Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.1, Feb. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.1.5

소수테이블을 이용한 실용적인 다중 키워드 검색가능 암호시스템

양 유 진,† 김 상 진‡
한국기술교육대학교

Practical Conjunctive Searchable Encryption Using Prime Table

Yu-jin Yang,† Sangjin Kim‡
Korea University of Technology and Education

요 약

검색가능 암호시스템(searchable encryption system)은 암호화된 자료의 기밀성이 보장된 상태로 원하는 자료의 검색을 가능하게 해주는 기술이다. 클라우드 서비스의 대중화로 데이터 아웃소싱에 대한 관심이 높아지면서 외부 서버의 신뢰 문제를 해결하는 방법으로 최근에 많은 연구가 진행되고 있다. 하지만 대부분의 검색가능 암호시스템에 대한 연구는 하나의 키워드를 이용한 부울 검색만 제안되었고, 다중 키워드 검색에 대한 연구결과는 상대적으로 적을 뿐만 아니라 이 연구들은 대부분 고정 필드 환경을 가정하는 제한적 기법들이다. 이 논문에서는 고정 필드를 사용하지 않으며, 랭킹 정보까지 제공할 수 있는 새로운 다중 키워드 검색가능 암호시스템을 제안한다. 이 시스템은 키워드와 소수를 매핑한 소수테이블과 검색 연산으로 최대공약수 연산을 사용하기 때문에 기존 시스템보다 상대적으로 효율적이며, 복잡한 암호모듈이 필요 없어 비교적 쉽게 구현이 가능하다.

ABSTRACT

Searchable encryption systems provide search on encrypted data while preserving the privacy of the data and the search keywords used in queries. Recently, interest on data outsourcing has increased due to proliferation of cloud computing services. Many researches are on going to minimize the trust put on external servers and searchable encryption is one of them. However, most of previous searchable encryption schemes provide only a single keyword boolean search. Although, there have been proposals to provide conjunctive keyword search, most of these works use a fixed field which limit their application. In this paper, we propose a field-free conjunctive keyword searchable encryption that also provides rank information of search results. Our system uses prime tables and greatest common divisor operation, making our system very efficient. Moreover, our system is practical and can be implemented very easily since it does not require sophisticated cryptographic module.

Keywords: searchable encryption, conjunctive keyword search, rank, prime table

I. 서 론

최근 클라우드 컴퓨팅의 발전과 더불어 제3의 서버에 데이터를 보관하는 서비스가 보편화되고 있다. 사용자들이 하나의 단말만 사용하는 것이 아니라 여러 단말을 사용하고 있고, 언제 어디서나 인터넷 접속이 가능해짐에 따라 데이터를 외부 저장 공간에 유지하는

접수일(2013년 7월 5일), 수정일(2013년 11월 19일), 게재확정일(2013년 12월 3일)
† 주저자, sunyujin@koreatech.ac.kr
‡ 교신저자, sangjin@koreatech.ac.kr (Corresponding author)

경향이 증가하고 있다. 개인뿐만 아니라 기업이나 기관도 구축, 관리 등의 비용을 줄이기 위해 데이터 아웃소싱의 필요성을 인식하고 있다.

데이터를 직접 관리하더라도 해킹 또는 내부자에 의한 유출 등 다양한 보안 문제가 발생할 수 있어 적절한 보안 메커니즘의 활용이 필수적이다. 하지만 직접 통제를 못하는 외부 저장 공간에 데이터를 유지할 경우 해당 업체가 보안에 대한 충분한 확신을 고객에게 주어야 활용률이 높아질 것이다. 특히 신뢰관계가 높더라도 데이터 자체가 보호되어 있지 않을 경우에는 중요하거나 사적인 데이터까지는 아웃소싱하기 힘들 것이다. 따라서 이를 해결하기 위한 방법으로 외부 저장 공간에 데이터를 유지할 때 데이터를 암호화하여 유지하는 것을 고려할 수 있다. 하지만 데이터가 암호화된 상태로 유지되어 외부 서버가 데이터 내용에 접근할 수 없을 경우 검색과 같은 기존 서비스를 그대로 적용하여 사용하기 힘들다. 이와 같은 문제를 해결하기 위해 연구되고 있는 것 중 하나가 검색가능 암호시스템[1]이다.

검색가능 암호시스템은 암호화된 데이터에 대해 이것을 복호화하지 않고 검색할 수 있게 해주는 시스템이다. 즉, 검색을 제공하는 서버에게 데이터와 질의 내용을 노출시키지 않고 검색 서비스를 받을 수 있게 해주는 시스템이다. 데이터를 저장하는 사용자와 검색하는 사용자가 같은 단일 사용자 환경에서는 브라우징을 통해 원하는 정보를 찾을 수 있지만 데이터의 양이 많아질 경우 또는 다중 사용자 환경에서는 검색을 통해 원하는 데이터를 찾는 기능이 필수적으로 필요하다.

검색가능 암호화는 데이터 아웃소싱 외에 데이터베이스 검색, 전자우편 등 다양한 응용에서 활용될 수 있다. 하지만 지금까지 제안된 대부분의 검색가능 암호시스템은 단일 키워드 부울 검색만 제공하기 때문에 검색의 질 측면에서 한계를 가지고 있다[2,3]. 이 때문에 이를 개선하기 위한 노력도 현재 많이 진행되고 있으며, 그 중 하나가 다중 키워드 검색 지원[4-7]이다. 다중 키워드 검색이란 여러 개의 키워드를 이용하여 AND 조건으로 검색하는 것을 말한다. 따라서 단일 키워드 검색을 여러 번 하여 결과 집합의 교집합을 구하여 동일 효과를 제공할 수도 있다. 이 때문에 다중 키워드 검색은 단일 키워드 검색을 여러 번 반복 수행하는 것보다 통신 및 연산 비용이 효율적이어야 한다.

지금까지 제안된 대부분의 다중 키워드 검색가능 암호기법은 고정 필드를 사용하는 제한적 기법[4-5]들이다. 실제 한 문서에 연관된 키워드 수가 k이고, 질의에 포함된 키워드 수가 t이면 각 문서마다 기본적으로 $k \cdot t$개의 비교가 필요하지만 l개의 고정 필드를 사용하면 각 문서마다 최대 l개의 비교만 필요하다. 최근에 고정 필드를 사용하지 않는 기법[6]도 제안되었지만 검색 연산이 지수와 겹선형(pairing) 연산이 필요하기 때문에 검색 비용이 비교적 비싸다. 또한 다중 키워드 검색임에도 불구하고 질의한 키워드 중에 몇 개의 키워드와 문서가 일치하였는지 알 수 없다.

이 논문에서는 필드를 고정하지 않고 검색 결과의 랭크를 제공할 수 있는 다중 키워드 검색가능 암호기법을 제안한다. 제안하는 기법은 기존 논문에서 많이 사용하고 있는 지수 연산이나 겹선형 연산을 사용하지 않고 소수테이블과 최대공약수를 사용한다. 따라서 복잡한 암호모듈이 필요 없기 때문에 쉽게 구현이 가능하며, 효율적이다. 최대공약수 연산에 기반하고 있기 때문에 근본적으로 문서간의 연관관계가 다른 기법보다 쉽게 노출될 수 있는 측면이 있지만 모든 검색가능 암호기법은 점진적으로 문서간의 연관관계의 노출이 불가피하기 때문에 실용적인 측면에서 제안한 시스템은 충분한 가치가 있다.

이 논문의 구성은 다음과 같다. 2장에서는 검색가능 암호시스템에 대해 보다 구체적으로 설명하고 이 논문의 주제인 다중 키워드 검색과 관련된 기존 연구들을 소개한다. 3장에서는 제안하는 기법의 기본 시스템을 상세히 설명하고, 4장에서는 기본 시스템의 문제를 극복하기 위한 방법들을 제안한다. 5장에서는 제안한 시스템의 안전성 및 효율성을 분석하고, 6장에서 이 논문의 결론과 향후 연구방향을 제시한다.

II. 관련 연구

검색가능 암호시스템은 Song 등[2]에 의해 처음 제안되었다. 초기에 제안된 검색가능 암호시스템은 대칭키 기반이었는데 그 후 Boneh 등[3]이 공개키 기반 시스템을 제안하면서 검색가능 암호시스템에 대한 연구가 본격화 되었다.

2.1 검색가능 암호시스템

검색가능 암호시스템에 참여하는 참여자는 크게 사용자와 검색서버로 구분된다. 사용자는 다시 데이터를 생성하여 저장하는 자와 검색하는 자로 나눌 수 있으

며, 저장 또는 검색 가능한 사용자의 다수 여부에 따라 크게 4 종류로 분류할 수 있다. 이 중 초기에는 SWSR(Single-Writer-Single-Reader) 모델에 대한 연구가 많았지만 Boneh 등[3]부터는 전자우편에 해당되는 단일 검색자의 공개키로 데이터를 암호화하는 MWSR(Multi-Writer-Single-Reader) 환경에 대한 연구가 많이 진행되었다, 하지만 보편적인 검색시스템이 되기 위해서는 MWMR(Multi-Writer-Multi-Reader)의 지원이 가능해야 한다.

MWMR에서는 검색서버 외에 그룹이 공동으로 공유해야 하는 키를 분배하고 관리하는 그룹서버가 필요할 수 있다. 이 경우 검색서버는 외부 서버이지만 그룹서버는 내부 서버로 구축 및 관리될 필요가 있다. 또한 그룹 멤버의 변화에 따라 전후방향 안전성을 제공하기 위한 키 갱신 및 저장된 데이터의 재암호화 등이 필요하지만 저장된 데이터를 모두 다운받아 키를 바꾸어 암호화하는 것은 현실적으로 어렵기 때문에 이를 효과적으로 해결하기 매우 힘들다.

검색가능 암호시스템은 사용자에 따른 분류와 상관없이 다음 3가지 단계로 동작한다.

- 저장단계: 사용자는 암호화된 데이터와 암호화된 키워드들을 서버에 전달하여 저장한다.
- 질의단계: 사용자는 검색에 사용할 키워드를 암호화하여 서버에 전달한다. 이 때 사용되는 질의문을 트랩도어(trapdoor)라 한다.
- 검색단계: 서버는 암호화된 데이터와 연관된 암호화된 키워드들과 질의자의 트랩도어를 이용하여 순차적으로 검색연산을 진행하여, 일치하는 암호화된 문서를 검색자에게 제공한다.

질의나 저장단계의 경우 서비스를 제공하는 외부 서버 측면에서는 접근하는 사용자의 인증은 필수적이다. 현재 인터넷에서 널리 사용되는 것처럼 간단하게 TLS(Transport Layer Security)와 패스워드 인증을 결합하여 사용할 수 있으며, 이 과정에서 세션키를 확립하여 안전한 채널로 상호 메시지를 교환할 수 있다. 기존에는 별도 보안채널을 사용하지 않아도 된다는 것[7]을 장점으로 부각시키는 경우도 있었지만 안전한 개체 인증이 필수적 요구되므로 데이터를 이중으로 암호화하는 비효율적인 측면도 있지만 암호화된 키워드나 트랩도어를 이중으로 암호화한다고 하여 서버나 클라이언트에 큰 부담은 되지 않는다.

검색가능 암호시스템에서 검색서버는 보통 궁금하지만 정직(curious-but-honest)하다고 가정한다. 이것의 의미는 검색서버는 사용자의 모든 요구에 대해 시스템에서 정해진 절차에 따라 서비스를 제공하여 주지만 데이터나 질의 내용에 대해서는 항상 궁금해 한다는 것을 말한다. 하지만 이 모델에 대한 정확한 정의가 없어 어디까지 정직한 것인지 단정하기 힘들다. 특히, 사용자와 검색서버 간의 공모를 통한 서버의 궁금증 해소 공격을 고려하는 논문은 거의 없다.

검색가능 암호시스템은 다음과 같은 근본적인 한계를 가지고 있다.

- 한계 1. 결정적 트랩도어를 사용할 경우(동일 질의어를 사용할 경우 결과 트랩도어가 항상 같은 경우[3]) 사용자의 질의 패턴이 검색 서버에게 노출된다.
- 한계 2. 확률적 트랩도어를 사용하더라도 검색서버는 결과 집합을 통해 동일 질의어의 사용여부와 사용자의 질의 패턴을 알 수 있다.
- 한계 3. 다양한 키워드로 검색을 많이 하면 할수록 검색서버는 검색 결과를 이용하여 저장된 문서간의 연관 관계를 파악할 수 있다. 즉, 문서간의 연관관계는 점진적으로 노출될 수밖에 없다.
- 한계 4. 다중 키워드 검색에서 랭킹 정보를 제공할 경우 검색에 사용한 키워드 개수가 노출될 수 있다.

2.2 기존 다중 키워드 검색가능 암호시스템

다중 키워드 검색가능 암호시스템은 Golle 등[4]이 처음 제안하였다. 이 시스템은 고정 필드를 사용한다. 고정 필드를 사용할 경우 각 문서마다 필드의 수만큼의 비교 연산이 필요하기 때문에 상대적으로 저렴한 비용으로 검색이 가능하지만 정해진 필드로 문서를 분류하는 것이 적절하지 않은 응용에는 적용하기 힘든 문제점이 있다. 또한 비교연산의 결과 모든 필드가 일치한 경우와 그렇지 않은 경우 두 가지 답만 얻을 수 있어 검색 결과에 대한 랭킹 정보를 제공하지 못한다. 더욱이 제안된 2개 시스템 중 첫 번째는 트랩도어의 크기가 저장된 문서에 비례하며, 두 번째는 검색 연산이 질의한 키워드 수에 비례한 겹선형 연산을 요구한다.

Hwang과 Lee[5]는 공개키 기반 다중 키워드 검색가능 암호시스템을 제안하였다. 하지만 이 시스템도

여전히 고정 필드를 사용하고 있으며, Golle 등의 시스템과 동일한 방식으로 동작하기 때문에 랭킹 정보는 제공하지 못한다. 하지만 Golle 등[4]과 달리 검색 연산은 항상 3개의 겹선형 연산만 요구한다는 측면에서 개선된 시스템이다.

최근 Wang 등[6]은 다항식을 이용하여 고정 필드를 사용하지 않는 다중 키워드 검색가능 암호시스템을 제안하였다. 이 시스템은 해당 문서와 연관된 키워드들이 어떤 차수 다항식의 해가 되도록 다항식을 구성하여, 이 다항식을 키워드 암호문으로 사용한다. 트랩도어는 사용자가 선택한 키워드들이 각 문서의 다항식에 해가 되는지 여부를 검사할 수 있도록 전달되는데, 각 키워드를 개별적으로 검사하는 것이 아니라 질의한 키워드들을 이용하여 생성한 하나의 값을 이용하여 검사하고 있어, 일부만 일치하더라도 검색이 성공한 것으로 간주하는 문제점이 있다. 이렇게 하는 이유는 사용자가 선택한 키워드의 개수를 숨기기 위함이다.

III. 소수테이블을 이용한 다중 키워드 검색가능 암호시스템

3.1 개요

이 장에서는 SWSR 모델을 가정하고 제안하는 시스템을 설명한다. MWMR로의 확장 가능 여부는 이 장 끝에 별도 설명한다. 제안하는 시스템의 기본 동작 방식은 다음과 같다. 사용자는 소수와 키워드가 매핑되어 있는 테이블을 가지고 있으며, 이 테이블에서 어떤 문서에 연관시키고 싶은 키워드들에 해당하는 소수들을 찾아 그들의 곱으로 키워드 암호문을 나타낸다. 트랩도어도 동일하게 생성되며, 검색은 키워드 암호문과 트랩도어 간에 최대공약수를 구하여 그 결과를 이용하여 판단한다.

이 때 소수는 일정한 크기의 소수를 사용하여 결과값의 크기에 따라 몇 개의 키워드가 일치하였는지 알 수 있도록 한다. 또한 특정 문서와 연관된 키워드의 수, 질의한 키워드의 수를 숨기기 위해 키워드 암호문과 트랩도어는 항상 같은 수의 소수를 사용하여 구성한다. 이를 위해 3개의 소수테이블, 키워드, 키워드 잉여, 트랩도어 잉여 소수테이블을 사용한다. 즉, 항상 5개의 소수로 트랩도어를 구성하도록 고정시킬 경우 사용자는 최대 4개의 키워드만을 이용하여 질의할 수 있으며, 필요한 임의의 소수는 트랩도어 잉여 소수테이블에서 선택하게 된다.

3.2 소수테이블

Table 1. Keyword prime table

색인	소수	키워드
1	p_1	zebra
2	p_2	null
⋮	⋮	⋮
s_k	p_{s_k}	monkey

제안하는 시스템은 [표 1]과 같은 소수와 키워드를 매핑한 테이블을 사용한다. 또 키워드 암호문과 트랩도어를 소수의 곱으로 나타낸다. 이 때 검색서버가 해당 소수의 곱을 쉽게 인수분해할 수 있으면 저장된 문서의 상호관계를 쉽게 파악할 수 있다. 이 정보는 키워드 암호문 간에 최대공약수를 계산하여 파악할 수도 있다. 2.1에서 제시된 한계 3에서 언급한 바와 같이 이 정보는 어떤 기법을 사용하더라도 점진적으로 노출될 수밖에 없지만 위 두 가지 방법 중 인수분해가 가능하면 너무 쉽게 노출되는 문제를 가지고 있다. 따라서 인수분해가 어렵도록 충분히 큰 소수를 사용해야 한다. 하지만 이 경우 암호화된 키워드 또는 트랩도어의 크기가 상대적으로 커지는 문제점을 가지고 있다.

이 논문에서는 512비트 크기의 소수를 사용한다고 가정한다. 이 경우 키워드 소수테이블의 실제 크기는 $s_k \times (|p|+|w|)$ 가 된다. 여기서 s_k는 키워드 소수테이블의 크기이다. 많은 종류의 키워드가 검색에 사용될 수 있지만 특정 문서 집합마다 다른 테이블을 사용할 수 있기 때문에 특정 소수테이블의 크기는 그렇게 크지 않을 것으로 예측된다. 예를 들어 $|p|=512bit$, $|w|=512bit$ (최대 32 문자, 1문자 $2byte$), $s_k=1000$이면 키워드 소수테이블의 실제 크기는 $125Kbytes$이다.

키워드 암호문 간에 최대공약수 계산에 의한 정보 노출은 잉여 소수테이블의 크기를 조절하여 극복할 수 있다. 키워드 암호문 크기가 len_k일 때 잉여소수가 암호문에 최대 $L=len_k-1$개 포함이 가능하다. 키워드 잉여 소수테이블의 크기가 s_{k_dummy}이고, 특정 개수의 잉여 소수가 포함될 확률은 같다고 가정할 때, 두 개의 키워드 암호문에 동일한 잉여 소수가 포함될 확률은 다음과 같다.

$$P=1-\frac{\sum_{j=1}^{L}\sum_{i=1}^{L}\frac{s_{k\_dummy-i}C_j}{s_{k\_dummy}C_j}}{L^2}$$

이 확률을 50%로 설정하고 싶고, $len_k = 5$이면 $s_{k_dummy} = 10$이다.

소수계량함수(prime counting function)에 의하면 512비트 소수는 약 11.36×10^{150}개 존재하기 때문에 사용하기 충분한 수의 소수는 늘 존재한다.

3.3 표기법

이 논문에서는 〔표 2〕에 제시된 표기법을 사용하여 제안된 시스템을 설명한다.

Table 2. Notation

K_m	문서 암호키(대칭키)
K_p	키워드 소수테이블 MAC 키
n	전체 문서의 개수
p	키워드에 할당된 소수
$\|p\|$	소수 크기(bit)
s_k	키워드 소수테이블 크기
s_{k_dummy}	키워드 잉여 소수테이블 크기
s_{t_dummy}	트랩도어 잉여 소수테이블 크기
q	키워드 잉여 소수
t	트랩도어 잉여 소수
len_k	키워드 암호문 크기
len_t	트랩도어 크기

3.4 시스템 설정

외부 서버와 사용자는 한 문서에 연관할 수 있는 최대 키워드 개수 $len_k - 1$에 동의한다. 이 경우 키워드 암호문의 크기는 $|p| \times len_k$로 고정된다. 트랩도어의 크기도 동일한 크기를 사용할 수 있고, 다른 크기를 사용할 수도 있지만 $|p| \times len_t$로 고정한다.

단일 사용자는 외부 서버에 가입하고 다음과 같은 단계를 통해 클라이언트 시스템을 설정한다.

- 단계 1. 데이터 암호키(대칭키) K_m을 생성하여 패스워드 암호기법을 통해 암호화된 상태로 유지한다.
- 단계 2. 소수테이블 색인키(대칭키) K_p를 생성하여 위와 마찬가지로 암호화된 상태로 유지한다.
- 단계 3. 키워드 소수테이블을 생성한다. 이를 위해 먼저 s_k만큼의 서로 다른 512비트 소수를 생성하여 테이블의 각 항을 채운다. 그 다음 OTP(One-Time Password)〔8〕에서 사용하는 방식과 유사하게 소수테이블 색인키와 사용할 키워드를 이용하여 hash MAC 값을 계산하고 이것을 색인으로 사용하여 키워드를 채운다.
- 단계 4. 소수테이블에서 사용하지 않은 같은 크기의 소수를 s_{k_dummy}과 s_{t_dummy}만큼 생성하여 2개의 잉여 소수테이블을 생성한다.
- 단계 5. 단계 3과 단계 4에서 생성한 테이블은 데이터 암호키로 암호화된 상태로 유지한다.

3.5 저장단계

문서 m_i을 저장하고 싶을 경우, 먼저 이 문서에 연관시키고 싶은 키워드들을 선택하여야 한다. 만약 해당 키워드가 키워드 소수테이블에 없으면 키워드 소수테이블에 먼저 등록해야 한다. 연관하고 싶은 키워드 개수가 w_k이면 각 키워드 w에 해당되는 소수 p_w를 키워드 소수테이블에서 얻고, 키워드 잉여 소수테이블에서 임의로 $len_k - w_k$만큼의 소수 q를 선택하여 다음과 같이 이들의 곱을 계산한다. 그 결과 W_i가 키워드 암호문이 된다. 사용자는 K_m을 이용하여 m을 암호화한 C_i를 W_i와 함께 서버에 전달한다.

$$S_i = [C_i, W_i]$$
$$C_i = E.K_m(m_i)$$
$$W_i = p_{w_1} p_{w_2} \cdots p_{w_k} q_1 \cdots q_{len_k - w_k}$$

이 때 서버는 사용자를 인증하여야 하며, 안전한 채널을 구축하여 이 채널로 정보를 교환한다.

3.6 질의단계

검색자는 먼저 질의할 키워드들을 선택하여야 한다. 이 시스템의 특성에 따라 키워드 소수테이블에 등록되어 있는 키워드들만 사용할 수 있다. w_t개의 키워드들을 이용하고 싶으면 해당 키워드 w에 해당되는 소수 p_w를 키워드 소수테이블에서 얻고, 트랩도어 잉여 소수테이블에서 임의로 $len_t - w_t$만큼의 소수 t를 선택하여 다음과 같이 이들의 곱을 계산한다. 그 결과 T가 트랩도어가 된다.

$$T = p_{w_t} p_{w_2} \cdots p_{w_t} t_1 \cdots t_{len_t - w_t}$$

사용자는 인증된 안전한 채널을 통해 T를 검색서버에게 전달하여 검색을 요청한다.

3.7 검색단계

사용자로부터 트랩도어 T를 받으면 해당 사용자의 모든 문서에 대하여 순차적으로 각 문서의 키워드 암호문 W_i와 T의 최대공약수를 계산한다. 그 결과가 1이면 해당 문서는 질의한 키워드들과 연관되어 있지 않다는 것을 말하며, 1보다 클 경우에는 그 크기를 계산하여 큰 순서대로 사용자에게 암호화된 문서를 제공한다.

3.8 소수테이블 갱신

2.1절 한계 3에서 언급한 바와 같이 정상적인 검색이 진행됨에 따라 문서 간의 연관관계가 검색서버에게 노출된다. 제안된 시스템은 정상적인 검색 외에 키워드 암호문간의 최대공약수를 계산하여 문서 간의 연관관계를 파악할 수도 있다. 이 문제는 소수테이블을 갱신하는 것으로 해결되지 않는다. 이것은 다른 시스템들도 마찬가지이다. 기존 데이터를 모두 다운받아 새 키로 암호화하여 다시 저장할 경우에만 검색서버가 지금까지 축적한 정보의 의미가 없어지게 된다. 이것은 매우 소모적인 일이며, 이 시스템만의 문제는 아니다.

제안한 시스템에서 소수테이블은 그 자체가 암호키 역할을 한다. 따라서 다른 일반적인 암호키들과 마찬가지로 갱신할 수 있어야 한다. 하지만 키워드 소수테이블을 갱신할 경우 기존에 저장된 모든 키워드 암호문을 갱신해야 하므로 효과적으로 할 수 없다. 이 역시 이 시스템만의 문제는 아니다. 다른 시스템에서도 사용된 키를 갱신하고자 하면 기존에 저장된 모든 키워드 암호문을 바꿀 수밖에 없다.

3.9 다중 사용자 환경으로의 확장

검색가능 암호시스템에서 MWMR 모델을 제공하기 위해서는 크게 다중 사용자에 의한 데이터 암호화/복호화, 다중 사용자에 의한 키워드 암호문과 트랩도어 생성, 사용자의 가입 및 탈퇴 3가지 측면에 대한 고려가 필요하다.

그룹 개념이므로 가장 쉽게 생각할 수 있는 방법은 그룹키 메커니즘을 사용하는 것이다. 이 경우 외부서버와 별도로 신뢰할 수 있는 그룹 키분배서버의 운영이 가능하면 SWSR 모델을 쉽게 MWMR 모델로 확장할 수 있다고 생각할 수 있다. 하지만 이 논문에서 고려하는 데이터 아웃소싱 응용의 경우에는 동적으로 교환되는 데이터를 보호하는 것이 핵심이 아니라 저장된 데이터에 대한 보호이기 때문에 기존에 암호화된 모든 데이터를 새로운 키로 다시 암호화해야 하는 문제가 있다. 이 문제는 근본적으로 해결하기 어렵다. 검색서버가 탈퇴한 사용자의 접근을 막는 방법을 생각할 수 있지만 이 경우 교환되는 암호화된 문서를 안전한 채널로 반드시 교환해야 하는 문제가 있다. 더욱이 궁금하지만 정직한 서버 모델을 사용하고 있기 때문에 검색서버가 탈퇴한 사용자와 공모하면 바로 내용을 볼 수 있게 되는 문제점도 있다.

제안한 시스템의 경우에는 이 문제 외에 소수테이블을 그룹 간의 공유해야 하는 문제가 있으며, 소수테이블은 정적이 아니라 새 키워드가 계속 추가될 수 있는 개념이기 때문에 동기화 문제까지 있다. 이 문제는 그룹서버가 소수테이블을 유지하고 새 문서를 검색서버에 올려야 할 경우 가장 최근 소수테이블을 받아 사용하는 방법을 고려할 수 있다. 이 과정에서 새 키워드를 테이블에 추가하고 싶으면 그룹서버에 요청하여 승낙을 받은 후에만 가능하다, 그룹서버는 특정 사용자에게 키워드 추가를 허용한 경우에는 다른 사용자의 추가 요청은 일시적으로 거부하게 된다. 거부된 사용자는 일정시간이 지난 후에 다시 키워드 추가를 요청할 수 있다. 이와 같은 키워드 소수테이블의 동기화는 키워드의 추가가 필요할 때만 이루어지며, 나머지 경우에는 각 클라이언트가 유지하고 있는 테이블을 이용하여 서비스를 사용하게 된다.

IV. 기본 시스템에 대한 개선

4.1 기본 시스템의 특징

3장에서 제안한 시스템을 기본 시스템이라 하며, 이 시스템은 다음과 같은 장점을 가지고 있다.

- 장점 1. 기존 시스템들에 비해 상대적으로 매우 저렴한 연산(최대공약수 계산)만 사용한다.
- 장점 2. 키워드 암호문과 트랩도어를 생성할 때마다 랜덤하게 잉여소수를 선택하여 생성되므로 같은 키워드들을 이용하여 이들을 만들더라도 서로 다른 값을 가질 확률은 매우 높다. 특히, 키워드 암호문과 트랩도어는 서로 다른 소수로 구성된 테이블에서 잉여소수를 선택하기 때문에 같아질 수 없다. 같은 키워드를 이용하여 생성한 두 개의 키워드 암호문 또는 두 개의 트랩도어가

같아질 확률은 잉여소수 테이블의 크기, 키워드 암호문과 트랩도어를 만들 때 포함되는 잉여소수의 개수에 의해 결정된다.
- 장점 3. 최대공약수 연산의 결과 값의 크기를 이용하여 검색 결과에 대한 랭킹 정보를 제공할 수 있다.
- 장점 4. 대칭키, 최대공약수 연산만 사용하기 때문에 복잡한 암호모듈이 필요 없어 비교적 쉽게 구현이 가능하다.

이와 같은 장점에도 불구하고 다음과 같은 문제점도 있다.

- 단점 1. 최대공약수 결과 크기가 1개 소수에 해당할 경우 해당 키워드 소수가 노출된다.
- 단점 2. 소수들의 곱으로 키워드 암호문과 트랩도어를 나타내기 때문에 두 값의 크기가 비교적 크다. 하나의 값으로 표현되지만 그 크기가 시스템에서 결정한 키워드 암호문에 포함할 수 있는 최대 키워드 개수에 비례한다.
- 단점 3. 시스템에서 정해진 개수 이상의 키워드들을 문서와 연관시킬 수 없다.

단점 1의 경우 키워드 소수테이블 자체가 노출되지 않을 경우 키워드 소수를 알게 되더라도 키워드 자체는 알 수 없다. 실제 검색서버는 다음과 같은 시도를 통해 문서들 간의 연관관계를 파악하고자 할 수 있다.

- 시도 1. 정상적인 검색 연산: 트랩도어와 키워드 암호문 간의 최대공약수 계산
- 시도 2. 키워드 암호문들 간의 최대공약수 계산
- 시도 3. 트랩도어 간의 최대공약수 계산
- 시도 4. 계산된 결과들을 이용한 반복적 최대공약수 계산

특히, 최대공약수를 계산하여 하나의 소수를 얻었을 경우, 이를 이용하여 추가적인 최대공약수 계산을 통해 또 다른 소수를 얻을 수 있다. 이것이 축적되면 사용되고 있는 대부분의 소수를 얻을 수도 있다.

기본 시스템에서 시도 1의 경우에는 키워드 잉여와 트랩도어 잉여는 서로소 집합이기 때문에 키워드 소수를 명확하게 구분할 수 있다. 시도 2와 시도 3은 잉여간 일치할 수 있는 가능성이 있기 때문에 최대공약수 연산 결과를 통해서는 이것이 키워드 소수인지 잉여소수인지 구분하기 어렵다.

단점 2의 경우 다음에 대한 고려가 필요하다. 제안하는 시스템은 문서에 연관하는 키워드 개수나 질의하는 키워드 개수를 숨기기 위해 항상 고정 크기의 키워드 암호문과 트랩도어를 사용하고 있다. 따라서 Golle 등의 시스템[4]이나 Hwang과 Lee[5] 시스템처럼 키워드 개수에 비례하는 시스템에 비해 평균적으로 키워드 암호문이나 트랩도어의 크기가 크지만 제안하는 시스템에서 최대 키워드 개수를 포함하였을 경우와 비교하면 차이가 크지 않다.

단점 3의 경우 현실적으로 한 문서와 연관하는 키워드 개수는 비교적 많지 않으므로 저장하는 문서의 종류 특성에 따라 초기에 적절하게 설정하면 큰 문제는 없다.

4.2 개선 시도

앞서 제시한 바와 같이 기본 시스템의 경우 최대공약수 연산의 특징을 이용하여 문서간의 연관관계를 점진적으로 알게 되는 문제점이 있다. 하지만 이를 부분적으로 극복할 수도 있다. 이를 위해서는 정상적 또는 비정상적으로 획득한 소수가 키워드 소수인지 잉여 소수인지 구분하기 힘들게 하여야 한다. 이렇게 되면 모든 문서와 연관된 키워드 암호문을 인수분해할 수 있어도 문서 간의 연관관계를 단정적으로 알 수 없게 된다.

구분이 어렵게 하기 위한 몇 가지 방법을 고려하여 보자. 가장 쉽게 생각해 볼 수 있는 방법은 첫째, 키워드 잉여와 트랩도어 잉여 테이블 2개를 사용하지 않고 하나만 사용하는 것이다. 이 방법을 축소잉여 소수테이블 기법이라고 한다. 이 경우 키워드 소수와 잉여소수간의 구분이 힘들지만 정상적인 경우에도 랭킹정보를 파악하기 힘든 문제가 있다. 특히 일치할 확률을 조절하거나 일치할 개수를 제한하기 힘들다.

또 다른 방법은 기본 시스템에 공통 잉여 소수테이블이라고 하는 테이블을 하나 더 사용하여 키워드 암호문과 트랩도어에 공통 잉여 소수테이블에서 항상 정해진 개수만큼 포함시키는 방법을 생각해 볼 수 있다. 이 때 일치할 개수를 최대 1로 제한하기 위해 키워드 암호문이나 트랩도어 중 하나는 1개만 포함하도록 할 수 있다. 예를 들어 공통 잉여 소수테이블의 크기를 10이라고 하면 키워드 암호문에는 공통 잉여로부터 2개의 소수를 임의로 포함하고, 트랩도어에는 1개의 소수를 임의로 포함하면 20%의 확률로 공통잉여가 일

치하게 된다. 이 기법을 공통 잉여 소수테이블 기법이라 한다.

공통 잉여 소수테이블 기법의 경우에도 잉여소수가 일치할 수 있어 정확한 랭킹정보를 파악할 수는 없지만 일치할 수 있는 잉여소수가 최대 하나이므로 랭킹 정보 활용에 큰 문제는 되지 않는다. 또한 불필요한 문서가 일부 검색되므로 이것이 문서 연관 관계를 파악하는데 혼란을 주기 때문에 정보노출 측면에서는 기본 시스템보다 이 측면에서 이점을 가지고 있다.

V. 분석

5.1 안전성 분석

5.1.1 정보노출 문제

기본적으로 검색가능 암호시스템은 다음 요구사항을 충족하여야 한다.

- R1. 저장된 문서의 내용이 노출되지 않아야 함
- R2. 문서와 연관된 키워드 정보가 노출되지 않아야 함
- R3. 트랩도어에 포함된 키워드 정보가 노출되지 않아야 함

R1의 경우 데이터 암호키와 관련된 요구사항이며, 이 논문에서는 이에 대해서는 고려하지 않는다. R2와 R3의 경우 키워드 소수테이블 자체가 노출되지 않으면 키워드 암호문과 트랩도어를 통해 사용된 키워드 정보가 노출되지는 않는다. 다만, 4장에서 언급한 바와 같이 최대공약수 연산을 이용하여 문서 간의 연관 관계가 점진적으로 노출되지만 4.2에서 설명한 공통 잉여 소수테이블 기법을 사용하면 연관 관계를 파악하는 것이 어려워진다.

5.2 기타 안전성 분석

5.2.1 키워드 추측 공격

키워드 추측공격은 트랩도어에 포함된 키워드를 알아내기 위한 공격으로 검색서버가 키워드 암호문을 직접 생성할 수 있는 시스템은 이 공격에 취약하다[9]. 예를 들어 MWSR 모델에서 단일 검색자의 공개키로 키워드 암호문을 생성하는 경우 검색서버는 다양한 키워드에 대한 암호문을 직접 만들 수 있으며, 이를 이용하여 검색자가 전달한 트랩도어 간의 검색 연산을 반복 수행하여 해당 트랩도어의 키워드를 알아낼 수 있다. 하지만 제안하는 시스템은 검색서버가 특정 키워드에 매핑된 소수를 얻어낼 수 없기 때문에 키워드 추측공격이 가능하지 않다.

5.2.2 차집합 공격

검색서버가 정당한 검색자로부터 수신한 트랩도어들을 이용하여 새로운 유효한 트랩도어를 만들 수 있을 경우 차집합(different set) 공격에 취약하다고 한다[10]. 제안한 시스템의 경우 소수곱으로 트랩도어를 나타내기 때문에 여러 개의 트랩도어로부터 아직까지 질의가 되지 않은 트랩도어를 만들 수 있을 가능성이 높다. 특히, 키워드 소수가 직접적으로 노출되는 기본 시스템의 경우 차집합 공격에 매우 취약하다고 볼 수 있다. 하지만 차집합 공격은 본질적으로 문제가 되지 않는 공격이다. 그 이유는 검색서버가 새롭게 만든 트랩도어와 동일한 능력을 가진 트랩도어를 이전에 이미 검색자가 질의하였을 수도 있고, 향후에 질의할 가능성이 있기 때문이다. 더욱이 새로운 트랩도어를 통해 검색서버가 알게 되는 문서간의 연관도는 이미 노출되어 있을 수도 있다. 예를 들어 w_1,w_2을 이용하여 만든 트랩도어 T_{w_1,w_2}와 w_1을 잉용하여 만든 T_{w_1}을 가지고 $T_{w_1,w_2}-T_{w_1}$을 계산할 수 있다는 것이 차집합 공격이지만 실제 T_{w_2}을 추가적으로 가지고 있었다면 해당 정보는 이미 T_{w_1}과 T_{w_2}을 이용한 검색 결과를 이용하여 알 수 있는 정보이다.

5.3 효율성 분석

기존의 시스템과 이 논문에서 제안한 기법의 기능적 비교는 [표 3]과 같다. 표에 알 수 있듯이 검색연산의 효율성 측면에서는 제안하는 시스템이 가장 우수하다. 제안하는 기법은 고정 크기의 키워드 암호문과 트랩도어를 사용하기 때문에 키워드 수 비례하는 시스템에 비해 평균적으로 크지만 이를 통해 추가적으로 문서와 연관된 키워드 수나 질의에 사용된 키워드 수를 숨길 수 있다. 또한 고정 필드를 사용하지 않으며, 랭킹 정보를 제공한다는 측면에서 다른 시스템들과 차별화된다.

Hwang과 Lee 기법[5]이나 Wang 등[6]의 기법

Table 3. Comparison with previous conjunctive searchable encryption schemes

	필드 사용유무	검색연산 효율	키워드 암호문 크기	트랩도어 크기	랭킹정보 제공	다중사용자 환경 지원
[4]-1	○	지수 연산	키워드 수 비례	저장된 문서에 비례	×	×
[4]-2	○	겹선형 연산 (키워드 수 비례)	키워드 수 비례	키워드 수 비례	×	×
[5]	○	겹선형 연산 (상수 비용)	키워드 수 비례	키워드 수 비례	×	△
[6]	×	겹선형 연산 (안전성 파라미터에 비례)	(안전성 파라미터에 비례)	안전성 파라미터에 비례	×	△
제안기법	×	GCD	고정(포함할 수 있는 최대 키워드 수 비례)	고정(포함할 수 있는 최대 키워드 수 비례)	○	△

모두 다중 사용자 환경을 고려하였지만 3.9에서 제시한 바와 같이 검색가능 암호시스템의 기본적 특성 때문에 동적 가입과 탈퇴 및 데이터 갱신에 대한 효과적인 솔루션을 제시하지는 못하고 있다.

VI. 결 론

이 논문에서는 소수테이블과 최대공약수 연산을 이용하는 새로운 다중 키워드 검색가능 암호시스템을 제안하였다. 기존 기법들에 비해 상대적으로 효율적인 연산을 사용하며, 복잡한 암호기술을 사용하지 않고 있기 때문에 구현 측면에서도 이득이 있어 실용적인 시스템이다. 더욱이 기존과 달리 랭킹 정보를 제공할 수 있다. 하지만 기존 시스템과 마찬가지로 2.1에서 설명한 문선 간 연관 관계가 점진적으로 노출되는 문제는 효과적으로 해결하지는 못하고 있다. 이에 향후 효과적으로 소수테이블과 암호화된 문서들을 갱신할 수 있는 방법에 대한 연구가 추가적으로 필요하다.

References

[1] S. Kim, J. Seo, P. Lee, "The state of the art in searchable encryption," Review of KIISC, 19(2), pp. 63-73, Apr. 2009.

[2] [2] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," Proc. of the IEEE Symp. on Security and Privacy, pp. 44-55, May 2000.

[3] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," Advances in Cryptology, Eurocrypt 2004, LNCS 3027, pp. 506-522, May 2004.

[4] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," Proc. of 2nd Int. Conf. on Applied Cryptography and Network Security, LNCS 3089, pp. 31-45, Jun. 2004.

[5] Y. Hwang and P. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," Proc. of the Pairing 2007, LNCS 4575, pp. 2-22, Jul. 2007.

[6] P. Wang, H. Wang, and J. Pieprzyk, "Keyword field-free conjunctive keyword searches on encrypted data and extension for dynamic groups," Proc. of 7th Int. Conf. on Cryptology and Network Security, LNCS 5339, pp. 178-195, Dec. 2008.

[7] J. Baek, R. Safiavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," Proc. of Int. Conf. on Computational Science and Its Application, LNCS 5072, pp. 1249-1259, Springer, Jun. 2008

[8] D. N'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, "HOTP: An HMAC-based one-time password algorithm," RFC 4225, Dec. 2005.

〔9〕 J. Byun, H. Rhee, H. Park, and D. Lee, "Off-Line keyword guessing attacks on recent keyword search schemes over encrypted data," Proc. of 3rd VLDB Workshop, SDM 2006, LNCS 4165, pp. 75 - 83, Sept. 2006.

〔10〕 H. Rhee, I. Jeong, J. Byun, and D. Lee, "Difference set attacks on conjunctive keyword search schemes," Proc. of 3rd VLDB Workshop, SDM 2006, LNCS 4165, pp. 64 - 74, Sept. 2006.

〈저자소개〉

양 유 진 (Yujin Yang) 정회원
2011년 8월: 한국기술교육대학교 컴퓨터공학부 졸업
2013년 8월: 한국기술교육대학교 컴퓨터공학과 석사
〈관심분야〉 암호기술응용

김 상 진 (Sangjin Kim) 종신회원
1995년 2월: 한양대학교 전자계산학과 졸업
1997년 2월: 한양대학교 전자계산학과 석사
2002년 8월: 한양대학교 전자계산학과 박사
2003년 3월~현재: 한국기술교육대학교 컴퓨터공학부 부교수
〈관심분야〉 암호기술응용