논문 2009-46CI-1-5

암호화 문서상에서 효율적인 키워드 검색 프로토콜 설계

(A Design of Efficient Keyword Search Protocol Over Encrypted Document)

변 진 욱*

(Jin Wook Byun)

요 약

본 논문에서는 공통의 키워드들을 포함하는 암호화 문서들을 검색하는 프로토콜에 대해서 연구한다. 공통의 키워드 검색 프로토콜은 자료 공급자 (data supplier), 자료 저장소 (database) 그리고 사용자 (user of database) 로 이루어진다. 자료 공급자는 암호화된 문서를 자료 저장소에 저장하게 되고 정당한 사용자는 원하는 키워드들을 질의하여 해당 키워드들이 공통으로 포함된 암호화 문서들을 얻을 수 있다. 최근, 많은 공통의 키워드 검색 프로토콜들이 다양한 환경에서 제안되었다. 하지만, 제안된 프로토콜들은 자료 공급자 및 자료 저장소 관점에서 많은 계산적 비용을 필요로 한다. 더욱이 지금까지 제안된 프로토콜들의 안전성은 랜덤 오라클 (random oracle) 모델에서만 증명되었다. 본 논문에서는 암호화 문서상에서 효율적인 공통의 키워드 검색 프로토콜을 랜덤 오라클 가정 없이 설계한다. 또한 사용자의 자료 저장량 그리고 자료 저장소의 계산량, 통신량 비용이 상수양의 비용을 가진다. 제안된 프로토콜의 안전성은 DBDH (Decisional Bilinear Diffie-Hellman) 문제의 어려움에 기반 한다.

Abstract

We study the problem of searching documents containing each of several keywords (conjunctive keyword search) over encrypted documents. A conjunctive keyword search protocol consists of three entities: a data supplier, a storage system such as database, and a user of storage system. A data supplier uploads encrypted documents on a storage system, and then a user of the storage system searches documents containing each of several keywords. Recently, many schemes on conjunctive keyword search have been suggested in various settings. However, the schemes require high computation cost for the data supplier or user storage. Moreover, up to now, their securities have been proved in the random oracle model. In this paper, we propose efficient conjunctive keyword search schemes over encrypted documents, for which security is proved without using random oracles. The storage of a user and the computational and communication costs of a data supplier in the proposed schemes are constant. The security of the scheme relies only on the hardness of the Decisional Bilinear Diffie-Hellman (DBDH) problem.

Keywords: 자료 검색 및 저장, 암호화문서 검색, 데이터베이스 보안, 프라이버시 보호, 개인정보보호기술,

I. 서 론

서버에서 관리되는 사용자 정보의 양이 급격히 증가됨에 따라 시스템 내부 관리자들에 의한 정보의 악용, 남용 및 누출은 심각한 사회적 문제로 이슈화되고 있다. 최근 CSI/FBI 보고서에 의하면, 악의적인 내부 공격자에 의한 개인정보 침해 사건은 전체 사건의 59%를 차지하는 것으로 보고되었으며[15], 그 위험성은 전 세계적으로 아주 심각하다. 또한 국내에서도 옥션 해킹사고 및 하나로 텔레콤의 개인정보유출등과 관련된 사고로 인해서 방송, 금융, 통신, 의료 사회 전반에 개인정보유출방지에 대한 논의가 활발히 진행 중이다.

특히, 올해 초 방송통신위원회의 출범과 더불어 방송과 통신의 융합 원년을 맞아 개인정보 유출방지기술은 새로운 국면을 맞이했다. 기존의 개인정보보호가 시스템과 네트워크 위주의 개념이었다면, 방송 통신 융합

* 정회원, 평택대학교 정보통신학과
(Department of Information and Communication, Pyeongtaek University)

※ 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2006-352-D00179)
접수일자: 2008년12월10일, 수정완료일: 2009년1월12일

환경에서는 방송과 사이버 윤리 영역까지 모두 수용하는 통합적인 개인정보유출방지 기술이 필요하다. 더욱이 최근 정부 여당에 의해 인터넷 종합 대책이 발의된 가운데, 방송 통신 융합 환경에서 사이버 모독죄, 최진실 법, 악성 댓글 위법성 여부 판별 문제 등, 방송 통신 융합 환경에서 프라이버시 보호를 비롯한 개인정보유출방지 기술에 대한 중요성이 더욱 높아지고 있다. 방송 통신 융합 환경에서는 개인정보의 저장 및 활용 범위가 더욱 광범위 해지므로, 그에 따른 적용가능한 개인정보 유출방지 기술에 대한 심도 있는 논의가 필요하다.

외부/내부 공격자에 의한 개인정보의 남용 및 오용을 막을 수 있는 가장 간단하고 널리 활용될 수 있는 방법은 사용자의 개인 정보를 암호화 하여 데이터베이스에 저장하는 것이다. 암호화는 평문과 암호화키를 입력 받아서 타인이 해독할 수 없도록 하며, 오직 복호화 키를 소유한 자만 해당하는 평문을 복호화 할 수 있다. 만약, 사용자가 자신의 암호화키 및 복호화 키를 안전하게 저장/관리한다면, 내부 관리자는 암호화된 문서를 복호화 할 수 없게 되므로 문서의 암호화 관리는 실질적인 개인정보보호 방법이 된다. 하지만, 현재까지 암호화 하는 방법이 현실성이 없었던 가장 큰 이유는 암호화된 데이터베이스 상에서는 효율적인 자료 검색을 보장 받지 못했기 때문이다. 암호화된 문서상에서 검색의 효율성을 충분히 보장 받을 수 있다면, 이것은 개인정보유출방지를 위한 가장 실질적인 기술이 될 것이다.

암호화문서상에 효율적인 키워드 검색 문제를 풀기 위하여 최근 몇 년 동안 많은 프로토콜들이 다양한 환경에서 제안되어져 왔다. 본 논문에서는 공통의 키워드 검색(conjunctive keyword search)에 중점을 둔다. 공통의 키워드 검색이란, 키워드를 여러 개 던질 수 있으며, 서버는 이러한 여러 개의 키워드가 동시에 포함된 암호화된 문서를 결과 값으로 반환한다. 공통의 키워드 검색은 기존의 단일 키워드 검색 시스템을 이용하여 구현할 수도 있지만, 이러한 접근 방법은 굉장히 비효율적이다. 왜냐하면, 사용자는 각각 하나의 키워드를 질의해서 그에 해당하는 문서를 각각 받은 다음, 사용자는 또 다시 교집합 되는 부분을 추가적인 비교작업을 통해 추려내야 하며 이에 대한 오버헤드 및 비용 손실이 상당하기 때문이다. 그러므로 단일 키워드 시스템을 사용하여 공통의 키워드 시스템을 설계하는 방법은 현실적인 방안이 못되어 왔었다. 즉, 공통의 키워드 검색 프로토콜은 사용자가 단 한 번의 질의를 통해서 공통의 키워드를 포함하는 암호화 문서들을 한 번에 얻을 수 있는 프로토콜을 의미한다.

본 논문에서는 계산 및 저장량 측면에서, 암호화 문서상에서의 효율적인 공통의 키워드 검색 프로토콜을 정형화된 모델을 통해서 설계하고 이에 대한 안전성을 증명한다.

1. 관련 연구

암호화 문서상에서의 공통의 키워드 검색 시스템은 자료저장소인 데이터베이스 그리고 데이터베이스의 사용자로 구성된다. 먼저, 사용자는 문서를 암호화 하여 데이터베이스에 저장한다. 이후 사용자는 원하는 키워드가 숨겨진 트랩도어(trapdoor)를 데이터베이스에 질의하고 서버는 이를 검색하여 해당 키워드가 포함된 암호화된 문서를 사용자에게 내려주게 된다. 이때 서버는 사용자의 트랩도어 질의들 및 이에 대한 결과 값인, 암호화 문서들로부터 원래 문서에 대한 어떠한 정보도 알 수 없어야 한다. 이러한 공통의 키워드 검색 프로토콜은 자료 공급자의 관계에 따라 다음 두 가지 환경으로 분류할 수 있다. 다음의 x-y-z 기호는 x는 자료 공급자, y는 자료 저장소, z는 탐색을 원하는 자를 의미한다.

가. User-Database-User (UDU) 환경

자료 공급자가 자료 검색 자가 되는 환경을 의미한다. 예를 들어, 웹 기반 저장 시스템에서는 시스템 사용자 자신이 기밀문서를 직접 암호화해서 웹 기반 저장시스템에 있는 database에 저장하게 된다. 또한 필요 시에 자신이 직접 암호문들을 키워드를 이용하여 검색한다. 이러한 UDU 환경에서, Bloom 필터[2], 스트림 암호 및 블록 암호 등 대칭키 암호 기법들을 이용하여 많은 키워드 검색 프로토콜이 제안되었다[11, 14, 20, 23]. Song 등은 스트림 암호와 블록 암호를 이용하여 증명가능한 키워드 검색 프로토콜을 제안하였으며[23], Goh는 Bloom 필터를 이용하여 키워드 검색을 설계하였다[14]. 최근에 Chang과 Mitzenmacher에 의해 통신량과 저장량 측면에서 효율적인 스킴이 제안되어졌다. 하지만, 위의 프로토콜들은 완전한 공통의 키워드 검색을 제공하지 못한다[11].

공통의 키워드 검색을 해결하기 위해 Golle는 처음으로 대칭키 및 공개키 암호 기법을 키워드 검색에 적용하였다[12]. 하지만, 제안된 두 개의 프로토콜들 (Golle et al. I 과 Golle et al. II) 은 통신량과 저장량 측면에서 효율적이지 못하다. 즉, 저장되는 자료들의 수에 의

존하여 통신량과 저장량도 함께 선형적으로 증가하기 때문에 고용량 및 많은 자료들을 처리하는 데이터베이스 환경에서는 굉장히 비효율적인 기술이다. 또한 참고문헌 [12]의 두 번째 프로토콜은 표준적인 가정 (CDH, DBDH 등) 이 아닌 전혀 알려지지 않은 가정을 기반으로 하여 안전성이 증명되었으므로 그 안전성에 신빙성을 확보하기 어렵다. 사실, 안전성을 보장할 수 없는 프로토콜은 얼마든지 설계 가능하다. 하지만, 무엇보다 중요한 것은 설계된 프로토콜의 안전성이 어떠한 잘 알려진 계산적 가정에 기반하고 있는가 하는 점이다. 새로운 계산적 가정을 제시했으면, 제시한 가정이 기존의 잘 알려진 어려운 가정과 어떠한 관련이 있는가 하는 것이 (예: 포함관계에 대한 수학적 증명) 반드시 규명되어야 한다. 이러한 정밀한 규명 없는 새로운 계산적 가정의 사용은 암호학적 관점에서 굉장히 위험한 일이며, 이러한 가정을 기반으로 한 프로토콜 설계 역시 무의미한 작업이다. Golle et al. I 프로토콜은 Golle et al. II 프로토콜에 비해, 비록 랜덤 오라클 모델에서 설계되었지만, 그 안전성이 정확히 DBDH (Decisional Bilinear Diffie-Hellman) 문제로 귀결되므로 의미 있는 작업이다.

나. Sender-Database-User (SDU) 환경

자료 공급자와 자료 검색자가 다른 환경이다. 예를 들어, 이메일 시스템에서, 메일 송신자가 기밀문서를 수신자의 공개키로 암호화해서 보내면, 수신자는 암호화 문서에서 원하는 키워드를 검색하는 환경이다. 그러므로 UDU 환경과는 다르게 오직 공개키 암호알고리즘이 사용된다.

UDU 환경은 기밀문서를 암호화 할 때 대칭키 암호 방식을 사용하고, SDU 환경은 공개키 암호 방식을 사용하므로, 일반적으로, UDU 환경이 효율적이라 말할 수 있다. 하지만, 개체들 간의 환경 및 역할이 틀리므로 그 환경에 맞는 안전성 모델 및 프로토콜 설계가 필요하다. 사실, SDU 환경에서 프로토콜 설계가 UDU 환경에서 보다 더 어려워보이나, 아직까지 두 검색 환경에 대해 정확한 관계 규명이 이루어지지 않았다.

Boneh et al. 에 의해서 처음으로 SDU 환경에서 단일 키워드를 이용한 프로토콜이 제안되어졌다. 하지만, 제안된 프로토콜은 공통의 키워드 검색 기능을 제공하지 않을뿐더러 그 안전성 모델이 랜덤 오라클에 기반하였다. 최근에 Boneh와 Waters에 의해서 공통의 키워드 검색과 비교 연산이 동시에 가능한 공통의 키워드 검색 프로토콜인 HVE 프로토콜이 제안되어졌다[6]. 하지만, 그 효율성이 현실에 사용되어지기에 많은 무리가 있다. 표 1에 기존 프로토콜과의 비교를 나타내었다.

표 1. 기존 공통의 키워드 검색 프로토콜 비교
Table 1. Comparison of the existing CKS protocols.

$|q|$: 큰 소수 q의 크기, m : 데이터베이스의 필드 (column)의 개수, n : 데이터베이스의 레코드 (row) 갯수, TSC : 사용자와 데이터베이스간의 통신 량, TSD : 데이터베이스의 전체 크기, SNPV : 검색 당 필요한 페어링 연산 수, NPS : 자료 공급자로부터 요구되어지는 페어링 연산 수, NS : 사용자의 비밀 키 개수, SM : 안전성 기반 모델 (RO : 랜덤 오라클 모델, ST : 표준적인 가정 모델), NST : 비표준적인 가정, CON : 공통의 키워드검색 가능 여부 (Y : 가능함, N : 불가능함)

Protocol	TSC	TSD	SNPV	NPS	NS	SM	CON
Golle et al I	$(n+1)\|q\|+\log m$	$n(m+1)\|q\|$	0	0	$n+2$	RO	Y
Golle et al II	$3\|q\|+\log m$	$n(2m+1)\|q\|$	3	0	2	NST	Y
HVE	$3m+\log m$	$n(2m+2)\|q\|$	$2m+1$	0	$3m+1$	ST	Y
ECKS	$4\|q\|+\log m$	$n(2m+2)\|q\|$	4	0	3	ST	Y

1. 논문의 공헌도

본 논문에서는 UDU 환경에서, 효율적인 공통의 키워드 검색 프로토콜, ECKS를 제안한다. 표 1에서 나타내었듯이, 제안된 프로토콜은 처음으로, TSD를 제외한 모든 비용 평가 항목에서 상수양의 비용을 요구한다.비록, TSC가 필드의 개수에 의존적이지만, 데이터베이스 필드의 개수는 이미 정해진 고정 값이므로 상수 값으로 간주 될 수 있다. 또한 본 논문에서는 제안된 프로토콜의 안전성을 표준적인 모델 (standard model) 에서 DBDH 문제로 귀결 시켜 증명하였다. 효율성 측면에서, 기존의 Golle et al. I 프로토콜보다 통신량과 저장량 측면에서 우수하다. Golle et al. II 프로토콜은 안전성이 알려지지 않은 문제에 귀결되므로 기존에 제안된 프로토콜들과의 비교가 합리적이지 않다. 비록, 제안된 환경은 틀리지만, 효율성 측면에서 HVE 프로토콜과의 비교도 흥미롭다. 즉, HVE 프로토콜은 검색 당 사용되어지는 페어링의 연산수 및 프로토콜의 통신량이 데이터베이스의 필드 개수에 비례하여 증가한다. 그러므로 데이터베이스 검색 시 요구되는 계산 량 및 탐색 시간이 현실적으로 사용되어지기에 비효율적이다. 결론적으로,

지금까지 알려진 공통의 키워드 검색 프로토콜 중에서, ECKS 프로토콜은 상수 량의 비용을 요구하며 랜덤 오라클 없이 증명된 프로토콜이다.

Ⅱ. 안전성 모델

1. UDU 환경의 정의

본 논문에서는 n개의 레코드와 m개의 필드로 구성된 실제적인 데이터베이스(예: Oracle 8.0)를 고려한다. 예를 들어, 마약 정보와 관련된 민감한 데이터베이스를 가정하면, 그림 2와 같이, "Name", "Sex", "Race", "Aid", "Fines", "Drugs"과 같은 키워드 필드를 고려할 수 있다. 하나의 레코드에는 암호화된 문서와 m개의 키워드 필드가 존재한다. 즉, 각각의 레코드를 R_i라 정의했을 때, R_i에는 i번 째 문서 $D_i=(W_i[1],...,W_i[m])$를 포함한다. 단, $W_i[j], (1 \le j \le m)$는 문서 D_i의 j번째 키워드 값을 의미한다. 그림 1에서 n=5, m=6으로 가정했을 때의 데이터베이스 스키마의 한 예를 나타내었다. 가령, 레코드 R_1는 $D_1=(Alice, F, C, 0., 45., 0)$ 로 이루어진다.

사용자는 자신의 개인키 및 공개키를 이용하여 암호화된 문서와 탐색정보 (CSI : conjunctive searchable information) 를 만든 다음, 데이터베이스에 업로드 한다. 그 다음, 사용자는 자신의 키워드들을 블라인드 (blind) 시킨 공통의 키워드에 대한 트랩도어 TCK (trapdoor for conjunctive keyword)를 만들어 데이터베이스에게 질의하게 된다. 단, TCK는 오직 개인키를 아는 사용자만이 만들 수 있다. 서버는 받은 TCK와 탐색정보들을 이용하여 데이터베이스에 저장된 암호화된 문서를 각각 테스트하여 합당한 암호화된 문서를 사용자에게 돌려준다.

Name	Sex	Race	Aid	Fines	Drugs
Alice	F	C	0.	45.	0
Bob	M	C	50.	0.	2
Chin	F	A	20.	20.	1
Dewitt	M	B	30.	35.	1
Eva	F	B	0.	0.	0

그림 1. 데이터베이스 스키마 예제
Fig. 1. An example of schemea for database.

가. 구성 알고리즘

공통의 키워드 검색 프로토콜은 크게 4가지 알고리즘들로 구성된다.

키 생성 알고리즘 KeyGen(1^k) : 키 생성 알고리즘은 암호화 문서상에서 탐색을 가능하게 하는 탐색 정보 및 트랩도어 정보를 생성하기 위해 사용자들의 키를 출력하는 알고리즘이다. 생성된 키를 가지고 있는 사용자만이 탐색 정보 및 트랩도어를 이용해 검색을 수행할 수 있다. 입력으로 안전성 파라미터 k를 받고, 공개키/개인키 쌍인 (pk, sk)를 생성한다. 이 키들은 탐색정보, CSI를 만드는데 사용된다.

- 탐색 정보 생성 알고리즘 CSI(pk, sk, D_i) : 탐색 정보 생성 알고리즘은 암호화 문서상에서 탐색을 가능하게 해주는 알고리즘이다. 암호화 문서상에서 검색은 사실 불가능하다. 하지만, 탐색 정보 생성 알고리즘을 통해 비록 암호문이지만 검색을 가능하게 만들어 준다. 개인키 sk와 공개키 pk 및 m개의 키워드가 포함된 문서 $D_i=(W_i[1],...,W_i[m])$를 입력으로 받아서 $CSI_i=(I_i, CSI_{i,1}(W_i[1]),...,CSI_{i,m}(W_i[m]))$를 출력한다. 단, I_i는 검색에 필요한 보조 정보들을 의미하며, $CSI_{i,j}(W_i[j])$는 해당 키워드 $W_i[j]$의 탐색 정보이다. 단, $1 \le j \le m$이다.
- 트랩도어 생성 알고리즘 TCK(pk, sk, $p_1,...,p_l$, Q_l) : 트랩도어는 비밀 값을 의미한다. 즉, 탐색정보가 만들어 졌어도 트랩도어 값을 통해 탐색을 수행할 수 가 있다. 즉, 키 생성 알고리즘을 통해 만들어진 비밀 값과 해당 키워드를 통해, 트랩도어 값을 생성하고 이 값을 소유한 사용자만이 해당 키워드에 대한 검색을 수행할 수 있다. 이 알고리즘은 개인키와 공개키 및 목적 키워드 필드의 리스트 $p_1,...,p_l$* 그리고 l개의 키워드인 Q_l을 입력으로 받아서 트랩도어 T_l을 출력한다. Q_l은 사용자가 검색하기 원하는 키워드들의 집합으로써, 데이터베이스에 존재하는 키워드들의 집합인 D_i와는 구별된다. 즉, $Q_l=(W[p_1],...,W[p_l])$이다.
- 테스트 알고리즘 Test(CSI_i, T_l) : 해당 키워드에 대해서 검색을 수행했을 때 암호화 문서상에서 해당 키워드가 있는지 없는지를 판별해주는 역할을 수행한다. 입력으로 $CSI_i=(I_i, CSI_{i,1}(W_i[1]),...,CSI_{i,m}(W_i[m]))$와 트랩도어를 받은 후에, 트랩도어 T_l에 대해서 만약 테스트 식이 $(W_i[p_1]=W[p_1]) \wedge ... \wedge (W[p_1]=W[p_l])$를 만족하면, Yes를 출력하고 아니면, No를 출력한다.

평문의 데이터베이스는 위의 키 생성 알고리즘 및 탐

* 주소, 이름, 성명 등과 같은 데이터베이스의 필드 명들을 의미한다.

색 정보 생성 알고리즘에 의해서 실제적으로 암호화 된다. 예를 들어, 그림 2에 나타낸 것처럼, 암호화된 데이터베이스에서의 열 R_i는 $D_i=(W_i[1],...,W_i[m])$ 에 대한 암호문 $E_S(R_i)$와 그에 해당하는 공통의 탐색 정보 $CSI_i=(I_i, CSI_{i,1}(W_i[1]),...,CSI_{i,m}(W_i[m]))$로 구성된다. 단, S는 대칭키 암호 알고리즘, E에서 사용되는 개인키이고, $CSI_{i,1}$는 해당 키워드 $W_i[j]$의 탐색 정보이다. 암호화는 문서를 랜덤화 시키기 때문에 서버는 암호화된 문서들만 이용하여 사용자의 키워드 질의에 대응하는 결과 값을 반환할 수 없다. 서버는 암호화된 문서 옆에 저장되어 있는 탐색 정보를 이용하여 검색을 수행하며, 사용자의 키워드 질의에 부합하는 암호화 문서들에 대한 답변을 한다. 그러므로 서버는 사용자의 키워드 질의 값 (TCK) 과 탐색정보들을 이용하더라도 암호화된 문서내의 키워드들에 대한 내용을 모르게 설계되어야 한다. $D_i=(W_i[1],...,W_i[m])$를 암호화 할 때에 안전한 대칭키 암호화 알고리즘이 사용되어지므로, 암호화를 통한 정보 노출은 고려하지 않아도 된다. 문제는 공격자가 사용자의 공통의 트랩도어 질의 값, TCK와 탐색정보 $CSI_i=(I_i, CSI_{i,1}(W_i[1]),...,CSI_{i,m}(W_i[m]))$를 통해서 문서 D_i에 대한 어떠한 정보도 노출되지 않도록 프로토콜이 설계되어져야 한다는 점이다. 안전성 정의를 공격자가 TCK 질의를 할 수 있는 상황에서, 주어진 CSI_i를 *구별할 수 없다* (indistinguishability) 로 정의를 내릴 것이며, 이에 대해 다음 절에서 상세히 설명되어질 것이다.

사실, Golle 기타 등이 참고문헌 [12]에서 처음으로 구별 불가능성 개념을 이용하여 UDU 환경에서 안전성 모델을 정의하였다. 본 논문에서 제안되는 안전성 모델의 기본 아이디어는 비록, 참고문헌 [12]의 안전성 모델을 기반으로 하고 있으나, 그 적용 모델이 더욱 구체적이다. 즉, 실제적인 레코드와 필드를 갖는 데이터베이스를 가정하였으며, 표기를 더욱 구체적으로 사용하여 정의하였다.

ED	AD	Name	Sex	Race	Aid	Fines	Drugs
$E_S(D_1)$	I_1	$CSI_{1,1}$(Alice)	$CSI_{1,2}$(F)	$CSI_{1,3}$(C)	$CSI_{1,4}$(0.)	$CSI_{1,5}$(45.)	$CSI_{1,6}$(0)
$E_S(D_2)$	I_2	$CSI_{2,1}$(Bob)	$CSI_{2,2}$(M)	$CSI_{2,3}$(C)	$CSI_{2,4}$(50.)	$CSI_{2,5}$(0.)	$CSI_{2,6}$(2)
$E_S(D_3)$	I_3	$CSI_{3,1}$(Chin)	$CSI_{3,2}$(F)	$CSI_{3,3}$(A)	$CSI_{3,4}$(20.)	$CSI_{3,5}$(20.)	$CSI_{3,6}$(1)
$E_S(D_4)$	I_4	$CSI_{4,1}$(Dewitt)	$CSI_{4,2}$(M)	$CSI_{4,3}$(B)	$CSI_{4,4}$(30.)	$CSI_{4,5}$(35.)	$CSI_{4,6}$(1)
$E_S(D_5)$	I_5	$CSI_{5,1}$(Eva)	$CSI_{5,2}$(F)	$CSI_{5,3}$(B)	$CSI_{5,4}$(0.)	$CSI_{5,5}$(0.)	$CSI_{5,6}$(0)

그림 2. 암호화된 데이터베이스 예제
Fig. 2. An example of encrypted database.

* 단, ED (Encrypted Data) : 암호화된 문서가 저장되는 필드를 의미한다. AD : 보조 탐색 정보를 의미한다.

1. UDU 환경의 안전성 정의

앞서, UDU 환경에 대해서 정의하였다. 이를 바탕으로 하여 UDU 환경의 안전성을 정의하려 한다. 안전성 정의를 위해서는 공격자에 대한 정의가 필요하다. 정의된 공격자에 대해서 안전성의 정의는 게임을 통해서 이루어진다. 게임을 통해, 공격자는 실제적인 문서와 랜덤한 문서를 구분하게 되며, 이러한 구분 이점에 대해서 작은 확률로 분석되어진다면, UDU 환경에 안전하다고 말 할 수 있다. 이를 구체적으로 정의하면 다음과 같다.

정의 1. [공통의 키워드 검색 프로토콜의 안전성] CKS={KeyGen(1^k), CSI(pk, sk, D_i), TCK(pk, sk, $p_1,...,p_l$, Q_l), Test(CSI_i, T_l)}를 공통의 키워드 검색 프로토콜이라 하고 A를 CKS에 대응하는 공격자라 정의하자. 안전성을 정의하기 위해 다음의 게임을 고려한다. A는 게임동안 트랩도어 오라클 $O_T(.)$ 및 탐색정보 (CSI)에 대한 오라클 $O_C(.)$ 질의를 각각 q_T, q_C 만큼 다항식 시간 내에 각각 질의할 수 있다. 즉, A는 원하는 키워드들을 선택해서 그에 해당하는 트랩도어 값인 TCK값 및 CSI 값을 얻을 수 있다. 공격자는 두 개의 문서 D_0^*와 D_1^*을 목적 문서로 출력한다. 이는 자신이 구분할 문서를 임의로 뽑아서, 이에 대해서 구분하기 위함이다. 게임은 랜덤한 비트 값 b를 뽑아서, CSI_b^*값을 A에게 준다. 공격자는 CSI_b^*값을 정확히 추측하게 되면 게임에서 이기게 된다. 단, 공격자가 트랩도어 및 탐색정보 오라클 $O_T(.)$ Oc(.) 들에게 키워드를 질의할 때 쉽게 CSI_b^* 값을 구별할 수 있는 키워드에 대해서는 질의 할 수 없다. 예를 들어, D_0^*와 D_1^*을 쉽게 구별 할 수 있는 키워드들에 대해서 트랩도어 오라클 및 탐색 오라클이 허용된다면 Test 식을 이용하여 쉽게 CSI_b^* 값을 구별할 수 가 있다. 구체적으로 다음과 같은 게임이 정의된다.

Game $\mathrm{Exp}_A^{cta}(k)$

1. $(pk, sk) \xleftarrow{R} KeyGen(1^k)$
2. $\left(D_0=(W_0^*[1],...,W_0^*[m]),\ D_1=(W_1^*[1],...,W_1^*[m])\right) \leftarrow A^{O_T(.)\,O_C(.)}$
3. $b \xleftarrow{R} [0,1]$
4. $d \leftarrow A^{O_T(.)\,O_C(.)}(CSI_b)\ s.t,\ CSI_b^*=(CSI_{b,1}^*(W_b^*[1]),...,CSI_{b,m}^*(W_b^*[m]))$
5. *If* $b=d$ *then return* 1. *Otherwise, return* 0.

게임에 의해 b가 (0 혹은 1) 선택되어지며, 선택된 b 값을 공격자는 위에 정의된 게임의 절차에 따라 추측한

다. 이러한 *ss-cta-udu* 이점(advantage)을 다음과 같이 정의한다.

$$Adv_A^{cta}(k,q_T,q_C,T_D)=\left|\begin{array}{l}\Pr\left[\mathrm{Exp}_A^{cta}(k)=1|b=1\right]\\ -\Pr\left[\mathrm{Exp}_A^{cta}(k)=1|b=0\right]\end{array}\right|$$

위의 이점이 무시할 수 있는 함수로 표현되어질 때, 주어진 CKS 프로토콜은 CTA (chosen trapdoor attack)에 안전 (semantic secure) 하다고 정의한다.

Ⅲ. Bilinear Map 과 계산적 가정

정의 2. [Admissible Bilinear Map] G_1과 G_2를 큰 소수 q에 대한 위수가 q인 두 그룹으로 정의하자. 다음의 세 가지 조건들을 만족할 때 $e: G_1 \times G_1 \rightarrow G_2$ 를 admissible bilinear map이라 정의한다. (1) Bilinear : $e(u^a,v^b)=e(u,v)^{ab}$ (2) Non-degenerate : 만약 g가 G_1의 생성자이면, $e(g,g)$는 G2의 생성자이다. (3) Computable : 모든 $u,v \in G_1$에 대해 $e(u,v)$는 효율적으로 계산된다.

정의 3. [DBDH 문제] DBDH(Decisional Bilinear Diffie-Hellman) 문제는 두 개의 형태로 정의되어 왔다. 하나는 구분하는 마지막 원소가 $e(g,g)^{abc}$와 $e(g,g)^d$ 이고, 다른 하나는 구분하는 마지막 원소가 g^{abc}, g^d형태이다[12]. 본 논문에서는 후자의 형태의 DBDH 문제를 따른다. Δ_D를 다항식 시간 T_D 이내에 DBDH 두 리스트를 구분하는 공격자를 가정하고, 다음의 $\mathrm{DBDH}_{\mathrm{real}}$ 과 $\mathrm{DBDH}_{\mathrm{rand}}$를 각각 포함하는 두 실험들을 고려해 보자.

Experiment $\mathrm{Exp}_{\Delta_D}^{real}(k)$	**Experiment** $\mathrm{Exp}_{\Delta_D}^{rand}(k)$
$x \leftarrow G_1; X \leftarrow g^x, y \leftarrow G_1; Y \leftarrow$	$x \leftarrow G_1; X \leftarrow g^x, y \leftarrow G_1; Y \leftarrow g^y$
$z \leftarrow G_1; Z \leftarrow g^z, R \leftarrow g^{xyz}$	$z,r \leftarrow G_1; Z \leftarrow g^z, R \leftarrow g^r$
$b \leftarrow \Delta_D(DBDH_{real} = (X,Y,Z,R))$	$b \leftarrow \Delta_D(DBDH_{rand} = (X,Y,Z,R))$
return b	return b

DBDH에 대한 공격자 Δ_D의 이점(advantage)을 다음과 같이 정의한다.

$$Adv_{\Delta_D}^{dbdh}(k,T_D)=\left|\begin{array}{l}\Pr\left[\mathrm{Exp}_{\Delta_D}^{real}(k)=1|b=1\right]\\ -\Pr\left[\mathrm{Exp}_{\Delta_D}^{rand}(k)=1|b=0\right]\end{array}\right|$$

정의 4. [DBDH 가정] 만약 DBDH 문제를 푸는 이점이 무시할 수 있는 함수로 표현되어 질 때 DBDH 가정이 G_1에서 만족된다고 정의한다.

정의 5. [MDBDH 문제] MDBDH(Multi-Decisional Bilinear Diffie-Hellman) 문제는 DBDH의 문제에서 입력 값을 m개로 확장한 문제이다. Δ_M를 다항식 시간 T_M 이내에 MDBDH 두 리스트를 구분하는 공격자로 가정하고, 다음의 두 실험들을 고려해 보자.

Experiment $\mathrm{Exp}_{\Delta_M}^{real}(k)$	**Experiment** $\mathrm{Exp}_{\Delta_M}^{real}(k)$
$u \leftarrow G_1; X \leftarrow g^u, v_0,..,v_m \leftarrow G_1;$	$u \leftarrow G_1; X \leftarrow g^u, v_0,..,v_m \leftarrow G_1$
$Y_0 \leftarrow g^{v_0},.., Y_m \leftarrow g^{v_m}$	$Y_0 \leftarrow g^{v_1},.., Y_m \leftarrow g^{v_m}$
$z \leftarrow G_1; Z \leftarrow g^z,$	$z,r_0,..,r_m \leftarrow G_1;$
$R_0 \leftarrow g^{uv_0z},..,R_m \leftarrow g^{uv_mz}$	$Z \leftarrow g^z, R_0 \leftarrow g^{r_0},..,R_m \leftarrow g^{r_m}$
$b \leftarrow \Delta_M(X,Y_0,..,Y_m, Z,R_0,..,R_m)$	$b \leftarrow \Delta_M(X,Y_0,..,Y_m, Z,R_0,..,R_m)$
return b	return b

MDBDH에 대한 공격자 Δ_M의 이점(advantage)을 다음과 같이 정의한다.

$$Adv_{\Delta_M}^{mdbdh}(T_M,q_t)=\left|\begin{array}{l}\Pr\left[\mathrm{Exp}_{\Delta_M}^{real}(k)=1|b=1\right]\\ -\Pr\left[\mathrm{Exp}_{\Delta_M}^{rand}(k)=1|b=0\right]\end{array}\right|$$

정의 6. [MDBDH 가정] : 만약 MDBDH 문제를 푸는 이점이 무시할 수 있는 함수로 표현되어 질 때 MDBDH 가정이 G_1에서 만족된다고 정의한다.

참고문헌 [5]에서 DBDH 및 MDBDH는 동일한 어려운 문제임을 보였다. 다음의 보조정리의 자세한 증명은 참고문헌 [5]를 참조 한다.

보조정리 2.1 위의 DBDH와 MDBDH의 입력 파라미터에 대해서 다음의 두 식이 성립한다.

(1) $Adv_{\Delta_M}^{mdbdh}(T_M,k) \le (m-1)Adv_{\Delta_D}^{dbdh}(T_D,k)$
(2) $Adv_{\Delta_D}^{dbdh}(T_D,k) \le (m-1)Adv_{\Delta_M}^{mdbdh}(T_M+4mT_{G1},k)$

단, T_{G1}은 G_1에서 지수 승에 대한 계산시간이다.

정의 7. [Pseudorandom function] 슈도 랜덤 함수 패밀리는 함수 $F_K: K(F) \times D(F) \rightarrow R(F)$의 모음(collection)이다. K(F)는 FK의 키들의 집합이고 D(F)는 FK의 도메인에 대한 집합이다. 슈도 랜덤 함수 FK는 다항식 시간 알고리즘 B에 대해서 다음을 만족해야 한다.

· 주어진 x ∈ D(F) 와 K ∈ K(F)에 대해서, F(x)를 계산하는 다항식 알고리즘이 존재해야 한다.
· 어떠한 다항식 알고리즘 B에 대해서 다음의 prf 이점에 대한 확률이 무시할 수 있을 만큼 작아야 한다.

$$\left|\Pr\left[B^{F_K}=1: K\xleftarrow{R}K(F)\right]-\Pr\left[B^{F}=1: F\xleftarrow{R}U_{F_K}\right]\right|\le \epsilon_{prf}$$

단, U_{F_K}는 전체 가능한 $F_K: K(F)\times D(F)\rightarrow R(F)$ 함수의 총 집합이다.

Ⅳ. 효율적인 키워드 검색 프로토콜 설계

1. ECKS 프로토콜 구성

본 장에서 DBDH에 기반을 둔 효율적인 ECKS 프로토콜을 제안한다. 제안된 프로토콜은 상수 비용의 계산량과 저장량을 요구하며 랜덤 오라클의 가정 없이 증명된 프로토콜이다. 키워드들을 순환 그룹으로 매핑 시키기 위해 슈도 랜덤 함수 $F_K: \{0,1\}^{l_k}\times\{0,1\}^{l_d}\rightarrow\{0,1\}^{l_r}$가 사용된다. 단, l_k는 키 비트이고, l_d는 키워드 비트이며, l_r는 그룹 G_1, G_2의 전체 비트 크기를 의미한다.

· 키 생성 알고리즘 KeyGen(1^k) : 입력으로 안전성 파라미터 k를 받고, 두 개의 그룹 G_1,과 G_2를 결정한다. 공개키/개인키, $sk=(\alpha,\beta,\theta)$, $pk=(y_1=g^{\alpha},y_2=g^{\beta})$를 생성한다.
· 탐색 정보 생성 알고리즘 CSI(pk, sk, D_i) : 입력으로 sk, pk와 문서 $D_i=(W_i[1],...,W_i[m])$를 받는다. 해당 문서에 대해서 $m+1$개의 랜덤 값, $v_i,s_{i,1},...,s_{i,m}$를 선택한 후 다음을 계산한다.

$$CSI_i=\begin{cases} I_i=g^{\theta v_i}, I_{i,0}=g^{\alpha\beta v_i}, I_{i,1}=g^{\alpha\beta s_{i,1}},...,I_{i,m}=g^{\alpha\beta s_{i,m}} \\ CSI_{i,1}(W_i[1])=g^{\alpha\beta\theta v_i F_K(W_i[1])}\times g^{\alpha\beta\theta s_{i,1}},... \\ CSI_{i,m}(W_i[m])=g^{\alpha\beta\theta v_i F_K(W_i[m])}\times g^{\alpha\beta\theta s_{i,m}} \end{cases}$$

· 트랩도어 생성 알고리즘 TCK(pk, sk, $p_1,...,p_l$, Q_l) : 비밀 키와 개인키 및 목적 키워드 필드의 리스트 $p_1,...,p_l$과 1개 여러 키워드 $Q_l=\{W_i[p_1],...,W_i[p_l]\}$를 입력 값들로 받아서 트랩도어 $T_l=[A,B,C,D,p_1,...,p_l]$를 출력한다.단, $A=g^r, B=g^{\alpha\beta r(\sum_{t=1}^{l}F_K(W_i[p_t]))}\times g^{\alpha\beta t}, C=g^{r\theta}, D=g^{t\theta}$ 이다.
· 테스트 알고리즘 Test(CSI_i, T_l) : 주어진 CSI_i, T_l를 이용하여, 모든 $1\le i\le n$에 대하여 다음 식을 검증한다.

$$\frac{e(I_i,B)\times e(C,\prod_{t=1}^{l}I_{i,p_t})}{e(A,\prod_{t=1}^{l}CSI_{i,p_t}(W_i[p_t]))}=e(D,I_{i,0})$$

식이 맞으면, Yes를 출력하고 아니면 No를 출력한다.

2. ECKS 프로토콜 효율성 분석

먼저, 사용자의 저장량 분석을 위해, 키 생성 알고리즘을 분석해 보자. 위의 알고리즘에 의해, 총 3개의 개인키와 2개의 공개키를 요구한다. 개인키만 안전하게 저장하면 되므로, 3개의 키를 저장할 공간이 필요하다. 사용자의 계산량 분석을 위해, 탐색 정보 생성 알고리즘, 트랩도어 생성 알고리즘을 분석해보자. 알고리즘에서 보듯이, 탐색 정보 생성 알고리즘은 페어링 연산을 요구하지 않는다. 페어링 기반의 암호화 프로토콜을 설계할 때에 페어링 연산을 줄이는 것은 대단히 의미 있는 작업이다. 더욱이, 트랩도어 생성 알고리즘에도 페어링 연산을 요구하지 않는다. 하지만, 사용자가 트랩도어 값을 서버에게 전달할 때에 $4|q|+logm$ 정도의 통신량이 필요하다. m값은 데이터베이스가 설계될 때에 고정되어진 값이므로 전체 통신량은 비록 m에 의존적이지만, 그 값은 상수양이라 할 수 있다.

데이터 저장소에 의해 테스트 작업이 수행되어 질 때에는 검색 당 총 4번의 페어링 연산이 필요하다. 데이터 저장소의 계산 부담을 줄여주는 것이 중요하므로, 테스트 작업 시에 페어링 수를 줄이는 것이 무엇보다 의미 있는 작업이다. 현재까지 제안된 프로토콜 중에는 페어링을 요구하지 않는 프로토콜은 아직 제안되지 않았으며, 향후에 반드시 연구되어져야 할 부분이다.

제안된 프로토콜은 데이터베이스의 크기를 제외하고는 모두 상수양의 계산량과 저장량을 필요로 한다. 엄밀히 말하면, 사용자의 계산량과 저장량 측면에서 상수량을 요구하고 데이터 저장소의 계산량 및 통신량 측면에서 상수양의 비용을 요구한다. 대부분의 프로토콜에서 비용측면에서 효율적이면, 안전성 증명이 랜덤 오라클 없이 이루어지기가 쉽지 않다. 즉, 비용이 좋으면, 안전성은 좋지 않게 설계되어진다. 하지만, 제안된 프로토콜은 효율적이면서 안전성 측면에서도 랜덤 오라클 없이 증명됨을 다음 절에 보인다. 제안된 프로토콜의 분석은 논문 앞 부부의 표 1에 나타내었다.

3. ECKS 프로토콜 안전성 증명

정리 3.1 A를 ECKS 프로토콜의 안전성을 깨려하는 PPT 공격자라 가정하자. A는 트랩도어 오라클 $O_T(.)$ 및 탐색정보 (CSI)에 대한 오라클 질의를 q_T, q_C 만큼 다항식 시간 T_D내에 각각 질의할 수 있다. MDBDH 문제가 어렵다고 가정 했을 때 제안된 ECKS 프로토콜은 A의 CTA에 안전하다. A의 프로토콜에 대한 공격 이점은 다음과 같이 수렴된다.

$$Adv_A^{cta}(k,q_T,q_C,T_M) \le 2Adv_{\Delta_M}^{mdbdh}(k,T_M)+\frac{2}{|q|}$$

단, $T_M > T+((m+4)q_T+(3m+2)q_C+3m+2)T_G$이고, T_G는 그룹 G에서 요구되는 지수 승 시간이다.

<증명> 안전성 증명은 대우명제를 사용하여 한다. 즉, MDBDH 문제가 풀기 어렵다면, 주어진 프로토콜이 안전하다를 증명하기 위해서는 대우명제를 이용하여 제안된 프로토콜의 안전성을 깨려는 공격자가 있다면, 그러한 공격자를 이용하여 주어진 가정인 MDBDH 문제를 깰 수 있음을 보이면 된다. 그러므로 공격자가 MDBDH 문제를 풀지 못하는 한 제안된 프로토콜은 안전하다. 먼저, MDBDH 문제를 깨는 알고리즘 Δ_M을 다음과 같이 구성하여 MDBDH문제를 깨는 확률을 구하려 한다. Δ_M는 입력으로 다음 값을 받는다.

$$MDBDH=(X=g^x, Y_0=g^{y_0},..., Y_{m+2}=g^{y_{m+2}}, Z=g^z, R_0,.., R_{m+2})$$

공격자가 CSI 및 TCK 질의를 할 때에 참 랜덤 함수 (truly random function)가 사용되어진다. 참 랜덤 함수를 시뮬레이트 (simulate) 하기 위해, 참고문헌 [21]에서 했던 것처럼, FK의 입출력 값을 유지하는 테이블, TB를 정의한다. 즉, A가 TB에 이미 존재하는 (W, r)에 대해서 TCK 및 CSI 질의를 요청할 경우, r을 랜덤하게 Z_q^*상에서 랜덤하게 뽑아서, 그 값을 이용하여 TCK 및 CSI 값을 계산한다. 만약 TB에 없는 키워드 W′ 일 경우는 새롭게 r'을 Z_q^*상에서 선택해서 새로운 리스트 (W, r') 를 TB에 기록한다. 그 후 TCK 및 CSI 답변이 있을 경우, TB에 있는 리스트를 이용하여 동일한 방법으로 답변을 계산한다. 구체적으로, 공격자 A와는 다음과 같은 방법으로 질의에 응답한다.

- 키 생성 알고리즘 KeyGen(1^k) : Δ_M는 공격자 A의 공개키 및 개인키를 Z_q^* 상에서 다음과 같이 정의한다.

$$pk=(g,y_1=X,y_2=Z),\ sk=(\cdot,\cdot,\theta)$$

단 표기 "·" 는 널 (null) 을 의미하는 것으로서 각각의 비밀 키 값들을 모르는 상태에서 공격자의 여러 질의에 답변하게 된다.

- CSI 질의에 대한 답변 : 공격자 A가 $D_i=(W_i[1],..., W_i[m])$에 대해서 CSI 질의를 할 때에 Δ_M는 먼저 테이블 TB를 통해서 각각의 $F_K(W_i[j])=x_{i,j}$값을 얻는다 $(1 \le j \le m)$. CSI에 대해 다음과 같은 방법으로 답변한다.

$$CSI_i=\begin{cases} I_i=(Y_0)^\theta, I_{i,0}=R_0, I_{i,1}=R_1,..., I_{i,m}=R_m \\ CSI_{i,1}(W_i[1])=(R_0)^{\theta x_{i,1}}\times(R_1)^\theta,..., \\ CSI_{i,m}(W_i[m])=(R_0)^{\theta x_{i,m}}\times(R_m)^\theta \end{cases}$$

- TCK 질의에 대한 답변 : $Q_l=(W[p_1],..., W[p_l])$에 대해서 공격자가 TCK 질의를 할 때에 Δ_M는 다음과 같은 방법으로 질의에 답변한다. 먼저, Δ_M는 $Q_l=(W[p_1],..., W[p_l])$에 대한 값, $F_K(W_i[j])=x_{i,j}$를 TB를 통해서 얻는다 $(p_1 \le j \le p_l)$. 랜덤 값 k_1, k_2를 선택해서 A, B, C, D를 다음과 같이 계산하여 반환한다.

$$A=Y_{m+1}, B=(R_{m+1})^{(x_{i,p_1}+...+x_{i,p_l})}\times R_{m+2},$$
$$C=(Y_{m+1})^\theta, D=(Y_{m+2})^\theta$$

위와 같은 방법으로 형성된 트랩도어는 정당한 트랩도어임을 Test를 통한 일관성 체크를 통해 쉽게 알 수 있다. 즉, 동일한 키워드에 대해서 CSI 답변과 트랩도어 답변을 형성한 다음, Test 식을 이용해서 Yes가 반드시 출력이 되는지 확인을 통해서, Δ_M 의해서 만들어진 CSI 및 TCK 질의가 올바름을 확인한다. 이에 대한 증명은 자명하므로 지면 관계상 생략한다.

- Challenge에 대한 답변 : $D_i=(W_i[1],..., W_i[m])$에 대해서 Δ_M는 먼저 테이블 TB를 통해서 각각의 $F_K(W_i[j])=x_{i,j}$값을 얻는다 $(1 \le j \le m)$. 그리고 랜덤 값들을 $\delta_{i,0}, \delta_{i,1},..., \delta_{i,m}$ 선택한 후에 동일한 방법으로 답변을 한다.

$$CSI_b^*=\begin{cases} I_i^*=(Y_0)^\theta, I_{i,0}^*=R_0, I_{i,1}^*=R_1,..., I_{i,m}^*=R_m \\ CSI_{b,1}^*(W_b^*[1])=(R_0)^{\theta x_{i,1}}\times(R_1)^\theta,..., \\ CSI_{b,m}^*(W_b^*[m])=(R_0)^{\theta x_{i,m}}\times(R_m)^\theta \end{cases}$$

- 출력 : 공격자가 d 값을 출력으로 내면 그 값을 이용한다. 즉, 만약 $d=b$ 이면 Δ_M는 1을 출력한다. 다른 경우는 0을 출력한다.

입력 받은 튜플이 MDBDH 튜플인 경우를 고려해 보

자. 공격자 A가 정확히 비트 d를 추측하면 (d=b), Δ_M는 1을 출력으로 내게 된다. 즉, MDBDH 튜플인 경우에 Δ_M가 1을 출력할 확률은 공격자 A에 대한 프로토콜 공격 이점에 의존하게 된다.

$$\Pr\left[\mathrm{Exp}^{real}_{\Delta_M}(k)=1\right]=\frac{1}{2}\Pr\left[\mathrm{Exp}^{cta}_{A}(k)=1|b=1\right]$$
$$+\frac{1}{2}(1-\Pr\left[\mathrm{Exp}^{cta}_{A}(k)=1|b=0\right])$$
$$\geq \frac{1}{2}+\frac{1}{2}Adv^{cta}_{A}(k,q_T,q_C,T_D)$$

입력 받은 튜플이 랜덤한 튜플임을 가정해 보자. 이러한 경우는 랜덤 튜플인 경우에도 A가 d=b임을 맞추는 경우이다. 이 경우는 우연찮게 랜덤한 튜플이 정당한 MDBDH 튜플이 되는 경우이다. 이러한 경우는 1/|q|의 확률로 발생되므로 다음을 얻는다.

$$\Pr\left[\mathrm{Exp}^{rand}_{\Delta_M}(k)=1|b=0\right]\geq\frac{1}{2}+\frac{1}{|q|}$$

위 두식에 의해서 다음 식을 얻고 이는 정리 3.1의 결과를 만족시킨다.

$$\Pr\left[\mathrm{Exp}^{rand}_{\Delta_M}(k)=1|b=0\right]-\Pr\left[\mathrm{Exp}^{rand}_{\Delta_M}(k)=1|b=0\right]$$
$$\geq \frac{1}{2}Adv^{cta}_{A}(k,q_T,q_C,T_D)-\frac{1}{|q|}$$

단, Δ_M 알고리즘 구축에 필요한 시간 T_M는 기존의 A가 요구하는 시간 T에서, CSI, TCK, Challenge 답변을 할 때 요구되는 시간을 합한 값이다. 그러므로 T_M는 $T+((m+4)q_T+(3m+2)q_C+3m+2)T_G$에 수렴된다.

V. 결론 및 향후 연구과제

본 논문에서는 상수 량의 비용을 가지는 효율적인 키워드 검색 프로토콜 ECKS를 UDU 환경에서 제안하였다. 그리고 그 안전성도 MDBDH 문제의 어려움에 기반 하여 증명하였다. 제안된 프로토콜은 암호화 문서 및 암호화 데이터베이스 혹은 방송 통신 융합 환경의 암호화된 데이터 집적소의 안전하고 효율적인 자료 검색을 위해 널리 활용될 수 있다. 핵심적인 부분에서의 계산, 통신비용이 상수양을 요구하므로, 충분히 실용화가 가능한 알고리즘이다. 다만, 좀 더 효율적인 설계를 위해, 데이터저장소의 페어링 개수를 4개에서 더 줄일 필요성이 있다. 궁극적으로 페어링 개수를 요구하지 않는 것이 이상적이지만, 프로토콜의 설계의 구조상 어려워 보인다. 하지만, 1개 혹은 2개까지 비용을 줄이는 것은 차후에 계속해서 연구해야 할 주제이다.

제안된 ECKS 프로토콜은 공통의 키워드들을 찾는 검색 기능만 제공하고 기타 여러 비교 연산들에 대한 기능은 제공하지 않는다. SDU 환경에서는 Boneh와 Waters가 처음으로 비교 연산이 가능한 키워드 검색 프로토콜을 제안하였다[6]. 하지만, 그 검색 비용이 키워드 필드 수에 증가하는 페어링 연산을 요구하므로 비효율적이다. UDU 환경에서, 비교 연산들 (예: SQL)이 가능한 효율적인 키워드 검색 프로토콜 설계 및 안전성 증명은 현재 시급히 연구되어야 할 주제이다.

참 고 문 헌

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Encryption with keyword search, revisited: consistency conditions, relations to anonymous IBE, and extensions" *In Proceedings of Crypto'05*, LNCS Vol. 3621, pp. 205-222, Springer-Verlag, 2005.

[2] B. Bloom, "Space/time trade-offs in hash coding with allowable errors", *Communications of the ACM*, 13(7):422-426, 1970.

[3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search", *In Proceedings of Eurocrypt'04*, LNCS Vol. 3089, pp. 31-45, Springer-Verlag, 2004.

[4] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing", *SIAM J. of Computing*, Vol. 32, No. 3, pp. 586-615, 2003.

[5] J. W. Byun, D. H. Lee, and J. Lim, "Efficient Conjuctive Keyword Search on Encrypted Data Storage System", *In Proceedings of EuroPKI'06*, LNCS Vol. 4043, pp. 184-196, 2006.

[6] D. Boneh and B. Waters, "Conjunctive, Subset, and Range Queries on Encrypted Data", *In Proceedings of TCC'07*, LNCS Vol. 4392, pp. 535-554, 2007.

[7] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private Information Retrieval", *In Proceedings of 29th STOC*, 1997.

[8] G. Di. Crescenzo, Y. Ishai, and R. Ostrovsky, "Universal Service-providers for Database Private Information Retrieval", *In Proceedings of 17th PODC*, 1998.

[9] G. Di. Crescenzo, T. Malkin, and R. Ostrovsky, "Single-database private information retrieval implies oblivious transfer", *In Proceedings of Eurocrypt'00*, LNCS Vol. 1807, pp. 122-139,

Springer-Verlag, 2000.
[10] C. Cachin, S. Micali, and M. Stadler, "Computationally Private Information Retrieval", *In Proceedings of Eurocrypt'99*, LNCS Vol. 1403, pp. 361-374, 1998.
[11] Y. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data", *In Proceedings of ACNS'05*, LNCS Vol. 3531, pp. 442-445, Springer-Verlag, 2005.
[12] P. Golle, J. Staddon, and B. Waters, "Secure Conjunctive Keyword Search Over Encrypted Data", *In Proceedings of ACNS'04*, LNCS Vol. 3089, pp. 31-45, Springer-Verlag, 2004.
[13] S. Goldwasser and M. Bellare, Lecture notes on cryptography", page 155, 2001. Available at http://www-cse.ucsd.edu/users/mihir/courses.html
[14] E. Goh, "Secure Indexes", *In Cryptology ePrint Archieve* on March 16, 2004, Availabe at http://eprint.iacr.org/2003/216
[15] A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, "2004 CSI/FBI Computer Crime and Security Survey", Ninth annual report of computer security society, CSI, 2004. For general information, refer to "http://gocsi.com or http://www.nipc.gov"
[16] Microsoft Developer Network (MSDN), in the part of Maximum Capacity Specifications for SQL Server 2005. Refer to http://msdn2.microsoft.com/en-us/library/ms143432(SQL.90).aspx
[17 M. Naor and M. Yung. "Universal One-way Hash Functions and Their Cryptographic Applications", *In Proceedings of the 21st ACM Symposium on Theory of Computing*, pp 33-43, ACM Press, 1989
[18] R. Ostrovsky and W. Skeith, "Private keyword search on streaming data", This paper is availabe at http://eprint.iacr.org/2005/242.
[19] W. Ogata and K. Kurosawa, "Oblivious keyword search" *Journal of Complexity*, Vol. 20, Issues 2-3, pp. 356-371, 2004.
[20] D. J. Park, K. Kim, and P. J. Lee, "Public Key Encryption with Conjunctive Field Keyword Search", *In Proceedings of WISA'04*, LNCS Vol. 3325, pp. 73-86, Springer-Verlag, 2004.
[21] V.Shoup, "Sequences of games: a tool for taming complexity in security proofs", *Cryptology ePrint Archive*, Report 2004/332, 2004.
[22] M. Scott and P. S. L. M. Barreto, "Compressed pairing", *In Proceedings of Crypto'04*, LNCS Vol. 3152, pp. 140-156, Springer-Verlag, 2004.
[23] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data", *In Proceedings of IEEE sysmposium on Security and Privacy*, 2000.
[24] V. D. R. Safavi-Naini, and, F. Zhang, "New traitor tracing schemes using bilinear map", *In 2003 ACM Workshop on Digital Rights Management (DRM 2003)*, 2003.

저 자 소 개

변 진 욱(정회원)
2001년 고려대학교 전산학과 이학사 졸업.
2003년 고려대학교 정보보호 대학원 공학석사 졸업.
2006년 고려대학교 정보보호 대학원 공학박사 졸업.
2007년 런던대학, ISG, 박사 후 연구원
2008년 평택대학교 정보통신학과 전임강사
<주관심분야 : 정보보호 프로토콜, 프라이버시 보호 기술, 패스워드 인증, 정보통신 프로토콜>