情報保護學會論文誌
第19卷 第2號, 2009. 4

# 다중 수신자 환경에서 키워드 검색 가능한 공개키 암호시스템*

이 현 숙,[1†] 박 종 환,[2] 이 동 훈[2‡]
[1]울릉공대학교, [2]고려대학교

# Public Key Encryption with Keyword Search in Multi-Receiver Setting*

Hyun Sook Rhee,[1†] Jong Hwan Park,[2] Dong Hoon Rhee[2‡]
[1]University of Wollongong, [2]Korea University

요 약

키워드 검색 가능한 공개키 암호 (PEKS)은 검색어에 대한 프라이버시를 제공하기 위해서 Boneh et. al. 에 의해서 처음으로 제안되었다. 키워드 검색 가능한 공개키 암호 (PEKS) 기술은 송신자가 수신자의 공개키로 암호화된 메일 메시지를 이메일 서버에 보내고 서버는 암호문과 송신자에 의해서 생성된 암호화된 쿼리를 이용하여 암호화된 메일 메시지와 암호화된 검색어와의 관련성을 얻는 것이 가능하도록 한다. 이러한 메일 시스템에서는 그룹메일과 같이 하나의 암호화된 메일을 다수의 수신자에게 전송하는 경우를 생각할 수 있다. Hwang과 Lee는 이러한 점을 고려하여 다중 수신자환경에서 PEKS 스킴을 제안하였다. 이러한 다수의 수신자의 환경에서는 전송되는 데이터의 사이즈와 서버의 계산량을 줄이는 것이 중요한 이슈이다. 본 논문에서는 서버측의 페어링(Pairing) 계산량을 줄인 좀더 효율적인 다수의 수신자를 고려한 mPEKS 스킴을 제안한다.

ABSTRACT

To provide the privacy of a keyword, a public key encryption with keyword search(PEKS) firstly was propsed by Boneh et al. The PEKS scheme enables that an email sender sends an encrypted email with receiver's public key to an email server and a server can obtain the relation between the given encrypted email and an encrypted query generated by a receiver. In this email system, we easily consider the situation that a user sends the one identical encrypted email to multi-receiver like as group e-mail. Hwang and Lee proposed a searchable public key encryption considering multi-receivers. To reduce the size of transmission data and the server's computation is important issue in multi-receiver setting. In this paper, we propose an efficient searchable public key encryption for multi-receiver (mPEKS) which is more efficient and reduces the server's pairing computation.

**Keywords:** Public key encryption with keyword search, Searchable encryption

## I. 서 론

컴퓨터와 인터넷의 발전으로 인해 전송, 저장 그리고 이용되어지는 정보의 양이 급격히 증가하고 있다. 이러한 정보들은 개인이 어떠한 서비스를 이용할 경우 개인정보들이 개인의 PC에만 저장되는 것이 아닌 서비스 데이터베이스(예, 카드회사 웹, 인터넷 쇼핑몰 등)나 웹 기반의 이메일 서비스(예, hanmail, hotmail 등)와 같이 우리가 자주 사용하고 있는 시스템 상에도 저장되어지고 사용자의 요구에 따라 이러한 정보들은 빈번히 이동·전송된다. 이렇게 전송 이동 저장되는 정

접수일(2008년 9월 26일), 수정일(2008년 12월 10일), 게재확정일(2009년 1월 30일)
* 이 연구에 참여한 연구자(의 일부)는 '2단계 BK21사업'의 지원비를 받았음.
† 주저자, hyunsook.rhee@gmail.com
‡ 교신저자, donghlee@korea.ac.kr

보들은 해킹 및 바이러스와 같은 침해로 사용자의 개인정보를 누출 및 악용되고 있으며 그로 인한 프라이버시 침해의 결과를 초래하고 있다.

이러한 프라이버시 노출로 인한 피해의 심각성은 최근 옥션이나 GS칼텍스의 개인정보 누출로 인한 피해 사례와 같이 시스템 관리자로부터 사용자의 개인정보가 노출되어 문제가 되는 사례를 간혹 볼 수 있는데도 불구하고 사용자의 대다수는 자신의 민감한 정보가 어떻게 관리되고 있는지 알 수 없다는데 있다[8]. 이러한 문제가 발생하였을 때 개인 사용자와 기업 모두에게 그 피해의 심각성이 큰 이유는 다음과 같다. 첫째, 사용자 측면에서는 개인정보의 주체인 자신에게도 제2 제3의 피해를 야기시킬수 있다는 점에서 그 문제가 심각하다. 둘째, 기업의 입장에서도 문제를 야기시킨것에 대한 피해 보상 뿐 아니라 기업의 이미지의 실추로 인한 소비자 감소 및 기타 이익의 감소로 인한 커다란 경제적 손실의 결과를 초래하게 된다. 앞에서 언급한 옥션의 사례의 경우 개인정보가 노출사고 이후 방문자가 급감하였고 옥션이 쇼핑몰의 특성을 갖는 점에서 보았을 때 그 심각성이 크다. 이러한 내부와 외부 공격자 모두에게 안전하도록 데이터를 저장하기 위해서는 개인정보를 사용자만 아는 암호화키로 암호화하여 데이터베이스에 저장하는 것을 생각할 수 있다. 하지만 데이터를 암호화된 데이터의 형태로 저장하게 되면 데이터베이스 관리가 어렵고 데이터 검색의 효율성이 떨어진다.

이러한 문제점을 해결하기 위해서 Song et. al. [9]은 암호화된 데이터에서의 키워드를 이용한 검색 프로토콜을 제안하였고 그 이후로 크게 3가지 환경(웹 하드모델, email모델, vendor모델)에서 각 환경에 적합하도록 제안되었다[1-7,9]. 이중 Boneh et. al. 가 제안하였던 email환경에서의 키워드를 이용한 검색기술은 메일 송신자가 수신자의 공개키로 메일내용을 암호화하고 암호화된 검색정보를 첨부한 형태로 전송하여 수신자가 검색 가능하도록 하는 기술이다[1]. 처음 Bonehet. al. 가 제안하였던 키워드 검색 가능한 공개키 암호 (PEKS) 스킴의 경우 메일 수신자를 한명으로 제안하고 있다. 하지만 우리가 사용하는 일반적인 메일의 경우 하나의 메시지를 여러명이 동시에 수신하도록 설정하는 것이 가능하다. 따라서, 하나의 메시지를 전송하여 다수의 수신자들이 동시에 암호화된 메일 수신이 가능하면서도 키워드를 이용한 검색이 가능하도록 한다면 데이터 프라이버시를 제공하면서도 사용자의 편리성은 증가할 것이다.

Hwang 과 Lee는 최근에 이러한 점을 고려하여 다수의 사용자를 고려한 PEKS 스킴을 제안하였다[6]. 그들은 다수의 수신자가 검색 가능하도록 하기 위해서 서버에 저장되어야 하는 검색정보의 양과 메일 송신자가 검색정보 생성 시 요구되어지는 계산량을 개선하는데 초점을 맞추었다. 하지만 수신자가 하나의 암호화된 검색어를 서버에게 제공하였을 때, 테스트 단계에서 요구하는 계산량은 PEKS 스킴보다 많은 pairing 계산량을 요구한다. 이때, 송신자가 서버에 전송하는 암호화된 검색정보의 경우 미리 계산(precomputation)하는 것이 가능하다. 하지만 테스트 단계의 계산은 실시간으로 이루어지며 그룹메일과 같이 송신자가 한명이 아닌 n명인 환경을 고려한다면 메일 서버의 특성상 다수의 사용자의 접근이 빈번한 상황이 고려되어지기 때문에 검색을 원하는 메일 수신자가 서버에 접속하여 검색할 때마다 요구되는 테스트 단계에서의 계산량을 줄이는 것은 의미가 있다. 또, 메시지의 수신자에 대한 정보를 제공하고 있지 않은 PEKS 스킴에서는 특히 메일 서버가 테스트 단계에서 데이터베이스에 저장된 암호화된 메시지 전부를 대상으로 전수조사를 해야 하기 때문에 테스트(test)단계에서의 계산량을 줄이는 것은 더욱 의미가 크다. 반면 검색을 위한 검색정보(searchable information)를 구성하는데 필요한 계산들은 오프라인 (off-line)과정에서 미리 계산하는 것이 가능하다.

본 논문에서 제안한 스킴과 Hwang 과 Lee가 제안한 스킴만이 다수의 수신자를 고려하고 있다. 이러한 측면에서 고려하였을 때, 본 논문에서 제안한 스킴은 다수의 수신자를 고려하면서도 테스트 단계의 계산량을 개선하여 같은 조건에서 제안된 Hwang 과 Lee 에 의해서 제안된 스킴보다 훨씬 적은 테스트 단계의 계산량만을 요구하는 "다수의 수신자를 고려한 키워드 검색가능한 공개키 암호 (mPEKS)" 스킴이다. 다음의 표는 n명의 수신자와 m개의 검색어를 고려한 환경에서 $w$개의 암호문이 저장된 암호화된 메일 서버상에서 계산량을 비교한 표이다 (메일환경에서 제안된 검색가능한 공개키 암호화 스킴들은 모두 암호화된 메일에 대한 수신자를 표기하는지 또는 표기하지 않는지에 따라서 요구되어지는 계산량의 증가정도가 유사하다.

따라서 본 논문에서는 일반성을 잃지 않고 암호화된 메일에 대한 수신자를 표기하지 않는 경우를 가정한다).

또, 본논문에서는 mPEKS 스킴의 안전성을 BDHI

〔표 1〕 효율성 분석

| 스킴 | Multi-receiver setting 여부 | 암호문 C의 크기 | Trapdoor의 크기 | 계산량 | | |
|---|---|---|---|---|---|---|
| | | | | 암호문 C 생성 | Trapdoor 생성 | Test 단계 |
| 제안된 스킴 | Multi Receiver setting | $nL_1+m\lambda$ | $\lambda$ | $ne+m(H_1+p)$ | $H_1+e$ | $\nu p$ |
| BCOP04 | One Receiver setting | $L_1+nm\lambda$ | $\lambda$ | $(n+1)e+mH_1+nmp$ | $H_1+e$ | $\nu p$ |
| BSS06 | One Receiver setting | $L_1+nmL_2$ | $\lambda$ | $(nm+1)e+mH_1+nmp$ | $H_1+e$ | $\nu p$ |
| PKL04 | One Receiver setting | $2L_1+nmL_2$ | $\lambda$ | $(m+1)e+m(H_1+p)$ | $H_1+e$ | $\nu p$ |
| HL07 | Multi Receiver setting | $(n+m+1)L_1$ | $3\lambda$ | $(n+2m)e+2mH_1$ | $3(H_1+e)$ | $3\nu p$ |

$H_1$: an admissible encoding 해쉬 함수
$p$: 페어링 계산 (a pairing computation)
$e$: 지수 연산 (an exponential computation)
$L_1$ (or $L_2$) : $G_1$ (또는 $G_2$)의 원소의 길이 (a bit length of an element of $G_1$ (or $G_2$))
$n$: 수신자의 수, $m$: 검색어의 수(검색어 필드의 수), $\nu$: 저장된 메시지의 수

가정 (Bilinear Diffie-Hellman Inversion Assumption) 에 기반을 두고 랜덤 오라클 (random oracle) 모델에서 증명한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 논문에서 안전성의 기반을 두고 있는 가정에 대한 정의를 살펴보고, 본 논문에서 제안하는 모델을 정의한다. 제 3장에서는 프로토콜을 제안하고 안전성을 증명한다. 마지막으로 제 4장에서는 결론을 맺는다 .

## II. 안전성 정의 및 hardness 가정

본 논문에서 제안하는 스킴의 안전성의 기반인 BDHI(Bilinear Diffie-Hellman Inversion Assumption) 와 다중 수신자 환경에서 검색가능한 공개키 암호화시스템(mPEKS)을 정의한다.

### 2.1 페어링 연산과 Hardness 가정

페어링 연산 (Bilinear Pairing). $G_1$ 과 $G_2$ 는 소수 $p$를 위수로 갖는 환군(cyclic group)이라 하고 $g$를 $G_1$의 생성원(generator) 라 하자. 함수 $e: G_1\times G_1\rightarrow G_2$가 다음의 성질을 만족하면 우리는 이 함수를 bilinear 함수 이라고 부르고, bilinear 함수 $e: G_1\times G_1\rightarrow G_2$과 그룹 $G_2$가 존재하면 $G_1$을 bilinear group이라고 부른다.

Bilinear : 임의의 $u,v\in G_1$과 $a,b\in Z$에 대해서 $e(u^a,v^b)=e(u,v)^{ab}$를 만족한다.

- Non-degenerate : $e(g,g)\neq 1$
- Computable : 함수 $e: G_1\times G_1\rightarrow G_2$를 계산하는 효율적인 알고리즘이 존재한다.

$e(,)$는 $e(g^a,g^b)=e(g,g)^{ab}=e(g^b,g^a)$ 를 만족하고 이를 페어링 연산의 대칭성(symmetric)이라 부른다.

Hardness 가정 (Hardness Assumption): 본 논문에서 제안한 mPEKS 스킴은 BDHI 가정 (Bilinear Diffie-Hellman Inversion Assumption) 에 안전성의 기반을 둔다.

BDHI 가정 (Bilinear Diffie-Hellman Inversion Assumption) : 주어진 입력값 $(g,g^x)\in(G_1^*)^2$ 에 대하여 $e(g,g)^{1/x}\in G_2$ 을 계산하는 문제를 그룹 $G$에서의 1-BDHI문제라고 정의한다. 이때, 알고리즘 $B$ 가 $\Pr[B(g,g^x)=e(g,g)^{1/x}]\geq\epsilon$ 을 만족하면 그룹 $G_1$에서 1-BDHI 문제를 푸는 이익(advantage) $\epsilon$을 갖는다고 정의한다. 이때, 모든 다항식 시간 알고리즘 $B$가 1-BDHI문제를 푸는데 있어서 negligible 한 이익(advantage)를 갖는다면 1-BDHI문제가 intractable 하다고 정의한다.

정의 1. 그룹 $G_1$에서 적어도 1-BDHI 문제를 푸

는 이익(advantage) $\epsilon$을 갖는 t-time 알고리즘이 존재하지 않는다면 그룹 $G_1$에서 $(t,1,\epsilon)$-BDHI 가정이 성립한다고 정의한다.

### 2.2 정의 및 안전성 모델

검색가능한 공개키 암호 시스템은 Boneh et al. 에 의해서 제안되어진 암호화된 메시지 전송시스템이다. 이 시스템은 송신자, 수신자, 그리고 서버로 구성되며 특정 수신자에게 전송하기 위해서 공개키를 이용해서 암호화된 메일을 서버가 수신자로부터 암호화된 검색어를 받아서 테스트를 대행하여 검색하는 것이 가능한 시스템이다. 이 시스템에서 전송되는 메시지는 암호화된 메시지와 암호화된 검색어 정보를 포함한 구성을 갖는다. 하지만 수신자의 수가 증가하게 된다면 이에 따라서 암호화된 메시지의 크기가 증가하게 될 것이고 또 이는 테스트 단계에서의 높은 계산량을 요구할 것이다.

$$[SE_{key}(message), E_{pk_1}(key), E_{pk_2}(key),\ldots,$$
$$E_{pk_n}(key), peks_{pk_1}(w), peks_{pk_2},\ldots, peks_{pk_n}(w)]$$

본 논문에서 정의한 다중 수신자 환경에서 검색가능한 공개키 암호화 시스템의 정의는 다음과 같다.

정의 2. 다중 수신자 환경에서 검색 가능한 공개키 암호화 시스템 (mPEKS) 은 다음의 다항식시간 알고리즘들로 구성된다.

- **키 생성 알고리즘** (Key Generation Algorithm): security parameter $\lambda$를 입력받은 후, 사용자 $i$의 공개키/ 개인키 쌍 $(y_i, x_i)$ 을 생성한다.
- **다중 수신자를 위한 암호화된검색정보 알고리즘** (mPEKS Algorithm): 수신자들의 공개키의 집합 $R=\{y_1, y_2,\ldots, y_n\}$과 검색어 $w$를 입력받은 후, 암호화된 검색정보 $c=mPEKS(\lambda, R, w)$를 생성한다.
- **트랩도어 알고리즘**(Trapdoor Algorithm): 주어진 적당한 수신자 $j$의 비밀키 $x_j$와 검색어 $w$를 입력받은 후, 암호화된 검색어인 트랩도어 $T_w$를 생성한다.
- **테스트 알고리즘** (Test Algorithm): 주어진 암호화된 검색정보 $c=mPEKS(\lambda, R, w)$와 트랩도어 $T_{w'}$을 입력받으면, $w=w'$이면 "yes"를 $w\neq w'$이면 "no"를 출력한다.

**안전성 모델.** 지금부터 다중 수신자를 위한 검색가능한 공개키 암호 시스템에 대한 안전성을 이전의 관련된 연구들 [1,6]에서와 마찬가지로 semantic-security 관점에서 정의한다. Hwang 과 Lee가 제안하였던 다수의 수신자와 여러개의 검색어를 고려한 검색 스킴에서의 안전성의 정의에서는 다수의 수신자와 여러개의 검색어를 고려한 검색 스킴에서의 안전성을 고려하고 있다. 이는 기존의 Golle et al. 에 의해서 제안된 안전성의 정의를 그대로 따른다[5]. 즉, 다수의 수신자를 고려한 환경에서 안전성의 정의는 단일 수신자를 고려한 환경에서의 안전성의 정의를 그대로 따른다.

본 논문에서는 mPEKS에 대한 안전성의 정의는 단일 검색어를 고려한 Boneh et al.에 의해서 제안된 PEKS 에서의 안전성에 기반을 둔다. PEKS 스킴에서와 마찬가지고 mPEKS 스킴에서의 안전성은 암호문이 트랩도어(trapdoor)없이는 정보를 노출하지 않는다는 것을 보장한다. 이때, 공격자는 자신이 선택한 검색어 $w$에 대한 트랩도어 $T_w$를 얻는 것이 가능한 능동적인 (active) 공격자를 가정한다. 공격자의 목적은 트랩도어를 얻을 수 없는 두 개의 검색어 $w_0, w_1$ 중 하나에 대한 암호문이 주어졌을 때 어떠한 검색어에 대한 암호문인지를 결정하는 것이다. mPEKS에서 정의된 안전성는 다음의 *Game* 을 이용한다.

Setup: Challenger는 각 수신자 $i$의 공개키/개인키 $(y_i, x_i)$쌍을 생성하여 수신자에게 주고 공개키$y_i$는 공개한다.

Phase 1 (Trapdoor queries): 공격자는 자신이 선택한 검색어 $w\in\{0,1\}^*$ 와 수신자 $j$에 대한 트랩도어 값을 질의하면 Challenger는 $T_{w^*}=Trapdoor(x_j, w)$를 답한다.

Challenge: 공격자가 수신자의 집합 $R$과 두 개의 target 검색어 $w_0, w_1$를 선택한다. 이 때, 공격자는 이전단계에서 트랩도어 쿼리를 질의하지 않았던 검색어를 target 검색어로 선택하고 공격자는 $R$에 포함되지 않는 것을 가정한다. Challenger는 랜덤하게 $b\in\{0,1\}$를 선택하여 $C^*=mPEKS(R, w_b)$ 값을 계산하여 공격자에게 전송한다.

Phase 2 (Trapdoor queries): 공격자는 $w\neq w_0, w_1$인 검색어 $w\in\{0,1\}^*$ 를 선택하여 수신자 $j$에 대한 트랩도어 값을 질의하면 Challenger는

$T_w = Trapdoor(x_j, w)$를 답한다.

Guess: 공격자는 $b' \in \{0,1\}$ 을 추측(guess)하여 결과를 낸다. 이때, $b=b'$이면 공격자는 *Game*에서 이긴다.(win)

우리는 mPEKS 스킴에서 공격자 $A$의 advantage를 다음과 같이 정의한다.

$Adv_A(\lambda) = |\Pr[b=b'] - 1/2|$

정의 3. 임의의 다항식 시간 공격자 $A$에 대해서 $Adv_A(\lambda)$가 negligible 하면 mPEKS 스킴은 선택한 검색어 공격 (adaptive chosen keyowrd attack) 에 대해서 안전 (semantically secure) 하다고 정의한다.

## III. 제안된 스킴

본 장에서는 다중 수신자 환경에서 검색 가능한 공개키 암호 시스템 (mPEKS)을 제안하고 제안된 스킴의 안전성을 랜덤 오라클 모델에서 분석한다. 제안된 스킴의 안전성은 1-BDHI(Bilinear Diffie-Hellman Inversion) 문제의 어려움에 기반을 둔다.

### 3.1 mPEKS 스킴

$G_1$을 $g$를 $G_1$의 생성원(generator)으로 갖는 그룹이라고 하고 $G_1$에서 1-BDHI 문제가 어렵다고 가정하자. 함수 $e: G_1 \times G_1 \rightarrow G_2$를 bilinear 함수라고 가정하자. 이때, $e(g,g)$는 $G_2$의 생성원이다. $\lambda$는 안전성 파라메터 (security parameter)라고 하고 $H_1 : \{\}^* \rightarrow G_1$ 과 $H_2 : G_2 \rightarrow \{0,1\}^{\lambda}$ 는 랜덤 오라클로 model 될 해쉬함수이다. mPEKS 스킴은 다음 4개의 다항식 시간 알고리즘으로 구성된다.

• *KeyGen*($\lambda$) : 수신자 $i$에 대하여 임의의 랜덤 값 $x_i \in Z_p^*$를 선택하여 $y_i = g^{x_i}$를 계산한 후, 공개키와 개인키 쌍 $(y_i, x_i)$를 생성하여 사용자 $i$에게 안전하게 전송하고 $y_i$를 공개한다.

• *mPEKS*($R$,$w$): 수신자의 공개키의 집합 $R=\{y_1, y_2, ..., y_n\}$과 검색어 $w$를 입력받은 후, 다음을 계산한다.

1. 임의의 $r \in Z_p^*$ 를 선택하여 $C_1 = \{y_1^r, y_2^r, ..., y_n^r\}$를 계산한다.
2. $C_2 = H_2(e(g, H_1(w)^r))$를 계산한다.
3. 암호문 $C=[C_1, C_2]$를 생성한다.

• *Trapdoor*($x_i$,$w$): 수신자 $i$의 개인키 $x_i$와 검색어 $w$를 입력받은 후, 트랩도어 $T_w = H_1(w)^{1/x_i}$를 출력한다.

• *Test*($C$,$T_w$): 암호문 $C=[C_1, C_2]$와 트랩도어 $T_w$를 입력받은 후 다음을 테스트 한다 .

1. $C_2 = H_2(e(y_j^r, T_w))$ $(j=1,...,n)$인지를 체크한다.
2. 만약 1의 등호가 성립하는 $j$값이 존재하면 "예"를 출력하고 그렇지 않으면 "아니오"를 출력한다.

### 3.2 안전성 분석

본 절에서는 mPEKS 스킴의 안전성을 1-BDHI 문제에 기반하여 증명한다.

정리 4. 다중 수신자 환경에서 검색가능한 공개키 암호 (mPEKS) 스킴은 1-BDHI(Bilinear Diffie-Hellman Inversion assumption) 이 intractable 하다는 가정아래 랜덤 오라클 모델에서 Chosen keyword attack 에 대하여 semantically secure 하다.

증명. 공격자 $A$를 mPEKS 스킴에 대한 공격의 advantage $\epsilon$을 갖는 공격자라고 가정하자. 이때, $A$는 기껏해야 $q_T$개의 트랩도어 쿼리를 만들 수 있다고 가정하자. 우리는 1-BDHI 문제를 푸는데 있어서 $\epsilon' = \epsilon / e\, q_T\, q_{H_2}$의 advantage를 갖는 공격자 $B$를 construt 할 수 있다는 것을 보임으로써 스킴에 대한 안전성을 증명한다. 여기서, $e$는 자연지수(natural logarithm) 이다.

$g$를 $G_1$의 생성원(generator)이고 $g, u_1 = g^{\alpha} \in G_1$이 1-BDHI 문제의 입력값으로 주어졌다고 가정하자. 알고리즘 $B$의 목적은 $e(g,g)^{1/\alpha} \in G_2$ 를 계산하는 것이다. 알고리즘 $B$는 다음과 같이 공격자 $A$와 interact 한다

*Setup*: 알고리즘 $B$는 랜덤값 $t_i \in Z_p^*$ 를 선택하여 $i$의 공개키를 $y_i = u_1^{t_i} = g^{\alpha \cdot t_i}$ $(i=1,...,n)$로 놓고 $y_i (i=1,...,n)$을 공개한다.

$H_1$ - *queries*: 공격자 $A$는 랜덤오라클 $H_1$에 언제든지 쿼리할 수 있다. 공격자 $A$가 쿼리했을 때 대답하기 위해서 알고리즘 $B$는 $H_1$-리스트 라 불리는 $< w_i, h_i, e_i, c_i >$의 tuple의 리스트를 저장 그리고 관리한다. $A$가 $w_i \in \{0,1\}^*$를 쿼리했을 때 알고리즘 $B$는 다

음과 같이 대답한다.

1. $w_i$ 가 이미 $<w_i, h_i, e_i, c_i>$의 형태로 $H_1$-리스트에 존재하면 알고리즘 $B$는 $H_1(w_i)=h_i \in G_1$로 답한다.

2. $w_i$ 가 $H_1$-리스트에 존재하지 않는다면 알고리즘 $B$는 $\Pr[c_i=0]=1/(q_T+1)$를 만족하는 확률로 랜덤 코인(random coin) $c_i \in \{0,1\}$을 생성한다.

3. 알고리즘 $B$는 랜덤 $e_i \in Z_p^*$를 선택하여 $c_i=0$이면 $h_i=g^{e_i} \in G_1$ 이라 하고 $c_i=1$이면 $h_i=u_1^{e_i} \in G_1$ 라고 한다.

4. 알고리즘 $B$는 $<w_i, h_i, e_i, c_i>$를 $H_1$-리스트에 추가하고 $H_1(w_i)=h_i \in G_1$로 셋팅하여 공격자 $A$에게 답한다. 여기서, $h_i$는 $G_1$에서 uniform 하고 $A$의 view에 독립이어야 한다.

*$H_2$ - queries*: $H_1$ 오라클 구성과 비슷한 방법으로, 공격자 $A$는 랜덤오라클 $H_2$에 언제든지 쿼리할 수 있다. 공격자 $A$가 $t$를 $H_2$에 쿼리하면 $t$가 이미 $H_2$-리스트에 $(t, V)$의 형태로 존재하면 알고리즘 $B$는 $H_2(t)=V$로 답한다. 만약 존재하지 않는다면 알고리즘 $B$는 랜덤값 $V \in \{0,1\}^\lambda$를 선택하여 $H_2(t)=V$로 놓고 답하고 $(t, V)$를 $H_2$-리스트에 추가한다. 초기단계의 $H_2$-리스트는 공집합(empty set) 이다.

*Phase 1 (트랩도어 쿼리)*: 공격자 $A$는 검색어 $w$에 대한 수신자 $\xi$ 의 트랩도어를 위한 쿼리를 생성을 요구할 때, 알고리즘 $B$는 다음과 같이 대답한다.

1. 알고리즘 $B$는 $H_1(w^*)=H_1(x_j)=h_j \in G_1$인 $h_j \in G_1$을 얻기 위해서 $H_1$ 함수와 $H_1$-리스트를 검색하다. $<w_j, h_j, e_j, c_j>$ tuple 이 $H_1$-리스트에 있는 $w_j=w^*$인 tuple이라고 가정하자. 만약 $c_j=0$이면 알고리즘 $B$는 실패하고 종료한다.

2. $c_j=1$ 라면 $h_j=u_1^{e_j} \in G_1$ 이기 때문에, 알고리즘 $B$는 수신자의 공개키 $y_\xi$에 대응하는 정당한 트랩도어 $T_{w^*}=(g^{\alpha \cdot e_j})^{1/(\alpha \cdot t_\xi)}=g^{\frac{e_j}{t_\xi}}$ 를 생성할 수 있다. 알고리즘 $B$는 트랩도어 $T_{w^*}$를 공격자 $A$에게 준다.

*Challenge*: 공격자 $A$는 수신자의 공개키의 집합 $R=\{y_1, \ldots, y_n\}$ 과 검색어 $w_0, w_1$를 알고리즘 $B$에게 제공한다. 알고리즘 $B$는 다음의 challenge 값 $C=[C_1, C_2]$를 생성한다.

1. 알고리즘 $B$는 $H_1(w_0)=h_0$와 $H_1(w_1)=h_1$을 만족하는 $h_0, h_1 \in G_1$을 얻기 위해서 $H_1$에 쿼리한다. 여기서, $<w_b, h_b, e_b, c_b>$ $(b=0,1)$를 $H_1$-리스트에 있는 대응하는 tuple이라고 하자. 만약, $c_0=1$이고 $c_1=1$이라면 challenge 값 생성에 실패한다.

2. $c_0=0$ 이거나 $c_1=0$ 이라면 알고리즘 $B$는 $b \in \{0,1\}$을 랜덤하게 선택하여 다음의 과정을 통해서 challenge 값을 생성한다. (일반성을 잃지않고, $c_i=0$인 $i$값을 $b$로 선택해도 된다.)

(1) 알고리즘 $B$는 랜덤값 $k \in Z_p^*$를 선택하여 $g^{t_j \cdot k}$를 계산하여 $C_1=g^{t_1 k}, \ldots, g^{t_l k}$ 을 생성한다. 여기서, 임의의 랜덤값 $k \in Z_p^*$에 대해서 $r=k/\alpha$를 만족하는 적당한 $r \in Z_p^*$이 존재한다. 따라서, $y_j^r=(g^{\alpha t_j})^r=(g^{\alpha t_j})^{k/\alpha}=(g^{t_j})^k$을 만족한다.

(2) 알고리즘 $B$는 랜덤값 $Z \in \{0,1\}^\lambda$를 선택하고 검색어 $w_b$에 대한 암호문 $C^*=[C_1, C_2]=[C_1, Z]$ 를 생성한다. 이때, $Z=H_2(e(g, H_1(w_b)r))$이 되고 $e(g, H_1(w_b)^r)=e(g, H_1(w_b))^{k/\alpha}=e(g, g^{e_b})^{k/\alpha}=(e(g,g)^{1/\alpha})^{e_b \cdot k}$ 이다.

*Phase 2 (트랩도어 쿼리)*: 공격자 $A$는 $w_j \neq w_0, w_1$인 검색어 $w_j$에 대한 트랩도어 쿼리를 질의한다. 알고리즘 $B$는 Phase 1에서와 동일한 방법으로 답한다.

*출력단계*: 공격자 $A$는 암호문 $C^*$ 가 $w_0$에 대한 암호문인지 $w_1$에 대한 암호문인지를 결정하여 $b'$을 결과로 출력한다. 알고리즘 $B$는 $H_2$-리스트로부터 $(t, V)$를 임의로 선택하여 $t^{1/(e_b \cdot k)}$를 결과인 $e(g,g)^{1/\alpha}$로 얻는다. 여기서, $e_b$와 $k$는 Challenge 단계에서 선택된 값이다.

알고리즘 $B$가 적어도 $\epsilon'$의 확률로 결과인 $e(g,g)^{1/\alpha}$를 얻는다는 것을 다음의 사건들을 이용하여 보일수 있다. 여기서 이용하는 사건들의 정의와 접근은 PEKS 스킴에서의 증명에서의 방법을 그대로 따른다 [1]. 따라서 자세한 증명은 생략한다.

$E_1$ : 알고리즘 $B$가 공격자 $A$의 트랩도어 쿼리를 abort하지 않는다 .

$E_2$ : Challenge 단계를 abort하지 않는다 .

$E_3$ : Output 단계에서 올바른 $e(g,g)^{1/\alpha}$의 값을 얻도록 $H_2$-리스트로부터 $(t, V)$를 제대로 선택한다.

보조정리 1. $\Pr[E_1] \geq 1/e$

보조정리 2. $\Pr[E_2] \geq 1/q_{d_T}$

보조정리 3. $\Pr[E_3] \geq 2\epsilon$

결과적으로 보조정리 3에 의해서 $e(g, H_1(w_b)^r)=e(g, H_1(w_b))^{k/\alpha}=e(g, g^{e_b})^{k/\alpha}=(e(g,g)^{1/\alpha})^{e_b \cdot k}$는 $H_2$-

리스트에 나타날 것이기 때문에 알고리즘 $B$는 적어도 $1/q_{H_2}$의 확률로 올바는 $(t, V)$쌍을 선택할 것이고 적어도 $\epsilon/q_{H_2}$의 확률로 올바른 값을 답을 얻을 것이다. 또, 적어도 $1/(eq_T)$의 확률로 도중에 실패해서 멈추지 않을 것이기 때문에 알고리즘 $B$ 는 적어도 $\epsilon/eq_T q_{H_2}$의 확률도 1- BDHI문제를 해결할 수 있다. □

## IV. 결 론

본 논문에서는 다중수신자를 고려한 효율적인 검색 가능한 공개키 암호시스템(mPEKS)을 제안하였다. 우리는 mPEKS에서 고려되어져야 할 안전성을 정의하고 안전성 모델에서 제안된 스킴의 안전성을 랜덤오라클 모델하에 증명하였다. 본 논문에서 제안한 스킴은 특히 다수의 사용자가 이용하는 서버측면의 계산량을 개선하였기 때문에 의미가 있다. 최근에 암호화된 데이터에서의 검색기술에 대한 연구가 활발해 지면서 특히 쿼리의 다양성에 대한 연구도 함께 진행되고 있다. 다중수신자를 고려한 효율적인 검색가능한 공개키 암호시스템에서도 쿼리의 다양성을 위한 연구가 필요하다.

## 참 고 문 헌


[1] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," In Advances in Cryptology-Eurocrypt 2004, LNCS 3027, pp. 506-522, 2004.

[2] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," In proceedings of TCC'07, LNCS 4392, pp. 535-554, 2007.

[3] Y.C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," In Third International Conference on Applied Cryptography and Network Security(ACNS 2005), LNCS 3531, pp. 442-455, 2005.

[4] E.J. Goh, "Secure Incexes," Technical report 2003-216, In IACR ePrint Cryptography Archive, 2003.

[5] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," In proceedings of ACNS 2004, LNCS 3089, pp. 31-45, 2004.

[6] Y.H. Hwang and P.J. Lee, "Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-user System," In proceedings of Pairing 2007, LNCS 4575, pp. 2-22, 2007.

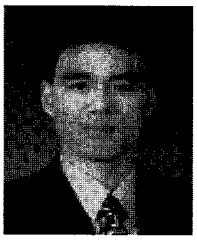
[7] W. Ogata and K. Kurosawa, "Oblivious Keyword Search," Journal of Complexity, Vol. 20, No. 2-3, pp. 356-371, Apr.-June, 2004.

[8] R. Richardson, "2007 CSI Computer Crime and Security Survey," The 12th Annual report of computer security society, CSI, 2007.

[9] D.X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," In Symposium on Security and Privacy IEEE, pp. 44-55, May 2000.

## 〈著者紹介〉

이 현 숙 (Hyun Sook Rhee) 정회원
1998년 2월: 단국대학교 수학과 졸업
2000년 2월: 단국대학교 응용수학 석사
2008년 2월: 고려대학교 정보경영공학전문대학원 정보보호전공 박사
2008년 3월~2008년 8월: 고려대학교 정보경영공학전문대학원, 박사후 연구원
2008년 9월~현재:University of Wollongong (Australia), 박사후 연구원
〈관심분야〉 정보보호, PET 기술, IPTV 와 Smart Card 관련 보안기술 등



박 종 환 (Jong Hwan Park) 정회원
1999년 2월: 고려대학교 수학과 졸업
2004년 2월: 고려대학교 정보보호대학원 암호프로토콜 석사
2008년 8월: 고려대학교 정보경영공학전문대학원 정보보호전공 박사
2008년 9월~현재: 고려대학교 정보경영공학전문대학원, 박사후 연구원
〈관심분야〉 Pairing based 암호, Broadcast 암호, 전자서명 등

이 동 훈 (Dong Hoon Lee) 정회원
1984년: 고려대학교 경제학 학사
1987년: University of Oklahoma 전산학과 석사
1992년: University of Oklahoma 전산학과 박사
1993년~1997년 고려대학교 전산학과 조교수
1997년~2001년 고려대학교 전산학과 부교수
2001년~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 암호이론, 암호프로토콜, USN 이론, 키 교환, 익명성 연구, PET 기술