슈도님 시스템은 근본적으로 사용자 익명성을 제공하기 위해 제안되었기 때문에, 슈도님으로부터 해당 소유주를 알 수 없다. 따라서 슈도님 증명을 위한 익명 인증서(anonymous credential) 시스템[10~11, 14]이 함께 사용되어야 한다. 이와는 반대로, 제안하는 스킴은 이미 사용하고 있는 식별 가능한 (의미 있는) 아이디를 그대로 사용함과 동시에, 이로부터 공개키를 생성함으로써, 아이디를 이용한 공개키 인증도 용이하도록 한다.

본 논문에서는 단 하나의 복호키만 유지하면서, 인증서의 사용 없이 다수의 인증된 공개키를 생성하여 사용할 수 있는 "불연계성을 갖는 다중 공개키 암호 시스템(MU-PKE: Multiple and Unlinkable Public Key Encryption)"을 소개한다. MU-PKE는 기존의 PKE와 IBE를 혼합한 것으로, 사용자 키 생성 및 인증을 돕기 위해 신뢰기관인 KGC가 존재한다고 가정한다. 하지만, 본 논문에서 가정하는 KGC는 IBE에서의 KGC와는 달리, 절대적으로 사용자의 키를 생성하는 것은 아니며, CBE나 CL-PKE에서 사용된 것처럼 함축적(implicit) 인증을 위해 필요하다. MU-PKE는 구체적으로 다음의 기능을 수행한다.

• 하나의 복호키에 연관된 복수의 공개키: 사용자는 이미 온-오프 라인에서 사용하고 있는 다수의 아이

디에 대한 공개키를 생성할 수 있는 반면, 단 하나의 복호키만을 생성하여 각 공개키에 의해 암호화된 암호문을 복호할 수 있다.

• 인증서 및 키 예탁 문제 해결: 사용자는 KGC와 함께 자신의 부분 복호키 및 부분 공개키를 생성하고, 최종적으로는 사용자가 선택한 랜덤 비밀 값을 이용하여 자신의 공개키 및 복호키를 완성시킨다. 따라서 사용자와 KGC는 독단적으로 사용자의 키를 생성할 수 없고, 반드시 서로의 공조를 필요로 하므로, 생성된 공개키 및 복호키는 이미 KGC의 인증을 포함하고 있다. 따라서 별도의 인증서 사용을 필요로 하지 않는다. 또한 KGC는 사용자의 최종 공개키 및 복호키 값을 알지 못하므로, 키 예탁 문제도 해결한다.

• 공개키의 불연계성: 각각의 아이디로부터 추출된 공개키는 서로 연계되지 않는다. 즉, 사용자의 아이디 정보를 사전에 알지 못하는 공격자가 임의의 두 쌍의 아이디-공개키 쌍을 얻었을 때, 이들이 동일한 사용자의 것인지 아니면 서로 다른 두 사용자의 것인지 구분할 수 없다.

• 복호키 갱신: 다수의 공개키에 대해서 단 하나의 복호키 사용은 복호키 노출에 따른 안전성에 취약할 수 있다. 따라서 사용자는 주기적으로 복호키를 갱신함으로서, 복호키의 안전성을 제공한다.

MU-PKE 스킴은 타원 곡선상의 Weil 페어링과 같은 변형된 bilinear map[3]을 이용하여 설계되었다. 제안한 스킴의 안전성을 증명하기 위해, 먼저 MU-PKE 스킴에 대한 선택적 암호문 공격 모델(adaptively chosen-ciphertext security)을 정의하고 랜덤 오라클 모델에서 이 공격에 대한 안전성을 증명한다.

Ⅱ장에서 제안한 스킴의 안전성을 제공하는 암호학적 가정들을 설명하고, MU-PKE 스킴과 보안 모델을 정의한다. Ⅲ장에서 MU-PKE 스킴의 구체적인 프로토콜을 설명하고, Ⅳ장에서 선택적 암호문 공격에 대한 안전성 및 불연계성을 분석한다. Ⅴ장에서 복호키 갱신 프로토콜을 제안하고 Ⅵ장에서 결론을 맺는다.

Ⅱ. 보안 모델 및 정의

1. 암호학적 가정

제안한 스킴의 기본적 안정성을 제공하며, 논문의 나머지 부분을 통틀어 사용될 수학적 정의 및 가정을 기술한다. MU-PKE는 변형된 bilinear map에 기반을 두어 설계되었으므로, 변형된 bilinear map 및 이를 바탕으로 한 계산상 어려운 문제인 bilinear Diffie-Hellman 문제에 대해서 간략히 기술한다.

• Bilinear Map: 소수 q에 대해서, 오더가 q인 두 개의 순환(cyclic) 그룹 G_1, G_2가 있다. G_1은 타원 곡선(elliptic curve)과 같은 덧셈 그룹이고, G_2는 곱셈 그룹이라고 하자. 주어진 그룹 사이에서의 변형된 bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$는 다음의 조건을 만족해야 한다.

(1) Bilinear: 모든 $Q, W, Z \in G_1$에 대해서, 다음의 조건을 만족하면, $\hat{e}: G_1 \times G_1 \rightarrow G_2$는 bilinear 하다.

$\hat{e}(Q, W+Z) = \hat{e}(Q, W)\hat{e}(Q, Z)$이고,

$\hat{e}(Q+W, Z) = \hat{e}(Q, Z)\hat{e}(W, Z)$이다.

따라서, 임의의 $a, b \in Z_q^*$에 대해서,

$\hat{e}(aQ, bW) = \hat{e}(Q, W)^{ab} = \hat{e}(abQ, W)$이다.

(2) Non-degenerate: 그룹 G_1, G_2가 소수 오더를 갖는 그룹이기 때문에, 만약 P가 그룹 G_1의 생성자라면, $\hat{e}(P, P) \neq 1$이고, $\hat{e}(P, P)$는 그룹 G_2의 생성자다.

(3) Computable: 모든 $Q, W \in G_1$에 대해서, $\hat{e}(Q, W)$를 계산하는 효율적인 알고리즘이 존재한다.

• Bilinear Diffie-Hellman (BDH) 문제: 위에서 정의된 $(G_1, G_2, \hat{e})$와 임의의 정수 $a, b, c \in Z_q^*$에 대해서, $<P, aP, bP, cP>$가 주어졌을 때, $W = \hat{e}(P, P)^{abc} \in G_2$를 계산하는 것이다.

• BDH 가정: IG를 $(G_1, G_2, \hat{e})$를 생성하는 BDH 파라미터 생성자라고 하자. 어떤 알고리즘 A가 매우 큰 k에 대해서, 식(1)의 확률로 BDH 문제를 해결한다면, A는 $\epsilon(k)$만큼의 advantage를 갖는다고 한다.

$$Adv_{IG,A}(k) = \Pr[A(q, G_1, G_2, \hat{e}, P, aP, bP, cP) = \hat{e}(P, P)^{abc} \mid (q, G_1, G_2, \hat{e}) \leftarrow IG(1^k), P \leftarrow G_1^*, a, b, c \leftarrow Z_q^*] \geq \varepsilon(k) \quad (1)$$

임의의 다항식 $f \in Z[x]$에 대해서, 어떤 다항식 시간 알고리즘 A도 $Adv_{IG,A}(k) < 1/f(k)$만큼의 advantage를 갖는다면, IG는 BDH 가정을 만족한다고 하며, 그 BDH 문제는 계산상 풀기 어렵다.

2. 정의 및 보안 모델

MU-PKE 스킴은 크게 암호문을 생성하여 송신하는

송신자와, 암호문을 복호할 수 있는 암호문 수신자 및 KGC로 구성된다. 여기서 수신자는 일반적으로 개인 사용자이고, 송신자는 수신자와 연관된 사람 또는 응용 서비스 등에 해당된다. 따라서 한 명의 수신자 R_A에 대해서 다수의 송신자들이 존재할 수 있다. 예를 들어, 수신자 R_A가 총 t개의 아이디를 사용한다면 R_A와 연관된 송신자들은 R_A의 아이디에 따라서 t개의 송신자 그룹 $S_A=\{S_1,...,S_t\}$으로 세분화될 수 있다. R_A가 각 송신자 그룹 S_i에서 아이디 $ID_{A,i}$를 사용한다면, S_i에 포함되는 송신자들은 $ID_{A,i}$가 R_A를 지칭함을 이미 알고 있다고 가정한다. 기본적으로 송신자와 수신자간 데이터 통신은 공개 네트워크를 통해서 이루어지므로, R_A는 MU-PKE 스킴을 이용하여 언제든지 각 송신자 그룹에서 사용하는 아이디에 대한 공개키를 생성한 후, 해당 송신자와 MU-PKE 시스템을 이용하여 안전한 통신을 수행할 수 있다. 단, KGC와 R_A간에는 안전한 네트워크 채널이 존재한다고 가정한다. k를 보안 파라미터라고 하고, IG를 BDH 파라미터 생성자라고 했을 때, MU-PKE 스킴은 다음과 같이 정의된다.

정의 1. MU-PKE 스킴은 다섯 개의 랜덤 알고리즘 (Setup, Gen-DK, Gen-PK, Encryption, Decryption)으로 구성되며, 각각의 알고리즘은 다음과 같이 정의된다.

(1) Setup: 확률적(probabilistic) 시스템 파라미터 생성 알고리즘으로, k와 IG를 입력받아서, 시스템 파라미터 $params$와 KGC의 매스터 키 s를 출력한다. KGC에 의해 한번 수행되며, s를 제외한 $params$는 공개된다.

(2) Gen-DK: 결정적(deterministic) 복호키 생성 알고리즘으로, Set-Private-Key, Extract-Partial-Decryption-Key, Set-Decryption-Key로 구성된다. 각 수신자 R_A와 KGC가 함께 수행한다.

(a) Set-Private-Key: $params$, R_A가 선택한 문자열, 랜덤 정수 x_A를 입력받아서, R_A의 개인키 쌍 (x_A,P_A) 및 매스터 아이디 MID_A를 출력한다. R_A에 의해 수행된다.

(b) Extract-Partial-Decryption-Key: MID_A와 s를 입력받아서, R_A의 부분 복호키 PDK_A를 출력한다. KGC에 의해 수행되며, MID_A와 PDK_A는 R_A에게 안전하게 전송된다.

(c) Set-Decryption-Key: PDK_A, x_A를 입력받아서, 최종 복호키 DK_A를 출력한다. R_A에 의해 수행되며, MID_A,x_A,DK_A는 안전하게 관리한다.

(3) Gen-PK: 확률적 공개키 생성 알고리즘으로, Extract-Partial-Public-Key와 Set-Public-Key로 구성된다. R_A의 임의의 아이디 $ID_{A,i}$에 대해서, R_A와 KGC가 함께 수행한다.

(a) Extract-Partial-Public-Key: 아이디 $ID_{A,i}$, 사용자 정보 Info_A, 아이디 증명 $PF_{A,i}$, s를 입력받아서, 부분 공개키 $PPK_{A,i}$를 출력하거나, 아이디 증명이 틀린 경우 $\perp$를 출력한다. KGC에 의해 수행되며, $PPK_{A,i}$는 R_A에게 안전하게 전송된다.

(b) Set-Public-Key: $ID_{A,i}$, $PPK_{A,i}$, DK_A, x_A 및 MID_A를 입력받아서, $ID_{A,i}$에 대한 최종 공개키 집합 $PKS_{A,i}=\{E1,E2,E3,E4\}$를 출력한다. R_A에 의해 수행된다.

(4) Encryption: 확률적 암호 알고리즘으로, $params$, 메시지 M, R_A의 $ID_{A,i}$ 및 $PKS_{A,i}$을 입력받아서, 암호문 C를 출력하거나 암호화에 실패했을 경우 $\perp$를 출력한다. 각 송신자에 의해 수행된다.

(5) Decryption: 결정적 복호 알고리즘으로, $params$, R_A의 DK_A, C를 입력받아서, 평문 M을 출력하거나, 복호화에 실패했을 경우 $\perp$를 출력한다.

다음으로, 공격자의 선택적 암호문 공격에 대한 MU-PKE 스킴의 보안 모델을 정의한다. 선택적 암호문 공격(IND-CCA)[2~3, 12]모델은 PKE 시스템의 안전성을 증명하기 위한 가장 대표적인 공격 모델로서, 전통적인 IND-CCA 모델은, 공격자가 도전해야(challenge)하는 암호문에 대한 평문을 제외하고, 공격자가 선택한 모든 암호문에 대한 평문을 얻을 수 있다고 가정한다. 이 공격 모델이 IBE 시스템에서는 공격자의 공격 능력이 조금 더 강화되어, 공격자가 도전하는 아이디에 대한 복호키를 제외하고, 공격자가 선택한 모든 아이디에 대한 복호키를 얻을 수 있다고 허용한다(IND-ID-CCA). 제안하는 스킴도 일종의 IBE 시스템이므로, IND-ID-CCA 공격 모델을 바탕으로 한다. 하지만, 제안하는 스킴은 한 명의 수신자가 다수의 아이디를 사용하기 때문에, IND-ID-CCA 모델을 조금 수정하여, 다음과 같이 MU-PKE 스킴에 대한 선택적 암호문 공격자(IND-MUP-CCA) A의 공격 능력을 정의한다.

- A는 자신이 선택하는 모든 아이디에 대한 유효한

공개키를 얻을 수 있다.

• A는 자신이 선택한 임의의 암호문에 대한 평문을 얻을 수 있지만, 공격자가 공격하는 도전자 R_{ch}의 특정 아이디 및 공개키 집합 $<ID_{ch,I}, PKS_{ch,I}>$로 암호화된 암호문 C_{ch}에 대한 평문은 얻을 수 없다.

• A는 자신이 선택한 아이디들에 대한 복호키를 얻을 수 있는데, 공격자가 도전하는 도전자 R_{ch}의 아이디들에 대한 복호키는 알 수 없다.

• A는 자신이 선택한 모든 아이디에 대한 부분 공개키 및 부분 복호키를 얻을 수 있다. 공격자가 도전하는 도전자 R_{ch}의 특정 아이디 $ID_{ch,I}$에 대한 부분 공개키 및 R_{ch}의 부분 복호키도 포함한다.

경우에 따라서는, KGC 또는 KGC에 소속된 직원이 악의적인 공격자가 될 수도 있으므로, KGC의 매스터 키 s에 접근할 수 있느냐 없느냐에 따라서 위에서 정의된 공격자를 다시 두 가지 타입으로 세분화한다. Type-1 공격자 A_I는 KGC의 매스터 키에 접근할 수 없지만, Type-2 공격자 A_{II}는 KGC의 매스터 키 및 수신자의 매스터 아이디에 대한 정보를 알 수 있다. 하지만, 두 타입의 공격자 모두 수신자의 개인키에 대한 정보는 알 수 없다.

정의 2. 만약 어떤 다항 시간 IND-MUP-CCA 공격자 $A(A_I$과 $A_{II})$도 다음의 IND-MUP-CCA 게임에서 $Adv_{IG,A}(k)$가 무시할 만큼 작다면, MU-PKE 스킴은 IND-MUP-CCA 공격에 안전하다.

[IND-MUP-CCA 게임]

• Setup: 도전자는 보안 파라미터 k에 대해서 Setup 알고리즘을 수행한 후, $params$를 A에게 제공한다. 단, Type-2 공격자 A_{II}에게는 s도 함께 제공한다.

• Phase 1: A는 자신이 선택한 아이디 $<ID_{i,j}>$ 또는 아이디-암호문 쌍 $<ID_{i,j}, C_{i,j}>$들에 대해서, 총 $q_1,...,q_m$개의 쿼리를 생성하는데, 각 쿼리 q_i는 다음 중 하나에 해당한다. 단, Type-2 공격자 A_{II}에게는, 도전자가 해당 쿼리에 대한 응답뿐 아니라, 쿼리로 질문되는 아이디에 대한 매스터 아이디도 함께 제공한다.

- $<ID_{i,j}>$에 대한 공개키 집합 쿼리: 도전자는 $ID_{i,j}$의 소유주 R_i를 결정한 다음, Gen-DK 알고리즘을 수행하여, R_i의 복호키, 개인키 및 매스터 아이디를 얻는다. 그런 다음, Gen-PK 알고리즘을 수행하여 $ID_{i,j}$에 대한 공개키 집합 $PKS_{i,j}$를 생성하여 공격자에게 $PKS_{i,j}$를 제공한다.
- $<ID_{i,j}>$에 대한 부분 공개키 쿼리: 도전자는 $ID_{i,j}$의 소유주 R_i를 결정한 다음, Gen-DK 알고리즘을 수행하여, R_i의 복호키, 개인키 및 매스터 아이디를 얻는다. 그런 다음, Gen-PK 알고리즘을 수행하여 $ID_{i,j}$에 대한 부분 공개키 $PPK_{i,j}$를 생성하여 공격자에게 $PPK_{i,j}$를 제공한다.
- $<ID_{i,j}>$에 대한 부분 복호키 쿼리: 도전자는 $ID_{i,j}$의 소유주 R_i를 결정한 다음, Gen-DK 알고리즘을 수행하여, R_i의 부분 복호키 $PDK_{i,j}$를 생성하여 공격자에게 제공한다.
- $<ID_{i,j}>$에 대한 복호키 쿼리: 도전자는 $ID_{i,j}$의 소유주 R_i를 결정한 다음, Gen-DK 알고리즘을 수행하여, R_i의 복호키 DK_i를 생성하여 공격자에게 제공한다.
- $<ID_{i,j}, C_{i,j}>$에 대한 복호 쿼리: 도전자는 $ID_{i,j}$의 소유주 R_i를 결정한 다음, Gen-DK 알고리즘을 수행하여, R_i의 복호키 DK_i를 생성한다. DK_i를 이용하여 Decryption 알고리즘을 수행하여, $C_{i,j}$에 대한 평문을 얻어서 공격자에게 제공한다.

• Challenge: 공격자가 Phase 1 단계를 끝내면, 동일한 길이의 두 평문 M_0, M_1과 아이디 $ID_{ch,I}$를 선택한다. 단, Phase 1에서 $ID_{ch,I}$의 소유주 R_{ch}에 대한 복호키 쿼리는 수행된 적이 없어야 한다. 도전자는 랜덤 $b \in \{0,1\}$를 선택하여, 암호문 $C_{ch,I}$를 생성한다.

$$C_{ch,I} = Encryption(params, ID_{ch,I}, PKS_{ch,I}, M_b)$$

• Phase 2: 공격자는 다시 $q_{m+1},...,q_n$의 쿼리를 생성하는데, 각 쿼리 q_i는 다음 중 하나에 해당한다. 단, (1) $<ID_{i,j}>$에 대한 복호키 쿼리 시, 해당 소유주 R_i가 R_{ch}와 같아서는 안 되며, (2) $<ID_{i,j}, C_{i,j}>$에 대한 복호 쿼리 시, $<ID_{i,j}, C_{i,j}>$가 $<ID_{ch,I}, C_{ch,I}>$와 동일해서는 안 된다.

- $<ID_{i,j}>$ 에 대한 공개키 집합 쿼리, 부분 공개키 쿼리, 부분 복호키 쿼리: Phase 1과 동일하게 응답한다.
- $R_i \neq R_{ch}$인 $<ID_{i,j}>$에 대한 복호키 쿼리: Phase 1과 동일하게 응답한다.
- $<ID_{i,j}, C_{i,j}> \neq <ID_{ch,I}, C_{ch,I}>$에 대한 복호 쿼리:

Phase 1과 동일하게 응답한다.

• Guess: 공격자는 자신의 추측인 $b' \in \{0,1\}$을 출력하는데, 만약 $b' = b$이면 게임에서 승리한다.

위와 같은 공격자 A를 IND-MUP-CCA 공격자라고 하고, advantage를 $Adv_{IG,A}(k) = |\Pr[b = b'] - \frac{1}{2}|$로 정의한다.

다음으로, 공개키의 불연계성에 대해서 정의한다. 불연계성을 개략적으로 설명하면, 공격자가 임의의 두 개의 공개키 집합을 얻었을 때, 해당 아이디의 소유주가 누구인지 알지 못한다면, 이 두 공개키 집합이 동일한 사용자의 것인지, 서로 다른 두 명의 사용자의 것인지 알 수 없음을 의미한다.

정의 3. 어떤 다항시간 IND-LINK 공격자 A_L도 다음의 IND-LINK 게임에서 무시할 수 없는 advantage를 갖지 못하면, MU-PKE 스킴은 불연계성을 만족한다.

[IND-LINK 게임]

• Setup: CH_0, CH_1으로 표기되는 두 명의 도전자가 있다. 보안 파라미터 k에 대해서 Setup 알고리즘을 수행하여 시스템 파라미터를 생성한 후, Gen-DK 알고리즘을 수행하여 매스터 아이디, 개인키 및 복호키를 생성한 후, 시스템 파라미터를 A_L에게 제공한다.

• Challenge: A_L은 임의의 두 문자열 $ID_1, ID_2 \in \{0,1\}^*$을 선택한 후, 도전자들에게 해당 공개키 집합을 요구한다. 도전자들은 다음과 같이 응답한다.

(1) CH_0가 랜덤 $c_0 \in \{0,1\}$을 선택하면, CH_1은 $c_1 = 1 - c_0$로 한다. $c_i = 1$인 CH_i가 ID_1에 대한 Gen-PK 알고리즘을 수행하여 PKS_1을 출력한다.

(2) CH_0가 다시 랜덤 $c_0' \in \{0,1\}$을 선택하면, CH_1은 $c_1' = 1 - c_0'$로 한다. $c_i' = 1$ 인 CH_i가 ID_2에 대한 Gen-PK 알고리즘을 수행하여 PKS_2을 출력한다.

(3) $c = c_0 \oplus c_0'$를 계산한 후, PKS_1와 PKS_2를 A_L에게 제공한다.

• Guess: A_L은 $c' \in \{0,1\}$을 추측한다. 두 공개키 집합이 동일한 도전자에 의해 생성되었다면, $c' = 0$으로 하고, 아니면, $c' = 1$로 한다. 만일 $c = c'$이면, A_L이 이긴다.

위와 같은 공격자를 IND-LINK 공격자라고 하며, advantage는 $Adv_{IG,A_L}(k) = |\Pr[c = c'] - \frac{1}{2}|$이다.

Ⅲ. 불연계성을 갖는 다중 공개키 암호(MU-PKE: Multiple and Unlinkable Public Key Encryption)

본 장에서는 bilinear map을 이용하여 제안한 스킴의 구체적인 프로토콜을 제시한다. KGC, $i = 1,..,t$에 대해서 $ID_{A,i}$로 표기되는 t개의 서로 다른 아이디를 가지고 있는 수신자 Alice, 그리고 Alice의 특정 아이디 $ID_{A,i}$를 알고 있는 송신자 S_i가 있다고 하자. MU-PKE 스킴을 구성하는 다섯 개의 알고리즘은 다음과 같다.

(1) Setup: KGC는 (k, IG)를 입력받아서, 소수 오더 q를 갖는 두 개의 그룹 G_1, G_2와 $\hat{e}: G_1 \times G_1 \rightarrow G_2$를 생성한다. G_1의 생성자 $P \in G_1$를 선택하고, 랜덤 매스터 키 $s \in Z_q^*$를 선택한 후, $P_0 = sP \in G_1$를 계산한다. 다섯개의 암호학적 해쉬 함수를 다음과 같이 선택한다.

$$H_0 : \{0,1\}^* \rightarrow Z_q^*, H_1 : \{0,1\}^* \rightarrow G_1^*, H_2 : G_2 \rightarrow \{0,1\}^n,$$

$$H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow Z_q^*, \quad H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$$

단, n은 평문 길이와 같다. s는 안전하게 보관하고, 시스템 파라미터 $params = < G_1, G_2, \hat{e}, n, P, P_0, H_0, H_1, H_2, H_3, H_4 >$는 공개한다.

(2) Gen-DK: Alice와 KGC 사이에 한번만 수행되며, 안전한 네트워크 채널을 통해 통신이 이루어진다.

(a) Alice는 자신의 사용자 정보 $\text{Info}_A \in \{0,1\}^*$를 생성한 후, $(params, \infty_A)$에 대해서, Set-Private-Key 알고리즘을 다음과 같이 수행한다.

- 랜덤 $x_A \in Z_q^*$ 선택 후, $P_A = x_A P \in G_1$ 생성
- $MID_A = \text{Info}_A \| P_A \in \{0,1\}^*$와 $M_A = H_1(MID_A) \in G_1$생성 후, 매스터 아이디 MID_A, M_A, 개인키 쌍 (x_A, P_A) 출력

(b) KGC는 $(params, MID_A, s)$에 대해서, Extract-Partial-Decryption-Key 알고리즘을 수행한다.

- $M_A = H_1(MID_A) \in G_1$ 계산
- 부분 복호키 $PDK_A = sM_A \in G_1$ 생성 후, Alice에게 안전하게 전송
- KGC는 MID_A를 안전하게 보관

(c) Alice는 $(params, PDK_A, x_A)$에 대해서 Set-Decryption-Key 알고리즘을 수행한다.

- 복호키 $DK_A = x_A PDK_A = x_A s M_A \in G_1$ 생성
- Alice는 MID_A, x_A, DK_A를 안전하게 보관

(3) Gen-PK: Alice와 KGC간에 안전한 형태로 수행된다. Alice는 필요할 때마다, 자신의 특정 아이디 $ID_{A,i}$에 대한 공개키 집합을 다음과 같이 생성한다.

(a) Alice는 자신이 $ID_{A,i}$의 소유주임을 증명하기 위해서, 아이디 증명 $PF_{A,i} = x_A H_1(\infty o_A \| ID_{A,i}) \in G_1$을 생성하여 $ID_{A,i}$ 및 사용자 정보 ∞o_A와 함께 KGC에게 보낸다.

(b) KGC는 $(params, ID_{A,i}, \infty o_A, PF_{A,i}, s)$에 대해 Extract-Partial-Public-Key 알고리즘을 수행한다.

- KGC가 관리하는 사용자의 매스터 아이디 리스트로부터 ∞o_A에 상응하는 P_A 검색 후, $\hat{e}(PF_{A,i}, P) = \hat{e}(H_1(\infty o_A \| ID_{A,i}), P_A)$인지 검사
- 등식이 성립하면, $Q_{A,i} = H_1(ID_{A,i}) \in G_1$ 생성 후, 부분 복호키 $PPK_{A,i} = s Q_{A,i} \in G_1$을 생성하여 Alice에게 전달하고, 아니면 $\perp$을 출력

(c) Alice는 $(params, ID_{A,i}, PPK_{A,i}, MID_A, x_A, DK_A)$에 대해서 Set-Public-Key 알고리즘을 수행한다.

- $Q_{A,i} = H_1(ID_{A,i}) \in G_1$ 생성
- $a_i = H_0(MID_A \| ID_{A,i}) \in Z_q^*$ 생성
- $E1 = a_i x_A M_A, E2 = \frac{1}{a_i} PPK_{A,i} = \frac{1}{a_i} s Q_{A,i}$, $E3 = \frac{1}{a_i} Q_{A,i} \in G_1$ 생성 후,
- $QC_{A,i} = H_1(E1 \| E2 \| E3 \| ID_{A,i}) \in G_1$ 계산
- $E4 = \frac{1}{a_i} QC_{A,i} \in G_1$ 계산 후, 공개키 집합 $PKS_{A,i} = < E1, E2, E3, E4 >$ 생성

Alice는 $PKS_{A,i}$를 송신자 S_i에게 공개한다.

(4) Encryption: Alice의 특정 아이디 $ID_{A,i}$ 및 공개키 집합 $PKS_{A,i}$를 알고 있는 S_i가 메시지 M을 암호화하기 위해, $(params, ID_{A,i}, PKS_{A,i}, M)$에 대해서 다음과 같이 수행한다.

- $Q = H_1(ID_{A,i}), QC = H_1(E1 \| E2 \| E3 \| ID_{A,i})$ 계산
- 두 등식 $\hat{e}(E4, Q) = \hat{e}(QC, E3)$와 $\hat{e}(E2, P) = \hat{e}(E3, P_0)$가 성립하는지 검사. 만약 하나라도 등식이 성립하지 않으면 공개키 집합이 잘못된 것으로, 암호 알고리즘을 중단
- $g = \hat{e}(E1, E2) = \hat{e}(a_i x_A M_A, \frac{1}{a_i} s Q_{A,i}) = \hat{e}(x_A s M_A, Q)$ 계산
- 랜덤 $\sigma \in \{0,1\}^n$ 선택 후, $r = H_3(\sigma, M) \in Z_q^*$ 계산
- $C = < U, V, T > = < rQ, \sigma \oplus H_2(g^r), M \oplus H_4(\sigma) >$를 암호문으로 생성

(4) Decryption: Alice는 $(params, DK_A, C)$에 대해서, 다음과 같이 복호한다.

- $V \oplus H_2(\hat{e}(DK_A, U)) = \sigma'$ 계산
- $T \oplus H_4(\sigma') = M'$ 계산
- $r' = H_3(\sigma', M')$ 계산 후, $U = r' Q_{A,i}$인지 검사 만약 등식이 성립하지 않으면, 암호문이 틀린 것으로 복호를 중단하고, 아니면 M'을 평문으로 출력

Ⅳ. 안전성 분석

본 장에서는 BDH 문제가 어렵다는 가정하에, MU-PKE 스킴이 IND-MUP-CCA 공격에 안전함을 보인다. 이를 위해서, 하나의 복호키에 대해서 t개의 서로 다른 공개키를 갖는, 아이디 기반이 아닌, 두 개의 다중 공개키 암호 시스템 **BasicMultiPub**과 **FullMultiPub**을 정의한다. **FullMultiPub**은 **BasicMultiPub**에 Fujisaki-Okamoto 변환[5]을 적용한 것으로, Fujisaki-Okamoto 정리는, **FullMultiPub**에 대해서 $\epsilon(k)$만큼의 advantage를 갖는 IND-CCA 공격자 A가 있다면, **BasicMultiPub**에 대해서 $\epsilon_1(k)$만큼의 advantage를 갖는 선택적 평문 공격자(IND-CPA) B가 있음을 보여준다.

Fujisaki-Okamoto 정리. A를 **FullMultiPub**에 대해서 $\varepsilon(k)$만큼의 advantage를 갖는 IND-CCA 공격자라고 하자. A가 총 $t(k)$시간 동안, 최대 q_D개의 복호 쿼리, H_3, H_4 해쉬 함수에 대해서, 최대 q_{H_3}, q_{H_4}개의 해쉬 쿼리를 수행한다면, **BasicMultiPub**에 대해서, 총 $t_1(k)$ 시간 동안, 식(2)의 advantage $\epsilon_1(k)$를 갖는 IND-CPA 공격자 B가 존재한다.

$$\epsilon_1(k) \geq FO_{adv}(\epsilon(k), q_{H_4}, q_{H_3}, q_D) = \frac{1}{2(q_{H_4}+q_{H_3})}[(\epsilon(k)+1)(1-2/q)^{q_D}-1], \quad (2)$$

$$t_1(k) \leq FO_{adv}(t(k), q_{H_4}, q_{H_3}) = t(k) + O((q_{H_4}+q_{H_3}) \cdot n)$$

단, q는 G_1, G_2의 크기이고, n은 σ의 길이이다.

Fujisaki-Okamoto 정리를 이용하여, 다음의 정리 1은, 만약 IND-MUP-CCA 공격자 A가 IND-MUP-

CCA 게임에서 무시할 수 없는 advantage를 가지면, A를 이용하여 BDH 문제를 해결할 수 있는 공격자 B가 존재함을 보인다.

정리 1. 네개의 해쉬 함수 H_1,H_2,H_3,H_4는 랜덤 오라클이고, MU-PKE 스킴에 대해서 총 $t(k)$시간 동안 $\epsilon(k)$만큼의 advantage를 갖는 두 형태의 IND-MUP-CCA 공격자 A_I과 A_{II}가 있다고 하자. A_I는 최대 $q_{PK}>0$ 만큼의 공개키 집합 쿼리, $q_{PPK}>0$만큼의 부분 공개키 쿼리, $q_{PD}>0$만큼의 부분 복호키 쿼리, $q_{DK}>0$만큼의 복호키 쿼리, $q_D>0$만큼의 복호 쿼리를 수행하고, A_{II}는 최대 $q_{PK}>0$만큼의 공개키 집합 쿼리, $q_{PD}>0$만큼의 부분 복호키 쿼리, $q_{DK}>0$만큼의 복호키 쿼리, $q_D>0$만큼의 복호 쿼리를 수행한다고 하자. 또한, 각 공격자는 H_2,H_3,H_4해쉬 함수에 대해서 최대 q_{H_2},q_{H_3},q_{H_4}만큼의 해쉬 쿼리를 수행한다. 그럼 적어도 식(3)의 advantage로 IG에 의해 생성된 그룹에서 BDH 문제를 해결하는 알고리즘 B가 존재한다.

$$Adv^{A_I}_{IG,B}(k) \geqq$$
$$2FO_{adv}\left(\frac{\epsilon(k)}{e(1+q_{PPK}+q_{PD}+q_{DK}+q_D)},q_{H_4},q_{H_3},q_D\right)/q_{H_2},$$
$$Adv^{A_{II}}_{IG,B}(k) \geqq$$
$$2FO_{adv}\left(\frac{\epsilon(k)}{e(1+q_{DK}+q_D)},q_{H_4},q_{H_3},q_D\right)/q_{H_2} \quad (3)$$

B의 총 수행 시간은 $t_B(k) \leqq FO_{time}(t(k),q_{H_4},q_{H_3})$이다. 단, $e \approx 2.71$는 자연 로그의 밑이고, FO_{adv}와 FO_{time}은 Fujisaki-Okamoto 정리에서 정의된 Fujisaki-Okamoto 함수이다.

위 정리를 증명하기 위해서, 먼저 MU-PKE 스킴에 대한 IND-MUP-CCA 공격이 **FullMultiPub**에 대한 IND-CCA 공격으로 변환될 수 있음을 보인다. 그런 다음, **FullMultiPub**에 대해서 IND-CCA 공격자가 존재하면, **BasicMultiPub**에 대해서 IND-CPA 공격자가 존재하고, BDH 가정하에서 **BasicMultiPub**이 INC-CPA 공격에 안전함을 보인다.

BasicMultiPub 알고리즘은 표 1과 같다. **FullMultiPub**은 **BasicMultiPub**의 Encryption 및 Decryption 알고리즘에 두 개의 해쉬 함수 $H_3:\{0,1\}^n\times\{0,1\}^n\rightarrow Z_q^*$와 $H_4:\{0,1\}^n\rightarrow\{0,1\}^n$가 추가적으로 사용되는 것을 제외하고는 **BasicMultiPub**과 동일하다. 암호 및 복호를 위해 두 해쉬 함수가 사용되는 방법은 MU-PKE 스킴과 동일하다.

표 1. **BasicMultiPub** 알고리즘
Table 1. **BasicMultiPub** Algorithm

- Setup: k와 IG에 대해서,
  - $(G_1,G_2,\hat{e})$ 생성 후, 생성자 $P\in G_1$ 선택
  - 랜덤 비밀 값 $M\in G_1^*$, $s,x\in Z_q^*$ 선택
  - $P_0=sP$와 $D=xsM\in G_1^*$ 계산
  - t개의 랜덤 $Q_1,Q_2,...,Q_t\in G_1^*$와 랜덤 비밀 정수값 $a_1,a_2,...,a_t\in Z_q^*$ 선택
  - t개의 공개키 $PKS_1,PKS_2,...,PKS_t$ 생성 후, $PKS=\{\mathrm{PKS}_1,\mathrm{PKS}_2,...,\mathrm{PKS}_t\}$ 설정
    $i=1,..,t$에 대해서,
    $PKS_i=<E0,E1,E2,E3,E4>$
    $=<Q_i,a_ixM,\frac{1}{a_i}sQ_i,\frac{1}{a_i}Q_i,\frac{1}{a_i}T_i>,$
    $T_i=Q_i+a_ixM+\frac{1}{a_i}sQ_i+\frac{1}{a_i}Q_i$
  - 암호학적 해쉬 함수 $H_2:G_2\rightarrow\{0,1\}^n$ 선택
  - 공개키 $K_{pub}=<G_1,G_2,\hat{e},n,P,P_0,PKS,H_2>$과 복호키 $D=xsM$ 설정
- Encryption: 하나의 공개키 집합 PKS_i 및 M에 대해서,
  - $T=E0+E1+E2+E3\in G_1^*$ 계산
  - 두 등식 $\hat{e}(E4,E0)=\hat{e}(T,E3)$와 $\hat{e}(E2,P)=\hat{e}(E3,P_0)$인지 검사. 성립하지 않으면 암호 중단
  - $g=\hat{e}(E1,E2)=\hat{e}(xM,sQ_i)$ 계산
  - 랜덤 $r\in Z_q^*$ 선택
  - 암호문 $C=<U,V>=<rE0,M\oplus H_2(g^r)>$ 생성
- Decryption: $C=<U,V>$에 대해서,
  - $M=V\oplus H_2(\hat{e}(D,U))$ 계산 후, 평문 M 출력

보조 정리 1. H_1은 랜덤 오라클이고, A_I는 MU-PKE 스킴에 대해서 $\epsilon(k)$만큼의 advantage를 갖는 Type-1 IND-MUP-CCA 공격자라고 하자. A_I는 최대 $q_{PK}>0$만큼의 공개키 집합 쿼리, $q_{PPK}>0$만큼의 부분 공개키 쿼리, $q_{PD}>0$만큼의 부분 복호키 쿼리, $q_{DK}>0$만큼의 복호키 쿼리, $q_D>0$만큼의 복호 쿼리를 수행한다. 그러면, **FullMultiPub**에 대해서 적어도

$\frac{\epsilon(k)}{e(1+q_{PPK}+q_{PD}+q_{DK}+q_D)}$ 만큼의 advantage를 갖는 IND-CCA 공격자 B가 존재하며, 수행시간은 $O(time(A_I))$이다.

증명. 먼저 A_I를 이용하여 **FullMultiPub**에 대해서 위의 advantage를 갖는 IND-CCA 공격자 B를 구성한다. B는 IND-MUP-CCA 게임에서 다음과 같이 도전자를 시뮬레이트하여 A_I의 쿼리에 응답한다.

Setup: 먼저, 도전자 CH는 **FullMultiPub**의 Setup 알고리즘을 수행한 다음, MU-PKE를 시뮬레이트하는 공격자 B에게 공개키 K_{pub}을 전달한다. B는 K_{pub}으로부터 $params = < G_1, G_2, \hat{e}, n, P, P_0, H_1, H_2, H_3, H_4 >$를 추출하여 A_I에게 전달한다. 단, H_1은 B에 의해 통제되는 랜덤 오라클이다. A_I는 자신의 공격 중에 각 해쉬 함수에 대한 해쉬 쿼리를 수행할 수 있다. 여기서, H_2, H_3, H_4는 B에 의해서 통제되는 랜덤 오라클일 필요가 없고, B는 이들에 대한 쿼리를 CH에게 넘겨서 CH가 생성한 응답을 다시 A_I에게 건네준다. B는 H_1해쉬 쿼리에 대해서 다음과 같이 응답한다.

H_1해쉬 쿼리: A_I는 두 가지 입력 형태에 대한 H_1쿼리를 수행할 수 있는데, Type 1 해쉬 쿼리는 $< ID > \in \{0,1\}^*$에 대한 해쉬 쿼리(MU-PKE 스킴에서 Q를 얻기 위한)이고, Type 2 해쉬 쿼리는 $< ID, PKS >$에 대한 해쉬 쿼리(MU-PKE 스킴에서 QC를 얻기 위한)이다.

B는 먼저 Z_q^*상에서 랜덤하게 선택한 $l < [q/t]$개의 랜덤 정수로 랜덤 풀 $MP = \{m_1, m_2, ..., m_l\}$를 구성한다. 이 랜덤 풀은 각각 최대 t개의 공개키를 갖는 l명의 서로 다른 사용자(수신자)를 나타낸다. B는 $< ID_j, Q_j, T_j, c_j, b_j, x_j, m_j, PKS_j >$ 형태의 튜플로 구성되는 H_1리스트를 관리하는데, 초기에는 비어있다. A_I가 Type 1 또는 Type 2 해쉬 쿼리를 수행하면, B는 다음과 같이 응답한다.

(1) ID_i가 이미 H_1리스트 상에 $< ID_i, Q_i, T_i, c_i, b_i, x_i, m_i, PKS_i >$튜플로 존재한다면, B는 Type 1 해쉬 쿼리에 대해서는 $H_1(ID_i) = Q_i$로 응답하고, Type 2 해쉬 쿼리에 대해서는 $H_1(ID_i, PKS_i) = T_i$로 응답한다.

(2) 그렇지 않으면, B는 랜덤하게 $c_i \in \{0,1\}$를 선택하는데, $\Pr[c_i = 0] = \delta$이다. δ는 뒤에서 설명한다.

(3) 만약, $c_i = 0$이면, B는 랜덤하게 $b_i, x_i, a_i \in Z_q^*$, $m_i \in MP$를 선택한 후, $Q_i = b_i P$, $E1 = a_i x_i m_i P$, $E2 = \frac{1}{a_i} b_i P_0, E3 = \frac{1}{a_i} b_i P, T_i = Q_i + E1 + E2 + E3$, $E4 = \frac{1}{a_i} T_i$를 계산한다.

(4) 만약, $c_i = 1$이면, B는 랜덤하게 $m_i \in MP$를 선택한 후, $b_i = m_i$와 $x_i = \perp$로 한다. PKS에서 랜덤하게 $PKS_I = < Q_I, a_I x M, \frac{1}{a_I} s Q_I, \frac{1}{a_I} Q_I, \frac{1}{a_I} T_I >$선택 후, $Q_i = b_i Q_I, E1 = b_i a_I x M, E2 = b_i \frac{1}{a_I} s Q_I$ $E3 = b_i \frac{1}{a_I} Q_I, E4 = b_i \frac{1}{a_I} T_I, T_i = Q_i + E1 + E2 + E3$를 계산한다.

(5) $PKS_i = < E1, E2, E3, E4 >$로 하고, H_1리스트에 튜플 $< ID_i, Q_i, T_i, c_i, b_i, x_i, m_i, PKS_i >$를 추가한다. Type 1 해쉬 쿼리에 대해서, $H_1(ID_i) = Q_i$로 응답하고, Type 2 해쉬 쿼리에서 대해서는 $H_1(ID_i, PKS_i) = T_i$로 응답한다.

Phase 1: A_I는 B로부터 $params$를 받은 후에, 다음의 쿼리를 수행하면서, Phase 1 공격을 수행한다. 각각의 쿼리에 대해서, B는 다음과 같이 응답한다.

- $< ID_i >$에 대한 공개키 집합 쿼리: $< ID_i >$에 대한 H_1 해쉬 쿼리를 수행하여 H_1리스트에서 해당 튜플을 얻은 후, 튜플에서 PKS_i를 공개키 집합으로 응답한다.
- $< ID_i >$에 대한 부분 공개키 쿼리: $< ID_i >$에 대한 H_1 해쉬 쿼리를 수행하여 H_1리스트에서 해당 튜플을 얻는다. 만약, $c_i = 1$이면, 시뮬레이션을 중단하고, 그렇지 않으면, $b_i P_0$로 응답한다.
- $< ID_i >$에 대한 부분 복호키 쿼리: $< ID_i >$에 대한 H_1 해쉬 쿼리를 수행하여 H_1리스트에서 해당 튜플을 얻는다. 만약, $c_i = 1$이면, 시뮬레이션을 중단하고, 그렇지 않으면, $m_i P_0$로 응답한다.
- $< ID_i >$에 대한 복호키 쿼리: $< ID_i >$에 대한 H_1 해쉬 쿼리를 수행하여 H_1리스트에서 해당 튜플을 얻는다. 만약, $c_i = 1$이면, 시뮬레이션을 중단하고, 그렇지 않으면, $x_i m_i P_0$로 응답한다.
- $< ID_i, C_i >$에 대한 복호 쿼리: $C_i = < U_i, V_i, W_i >$는 ID_i에 대한 유효한 공개키 집합 PKS_i에 의해 암호화된 암호문이다. B는 다음과 같이 응답한다.

(1) ID_i에 대한 H_1 해쉬 쿼리를 수행하여 H_1리스트에

서 튜플 $< ID_i, Q_i, T_i, c_i, b_i, x_i, m_i, PKS_i >$를 얻는다.

(2) $c_i = 0$이면 B는 공개키 $< ID_i, PKS_i >$에 대한 복호키 쿼리를 수행하여 복호키 $D_i = x_i m_i P_0$를 얻은 다음, C_i를 복호하는데 사용한다.

(3) $c_i = 1$이면 $Q_i = b_i Q_I$이고 $PKS_i = < Q_i, b_i a_I x M, \frac{1}{a_I} s b_i Q_I, \frac{1}{a_I} b_i Q_I, \frac{1}{a_I} b_i T_I >$이며, 이에 상응하는 복호키는 $D_i = b_i s x M$이다. B는 $C_i' = < b_i U, V, W >$를 생성하여 CH에게 전달한다. CH의 복호키는 $D = xsM$이므로, 다음과 같이 복호된다.

$$\hat{e}(D, b_i U) = \hat{e}(xsM, b_i U) = \hat{e}(b_i xsM, U) = \hat{e}(D_i, U)$$

즉, MU-PKE 스킴에서 D_i를 이용하여 C_i를 복호하는 것은 **FullMultiPub**에서 D를 이용하여 C_i'를 복호하는 것과 같다. 따라서, CH는 C_i에 대한 올바른 복호문을 생성하고, B는 이를 A_I에게 전달한다.

Challenge: A_I가 Phase 1 수행을 종료하면, 도전되어질 ID_{ch} 및 두 개의 메시지 M_0, M_1를 선택한다. B는 다음과 같이 응답한다.

(1) ID_{ch}에 대한 H_1해쉬 쿼리를 수행하여 해당 튜플 $< ID_{ch}, Q_{ch}, T_{ch}, c, b, x, m, PKS_{ch} >$를 얻는다.

(2) $c = 0$이면, B는 시뮬레이션을 중단한다.

(3) $c = 1$이면, B는 M_0, M_1과 $Q_I = b^{-1} Q_{ch}$를 CH에게 전달한다. CH는 도전 암호문 $C_{ch} = < U', V', W' >$를 생성한다. 여기서, C_{ch}는 랜덤 $g \in \{0,1\}$에 대해서, 공개키 PKS_I 및 K_{pub}을 이용한 M_g에 대한 **FullMultiPub** 암호문이다. B는 $C^* = < b^{-1} U', V', W' >$를 계산하여, C^*를 A_I에게 전달한다.

$Q_{ch} = bQ_I$이고, $PKS_{ch} = < Q_{ch}, ba_I x M, \frac{1}{a_I} sbQ_I, \frac{1}{a_I} bQ_I, \frac{1}{a_I} bT_I >$이므로, MU-PKE 스킴에서 공개키에 상응하는 복호키는 $D_{ch} = bsxM$이다.

$\hat{e}(b^{-1}U', D_{ch}) = \hat{e}(b^{-1}U', bD) = \hat{e}(U', D)$ 이므로, C^*는 M_g에 대하여 ID_{ch} 및 PKS_{ch}를 이용한MU-PKE 암호문이다.

Phase 2: A_I는 공개키 집합 쿼리, 부분 공개키 쿼리, 부분 복호키 쿼리, 복호키 쿼리 및 복호 쿼리를 수행하는데, B는 복호 쿼리를 제외하고는 Phase 1과 똑같이 응답한다.

• 복호 쿼리: Phase 1에서와 같이 응답하는데, 만약 도전자에게 전달되는 암호문이 C_{ch}와 동일하면, 시뮬레이션을 중단한다.

Guess: A_I는 g에 대한 추측 g'을 출력한다.

만약 B가 시뮬레이션을 중간에 중단하지 않으면, A_I가 봤을 때, A_I의 공격은 실제 공격과 동일하고, $|\Pr[g = g'] - \frac{1}{2}| \geq \epsilon(k)$이다. B는 모든 H_1 쿼리에 대해서, 실제 공격에서와 마찬가지로 균등하고(uniformly) 독립적인(independently) 분포로 응답하고, A_I의 쿼리에 대한 응답은 모두 유효하다. 또한, 도전자가 생성하여 A_I에게 주는 암호문 C^*는 MU-PKE 스킴에서 유효한 ID_{ch} 및 PKS_{ch}에 의한 암호문이므로, 알고리즘 A_I의 정의에 따라서, $g = g'$일 확률은 적어도 $\epsilon(k) + \frac{1}{2}$이다.

Probability: B가 시뮬레이션을 중단하지 않을 확률을 분석하기 위해, B가 시뮬레이션을 중단하는 경우를 분석한다. 다음 중 하나의 경우가 발생하면 B는 시뮬레이션을 중단한다.

(1) EV1: Phase 1 및 Phase 2에서 A_I가 ID_i에 대한 부분 공개키 쿼리를 수행했을 때, B가 중단하는 경우로, ID_i에 대해서 $c_i = 1$이면 B는 중단한다.

(2) EV2: Phase 1 및 Phase 2에서 A_I가 ID_i에 대한 부분 복호키 쿼리를 수행했을 때, B가 중단하는 경우로, ID_i에 대해서 $c_i = 1$이면 B는 중단한다.

(3) EV3: Phase 1 및 Phase 2에서 A_I가 ID_i에 대한 복호키 쿼리를 수행했을 때, B가 중단하는 경우로, ID_i에 대해서 $c_i = 1$이면 B는 중단한다.

(4) EV4: A_I가 도전해야 할 아이디 ID_{ch}에 대한 쿼리를 수행했을 때, B가 중단하는 경우로, ID_{ch}에 대해서 $c_{ch} = 0$이면 B는 중단한다.

(5) EV5: Phase 2에서, A_I가 $< ID_i, C_i >$에 대한 복호 쿼리를 수행했을 때, B가 중단하는 경우로, 도전자에게 넘겨질 암호문 C_i'이 C_{ch}와 동일하면 B는 중단한다.

결국, B가 시뮬레이션을 중단하지 않을 확률은 위의 모든 경우가 한 번도 발생하지 않는 경우로, $\Pr[c = 0] = \delta$이므로,

$$P[-EV1 \wedge -EV2 \wedge -EV3 \wedge -EV4 \wedge -EV5]$$
$$\geq \delta^{q_{PPK} + q_{PD} + q_{DK} + q_D}(1 - \delta)$$

이다.

이 확률을 계산하는 방법은 Boneh-Franklin의 IBE 스킴에서 $BasicPub^{hy}$ 공격자가 시뮬레이션을 중단하는 확률 분석과 유사하므로 생략한다. 위의 확률 값은 $\delta_O = 1 - \frac{1}{q_{PPK}+q_{PD}+q_{DK}+q_D+1}$일 때 최대화되므로, δ_O를 위의 확률 값에 적용하면, B가 시뮬레이션을 중단하지 않을 확률은 적어도 $1/e(1+q_{PPK}+q_{PD}+q_{DK}+q_D)$이다. 따라서 A_I가 MU-PKE에 대해서 $\epsilon(k)$만큼의 advantage를 가질 때, A_I를 이용하여 **FullMultiPub**에 대해서 적어도 $\frac{\epsilon(k)}{e(1+q_{PPK}+q_{PD}+q_{DK}+q_D)}$만큼의 advantage를 갖는 IND-CCA 공격자 B가 존재한다.

보조 정리 2. H_1은 랜덤 오라클이고, A_{II}는 MU-PKE 스킴에 대해서 $\epsilon(k)$만큼의 advantage를 갖는 Type-2 IND-MUP-CCA 공격자라고 하자. A_{II}는 최대 $q_{PK}>0$만큼의 공개키 집합 쿼리, $q_{PD}>0$만큼의 부분 복호키 쿼리, $q_{DK}>0$만큼의 복호키 쿼리, $q_D>0$만큼의 복호 쿼리를 수행한다. 그러면, **FullMultiPub**에 대해서 적어도 $\frac{\epsilon(k)}{e(1+q_{DK}+q_D)}$만큼의 advantage를 갖는 IND-CCA 공격자 B가 존재하며, 수행시간은 $O(time(A_{II}))$이다.

증명. 보조 정리 2의 증명은 다음의 두 가지 사실만 제외하고 보조 정리 1의 증명과 동일하다. A_{II}는 KGC의 매스터 키를 알고 있으므로, 모든 공개키에 대해서 부분 공개키를 생성할 수 있기 때문에 (1) A_{II}는 부분 공개키 쿼리를 수행하지 않는다. 또한, (2) A_{II}의 부분 복호키 쿼리에서 시뮬레이션이 중단되는 일이 발생하지 않는다. 따라서 B가 시뮬레이션을 중단할 경우는 $EV3, EV4, EV5$로 한정되며, 그 확률은 다음과 같다.

$$\begin{aligned}&\Pr[-EV3 \wedge -EV4 \wedge -EV5]\\&= \Pr[-EV3 \wedge -EV5 | -EV4]\Pr[-EV4]\\&\geq \delta^{q_{DK}+q_D}(1-\delta)\end{aligned}$$

위의 확률을 비롯하여, A_{II}를 이용하여 B가 **FullMultiPub**에 대한 IND-CCA 공격을 성공할 확률에 대한 분석은 보조 정리 1과 동일하므로 생략한다.

보조 정리 3. H_2는 랜덤 오라클이고, A는 **BasicMultiPub**에 대해서 $\epsilon(k)$만큼의 advantage를 갖는 IND-CPA 공격자라고 하자. A는 H_2에 대해서 최대 $q_{H_2}>0$만큼의 쿼리를 수행한다. 그러면, 적어도 $2\epsilon(k)/q_{H_2}$만큼의 advantage를 갖고 BDH 문제를 풀 수 있는 알고리즘 B가 존재한다. B의 수행 시간은 $O(time(A))$이다.

증명. B는 IG에 의해 생성된 BDH 파라미터 $(G_1, G_2, \hat{e})$와 BDH 문제에 대한 랜덤 인스턴스 $<P, aP, bP, cP> = <P, P_1, P_2, P_3>$를 알고 있다고 가정한다. 여기서, P는 그룹 G_1의 생성자이고, a,b,c는 Z_q^* 상에서 랜덤하게 선택된 정수이다. 결국, $D = \hat{e}(P,P)^{abc} \in G_2$가 BDH 문제의 해답인데, 다음에서, B가 A를 이용하여 D를 찾아내는 방법을 보인다.

Setup: B는 **BasicMultiPub**의 공개키 $K_{pub} = <G_1, G_2, \hat{e}, n, P, P_0, PKS, H_2>$을 생성하여 A에게 제공한다. 단, $P_0 = P_1$이고, $I = 1,...,t$에 대해서,

$$\begin{aligned}PKS_I &= <E0, E1, E2, E3, E4>\\&= <b_I P, a_I P_2, \frac{1}{a_I} b_I P_0, \frac{1}{a_I} b_I P, \frac{1}{a_I} T_I>\end{aligned}$$

이다.

a_I와 b_I는 Z_q^*에서 랜덤하게 선택한 정수이다. **BasicMultiPub** 알고리즘의 정의에 따라서, 공개키 K_{pub}에 해당하는 복호키는 $DK = aP_2 = abP$이다. H_2는 B에 의해 통제되는 랜덤 오라클로써, A는 언제든지 해쉬 쿼리를 수행할 수 있는데, B는 다음과 같이 처리한다.

H_2 해쉬 쿼리: B는 $<X_i, H_i>$튜플로 구성되는 H_2리스트를 관리한다. 초기에는 비어있고, A가 X_i에 대해서 H_2 해쉬 쿼리를 수행하면, 다음과 같이 응답한다.

(1) 만약, X_i가 이미 H_2리스트 상에 $<X_j, H_j>$ 튜플로 존재하면, $H_2(X_i) = H_j$로 응답한다.

(2) 그렇지 않으면, 랜덤 문자열 $H_i \in \{0,1\}^n$를 선택한 후, H_2리스트에 새로운 튜플 $<X_i, H_i>$를 추가하고, $H_2(X_i) = H_i$로 응답한다.

Challenge: A는 임의의 공개키 PKS_I와 도전할 두 개의 메시지 M_0, M_1을 출력한다. B는 랜덤 문자열 $R \in \{0,1\}^n$을 선택하여, 암호문 $C = <b_I P_3, R>$를 생성한 후, C를 A에게 전달한다. C에 대한 복호과정은

$$R \oplus H_2(\hat{e}(DK, b_I P_3)) = R \oplus H_2(D^{b_I})$$

이다.

Guess: A는 자신의 추측 $c' \in \{0,1\}$를 출력한다. 이

시점에서, B는 H_2리스트에서 임의로 하나의 튜플 $< X_j, H_j >$를 선택한 다음, $D' = X_j^{b_l^{-1}}$를 주어진 BDH 인스턴스에 대한 해답으로 출력한다.

B는 도전자와 H_2에 대한 랜덤 오라클을 시뮬레이트 하는데, A의 시각에서는 실제 공격 상황과 동일하다. 만약, A가 최대 q_{H_2}만큼의 H_2해쉬 쿼리를 수행한다고 했을 때, B가 올바른 D를 출력할 확률은 적어도 $2\epsilon(k)/q_{H_2}$이다. 이 확률에 대한 분석은 [3]에서의 Lemma 4.3과 동일하므로 생략한다.

정리 1에 대한 증명. 보조 정리 1과 보조 정리 2는 MU-PKE 스킴에 대해서 $\epsilon(k)$만큼의 advantage를 갖는 IND-MUP-CCA 공격자가 존재하면, **FullMultiPub**에 대해서 $\epsilon_1(k)$만큼의 advantage를 갖는 IND-CCA 공격자가 존재함을 보인다. Fujisaki-Okamoto 정리와 보조 정리 3을 통해서, 만약 **FullMultiPub**에 대해서 $\epsilon_1(k)$만큼의 advantage를 갖는 IND-CCA 공격자 A가 존재하면, 적어도 $2FO_{adv}(\epsilon_1(k), q_{H_1}, q_{H_3}, q_D)/q_{H_2}$만큼의 advantage로 BDH 문제를 해결할 수 있는 공격자 B가 존재함을 알 수 있다. 결과적으로, 만약 MU-PKE 스킴에 대해서 $\epsilon(k)$만큼의 advantage를 갖는 Type-1 IND-MUP-CCA 공격자와 Type-2 IND-MUP-CCA 공격자가 존재하면, 적어도 식(3) 만큼의 advantage로 BDH 문제를 해결하는 알고리즘 B가 존재한다.

정리 2. 해쉬 함수 H_1이 랜덤 오라클이면, MU-PKE 스킴에서 공개키는 불연계성을 만족한다.

정리 2에 대한 증명. 위 정리를 증명하기 위해서, **UnLinkPub**으로 표기되는 불연계성을 갖는 공개키 생성 알고리즘을 정의하는데, 표 2와 같다.

만약 IND-LINK 게임에서 $\epsilon(k)$만큼의 advantage를 갖는 IND-LINK 공격자 A_L이 존재한다면, A_L은 **UnLinkPub**에 대해서도 적어도 동일한 advantage를 가짐을 보인다.

Setup: 먼저 두 도전자 CH_0, CH_1는 **UnLinkPub**의 Setup 알고리즘을 수행한다. DK_0는 CH_0의 복호키이고, DK_1는 CH_1의 복호키라고 하자. 도전자들은 $sparams$를 공유한다. 도전자들은 $params = < G_1, G_2, \hat{e}, n, P, P_0, H_1, H_2, H_3, H_4 >$를 A_L에게 제공하는데, H_1은 도전자들에 의해 통제되는 랜덤 오라클이다.

표 2. **UnLinkPub** 알고리즘
Table 2. **UnLinkPub** Algorithm

- Setup: k 및 IG에 대해서,
- $(\hat{e}, G_1, G_2)$ 생성
- 랜덤 정수 $s, b_0, b_1, d_1, d_2, q_1, q_2 \in Z_q^*$ 와 랜덤 $P, P_1, T \in G_1$ 선택
- $P_0 = sP$ 계산 후, 시스템 파라미터 $sparams = < \hat{e}, G_1, G_2, P, P_0, P_1, T, b_0, b_1, d_1, d_2, q_1, q_2, s >$ 설정
- 두 개의 복호키 $DK_0 = sb_0P_1$, $DK_1 = sb_1P_1$ 생성
- MU-PKE에서 정의된 대로 암호학적 해쉬 함수 H_2, H_3, H_4를 선택
- 공개 파라미터 집합 $K_{pub} = < \hat{e}, G_1, G_2, n, P, P_0, H_2, H_3, H_4 >$생성 후, 공개
- $< ID_j, CH_i, sparams >$에 대한 G_{PKS}: 아이디 ID_j, 도전자 CH_i의 정보 및 $sparams$에 대해서,
- $H_1(ID_j) = Q_j = q_j\overline{b_i}P$ 와 $H_1(ID_j, PSK_j) = T_j = \overline{b_i}T$ 계산 (단, $\overline{b_i} = b_{1-i}$)
- $a_j = d_j\overline{b_i} \in Z_q^*$ 계산 후, 공개키 $PKS_j = < a_jb_iP_1, \frac{1}{a_j}sQ_j, \frac{1}{a_j}Q_j, \frac{1}{a_j}T_j >$ 생성

Challenge: A_L는 두 개의 랜덤 아이디 $ID_1, ID_2 \in \{0,1\}^*$를 생성한 다음, 도전자에게 각각에 대한 공개키 집합 쿼리를 생성한다. 도전자들은 다음과 같이 응답한다. 단, 도전자는 $< ID_j, CH_i, Q_j, T_j, d_j, q_j >$ 튜플로 구성되는 PKS 리스트를 관리한다.

(1) CH_0가 랜덤 $c_0 \in \{0,1\}$를 선택하면, CH_1는 $c_1 = 1 - c_0$로 정한다.

(2) $c_i = 1$인 CH_i가 $< ID_1, CH_i, sparams >$에 대한 G_{PKS}를 수행하여 PKS_1을 출력한다. CH_i는 튜플 $< ID_1, CH_i, Q_1, T_1, d_1, q_1 >$를 PKS에 추가한다.

(3) CH_0가 다시 랜덤 $c_0' \in \{0,1\}$을 선택하면, CH_1는 $c_1' = 1 - c_0'$로 정한다.

(4) $c_i' = 1$인 CH_i가 $< ID_2, CH_i, sparams >$에 대한 G_{PKS}를 수행하여, PKS_2를 출력한다. CH_i는 튜플 $< ID_2, CH_i, Q_2, T_2, d_2, q_2 >$를 PKS에 추가한다.

(5) 도전자들은 $c = c_0 \oplus c_0'$를 계산한 후, A_L에게 PKS_1, PKS_2를 제공한다.

A_L은 $< ID_j >$ 또는 $< ID_j, PKS_j >$에 대한 H_1해쉬 쿼리를 수행할 수 있는데, 도전자들은 PKS 리스트를 검사하여, A_L에게 각각에 대한 쿼리 응답 $H_1(ID_j) = Q_j$과 $H_1(ID_j, PKS_j) = T_j$를 제공한다.

Guess: A_L은 c에 대한 자신의 추측 c'를 생성한다.

Claim: A_L의 시각은 실제 공격 상황과 동일하며, $|\Pr[c = c'] - \frac{1}{2}| \geq \epsilon(k)$이다.

Claim에 대한 증명: 도전자들은 균등하고 독립적인 분포로 b_i, d_j, q_j를 선택한다. 따라서 $Q_j = H_1(ID_j)$, $T_j = H_1(ID_j, PKS_j)$ 및 a_j 또한 랜덤하게 계산되고, 생성되는 공개키는 모두 유효하다. 따라서 A_L는 적어도 $\epsilon(k)$의 확률로 $c' = c$를 생성한다.

결과적으로, A_L는 **UnLinkPub**에 대해서 무시할 수 없는 advantage를 가질 수 없음을 증명한다. 두 명의 도전자가 위의 Challenge 단계에서 ID_1, ID_2에 대한 두 개의 공개키 집합을 생성할 수 있는 모든 경우는 총 네가지로, 각각의 경우에 대해서, A_L에게 주어지는 두 개의 공개키 집합은 다음과 같다.

• Case 1 (CH_0, CH_0): CH_0가 PKS_1과 PKS_2를 모두 생성한 경우로, A_L은

$$PKS_1 = < d_1 b_1 b_0 P_1, \frac{1}{d_1 b_1} s q_1 b_1 P, \frac{1}{d_1 b_1} q_1 b_1 P, \frac{1}{d_1 b_1} b_1 T >,$$

$$PKS_2 = < d_2 b_1 b_0 P_1, \frac{1}{d_2 b_1} s q_2 b_1 P, \frac{1}{d_2 b_1} q_2 b_1 P, \frac{1}{d_2 b_1} b_1 T >$$

를 받고, $g_1 = \hat{e}(b_0 P_1, s q_1 b_1 P)$, $g_2 = \hat{e}(b_0 P_1, s q_2 b_1 P)$를 계산

• Case 2 (CH_0, CH_1): CH_0가 PKS_1를 생성하고, CH_1이 PKS_2를 생성한 경우로, A_L은

$$PKS_1 = < d_1 b_1 b_0 P_1, \frac{1}{d_1 b_1} s q_1 b_1 P, \frac{1}{d_1 b_1} q_1 b_1 P, \frac{1}{d_1 b_1} b_1 T >,$$

$$PKS_2 = < d_2 b_0 b_1 P_1, \frac{1}{d_2 b_0} s q_2 b_0 P, \frac{1}{d_2 b_0} q_2 b_0 P, \frac{1}{d_2 b_0} b_0 T >$$

를 받고, $g_1 = \hat{e}(b_0 P_1, s q_1 b_1 P), g_2 = \hat{e}(b_1 P_1, s q_2 b_0 P)$를 계산

• Case 3 (CH_1, CH_0): CH_1이 PKS_1를 생성하고, CH_0가 PKS_2를 생성한 경우로, A_L은

$$PKS_1 = < d_1 b_0 b_1 P_1, \frac{1}{d_1 b_0} s q_1 b_0 P, \frac{1}{d_1 b_0} q_1 b_0 P, \frac{1}{d_1 b_0} b_0 T >,$$

$$PKS_2 = < d_2 b_1 b_0 P_1, \frac{1}{d_2 b_1} s q_2 b_1 P, \frac{1}{d_2 b_1} q_2 b_1 P, \frac{1}{d_2 b_1} b_1 T >$$

를 받고, $g_1 = \hat{e}(b_1 P_1, s q_1 b_0 P)$, $g_2 = \hat{e}(b_0 P_1, s q_2 b_1 P)$를 계산

• Case 4 (CH_1, CH_1): CH_1이 PKS_1와 PKS_2를 모두 생성한 경우로, A_L은

$$PKS_1 = < d_1 b_0 b_1 P_1, \frac{1}{d_1 b_0} s q_1 b_0 P, \frac{1}{d_1 b_0} q_1 b_0 P, \frac{1}{d_1 b_0} b_0 T >,$$

$$PKS_2 = < d_2 b_0 b_1 P_1, \frac{1}{d_2 b_0} s q_2 b_0 P, \frac{1}{d_2 b_0} q_2 b_0 P, \frac{1}{d_2 b_0} b_0 T >$$

를 받고, $g_1 = \hat{e}(b_1 P_1, s q_1 b_0 P)$, $g_2 = \hat{e}(b_1 P_1, s q_2 b_0 P)$를 계산

위에서, A_L이 받는 공개키 정보 $< PKS_1, PKS_2 >$는 도전자가 누구인가와 상관없이 항상 동일하다. 따라서 A_L이 두 공개키 집합이 동일한 도전자에 의해 생성되었는지 아닌지를 구분하는 것은 불가능하다. A_L이 동일한 도전자에게 속하는 공개키를 서로 연결하지 못하므로, A_L는 IND-LINK 게임에서 무시할 수 없는 advantage를 가질 수 없다.

V. 복호키 갱신 프로토콜

MU-PKE 스킴에서는 하나의 복호키로 모든 공개키에 의해 암호화된 암호문을 모두 복호할 수 있기 때문에, 복호키가 도난당하거나 공격자에게 노출되게 되면, 상응하는 공개키가 사용되는 모든 응용 서비스들과 더 이상 안전한 통신을 할 수 없게 되므로, 주기적인 복호키 갱신이 요구된다. 따라서 복호키 갱신 기능을 제공하는 확장된 MU-PKE(ExMU-PKE) 스킴을 제안한다. ExMU-PKE 스킴에서는, 시간 구간을 정해서, 매 시간 구간마다 복호키를 갱신하기 위해서, 두 가지 형태의 복호키를 사용한다. 하나는 장기(long-term) 복호키로 MU-PKE 스킴에서의 DK와 같고, 다른 하나는 ExMU-PKE 스킴에서 새롭게 정의되는 단기(short-term) 복호키 SDK로 매 시간 구간에 새롭게 생성된다. 즉, SDK_j는 특정 시간 구간 j에서만 효력을 갖는다. 시간 구간은 KGC가 정하며, 매 시간 구간이 시작되는 시점에, KGC는 원래 비밀키-공개키 쌍 이외에, 새로운 단기 비밀키-공개키 쌍을 생성한 후, 공개키를 공개한다. 그리고 KGC는 새로 생성한 비밀키를 이용하여 KGC가 관리하고 있는 사용자들의 부분 복호키를 재생성하여 일괄적으로 전송한다. 사용자들은 새로운 부분 복호키를 이용하여 최종 복호키를 주기적으로 갱신한다. 복호키 갱신을 허용하는 ExMU-PKE 스킴의 구체

적인 프로토콜은 다음과 같다.

(1) Setup: t개의 시간 구간 $i=0,...,t-1$을 생성한 후, 매 시간 구간 i가 시작될 때, 랜덤 값 $s_i \in Z_q^*$를 생성하여 단기 공개키 $P_i = s_i P$를 생성하는 것 외에는, MU-PKE 스킴의 Setup 알고리즘과 동일하다. 결국, KGC는 장기 공개키 $P_0 = sP$와 함께 각 시간 구간에서만 사용되는 단기 공개키 $P_i = s_i P$를 관리하며, 매 시간 구간이 시작될 때마다 P_i를 공개한다.

(2) Gen-DK: 수신자는 단기 복호키 $SDK_{A,i}$를 추가로 생성하여 안전하게 관리하는 것 이외에는 MU-PKE 스킴의 Gen-DK와 동일하다. 매 시간 구간 $i \geq 0$가 시작될 때, KGC는 $(params, MID_A, s_i)$에 대해서 Extract-Partial-Decryption-Key 알고리즘을 수행하여 주기적 부분 복호키 $PSDK_{A,i} = s_i M_A$를 생성하여 수신자에게 전송한다. 수신자는 $(params, PSDK_{A,i}, x_A)$에 대해 Set-Decryption-Key 알고리즘을 수행하여 단기 복호키 $SDK_{A,i} = x_A PSDK_{A,i} = x_A s_i M_A$ 생성한다.

(3) Gen-PK: MU-PKE 스킴의 Gen-PK 알고리즘과 동일하다.

(4) Encryption: 시간 구간 $i \geq 0$에, 송신자 S_i는 해당 시간 구간에 대한 KGC의 단기 공개키 P_i를 얻는다. MU-PKE의 Encryption 알고리즘과 같이, 수신자의 $ID_{A,j}$와 $PKS_{A,j}$의 유효성을 검사한 다음, 유효하면

$$g = \hat{e}(E1, E2)\hat{e}(E1, P_i) = \hat{e}(x_A s M_A, Q)\hat{e}(a_j x_A s_i M_A, P)$$

를 계산한다. 랜덤 $\sigma \in \{0,1\}^n$를 선택하여, $r = H_3(\sigma, M)$을 계산한 후, 암호문 C을 생성한다.

$$C = < U, W, V, T > = < rQ, rP, \sigma \oplus H_2(g^r), M \oplus H_4(\sigma) >$$

(5) Decryption: 수신자는 $(params, DK_A, SDK_i, C)$에 대해서, 먼저 $a = H_0(MID_A \| ID_{A,j})$와 $V' = aW$를 계산한 후, $V \oplus H_2(\hat{e}(DK_A, U)\hat{e}(SDK_{A,i}, V')) = \sigma'$과 $T \oplus H_4(\sigma') = M'$을 계산한다. $r' = H_3(\sigma', M')$을 계산한 다음, $W = r'P$가 성립하는지 검사한다. 만약 등식이 성립하지 않으면 복호를 중단하고, 그렇지 않으면 M'을 평문으로 출력한다.

장기 복호키와 단기 복호키가 모두 있어야만, 암호문을 복호할 수 있으므로, 장기 복호키만 있고, 매 시간 주기에 유효한 단기 복호키를 얻지 못한 수신자 또는 공격자는 암호문을 제대로 복호할 수 없다. 단기 복호키는 매번 랜덤하게 생성되므로, 공격자가 이전 시간 구간에 유효한 단기 복호키를 얻었다 할지라도, 현재 유효한 단기 복호키 정보를 알 수 없으며, 현재 시간 주기에서 생성된 암호문을 복호하는 것 또한 불가능하다.

VI. 결 론

본 논문에서는 다수의 아이디 기반 공개키를 사용하되, 하나의 복호키로 모든 공개키로 암호화된 암호문을 복호할 수 있는 불연계성을 갖는 다중 공개키 암호(MU-PKE) 시스템을 새롭게 제안하였다. 각 공개키는 이미 다른 사용자 또는 응용 서비스에서 사용되고 있는 아이디에 기반을 두어 생성되며, 공개키와 복호키 생성 시, 키 소유주와 KGC가 반드시 함께 생성해야 하므로, 별도의 인증서의 사용 없이 공개키 인증이 가능하고, 키 예탁 문제도 해결하였다. 생성된 공개키들은 서로 불연계성을 가지므로, 공격자가 알려진 아이디 또는 공개키를 이용하여 사용자의 다른 의미 있는 정보의 수집 및 추적을 어렵게 하므로, 개인 프라이버시를 보장해준다. 제안한 스킴의 안전성 증명을 위해 MU-PKE 스킴에 대한 선택적 암호문 공격 모델(IND-MUP-CCA)을 정의하였고 랜덤 오라클 모델에서 안전함을 증명하였다. 또한, 하나의 복호키 사용에 따른 안전성 문제를 해결하기 위해 주기적으로 복호키를 갱신할 수 있는 확장된 ExMU-PKE 스킴도 함께 제안하였다.

본 논문에서 다중 공개키 암호 시스템에 대한 선택적 암호문 공격(IND-MUP-CCA) 모델을 새롭게 정의하였는데, 추후 제안한 보안 모델에 대한 지속적인 검증과정이 필요하다. 또한, 선택적 암호문 공격 이외에 다중 공개키 사용에 따라 발생 가능한 여러 다른 공격 형태 및 이들에 대한 안전성 분석이 요구된다. 그리고 제안한 스킴의 효율성을 증대시킬 수 있는 연구가 더 필요하다. 특히, 복호키 갱신에 따른 연산 량을 줄이고, KGC의 역할을 최소화할 수 있는 방안이 요구된다. 또한, 다수의 공개키 중에서 더 이상 사용하지 않는 특정 공개키를 파기시킬 수 있는 방법도 향후 필요하다.

참 고 문 헌

[1] S. Al-Riyami and K. Paterson, "Certicateless Public Key Cryptography," Advances in Cryptology-ASIACRYPT'03, LNCS 2894, pp. 452-473, 2003.

[2] M. Bellare, A. Desai, D. Pointcheval and P.

Rogaway, "Relations among Notions of Security for Public-Key Encryption Schemes," Advances in Cryptology-Crypto'98, LNCS 1462, pp. 26-45, 1998.
[3] D. Boneh and M. Flanklin, "Identity-Based Encrytion from the Weil pairing," SIAM J. of Computing, vol. 32, no. 3, pp. 586-615, 2003.
[4] W. Die and M. E. Hellman, "New Directions in Cryptography," IEEE Trans. on Information Theory, vol. IT-22, no. 6, pp. 644-654, 1976.
[5] E. Fujisaki and T. Okamoto, "Secure Integration of Asymmetric and Symmetric Encryption Schemes," Advances in Cryptology-Crypto 99, LNCS 1666, pp. 537-554, 1999.
[6] C. Gentry, "Certicate-Based Encryption and the Certicate Revocation Problem," Advances in Cryptology-Eurocrypt'03, LNCS 2656, pp. 272-293, 2003.
[7] Kohnfelder, "Toward a Practical Public Key Cryptosystems," Bachelor's thesis, MIT Department of Electronic Engineering, 1978.
[8] A. Lysyanskaya, R. Rivest, A. Sahai and S. Wolf, "Pseudonym Systems," Selected Areas in Cryptography, vol. 1758, 1999.
[9] R. C. Merkle, "Secure Communication Over Insecure Channels," Communications of the ACM, vol. 21, no. 4, pp. 294-299, 1978.
[10] P. Persiano and I. Visconti, "An Anonymous Credential System and a Privacy-Aware PKI," in Proc. of ACISP 03, LNCS 2727, pp. 27-38, 2003.
[11] P. Persiano and I. Visconti, "An Ecient and Usable Multi-show Non-transferable Anonymous Credential System," in Proc. of FC 04, LNCS 3110, pp. 196-211, 2004.
[12] C. Racko and D. Simon, "Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attacks," Advances in Cryptology-Crypto'91, LNCS 576, pp. 433-444, 1991.
[13] Y. Tamura and A. Miyaji, "Anonymity-enhanced Pseudonym System," in Proc. of ACNS 03, LNCS 2846, pp. 33-47, 2003.
[14] E. R. Verheul, "Self-Blindable Credential Certicates from the Weil Pairing," Advances in Cryptology - Asiacrypt 01, LNCS 2248, pp. 533-551, 2001.

저 자 소 개

박 소 영(정회원)
1998년 이화여자대학교 컴퓨터학과 학사 졸업.
2000년 이화여자대학교 컴퓨터학과 석사 졸업.
2006년 이화여자대학교 컴퓨터학과 박사 졸업.
2006년~현재 Univ. of Central Florida 박사 후 연구원
<주관심분야 : 암호 프로토콜, 정보 보호, 네트워크 보안>

이 상 호(정회원)
1979년 서울대학교 계산통계학과 학사 졸업
1981년 한국과학기술원 전산학과 석사 졸업
1987년 한국과학기술원 전산학과 박사 졸업
1983년~현재. 이화여자대학교 컴퓨터공학과 교수
<주관심분야 : 알고리즘 설계, 암호 및 보안기술 응용 등>