情報保護學會論文誌
第 21 卷 第 1 號, 2011. 2

다자간 환경에서 사용자 탈퇴가 가능한 프라이버시 보호 키워드 검색 기법*

김 동 민†, 천 지 영, 노 건 태, 정 익 래‡
고려대학교 정보경영공학전문대학원

Secure Searchable Encryption with User-Revocability in Multi-User Settings*

Dong Min Kim†, Ji Young Chun, Geon Tae Noh, Ik Rae Jeong‡
Graduate School of information Management and Security, Korea University

요 약

공공의 서버를 통한 자료의 공유는 사용자들의 편리성을 증가시킨 반면 저장된 데이터에 대한 접근통제 문제와 신뢰할 수 없는 서버에 대한 사용자들의 프라이버시 노출 문제 등의 새로운 문제를 야기한다. 다자간 환경에서 사용자 프라이버시를 보장하면서도 사용자간의 데이터 공유를 하기 위해서 키워드 검색이 가능한 암호화 방식이 사용된다. 특히, 이런 다자간 환경에서는 탈퇴한 사용자들이 더 이상 키워드 검색이나 저장된 데이터를 접근할 수 없도록 탈퇴한 사용자에 대한 안전성을 고려해야 한다. 하지만 기존의 다자간 환경에서 제안된 키워드 검색 암호화 방식은 탈퇴한 사용자가 정당한 사용자와 서버가 통신하는 암호화 메시지를 보고서 공유 데이터를 복구할 수 있는 문제점을 가지고 있다. 본 논문에서 제안하는 기법은 이런 문제점들을 해결한 탈퇴한 사용자에 대한 안전성을 보장하는 키워드 검색기법이다.

ABSTRACT

In recent days, people used to store and share the data with other users through the web storage services. It is more convenient for using the data, but it raise problems such as access control of stored data and privacy exposure to untrusted server. Searchable encryption is used to share the data securely in multi-user setting. Especially in the multi-user setting, the revoked users should not be able to search the data and access the stored data. That is, it should be considered the security from revoked users. However in the existing schemes, the revoked users can decrypt the shared data by passive attack. Proposed scheme is the secure searchable encryption that resolves the problem and guarantees the security for revoked users.

Keywords: Keyword search, Multi-user Setting, Encrypted data, user-revocability, Searchable encryption

I. 서 론

최근 클라우드 컴퓨팅(cloud computing) 기술의 발달과 더불어 인터넷을 통해 자료를 공유할 수 있는 클라우드 기반 스토리지(storage) 서비스가 주목받고 있다. 사용자들은 이러한 스토리지 서비스를 통해 공공의 서버(public server)에 자신의 자료를 업로드(upload)하고 관심 있는 자료를 다운로드(download) 받는다. 공공의 서버를 통한 자료의 공유는 사

접수일(2010년 5월 10일), 게재확정일(2010년 10월 11일)
* 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발사업(정보통신 [KI002113, car-헬스케어 보안 기술개발]과 지식경제부 및 정보통신산업진흥원의 "대학 IT연구센터 육성·지원사업" 사업의 일환으로 수행하였음 (NIPA-2010-C1090-1001-0004)
† 주저자, kkomang03@naver.com
‡ 교신저자, irjeong@korea.ac.kr

용자들의 편리성을 증가시킨 반면 사용자들의 프라이버시에 대한 새로운 문제를 야기한다. 공공의 서버에 저장된 자료는 사용자들이 아닌 스토리지 서비스 제공자에 의해 관리되므로 신뢰할 수 없는 서비스 제공자에 의한 데이터 노출은 단순한 접근통제만으로 해결하기 힘들다. 신뢰할 수 없는 서비스 제공자에 대한 데이터 노출을 막기 위해서는 데이터를 암호화 하여 저장하는 방법이 최선이겠으나 데이터 암호화는 검색의 효율을 급격히 저하시킨다.

이러한 검색 문제를 해결하기 위해 최근 암호화된 데이터에서의 키워드 검색에 대한 연구가 활발히 진행되고 있다[1-12]. 암호화된 데이터에서의 키워드 검색에 대한 연구는 크게 대칭키(symmetric key) 기반과 공개키(public key) 기반으로 나뉜다. Song 등의 논문[2]을 시작으로 한 대칭키 기반의 키워드 검색[2-5]은 자신의 대칭키로 문서를 암호화하여 서버에 저장한 후 자신의 대칭키를 이용하여 문서를 검색하는 방법으로 문서 암호화에 사용한 대칭키를 소유한 사람만이 검색이 가능하다. 반면 공개키 기반의 키워드 검색 기법[6-9]은 암호화된 문서와 함께 문서검색 정보를 어떤 사용자의 공개키로 암호화하여 서버에 저장하는데 암호화에 사용된 공개키에 대한 비밀키를 소유한 사람만이 검색이 가능한 기법으로 Boneh 등의 논문[6]을 시작으로 연구가 활발히 진행되고 있다. 그러나 대칭키, 공개키 기반 모두 문서 암호화시 특정 사용자를 염두에 두고 암호화하기 때문에 특정 사용자만이 검색이 가능하게 된다. 따라서 이러한 기법은 여러 사용자가 공공의 서버를 통해 문서를 공유하는 환경에는 적합하지 않다. 만약 n명의 사용자가 서로의 문서를 공유하는 환경이라면 문서를 업로드하는 사용자는 n명의 사용자 모두 문서 검색이 가능하도록 같은 문서를 n번 암호화하여 저장하여야 한다. 이는 저장량 측면에서 매우 비효율적이다. 따라서 다자간 문서 공유 환경에서 이러한 문제를 해결하기 위해 여러 기법들[5,10,11,12]이 제안되었다. 문헌 [5,10,12]은 다자간 문서 공유 환경에서 여러 사용자들이 검색 가능한 기법이지만 문서 업로드가 가능한 사람은 특정한 한명 뿐이다. 따라서 이러한 기법은 여러 사용자들이 자신의 문서를 업로드하는 환경에는 적합하지 않다. Bao 등의 논문[11]은 암호화된 문서에 대해 여러 사용자들이 자신의 문서를 업로드하고 다운로드받을 수 있는 환경으로 다자간 문서 공유 환경에 적합한 기법이기는 하나 탈퇴자의 수동적 공격(passive attack)에 대해 안전하지 못하다. 이 기법에서 기존 사용자가 탈퇴할 경우 문서를 검색할 수는 없으나 다른 사용자가 검색하는 문서를 도청(eavesdropping)할 경우 도청한 모든 문서에 대한 복호화가 가능하게 된다.

따라서 본 논문에서는 다자간 문서 공유 환경에서 기존 기법들이 갖는 문제점 들을 개선한 프라이버시를 보호하는 다자간 키워드 검색 기법을 제안한다. 제안하는 기법은 여러 사용자가 자신의 문서를 업로드하고 다운로드받을 수 있는 기법으로 이러한 과정에서 신뢰할 수 없는 서버에 대해 사용자들의 프라이버시를 보호한다. 또한 제안하는 기법은 사용자들의 가입과 탈퇴가 용이하며 특히 탈퇴자에 대한 안전성을 보장한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 연구, 3장에서는 본 논문과 관련된 배경지식에 대해서 살펴본다. 4장에서 시스템 모델 및 안전성 모델을 정립한 후 5장에서 정립된 안전성 모델에 대해 안전한 기법을 제안하고 분석한다. 6장에서는 결론을 통해서 논문을 마무리한다.

II. 관련 연구

최근 다자간 문서 공유 환경에서 사용자들의 프라이버시를 보호하면서도 문서 검색이 가능한 여러 기법들[5,10,11,12]이 제안되었다. 서론에서 살펴본 바와 같이 문헌 [5,10,12]은 다자간 문서 공유 환경에서 여러 사용자들이 검색 가능한 기법이지만 문서 업로드가 가능한 사람은 특정한 한명 뿐이다. 문서 업로드가 가능한 사람은 문서를 업로드하기 전에 먼저 문서를 다운로드 받을 사람들을 정한 후 그 사람들만이 문서를 다운로드 받을 수 있도록 문서를 암호화하여 저장한다. 따라서 이러한 기법들은 여러 사용자들이 자신의 문서를 업로드하는 환경에는 적합하지 않다. 또한 사전에 검색 가능한 사용자들을 지정하기 때문에 추후 가입자나 탈퇴자가 발생하였을 경우 처리가 어렵다. 문헌 [5]의 경우 탈퇴자 처리(revocation)를 위해 브로드캐스트 암호화(broadcast encryption) 기법을 사용하기 때문에 탈퇴자 발생 시마다 다른 모든 사용자들이 새로운 값을 갱신해야하는 번거로움이 따른다. 또한 문헌 [10,12]의 경우 가입자나 탈퇴자 처리를 위해 모든 문서에 대한 암호화된 인덱스를 수정해야하기 때문에 가입자나 탈퇴자 처리가 매우 비효율적이다.

이 후 기존 기법들에서의 문서 업로드 문제와 가

입·탈퇴자 처리 문제를 해결하기 위해 Bao 등은 다자간 문서 공유 환경에 적합한 기법[11]을 제안하였다. Bao 등의 기법은 권한이 있는 사람은 누구나 문서 업로드, 다운로드가 가능한 기법으로 신뢰할 수 없는 서버에 대해 사용자들의 프라이버시를 보호한다. Bao 등의 기법에서는 신뢰할 수 있는 사용자 관리자(user manager)가 존재하여 각각의 사용자에게 검색에 사용할 비밀키와 문서 암호화에 사용할 대칭키 e를 제공한다. 각 사용자의 비밀키는 사용자마다 모두 다르지만 문서 암호화에 사용되는 대칭키 e는 모든 사용자가 공유한다. 따라서 각각의 사용자는 자신의 문서를 모든 사용자가 공유한 대칭키 e로 암호화하여 업로드하게 된다. 자신이 원하는 키워드를 포함하는 문서를 검색하기 위해 자신의 비밀키를 사용하여 암호화한 검색정보를 서버에 보내면 서버는 이 값을 이용하여 사용자가 원하는 문서를 찾아 보내준다. 이 때 사용자가 받게 되는 문서 또한 모든 사용자가 공유한 대칭키 e로 암호화되어 있다. 탈퇴자를 처리하기 위해 사용자 관리자는 서버에게 탈퇴자 리스트를 보내고 서버는 자신의 사용자 리스트에서 탈퇴자들의 정보를 삭제한다. 추 후 탈퇴자가 문서 검색을 시도하더라도 서버에는 탈퇴자에 대한 정보가 남아 있지 않으므로 탈퇴자는 검색할 수 없게 된다.

Bao 등의 기법이 탈퇴자 처리에 대한 고려를 하였음에도 불구하고 탈퇴자의 수동적 공격(passive attack)에 대해 안전하지 못하다. 이는 탈퇴자가 발생하더라도 문서 암호화에 사용된 대칭키 e를 갱신하지 않기 때문이다. 따라서 기존 사용자가 탈퇴할 경우 이 사용자에 대한 정보가 서버에서 삭제되므로 기존 비밀키를 사용하여 문서를 검색할 수는 없으나 다른 사용자가 검색하는 문서를 도청(eavesdropping)할 경우 도청한 모든 문서에 대해 복호화가 가능하게 된다.

III. 배경지식

3.1 의사난수 함수(pseudorandom function)와 의사난수 치환(pseudorandom permutation)

난수 함수(random function)의 출력 값과 효율적으로 구별이 불가능한 출력 값을 가지는 함수를 의사난수 함수(pseudorandom function)라고 한다. 즉, 의사난수 함수 $f:\{0,1\}^j\times\{0,1\}^m\rightarrow\{0,1\}^n$이고, 여기서 j는 시드(seed) 값의 비트수, m은 입력 값의 비트수, n은 출력 값의 비트수이다. 본 논문에서 사용하는 암호학적 해시 함수는 일반적으로 의사난수 함수로부터 모델링된다. 의사난수 함수 중에서 입력 값의 비트수와 출력 값의 비트수가 같을 경우, 이것을 의사난수 치환(pseudorandom permutation)이라고 한다. 즉, 의사난수 치환 $p:\{0,1\}^j\times\{0,1\}^n\rightarrow\{0,1\}^n$이다. 일반적으로 안전한 대칭키 암호화 기법은 의사난수 치환을 사용하여 모델링된다.

3.2 겹선형 사상(bilinear maps)

제안하는 기법은 겹선형 사상을 제공하는 소수 위수의 어떤 유한군을 사용한다. G와 G_1은 같은 소수 위수 p를 가지는 두 개의 군이라 하고, 겹선형 사상을 $e:G\times G\rightarrow G_1$라고 하면, 다음의 성질을 만족한다.

1) 모든 $g_1,g_2\in G$와 모든 $a,b\in Z_p^*$에 대해 $e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$를 만족한다.
2) 만약 g가 군 G의 생성자이면, $e(g,g)$는 군 G_1의 생성자이다.
3) 임의의 원소 $g_1,g_2\in G$에 대해 $e(g_1,g_2)$는 계산하기 쉬워야한다.

3.3 BLS 짧은 서명(short signature)[13]

BLS 짧은 서명(short signature)은 Boneh 등에 의해 제안된 겹선형 사상에 기반을 둔 서명 기법이다. 이것을 간단히 살펴보면 다음과 같다: (G,G_1,e)를 3.2절과 동일하게 정의하고, g를 군 G의 생성자, $h:\{0,1\}^*\rightarrow G$를 충돌 저항성 해시 함수, 사용자의 서명키를 $k\in Z_p^*$, 검증키를 $y=g^k\in G$라고 하자. 이 때 메시지 m에 대한 서명을 $\sigma=h(m)^k$라고 하며, 이에 대한 검증은 $e(g,\sigma)=e(y,h(m))$를 통해 가능하다. BLS 짧은 서명은 CDH(Computational Diffie- Hellman) 문제의 어려움에 기반하여 랜덤 오라클 모델에서 서명 위조 불가능성을 만족한다.

IV. 시스템 모델 및 정의

4.1 시스템 모델

우리는 $\{D, UM, U, S\}$로 구성된 데이터베이스 시스템 Γ을 고려한다. 여기서 D는 데이터베이스, UM은 사용자 관리자, U는 사용자 집합, S는 서버이다. 데이터베

이스 D는 다수의 속성들로 구성된 m개의 레코드 $\{d_1,\cdots,d_m\}$로 구성되어 있으며, 속성 중의 하나는 검색을 위한 키워드이다. 사용자 관리자 UM은 사용자의 관리를 책임지는 신뢰 기관으로써 사용자 등록 및 탈퇴, 사용자와 서버의 개인키를 발급하는 역할을 수행하며, 본 논문에서 UM과 통신하는 모든 채널은 안전하다고 가정한다. 사용자 집합 U에 속하는 모든 사용자는 서버 S를 통해 데이터베이스 D에 자유롭게 문서를 저장하고 검색할 수 있으며, 문서는 키워드를 가지는 암호화된 형태로 저장되고, 키워드를 사용하여 검색할 수 있다. 결과적으로 서버 S는 데이터베이스 D에 m개의 암호화된 형태의 레코드 $EI_D=\{EI_1,\cdots,EI_m\}$를 저장한다.

본 논문에서 우리는 다음의 표기를 사용한다. $x\in_R X$는 집합 X의 원소 x를 랜덤하게 선택한다는 표기이며, $|X|$는 집합 X의 크기를 말하는 것이고, $x\leftarrow A$는 알고리즘 A에서 x를 출력한다는 표기이다. 마지막으로 $v(k):N\rightarrow R$는 모든 양의 정수 c에 대하여 정수 N_c가 존재해서 $k>N_c, |v(k)|<1/k^c$를 만족할 때 무시할 수 있을 만큼 작은 값으로 수렴하는 함수(negligible function)라고 한다.

4.2 정의

데이터베이스 시스템 $\Gamma=\{D,UM,U,S\}$에서는 다음과 같은 프로토콜들이 존재한다.

- $Setup(1^\kappa)$: 사용자 관리자 UM은 제안하는 기법에 사용될 공개 파라미터 $params$와 자신의 개인키 sk_{UM}를 생성한다.
- $Join(u;sk_{UM};SU_{list})$: 이 프로토콜은 사용자 u, 관리자 UM, 서버 S가 참여하는 프로토콜이다. 사용자 u는 관리자 UM에게 등록을 요청하고, UM은 자신의 비밀값 sk_{UM}을 사용하여 사용자 u에게 비밀키 qk_u를 발급한다. 이 때 서버는 사용자 등록 리스트인 SU_{list}에 사용자 u의 정보를 등록한다.
- $Store(qk_u,d,w;SU_{list},EI_D)$: 이 프로토콜은 사용자 u와 서버 S가 참여하는 프로토콜이다. 사용자 u는 문서 d와 키워드 w를 자신의 개인키 qk_u를 이용해서 암호화한 문서 EI_d를 서버에 전송한다. 서버 S는 사용자 u가 등록된 사용자인지를 체크하고, EI_d를 EI_D에 저장한다.
- $Query(qk_u,w)$: 사용자 u는 키워드 w와 자신의 개인키 qk_u를 이용하여, 서버에 요청할 질의 $q_u(w)$를 생성한다.
- $Search(qk_u,q_u(w);SU_{list},EI_D)$: 이 프로토콜은 사용자 u와 서버 S가 참여하는 프로토콜이다. 사용자 u는 서버에 $q_u(w)$를 보내서 $ans_u(w)$를 받는다. 사용자 u는 자신의 개인키 qk_u를 이용해서 $ans_u(w)$로부터 키워드 w와 관련된 암호화된 문서들을 복호화한다.
- $Revoke(u;SU_{list})$: 이 프로토콜은 관리자 UM과 서버 S가 참여하는 프로토콜이다. 관리자 UM은 서버 S에게 삭제할 사용자 u를 보내며, 서버는 사용자 u의 정보를 SU_{list}로부터 삭제한다.

앞으로 우리는 질의 프라이버시(query privacy), 질의 위조 불가능성(query unforgeability), 탈퇴한 사용자로부터의 안전성(full-revocability)을 통해 제안하는 기법의 안전성을 정의한다.

- 질의 프라이버시(query privacy): 사용자와 서버의 통신을 통해서는 문헌 [4]에서 정의한 흔적(trace) 이외의 어떠한 정보도 노출하지 않는다. 즉, 질의에 대한 안전성을 제공한다.
- 질의 위조 불가능성(query unforgeability): 정당한 사용자만이 정당한 질의를 생성할 수 있다.
- 탈퇴한 사용자로부터의 안전성(full- revocability): 탈퇴한 사용자는 정당한 사용자인척 검색할 수 없고, 다른 정당한 사용자가 검색하는 문서에 대해 어떠한 정보도 얻을 수 없어야 한다.

4.2.1 질의 프라이버시(Query Privacy)

모든 검색 가능한 암호화 기법들은 안전성의 조건으로 질의 프라이버시를 만족해야 한다. 이것은 사용자가 서버에 질의를 하였을 때 서버에 노출되는 정보의 양을 표현하는 안전성에 대한 개념이다. 어떠한 기법이든 PIR(Private Information Retrieval) 프로토콜을 적용하지 않는 이상은 질의를 할 때에 서버에 특정한 질의의 흔적이 남게 된다. 비록 서버가 질의로부터 어떠한 키워드가 검색이 되었는지는 알 수가 없더라도 사용자 질의의 접근 유형(access pattern)은 항상 관찰할 수 있다. 하지만 이러한 흔적이나 흔적들로부터 알아낸 정보로부터 사용자의 다른 정보들이 서버에게 노출되어서는 안 된다. 다시 말하면 질의 프라이버시를 만족하기 위해서는 정당한 질의 과정을 통해서 공격자가 얻을 수 있는 정보이상으로는

어떠한 정보도 노출되지 않아야 한다는 것이다.

정의 1. 다수의 사용자에 대한 암호화된 데이터베이스 시스템 Γ는 모든 데이터베이스 D와 모든 다항시간 (PPT, Probabilistic Polynomial Time) 알고리즘 A에 대해서, 다음을 만족하는 다항시간 알고리즘 A^*가 존재하면 질의 프라이버시를 만족한다. $v(\kappa)$는 무시할 수 있을 만큼 작은 값으로 수렴하는 함수이다(negligible function).

$$\left|\Pr[A(V_t)=f(D,W_t)]-\Pr[A^*(T_t)=f(D,W_t)]\right| < v(\kappa) \quad (1)$$

서버에 (임의의 사용자에 의해서) i번째 저장되는 데이터와 키워드를 각각 (d_i,w_i)라고 하며, (d_i,w_i)를 암호화한 암호문을 EI_{d_i}라고 한다. 이런 EI_{d_i}들의 집합은 EI_D이다. SU_{list}는 서버에 등록된 사용자들의 정보 목록이다. 그리고 $Q_t=(q_1,...,q_t)$를 사용자 그룹 내에서 발생한 t개의 질의, $W_t=(qw_1,...,qw_t)$를 질의된 키워드, $ANS_t=(ans_1,...,ans_t)$를 질의에 대한 결과라고 하자. 우리는 $\Omega(q_i)=(qu_i,\{j\}_{qw_i})$를 정의하며, 이 때 qu_i는 i번째 질의를 하는 사용자이며, $\{j\}_{qw_i}$는 i번째 질의에 들어있는 키워드 qw_i를 포함하는 EI_{d_j}들의 인덱스 집합을 말한다. [5]의 정의를 이용해서 V_t를 서버가 질의를 통해서 얻을 수 있는 모든 실질적인 정보, 뷰(view)라 하면 이러한 V_t는 실제 질의와 해당하는 결과를 포함하고 있다. V_t를 $V_t=\{EI_D,SU_{list},Q_t,ANS_t\}$로 정의한다. 그리고 T_t를 t개의 질의에 대한 흔적, 즉 어쩔 수 없이 노출되어야 하는 정보들이라고 하고 $T_t=\{|EI_D|,|SU_{list}|,\Omega(q_1),...,\Omega(q_t)\}$로 정의한다. $|EI_D|$, $|SU_{list}|$는 각각 암호화된 데이터의 개수, SU_{list}에 포함된 사용자의 수를 의미한다.

정의에 따라 질의 프라이버시를 증명하기 위해서는, 주어진 흔적 T_t를 가지고서 V_t를 시뮬레이션 할 수 있는 시뮬레이터 S가 존재함을 보인다. 그러면 S를 이용해서 $A^*(T_t)=A(S(T_t))$를 구성할 수가 있다. S는 질의를 하고 그 결과를 대답하는 과정에서 어쩔 수 없이 노출되는 정보를 가지고, 서버가 얻을 수 있는 모든 정보를 시뮬레이션 할 수 있다는 것을 의미한다. 그러면, A가 자신이 가진 V_t를 이용해서 (D,W_t)에 대해 얻을 수 있는 정보와 A^*가 자신이 가진 T_t를 이용해서 (D,W_t)에 대해 얻을 수 있는 정보의 차이가 없다는 것을 알 수 있다.

4.2.2 질의 위조 불가능성(Query unforegeability)

모든 검색 가능한 암호화 기법들은 질의의 생성이 정당한 사용자에 의해서만 이루어져야 한다. 정당한 사용자는 UM으로부터 받은 자신의 비밀키를 이용해서 질의를 생성하고, 서버는 이 질의가 정당한 사용자의 것임을 확인하고 질의에 대한 답을 한다. 이러한 질의는 다른 사용자는 물론이고 서버 또한 정당한 사용자인 것처럼 질의를 생성할 수 없어야 한다. 검색 가능한 암호화 기법에서 이러한 안전성 조건을 질의 위조 불가능성이라고 한다.

정의 2. 다수의 사용자에 대한 암호화된 데이터베이스 시스템 Γ는 임의의 사용자 $\hat{u}$와 PPT 공격자 A_U, A_S에 대해서 다음을 만족할 때 질의 생성 불가능성을 만족한다.

$$\begin{aligned}&\Pr[q\in Q_{\hat{u}}\setminus Q_{\hat{u}}^{'}:\\ &\quad (params,sk_{UM})\leftarrow Setup(1^{\kappa});\\ &\quad \forall u\in U,(qk_u;.;SU_{list})\\ &\quad \leftarrow Join(u;sk_{UM};SU_{list});\\ &\quad q\leftarrow A_U^{Store,Query,Search}(\{qk_u|u\in U\setminus\{\hat{u}\}\})\\ &\quad \text{or } q\leftarrow A_S^{Query}(SU_{list})\\ &]<\nu(\kappa)\end{aligned} \quad (2)$$

qk_u는 사용자 u가 질의를 생성할 때 사용하는 비밀키이고, SU_{list}는 서버에 저장되는 등록된 사용자들에 대한 정보 목록이다. 질의 위조 불가능성을 설명하기 위해서 우선 정당한 사용자를 정의해야 한다. 정당한 사용자란 UM으로부터 개인키를 받아서 서버에 정당한 검색을 할 수 있는 능력을 가진 사용자들을 말한다. 정당한 사용자 질의의 집합을 $Q_u=\{q_u(w)|q_u(w)\leftarrow Query(qk_u,w),w\in W\}$로 정의한다. 여기서 W는 가능한 키워드들의 집합이다. 즉, 정당한 질의를 생성하기 위해서는 정당한 사용자의 개인키와 키워드가 $Query$라는 단계를 거쳐야 한다는 것이다. 직관적으로 개인키를 가지지 않은 사용자는 정당한 질의를 생성할 수 없으므로 질의 위조 불가능성을 만족한다고 할 수 있다.

질의 위조 불가능성을 증명하기 위해서 공격자(adversary)와 챌린저(challenger)의 게임을 정의한다. 우선 공격자의 유형을 두 가지로 나눈다. A_U를 임의의 사용자가 타겟 사용자 $\hat{u}$에 대한 질의를 위조하는 공격자라 하고, A_S를 서버가 타겟 사용자 $\hat{u}$에 대한

〔표 1〕 제안하는 기법

Phase	User	User Management	Server
『*Setup*』		$sk_{UM}=(k_m,s)$ $params=(G,G_1,e)$	
『*Join*』	u →	$U_{list}=U_{list}\cup\{u\}$, $k_u\in Z_p^*$, $s_u=g^{k_{um}/k_u}$	u, s_u → $SU_{list}=SU_{list}\cup\{(u,s_u)\}$
	k_u, s ←		
	$qk_u=(k_u,s)$ ←		
『*Revoke*』		$U_{list}=U_{list}\setminus\{u\}$	u → $SU_{list}=SU_{list}\setminus\{(u,s_u)\}$
『*Store*』	$t_i,r_i\in Z_p^*$,	$u, g^{t_i}, h_s(w_i)^{r_i}$ →	Check u $\overline{k_{w_i}}=e(h_s(w_i)^{r_i},s_u)$, $\overline{k_{d_i}}=e(g^{t_i},s_u)$
	$k_{w_i}=H((\overline{k_{w_i}})^{k_i/r_i})$, $k_{d_i}=H((\overline{k_{d_i}})^{k_u})$	← $\overline{k_{w_i}}, \overline{k_{d_i}}$	
	$R_i\in Z_p^*$ $EI_i=\{g^{t_i}, E_{k_{d_i}}(d_i), R_i, E_{k_{w_i}}(R_i)\}$	EI_i →	set $EI_D=EI_D\cup\{EI_i\}$
『*Query*』	$q_u(w)=\{u, h_s(w)^{k_u}\}$		
『*Search*』		$q_u(w)$ →	Check u $k_w=H(e(h_s(w)^{k_u}, s_u))$ For all $EI_i\in EI_D$, if $D_{k_w}(E_{k_{w_i}}(R_i))=R_i$, then $ans_u(w)=ans_u(w)\cup\{(E_{k_{d_i}}(d_i), \overline{k_{d_i}})\}$
	For all $(E_{k_d}(d), \overline{k_d})\in ans_u(w)$, $k_d=H((\overline{k_d})^{k_u})$ $d=D_{k_d}(E_{k_d}(d))$	← $ans_u(w)$	

질의를 위조하는 공격자라고 하자. 각각은 서로 알고 있는 것이 다르며 공격능력 또한 다르다. A_U의 게임에서는 챌린저가 I의 실행을 시뮬레이션하고, 임의의 사용자 $\hat{u}$에 대한 키워드 질의를 생성할 수 있게 해주는 *Query*를 포함한 *Search*, *Store* 오라클을 제공한다. A_S의 게임에서는 공격자가 서버이기 때문에 SU_{list}에 저장된 값을 알 고 있고, $\hat{u}$의 질의의 집합을 알고 있다고 가정한다. 그래서 챌린저는 A_S에게 *Query* 오라클을 제공해야 한다. 두 가지 게임은 다음과 같이 이루어진다. 공격자는 우선 타겟 사용자 $\hat{u}$를 설정한다. 그리고 챌린저는 타겟 사용자 $\hat{u}$를 제외한 모든 사용자의 키를 시뮬레이션해서 공격자 A_U에게 주고 타겟 사용자 $\hat{u}$에 대한 모든 오라클 질의를 시뮬레이션 해준다. A_S의 경우에도 마찬가지로 챌린저가 서버가 알아야 할 SU_{list}에 포함된 값을 시뮬레이션 해주고 *Query* 오라클도 시뮬레이션 해준다. $Q_{\hat{u}}$를 사용자 $\hat{u}$의 생성가능한 질의의 집합이라고 하고, $Q_{\hat{u}}'$을 공격자가 *Query*를 통해서 얻은 사용자 $\hat{u}$의 질의의 집합이라고 했을 때, 공격자는 $q\in Q_{\hat{u}}\setminus Q_{\hat{u}}'$인 경우에 게임에서 이길 수 있다.

4.2.3 탈퇴한 사용자로부터의 안전성(Full-revocability)

사용자의 가입과 탈퇴가 자주 일어나는 현재의 데

이터베이스 환경에서는 사용자를 탈퇴시키는 과정은 필수적이다. 신뢰할 수 있는 *UM*이 정당한 사용자 목록에서 사용자를 삭제하고 이를 서버에게 알려주어서 탈퇴한 사용자가 정당한 질의를 생성할 수 없어야 하고, 검색을 할 수 없어야 한다.

정의 3. 다수의 사용자에 대한 암호화된 데이터베이스 시스템 Γ는 사용자 u와 PPT 공격자 $A=(A_1,A_2)$에 대해서 다음을 만족할 때 탈퇴한 사용자로부터의 안전성을 만족한다.

$$\begin{aligned}&\Pr[b'=b:(params,sk_{UM})\leftarrow Setup(1^{\kappa});\\&\quad\forall u\in U,(qk_u;\cdot;SU_{list})\leftarrow Join(u;sk_{UM};SU_{list});\\&\quad(state1,u^*)\leftarrow A_1(U);\\&\quad Revoke(u^*);\\&\quad(state2,\hat{u},d_0,d_1,w_0,w_1)\leftarrow A_2(state1,qk_{u^*});\\&\quad b\in_R\{0,1\};\\&\quad(\cdot;\widetilde{EI_D})\leftarrow Store(qk_{\hat{u}},d_b,w_b;SU_{list},EI_D);\\&\quad EI_d=\widetilde{EI_D}-EI_D;\\&\quad b'\leftarrow A_3(state2,EI_d)\\&]<1/2+\nu(\kappa)\end{aligned}\tag{3}$$

공격자 $A=(A_1,A_2,A_3)$의 공격 성공을 다음의 게임에서 이기는 것으로 정의한다. A_1은 *UM*에 등록된 정당한 사용자 중 한명인 u^*를 선택한다. A_2는 선택된 사용자 u^*의 개인키인 qk_{u^*}와 A_1의 상태 정보인 $state1$을 받는다. u^*는 정당한 사용자에서 삭제된다. A_2는 두 개의 키워드 (w_0,w_1)와 각각의 키워드와 연결된 데이터 (d_0,d_1)을 선택하며, *UM*에 등록된 정당한 사용자 중 한명인 $\hat{u}$을 선택한다. A_3는 두 개의 키워드-데이터 쌍 중에서 랜덤하게 선택된 하나의 키워드-데이터 쌍을 암호화한 EI_d와 A_2의 상태 정보인 $state2$를 받고서 어떤 키워드-데이터 쌍이 암호화되어있는지를 추측하는 b'을 출력한다. 여기서 $b'=b$를 만족하면 A는 게임에서 이기게 된다.

[11]에서 제안하는 기법은 탈퇴한 사용자로부터의 안전성을 만족하지 않는다. 왜냐하면 [11]의 기법에서는 데이터를 암호화하는 키가 모두 동일하기 때문에 u^*가 탈퇴한 상태가 되더라도 주어진 EI_d를 qk_{u^*}로 복호화해볼 수 있기 때문에 b를 알아낼 수 있다.

V. 제안하는 기법

본 논문에서 제안하는 기법은 *Setup*, *Join*, *Revoke*, *Store*, *Query*, *Search* 단계로 구성되어 있다. 제안하는 기법의 구체적인 설명은 다음과 같다.

5.1. 구성

【$Setup(1^{\kappa})$】 *UM*은 시큐리티 파라미터(security parameter) 1^{κ}를 입력받아서 공개 파라미터 $params=\{G,G_1,e:G\times G\rightarrow G_1\}$를 설정하고, *UM*의 개인키 $k_m\in_R Z_p^*$를 랜덤하게 선택한다. 그리고 안전한 대칭키 암호화 기법 E, 충돌 저항성(collision resistance) 해시함수 $H:G_1\rightarrow K$, 해시함수의 시드 값 $s\in_R S$를 랜덤하게 선택하고, 시드를 가지는 충돌 저항성 해시함수 $h_s:S\times W\rightarrow G$를 설정한다. 여기서 K는 가능한 키들의 집합, S는 시드들의 집합, W는 가능한 키워드들의 집합이다.

【$Join(u;sk_{UM};SU_{list})$】 등록을 원하는 사용자 u는 *UM*에게 u를 전송하고, *UM*은 이 값을 사용자 목록 U_{list}에 포함시킨다. 그리고 *UM*은 사용자 u의 개인키 $k_u\in_R Z_p^*$를 랜덤하게 선택하고, 서버에 저장되는 사용자 u에 대한 키 $s_u=g^{k_m/k_u}$를 계산한다. 마지막으로 *UM*은 사용자 u에게 $\{k_u,s\}$, 서버에게 $\{u,s_u\}$를 전송하고, 서버는 서버의 사용자 목록 SU_{list}에 $\{u,s_u\}$를 포함시킨다.

【$Revoke(u;SU_{list})$】 *UM*은 탈퇴하고자 하는 사용자 u를 사용자 목록 U_{list}에서 제거하고, 서버에 u를 전송한다. 서버 역시 서버의 사용자 목록 SU_{list}에서 $\{u,s_u\}$를 제거한다.

【$Store(qk_u,d,w;SU_{list},EI_D)$】 사용자 u는 문서를 암호화하기 위한 키 k_{d_i}와 키워드 w에 대한 인덱스 생성키 k_{w_i}를 생성하기 위해 먼저 $r_i\in_R Z_p^*$와 $t_i\in_R Z_p^*$를 랜덤하게 선택하고, $\{u,g^{t_i},h_s(w)^{r_i}\}$를 계산해서 서버에게 전송한다. 그러면 서버는 사용자 u가 k_{d_i}와 k_{w_i}를 생성할 수 있도록 $\overline{k_{d_i}}$와 $\overline{k_{w_i}}$를 아래와 같이 생성해서 사용자 u에게 전송한다.

$$\begin{aligned}\overline{k_{d_i}}&=e(g^{t_i},s_u)=e(g^{t_i},g^{k_m/k_u}),\\\overline{k_{w_i}}&=e(h_s(w)^{r_i},s_u)=e(h_s(w)^{r_i},g^{k_m/k_u})\end{aligned}\tag{4}$$

사용자 u는 서버에게 받은 $\overline{k_{d_i}}$와 $\overline{k_{w_i}}$로부터 k_{d_i}와 k_{w_i}를 다음과 같이 생성한다.

$$k_{d_i} = H(e(g^{t_i}, g^{k_m/k_u})^{k_u}) = H(e(g^{t_i}, g^{k_m})),$$
$$k_{w_i} = H(e(h_s(w)^{r_i}, g^{k_m/k_u})^{k_u/r_i}) = H(e(h_s(w), g^{k_m})) \quad (5)$$

그리고 사용자 u는 저장하기 원하는 문서 d_i를 위에서 생성한 키 k_{d_i}를 사용하여 $E_{k_{d_i}}(d_i)$로 암호화하고, 이 문서에 대해 키워드 w로 검색 가능하도록 랜덤한 수 $R_i \in Z_p^*$를 선택하여 위에서 생성한 키 k_{w_i}를 사용하여 $E_{k_{w_i}}(R_i)$로 암호화한다. 마지막으로 사용자 u는 $EI_i = \{T_i = g^{t_i}, C_i = E_{k_{d_i}}(d_i), I_i = \{I_i[1], \quad I_i[2]\} = \{R_i, E_{k_{w_i}}(R_i)\}\}$를 서버에 저장한다.

【$Query(qk_u, w)$】 키워드 w를 사용하여 문서를 검색하고자 하는 사용자 u는 $q_u(w) = \{u, h_s(s)^{k_u}\}$를 계산한다.

【$Search(qk_u, q_u(w); SU_{list}, EI_D)$】 사용자 u로부터 $q_u(w) = \{u, h_s(s)^{k_u}\}$를 받은 서버는 SU_{list}에 u가 있는지 확인하고, 만약 없으면 정당한 사용자가 아니므로 종료한다. SU_{list}에 u가 있다면, 서버는 SU_{list}의 u와 대응되는 s_u를 사용하여 k_w를 다음과 같이 계산한다.

$$k_w = H(e(h_s(w)^{k_u}, s_u)) = H(e(h_s(w)^{k_u}, g^{k_m/k_u})) = H(e(h_s(w), g^{k_m})) \quad (6)$$

서버는 저장된 $EI_i = \{g^{t_i}, E_{k_{d_i}}(d_i), R, E_{k_{w_i}}(R)\}$들의 집합 EI_D 중에서 $\{R_i, E_{k_{w_i}}(R_i)\}$이 지금 계산된 k_w를 사용하여 암호화된 것들을 모두 스캔하고, 맞다면 $E_{k_{d_i}}(d_i)$와 $\overline{k_{d_i}} = e(g^{t_i}, s_u)$를 계산하여 사용자 u에게 전송한다. 사용자 u는 이 값을 사용하여 k_{d_i}를 다음과 같이 생성할 수 있다.

$$k_{d_i} = H((\overline{k_{d_i}})^{k_u}) = H(e(g^{t_i}, s_u)^{k_u}) = H(e(g^{t_i}, g^{k_m/k_u})^{k_u}) = H(e(g^{t_i}, g^{k_m})) \quad (7)$$

결과적으로 사용자 u는 이렇게 계산된 k_{d_i}를 사용하여 $E_{k_{d_i}}(d_i)$를 복호화할 수 있다.

5.2. 정확성(correctness)

제안하는 기법의 정확성은 다음과 같다. 어떤 사용자 u가 어떤 문서 d_i에 대해 키워드 w를 사용하여 서버에 저장한 값을 $EI_i = \{g^{t_i}, E_{k_{d_i}}(d_i), R_i, E_{k_{w_i}}(R_i)\}$라고 하자. 그러면 이 값에 대해 $k_w = H(e(h_s(w), g^{k_m}))$이고, $k_{d_i} = H(e(g^{t_i}, g^{k_m}))$이다. 그 이후 다른 사용자 u'이 동일한 키워드 w를 사용하여 생성된 질의 $q_{u'}(w) = \{u', h_s(w)^{k_{u'}}\}$를 서버에 보내면, 서버는 $k_w' = H(e(h_s(w)^{k_{u'}}, g^{k_m/k_{u'}})) = H(e(h_s(w), g^{k_m}))$을 생성한다. 그러면 결국 $k_w = k_w'$를 만족하므로, 사용자 u'은 사용자 u가 저장한 암호화된 문서를 받을 수 있다. 그리고 사용자 u'은 서버에게 전송받는 $ans_{u'}(w) = \{E_{k_{d_i}}(d_i), \overline{k_{d_i}} = e(g^{t_i}, s_{u'})\}$로부터 $k_{d_i}' = H((\overline{k_{d_i}})^{k_{u'}}) = H(e(g^{t_i}, s_{u'})^{k_{u'}}) = H(e(g^{t_i}, g^{k_m/k_{u'}})^{k_{u'}}) = H(e(g^{t_i}, g^{k_m}))$를 계산할 수 있다. 그러면 $k_{d_i} = k_{d_i}'$이므로, 사용자 u'은 사용자 u가 저장한 암호화된 문서를 복호화할 수 있다. 제안하는 기법에서 사용되는 해시함수 H, h_s는 충돌 저항성 해시함수이기 때문에, 서로 다른 키워드에 대해 동일한 키를 생성할 확률은 무시해도 좋을 정도이다.

5.3. 안전성 증명

본 논문에서 제안하는 기법에 대한 안전성을 4장에서 정의한 안전성 정의를 바탕으로 증명한다.

정리 1. 제안하는 기법은 질의 프라이버시, 질의 위조 불가능성, 탈퇴한 사용자로부터의 안전성을 만족하는 안전한 키워드 검색 기법이다.

아래의 부명제 1, 2, 3을 증명함으로써 본 논문에서 제안하는 기법이 질의 프라이버시, 질의 위조 불가능성, 탈퇴한 사용자로부터의 안전성을 만족함을 보인다.

5.3.1 질의 프라이버시(Query Privacy)

부명제 1. $E(\cdot)$이 의사 난수 치환이고 $h_s(\cdot)$가 의사 난수 함수라면 제안하는 기법은 질의 프라이버시를 만족한다.

증명) 주어진 t개의 질의에 대한 흔적 T_t를 이용해서 시뮬레이터 S가 V_t와 구별 불가능한 V_t^*를 시뮬레이션 한다면 질의 프라이버시를 제공한다는 것을 증명하기에 충분하다.

시뮬레이터 S는 질의에 대한 흔적 $T_t = \{|EI_D|, |SU_{list}|, \Omega(q_1), ..., \Omega(q_t)\}$가 주어진다. 이런 흔적 T_t가 $V_t = \{EI_D, SU_{list}, Q_t, ANS_t\}$로부터 발생했다고 하자. 또

한 이런 V_t가 실제 데이터베이스 $D=\{(d_i,w_i)\}_{i=1}^{|EI_D|}$, 사용자 집합 $U=(u_1,...,u_{|SU_{list}|})$와 질의 키워드 집합 $W_t=(qw_1,...,qw_t)$로부터 $\Omega(q_i)=(qu_i,\{j\}_{qw_i})$가 되도록 발생했다고 하자.

시뮬레이터 S는 D와 W_t를 모르지만 $V_t=\{EI_D, SU_{list}, Q_t, ANS_t\}$와 구별 불가능한 $V_t^*=\{EI_D^*, SU_{list}^*, Q_t^*, ANS_t^*\}$를 생성하기 위해서 임의로 $D^*=\{(d_i^*,w_i^*)\}_{i=1}^{|EI_D|}$와 $W_t^*=(qw_1^*,...,qw_t^*)$를 $\Omega(q_i^*)=\Omega(q_i)$가 만족되게끔 생성한다. S는 또한 임으로 $(s^*,k_m^*,k_{u_i}^*)$를 생성한다. 그런 이후에 S는 $(s^*,k_m^*,k_{u_i}^*)$, D^*, W_t^*를 이용해서 $\Omega(q_i^*)=\Omega(q_i)$가 되도록 제안된 프로토콜을 그대로 실행한다. S는 이렇게 실행한 후에 만들어지는 $V_t^*=\{EI_D^*, SU_{list}^*, Q_t^* ANS_t^*\}$를 출력한다.

(D^*, W_t^*)과 (D, W_t)는 비록 다를지라도, $V_t=\{EI_D, SU_{list}, Q_t, ANS_t\}$와 $V_t^*=\{EI_D^*, SU_{list}^*, Q_t^*, ANS_t^*\}$는 같은 질의에 대한 흔적을 가지고 있다. 즉, $T_t=T_t^*$이다. 또한, $E(\bullet)$가 의사 난수 치환이고, $h_s(\bullet)$가 의사 난수 함수라면 $V_t=\{EI_D, SU_{list}, Q_t, ANS_t\}$와 $V_t^*=\{EI_D^*, SU_{list}^*, Q_t^*, ANS_t^*\}$는 계산적으로 구별 불가능하다는 것을 알 수 있다.

5.3.2. 질의 위조 불가능성(Query unforgerability)

부명제 2. BLS short signature가 안전하면 제안하는 기법은 질의 위조 불가능성을 만족한다.

증명) 공격자를 사용자가 공격자가 되는 경우인 악의적인 사용자 A_U와 서버가 공격자가 되는 경우인 악의적인 서버 A_S로 정의하고, 이 두 가지 공격자에 대해서 제안하는 기법의 안전성을 증명을 한다.

i) 공격자 A_U

제안하는 기법의 질의를 위조하는 공격자 A_U를 이용해서 BLS 서명을 위조하는 공격자 B를 만들 수 있음을 보임으로써 제안하는 기법의 질의 위조 불가능성에 대한 안전성을 증명한다. 공격자 A_U는 타겟 사용자인 $\hat{u}$의 질의의 집합 $Q=\{h_s(w_1)^{k_{\hat{u}}}, h_s(w_2)^{k_{\hat{u}}}, ...\}$을 $Query(qk_{\hat{u}}, \bullet)$를 통해서 얻을 수 있다.

- *Setup* : 준비과정에서 B는 BLS short signature의 변수들인 G_1, G_2, e를 사용한다. B는 임의의 시드(seed)값 s를 선택한다. k_m을 선택하기 위해서 B는 $k_j \in_R Z_p^*$를 택해서 $K=\{k_1,...,k_{\max}\}$를 설정한다. 여기서 max는 최대 사용자 수를 나타낸다. 그리고 나서 B는 $\widetilde{k_m}=y^{\prod_{j=1}^{R} k_j}=g^{k\prod_{j=1}^{uR} k_j} \in G_1$를 계산한다.

- *Join* : 사용자 u를 등록하기 위해서 B는 K로부터 임의의 값을 뽑아서 사용자 u의 키로 지정한다. 각각의 사용자 $u_i \in U \setminus \{\hat{u}\}$에 대해서 다른 $k_j \in K$를 선택해서 $(qk_{u_i}, s_{u_i})=((k_i,s), y^{k_1...k_{i-1}k_{i+1}...k_{uR}})$를 설정한다. 타겟 사용자 $\hat{u}$에 대해서는 $s_{\hat{u}}=g^{\prod_{i=1}^{R} k_i}$로 설정한다. B는 $u_i \in U \setminus \{\hat{u}\}$에 대해서 qk_{u_i}를 A_U에게 준다. 이렇게 설정하면 타겟 사용자 $\hat{u}$의 개인키 $k_{\hat{u}}=k$가 된다.

- *Store, Search* : 이 세 단계에서는 B가 이전 단계에서 모든 사용자의 SU_{list}를 생성했기 때문에 그 값을 이용해서 B가 A_U의 모든 요청에 응답할 수가 있다.

- *Query* : B는 서명을 위조하는 공격자이기 때문에 서명 오라클 $O(\bullet)$을 이용할 수 있다. 그리고 B는 A_U의 $h_s(\bullet)^k$를 시뮬레이션 해주기 위해서 서명 오라클 $O(\bullet)$을 사용한다. B가 키워드 w를 A_U로부터 받으면 B는 w를 $O(\bullet)$에게 주어서 결과를 받는다. 이러한 결과가 유효한 것인지 서명 기법의 확인 알고리즘을 통해서 확인한 후 유효하다면 그 결과를 A_U에게 준다. A_U에 의해서 얻은 집합 $\{h_s(w_1)^{k_u}, h_s(w_2)^{k_u}, ...\}$은 $\{h(w_1)^{k_u}, h(w_2)^{k_u}, ...\}$가 되고 이러한 값은 BLS signature의 형태가 된다.

- A_U가 질의 위조에 성공해서 위조된 질의를 출력하면 B는 A_U가 위조한 값을 그대로 출력함으로써 BLS short signature를 위조할 수 있다.

〔표 2〕 기존 기법과 제안하는 기법의 비교

	[5]	[10]	[12]	[11]	제안하는 기법
동적인 환경	O	X	X	△*	O
접근제어	△	O	O	△	△
데이터 공급자의 수	1	1	1	n	n
데이터 검색자의 수	n	n	n	n	n

ii) 공격자 A_S

증명방법은 i)과 동일하다. 다른 점은 위의 공격자 A_U와는 다르게 A_S는 서버가 공격자인 형태이므로 시드값 s를 알지 못하는 대신에 s_{u_i}를 알게 된다. 자세한 증명은 다음과 같다.

- *Setup* : i)과 동일하다.
- *Join* : 기본적인 키의 생성방법은 i)과 동일하지만 A_S는 서버이기 때문에 qk_u를 주는 것이 아니라, s_{u_i}를 공격자인 A_S에게 준다는 점만 다르다.
- *Store, Search* : i)과 동일하다.
- *Query* : A_S는 $Query(qk_u, \bullet)$ 오라클을 사용해서 타겟 사용자 $\hat{u}$에 대한 질의를 얻을 수 있다. 나머지 과정은 i)과 동일하다.
- A_S가 질의 위조에 성공해서 위조된 질의를 출력하면 B는 A_S가 위조한 값을 그대로 출력함으로써 BLS short signature를 위조할 수 있다.

5.3.3 탈퇴한 사용자로부터의 안전성(Full-revocability)

부명제 3. $E(\bullet)$이 안전한 암호화 기법이라면 제안하는 기법은 탈퇴한 사용자로부터의 안전성을 만족한다.

증명) $R_0, R_1 \in_R Z_p^*$에 대해서 k_{w_0}, k_{w_1}를 각각 키워드 w_0, w_1에 대한 인덱스를 생성하는 키라고 했을 때 두 개의 키워드에 대한 인덱스는 $I_0 = \{R_0, E_{k_{w_0}}(R_0)\}$, $I_1 = \{R_1, E_{k_{w_1}}(R_1)\}$로 구성된다. 탈퇴한 사용자는 SU_{list}에서 s_u값이 삭제되었기 때문에 k_{w_0}, k_{w_1}를 생성할 수가 없다. 그리고 $E(\bullet)$이 안전한 암호화 기법이기 때문에 w_0, w_1에 관한 정보를 얻을 수 없다. 마찬가지로 g^{t_0}, $E_{k_{w_0}}(d_0), g^{t_1}, E_{k_{w_1}}(d_1)$에 대해서도 복호화 해 볼 수 있는 키를 생성할 수 없고, $E(\bullet)$가 안전한 암호화 기법이기 때문에 d_0, d_1에 관한 정보도 얻을 수 없다. 그러므로 EI_d는 $w_0 - d_0$쌍에 대한 암호문인지 $w_1 - d_1$쌍에 대한 암호문인지 탈퇴한 사용자 입장에서는 구별 불가능하다. 즉, 4장에서 정의한 수식 (3)이 만족함을 보일 수 있다.

5.4. 분석

이번 절에서는 다수의 사용자를 고려한 다른 기법들[5,10,11,12]과 제안하는 기법을 [표 2]와 같이 비교 분석한다. (* : 동적인 환경을 고려했으나 안전하지 않음)

문헌 [5], [10]과 [12]는 유일한 데이터 공급자를 가지며, 데이터 공급자가 다수의 데이터 검색자에 대해 접근제어 정책을 설정하는 환경에서 기법을 제안하였다. 하지만 데이터 공급자가 문서를 서버에 저장할 때 접근제어 정책이 결정되는 구조이기 때문에 데이터 검색자의 추가적인 등록 및 탈퇴가 불가능한 단점을 가진다. 그리고 문헌 [11]은 다수의 데이터 공급자와 다수의 데이터 검색자를 고려한 환경에서 기법을 제안하였다. 하지만 이것은 탈퇴자로부터의 안전성에 결함을 가진다. 다시 말하면, 서버에 저장되는 모든 문서의 암호화 키가 동일하기 때문에 탈퇴한 사용자도 도청을 통해 얼마든지 추가적인 정보의 획득이 가능하다. 우리가 제안하는 기법은 문헌 [11]에서의 장점을 모두 수렴하면서도, 모든 문서의 암호화 키를 다르게 설정하였다. 즉, 본 논문에서 최초로 동적 환경에서 안전한 기법을 제안하였다. 또한 제안하는 기법에서 서버를 반-신뢰(semi-trust)한다는 가정하에 데이터 공급자가 문서를 암호화하여 저장할 때 접근 가능한 사용자 목록을 추가적으로 저장하는 방법을 사용하여 접근제어 정책이 가능하다.

VI. 결 론

본 논문에서는 동적 환경에서 프라이버시를 보호하는 키워드 검색 기법을 제안하였다. 기존 연구에서 모든 문서의 동일한 암호화 키를 사용하는 문제점을 해결하기 위해 본 논문에서는 암호화된 문서별로 다른 암호화 키를 설정하였으며, 사용자가 문서를 검색할 때 문서에 대한 암호화 키를 계산할 수 있기 때문에 암호화된 문서의 복호화가 가능하다. 결과적으로 제안하는 기법은 다수의 사용자 환경에서 탈퇴한 사용자에 대한 안전성을 가지는 최초의 기법이며, 제안하는 기법을 사용하여 동적 환경에서 사용자의 프라이버시를 보호하는 키워드 검색이 가능하다.

참고문헌

[1] 김선영, 서재우, 이필중, "검색가능 암호기술의 연구 동향," 정보보호학회지, 19(2), pp. 63-73, 2009년 4월.

[2] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," IEEE Symposium on

Security and Privacy, pp. 44-55, May 2000.

[3] E. Goh, "Secure Indexes," Technical report 2003/216, In IACR ePrint Cryptography Archive, Oct. 2003.

[4] Y. Chang, and M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," ACNS, LNCS 3531, pp. 442-455, 2005.

[5] R. Curtmola, J. Garay, S. Kamara, R. Ostrovskey, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," ACM CCS, pp. 79-88, Oct. 2006.

[6] D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," EUROCRYPT, LNCS 3027, pp. 506-522, 2004.

[7] P. Golle, J. Staddon, and B. Waters, "Secure Conjunctive Keyword Search over Encrypted Data," ACNS, LNCS 3089, pp. 31-45, 2004.

[8] D. Boneh and B. Waters, "Conjunctive, Subset, and Range Queries on Encrypted Data," TCC, LNCS 4392, pp. 535-554, 2007.

[9] J. Katz, A. Sahai, and B. Waters, "Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products," EUROCRYPT, LNCS 4965, pp. 146-162, 2008.

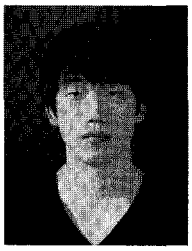
[10] S. Yau and Y. Yin, "Controlled privacy preserving keyword search," ASIACCS, pp. 321-324, Mar. 2008

[11] F. Bao, R. Deng, X. Ding, and Y. Yang, "Private query on encrypted data in multi-user settings," Information Security Practice and Experience, 4th International Conference, LNCS 4991, pp. 71-85. 2008.

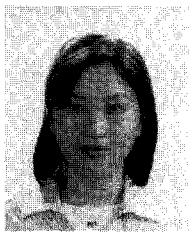
[12] 노건태, 천지영, 정익래, 이동훈, "프라이버시를 보호하는 접근제어가 가능한 키워드 검색 기법," 정보보호학회논문지, 19(5), pp. 35-44, 2009년 10월.

[13] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil Pairing," ASIACRYPT, LNCS 2248, pp. 514-532. 2001.

〈著者紹介〉

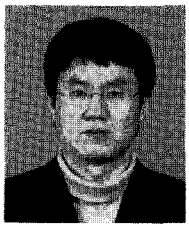


김 동 민 (Dong Min Kim) 학생회원
2009년 2월: 서울시립대학교 수학과 졸업
2009년 3월~현재: 고려대학교 정보경영공학과 석사과정
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 보안, 암호 이론



천 지 영 (Ji Young Chun) 학생회원
1997년 2월: 이화여자대학교 수학과 학사 졸업
2006년 2월: 고려대학교 정보경영공학과 석사 졸업
2006년 3월~현재: 고려대학교 정보경영공학과 박사과정
〈관심분야〉 암호 이론, 프라이버시향상기술(PET), 유비쿼터스 보안

노 건 태 (Geon Tae Noh) 학생회원
2008년 2월: 고려대학교 산업시스템정보공학과 학사 졸업
2010년 2월: 고려대학교 정보경영공학과 석사 졸업
2010년 3월~현재: 고려대학교 정보보호학과 박사과정
〈관심분야〉 암호 이론, 프라이버시향상기술(PET), 유비쿼터스 보안



정 익 래 (Ik Rae Jeong) 정회원
1998년 2월: 고려대학교 전산학과 학사 졸업
2000년 2월: 고려대학교 전산학과 석사 졸업
2004년 8월: 고려대학교 정보보호학과 박사 졸업
2006년 6월~2008년 2월: 한국전자통신연구원 암호기술연구팀 선임연구원
2008년 3월~현재: 고려대학교 정보경영공학부 조교수
〈관심분야〉 프라이버시향상기술(PET), 데이터베이스 암호, 암호 이론