Vol.6, No.10 pp.429~436 pISSN: 2287-5891 https://doi.org/10.3745/KTCCS.2017.6.10.429

An Improved Multi-Keyword Search Protocol to Protect the Privacy of Outsourced Cloud Data

Tae-Yeon Kim† · Ki-Hwan Cho†† · Young-Lok Lee†††

ABSTRACT

There is a growing tendency to outsource sensitive or important data in cloud computing recently. However, it is very important to protect the privacy of outsourced data. So far, a variety of secure and efficient multi-keyword search schemes have been proposed in cloud computing environment composed of a single data owner and multiple data users. Zhang et. al recently proposed a search protocol based on multi-keyword in cloud computing composed of multiple data owners and data users but their protocol has two problems. One is that the cloud server can illegally infer the relevance between data files by going through the keyword index and user's trapdoor, and the other is that the response for the user's request is delayed because the cloud server has to execute complicated operations as many times as the size of the keyword index. In this paper, we propose an improved multi-keyword based search protocol which protects the privacy of outsourced data under the assumption that the cloud server is completely unreliable. And our experiments show that the proposed protocol is more secure in terms of relevance inference between the data files and has higher efficiency in terms of processing time than Zhang's one.

Keywords : Cloud, Data Privacy, Outsource, Keyword, Multiple Owner

아웃소싱된 클라우드 데이터의 프라이버시를 보호하기 위한 멀티 키워드 검색 프로토콜의 개선

김 태 연† · 조 기 환†† · 이 영 록†††

요 약

최근 클라우드 컴퓨팅 환경에서 민감하거나 중요한 데이터를 아웃소싱하는 경향이 늘어나고 있다. 하지만 아웃소싱된 데이터의 프라이버시 보호는 매우 중요하다. 지금까지 단일 데이터 소유자와 다수의 데이터 사용자로 구성된 클라우드 컴퓨팅 환경에서 안전하고 효율적인 멀티 키워드 검색 구조들이 다양하게 제안되었다. Zhang 등은 다수의 데이터 소유자들과 사용자들로 연결된 클라우드 컴퓨팅 환경에서 멀티 키워드를 기반으로 하는 검색 프로토콜을 제안하였다. 그들의 프로토콜은 두 가지 문제점을 동시에 안고 있다. 하나는 클라우드 서버가 키워드 인덱스와 사용자의 트랩도어를 통해 데이터 파일들 간의 연관성을 불법적으로 추론할 수 있다는 것이고, 다른 하나는 키워드 인덱스의 크기만큼 복잡한 연산을 수행해야 하기 때문에 사용자의 요청에 대한 응답이 지연된다는 것이다. 본 논문에서는 클라우드 서버를 전적으로 신뢰할 수 없는 노드라는 가정 하에서 우리는 아웃소싱된 데이터의 프라이버시가 보호되는 개선된 멀티 키워드 기반 검색 프로토콜을 제안한다. 그리고 제안된 프로토콜이 Zhang의 프로토콜보다 데이터 파일들 간의 연관성 추론 측면에서 더 안전하고, 처리 시간의 측면에서 더 효율성이 높음을 실험을 통해 보인다.

키워드 : 클라우드, 데이터 프라이버시, 아웃소스, 키워드, 다수 소유자

1. 서 론

클라우드 컴퓨팅은 IT 인프라의 이용성과 확장성, 비용 측면에서 많은 이점을 가지고 있다. 클라우드 사용자는 필요한 IT자원을 물리적으로 소유하지 않는 상태에서 필요한 만큼의 자원을 시간의 제약 없이 할당받아서 원격접속을 통해 사용할 수 있다. 이와 같은 이점에도 불구하고 많은 사용자들은 클라우드 자원을 광범위하게 사용하는 것을 주저한다. 그 이유는 클라우드 사용자가 민감하고 중요한 데이터의 계산이나 관리 등을 아웃소싱하는 과정에서 다양한 보안 공격이 발생할 수 있기 때문이다. 이를 방지하기 위해 서비스 공급자(CSP, 서버)는 시스템 하드웨어 및 소프트웨어 스택의 낮은

※ 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음(IITP-2017-2015-0-00378).
† 종신회원 : 서남대학교 컴퓨터정보학과 조교수
†† 정 회 원 : 전북대학교 컴퓨터공학부 부교수
††† 정 회 원 : 전남대학교 정보보안협동과정 초빙교수
Manuscript Received : March 13, 2017
First Revision : May 12, 2017
Second Revision : June 20, 2017
Accepted : August 2, 2017
* Corresponding Author : Ki-Hwan Cho(ghcho@chonbuk.ac.kr)

수준까지 제어하는 방어벽 및 가상화와 같은 메커니즘을 적용하고 있다[1]. 그럼에도 불구하고 사용자가 아웃소싱한 데이터를 관리하는 서버를 우리가 절대적으로 신뢰할 수는 없기 때문에 데이터 프라이버시를 전적으로 보장받는다고 할 수 없다.

이러한 문제를 해결하기 위해 사용자는 데이터를 평문이 아닌 암호문으로 변환해서 서버에게 아웃소싱하는 방식들이 제안되었다[2]. 또한 데이터 프라이버시의 보호와 효율적인 검색을 위해 멀티 키워드 검색이나 관련 순위가 고려된(ranked) 키워드 검색, 키워드 유사성 검색, 의미론적 키워드 검색 등의 기능이 지원되는 방식들도 제안되었다[3-5].

앞에서 기술한 방식들은 단일 데이터 소유자와 다수(multiple) 사용자로 구성되는 클라우드 컴퓨팅 환경을 가정하고 있다. 하지만 대부분의 실제 환경은 이와 다르게 다수의 데이터 소유자들과 다수의 사용자들로 구성되며, 각 소유자들은 키워드들을 암호화하는데 사용되는 비밀키를 다른 소유자들과 공유하는 것을 원치 않는다.

따라서 각 소유자는 자신만이 아는 비밀키로 데이터와 키워드들을 암호화하고, 사용자 역시 소유자의 비밀키를 모른 상태에서 질의를 생성할 수 있어야 한다. 그리고 서버 역시 서로 다른 키로 암호화된 키워드들 중에서 같은 키워드를 찾아낼 수 있어야 한다.

Zhang 등[6]은 다수의 데이터 소유자들과 사용자들이 연결된 클라우드 환경에서 관련 순위가 고려된 멀티 키워드를 기반으로 하는 검색 프로토콜을 제안하였다. 하지만 그들의 방식은 사용자의 요청에 대한 응답이 늦어지는 단점을 가지고 있다. 왜냐하면 키워드 인덱스와 사용자가 전송한 트랩도어(trapdoor)를 통해 신뢰할 수 없는 서버가 데이터들 간의 관련성을 불법적으로 추론할 수 있기 때문이고, 키워드 인덱스 내에 일치하는 키워드가 있는 지를 검색하기 위해 복잡한 쌍 선형 페어링(bilinear paring) 연산들을 반복적으로 수행해야 하기 때문이다.

다수의 데이터 소유자와 다수의 사용자들로 구성된 실제 클라우드 환경에서는 서버가 전적으로 신뢰할 수 없는 노드라는 가정 하에서 데이터 파일들 간 관련성 추론의 가능성을 최소화하고 사용자의 요청에 대한 응답이 늦어지는 문제를 줄일 수 있는 메커니즘이 필요하다. 본 논문에서는 아웃소싱된 데이터의 프라이버시를 보호하고 검색 효율성을 고려한 멀티 키워드 기반 검색 프로토콜을 제안한다.

본 논문의 구성은 1장의 서론에 이어 2장에서는 관련 연구에 대해 살펴본다. 3장에서는 Zhang[6]의 멀티 키워드 기반 검색 프로토콜을 기술하고, 4장에서는 제안된 멀티 키워드 기반 검색 프로토콜을 제시한다. 5장에서는 보안 및 성능 분석을 기술하고, 마지막 6장에서는 결론을 맺는다.

2. 관련 연구

Song 등[7]은 암호화된 데이터를 키워드 기반으로 검색하는 방식을 처음으로 제안하였다. 그들은 데이터 소유자가 문서내의 모든 워드들을 비밀키로 암호화한 후에 서버에게 전송하고, 서버는 워드들에 대해 패턴 매칭하는 방식으로 검색을 하기 때문에 매우 비효율적이었다. 이러한 문제를 해결하기 위해 Goh[8]은 Bloom 필터[9]를 사용하여 파일에 대한 키워드 인덱스를 생성하여 서버에게 전송하고, 서버는 사용자로부터 트랩도어를 수신하면 인덱스를 통해서 원하는 데이터를 검색하는 방식을 제안하였다. 하지만 Bloom 필터 기반 방식에서 필터의 오작동을 유도하여 사용자가 불필요한 데이터를 다운로드하도록 하는 문제가 발생하였다..

Curtmola 등[4]는 효율적인 키워드 기반 검색을 위해 각 데이터 파일들은 소유자별로 서로 다른 비밀키로 암호화하고, 모든 파일의 키워드들 중에서 같은 키워드는 동일한 비밀키로 암호화하는 방식으로 해쉬 테이블을 생성하는 구조를 제안하였다. 해쉬 테이블의 각 항목은 하나의 키워드와 그와 관련된 모든 데이터 파일들의 식별자를 나열하는 방식이었다.

위에서 기술한 방식들은 키워드 기반 검색은 지원되었지만 관련 순위가 고려되지 않았기 때문에 사용자는 불필요한 데이터까지 다운로드 받아야 하는 문제가 있었다. 따라서 Sun 등[2]는 데이터의 프라이버시를 보호하고 효율성을 향상시키기 위해 순위가 고려된 멀티 키워드 검색 기법을 제안하였다. 키워드와 데이터 파일들 간에 더 많은 매치가 되는 것을 찾기 위해 등위 매칭(coordinate matching)의 원리를 사용하였다.

Chen 등[5]은 암호화된 클라우드 데이터에 대해 의미론적 멀티 키워드 기반과 순위가 고려된 파일 검색이 가능한 방식을 제안하였다. 이 방식은 검색 키워드와 정확히 일치하는 파일뿐만 아니라 의미론적으로 키워드와 관련된 용어를 포함하는 파일도 다운로드할 수 있도록 하였다.

Boneh 등[10]은 공개키 기반에서 순위가 고려된 검색 구조를 제안하였다. 그들의 구조는 쌍 선형 사상과 트랩도어 순열(permutation)을 사용하기 때문에 처리 시간이 많이 걸리는 단점이 있었다. 그리고 Zhang 등[6]은 다수의 데이터 소유자가 연결된 클라우드 환경에서 데이터의 프라이버시와 사용자의 효율성을 보호할 수 있는 순위가 고려된(ranked) 멀티 키워드 검색 프로토콜(PRMSM)을 제안하였다. 사용자의 요청 질의 내의 암호화된 키워드가 인덱스 테이블 내의 키워드와 서로 일치하는 것이 있는지를 판단하기 위해 쌍 선형 페어링 연산을 수행하고, 특수 함수를 사용하여 키워드들과 관련된 데이터 파일들 간에 관련성 점수(relevance score)을 계산한 다음에, 관련성 점수가 높은 k개의 데이터를 다운로드하도록 하는 방식이다. 본 논문에서 제안한 구

조는 Zhang 구조와 유사하지만 키워드 인덱스의 생성과 관리를 클라우드 서버가 아닌 중간 노드에서 하도록 함으로써 데이터 파일들 간의 관련성 추론의 가능성을 최소화하고 과중한 계산으로 인해 발생되는 응답 지연의 문제를 줄일 수 있는 구조라는 측면에서 차이가 있다.

3. Zhang[6]의 멀티 키워드 기반 검색 프로토콜

다수의 데이터 소유자와 사용자가 연결된 클라우드 컴퓨팅 환경에서 데이터 프라이버시와 순위가 고려된 멀티 키워드 기반 검색 프로토콜을 요약하면 다음과 같다[6].

3.1 키워드 암호화

1) 데이터 소유자(i)는 아웃소싱하려는 파일 F을 암호화한 결과인 $[Data(F)]_{k_O}$을 클라우드 서버에게 전송하고, 관련된 키워드들(s) 중에서 h번째 키워드가 $w_{i,h}(\in s)$라고 했을 경우에 $E_{a'} = g^{k_{i,w} \cdot r_o \cdot H(w_{i,h})}$ 와 $E_o = g^{k_{i,w} \cdot r_o}$ 을 생성한 다음에 $\hat{w}_{i,h} = (E_{a'}, E_o)$ 을 관리 서버(administration server)에게 전송한다. 여기에서 k_O는 소유자의 비밀키이고, $k_{i,h}$는 키워드 $w_{i,h}$을 위한 소유자의 비밀키, r_o는 소유자가 생성한 난수, $H(\cdot)$는 해쉬 함수이다.

2) 관리 서버는 $E_a = (E_{a'} \cdot g^{k_{a1}})^{k_{a2}}$ 을 생성한 다음에 $\hat{w}'_{i,h} = (E_a, E_o)$ 을 클라우드 서버에게 전송한다. 여기에서 k_{a1}와 k_{a2}는 관리 서버의 비밀키이다.

3) 클라우드 서버는 암호화된 키워드들을 키워드 인덱스에 추가한다. 인덱스에 저장된 키워드들 중에 서로 같은 키워드가 있는 경우에도 소유자들의 비밀키가 서로 다르기 때문에 서로 다른 값으로 취급된다.

3.2 트랩도어 생성과 키워드 검색

1) 데이터 사용자는 원하는 데이터를 다운로드하기 위해 해당 파일에 관련된 키워드들을 트랩도어 내에 기술한다. 관련된 키워드들(d) 중에서 h'번째 키워드가 $w_{h'}(\in d)$인 경우에 암호화한 결과($T'_{w_{h'}} = (g^{H(w_{h'}) \cdot r_u}, g^{r_u})$)가 포함된 트랩도어를 관리 서버에게 전송한다. 여기에서 r_u는 데이터 사용자가 생성한 난수이다.

2) 트랩도어를 수신한 관리 서버는 키워드가 $w_{h'}$인 경우에 $T_1 = g^{H(w_{h'}) \cdot r_u \cdot k_{a1} \cdot k_{a2} \cdot r_a}$와 $T_2 = g^{r_u \cdot k_{a1}}$, $T_3 = g^{r_u \cdot k_{a1} \cdot r_a}$을 생성한 다음에 $T_{w_{h'}} = (T_1, T_2, T_3)$을 서버에게 전송한다. 여기에서 r_a는 관리 서버가 생성한 난수이다.

3) 클라우드 서버는 트랩도어를 수신하면 암호화된 키워드들이 키워드 인덱스에 있는지를 찾는 과정은 Equation (1)에 의해 이루어진다. 이 과정은 트랩도어 내에 기술된 키워드의 수만큼 반복 수행되고, Equation (1)을 만족하는 키워드와 관련된 데이터 파일들에 대한 관련성 점수를 계산한 다음에 점수가 가장 높은 k개의 파일을 사용자에게 전송하는 구조이다.

$$e(E_a, T_3) = e(E_o, T_1) \cdot e(E_a, T_2) \qquad (1)$$

4. 제안한 멀티 키워드 기반 검색 프로토콜

4.1 시스템 모델

소유자들은 클라우드 서버에게 자신의 파일들을 안전하게 아웃소싱하고, 정당한 사용자들은 서버에게 저장된 파일들을 효율적으로 접근할 수 있어야 한다. 본 논문에서 제안한 멀티 키워드 기반 검색을 위한 구조는 Fig. 1과 같이 다수의 데이터(파일) 소유자와 다수의 데이터(파일) 사용자, 신뢰된 에이전트(TA: Trusted Agent), 클라우드 서버(CS: Cloud Server)로 구성된다.

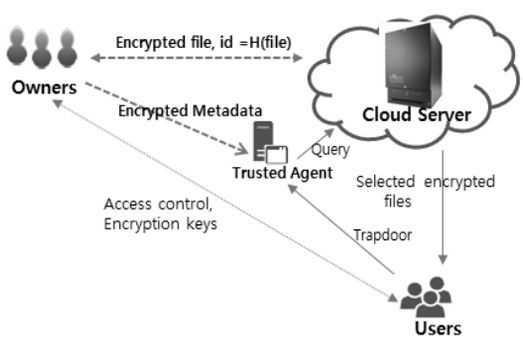

Fig. 1. Structure of the Proposed Cloud Computing

첫째, 파일 소유자는 CS에게 암호화된 데이터 파일($[Data(F)]_{k_O}$)과 파일 식별자 $H(file)$을 전송하고, 신뢰된 에이전트(TA)에게 아웃소싱할 파일에 대한 메타데이터를 전송한다(Table 1). 파일 사용자들에 대한 접근제어 정책과 각 파일에 사용된 비밀키의 관리에 대해서는 본 논문의 범위를 벗어나기 때문에 언급하지 않는다. 둘째, 파일 사용자는 소유자의 파일에 접근할 수 있는 정당한 노드로서, 원하는 파일을 다운로드하기 위해 키워드를 기반으로 하는 트랩도어를 생성한 후에 TA에게 전송하고, CS로부터 데이터 파일을 수신한다. 그리고 다운로드한 파일은 소유자로부

터 받은 비밀키를 사용하여 복호화한 다음에 내용을 읽는다. 셋째, TA는 CS와는 달리 보안상 전적으로 신뢰할 수 있는 노드로서, CS에게 아웃소싱할 파일들에 대한 메타데이터를 소유자로부터 받아 키워드를 기반으로 하는 인덱스를 생성하여 관리하는 역할과 사용자로부터 받은 트랩도어를 파일 식별자 기반으로 하는 질의로 변환하여 CS에게 전송하는 역할을 한다. 다시 말해서 TA는 사용자로부터 트랩도어를 수신하면 인덱스 내에 일치된 키워드들이 있는 지를 검색한 후에 검색된 파일 중에서 관련성 점수가 높은 파일들에 대한 식별자 기반 질의로 변환하여 CS에게 전송한다. 마지막으로 CS는 전적으로 신뢰할 수 없는 노드로서, 소유자의 파일을 안전하게 관리하고 Zhang 등[6]이 제안한 구조와 달리 소유자의 파일들에 대한 파일 기반 인덱스를 관리한다. 그리고 TA가 요청한 파일들을 해당 사용자에게 전송하는 역할을 한다. 앞으로 사용될 용어 $K_{X,Y}$는 노드X와 Y만이 공유하는 비밀키이다.

4.2 메타데이터(Metadata)

제안된 구조에서 키워드 기반 인덱스의 관리를 CS가 아닌 TA에서 하도록 한다. 따라서 소유자가 전송한 메타데이터는 Table 1과 같이 해당 파일의 id($H(F_j)$)와 크기($|F_j|$), 키워드의 id($H(w_{j,h})$, $h=1,...,s$)와 암호화된 결과($\hat{w}_{i,h}=\left(E_{a'},E_o\right)$, $h=1,...,s$), 빈도수($f_{w,h}$, $h=1,...,s$)로 구성된다.

Table 1. Metadata Format

File id		$H(F_j)$		
File size		$\|F_j\|$		
Keyword	id	$H(w_{j,1})$	···	$H(w_{j,s})$
	Encrypted values	$\hat{w}_{i,1}$	···	$\hat{w}_{i,s}$
	Frequency	000	···	000

여기에서 데이터 파일을 해쉬 함수의 입력으로 한 결과를 파일 id로 사용한 이유는 불법적으로 데이터를 위/변조하거나 소유자의 정당한 변경 요청에 대한 서버의 불이행 등을 감사(auditing)하기 위함이다. 따라서 서버에 저장된 파일의 내용이 변경되면 파일 id는 다른 값으로 변경된다. 파일의 크기는 파일내의 포함된 워드들의 수이다. 그리고 키워드 id를 $H(w_{j,h})$로 사용하는 이유는 암호화된 키워드들을 일정한 규칙이나 그룹으로 관리하기 위함이다. 그렇지 않은 경우에는 사용자의 트랩도어내의 키워드들과 일치하는 암호화된 키워드들을 찾는데 시간이 많이 길어진다. 암호화된 결과($\hat{w}_{i,h}$)은 3.1절에서 설명한 것과 같다. 빈도수는 키워드가 해당된 파일 내에서 반복되어 나타나는 횟수이다.

4.3 키워드 인덱스와 관련성 점수(Relevance Score)

TA는 데이터 사용자가 트랩도어 내에 기술한 키워드를 포함하고 있는 파일들을 효율적으로 찾기 위해 소유자들로부터 받은 메타데이터에 대한 키워드 인덱스를 일정한 규칙에 의해 구성한다. 파일 F_j에 대한 키워드 인덱스는 키워드 id($H(w_{j,h})$)와 암호화 결과($\hat{w}_{i,h}$), 파일 id($H(F_j)$)와 크기($|F_j|$), 키워드 빈도수($f_{w,h}$)로 구성된다. 여기에서 $h=1,...,s$이다.

관련성 점수 계산은 통신과 사용자의 서비스 비용을 줄이기 위한 것으로, 트랩도어 내의 키워드들과 관련이 있는 파일들 중에서 가장 관련성 점수가 높은 k개의 파일을 선택하는데 사용된다. 관련성 점수를 계산하는 수식은 Equation (2)와 같다[3].

$$r_Score(w_{j,i}, id(F_j)) = \sum_{w_i \in d} \frac{1}{|F_j|} \cdot \left(1+\ln f_{w_{j,i}}\right) \cdot \ln\left(1+\frac{N}{F^n_{w_{j,i}}}\right) \quad (2)$$

여기에서 d는 트랩도어 내의 키워드 집합니다.

4.4 키워드 기반 검색 프로토콜

CS에 의해 파일들 간의 관련성 추론의 가능성을 최소화하고 CS의 과중한 계산으로 인해 발생하는 응답 지연을 줄일 수 있는 프로토콜을 기술한다. 키워드 인덱스를 관리하고 키워드에 관련된 파일들을 찾는 노드가 CS가 아닌 TA라는 것이 Zhang[6] 구조와 차이가 있다.

1) 준비 단계

소유자($Owner$)는 데이터 파일(F_j)에 대한 식별자와 데이터를 비밀키로 암호화한 후에 [절차1]과 같이 CS에게 전송하고, 파일에 대한 메타데이터를 비밀키로 암호화한 후에 [절차2]와 같이 신뢰된 에이전트(TA)에게 전송한다.

[절차1] $Owner \rightarrow CS$: $\left[H(F_j), [Data(F_j)]_{K_O}\right]_{K_{O,CS}}$

여기에서 $H(F_j)$는 파일의 식별자로서 전체 데이터의 해쉬 값을 취한 것이고, $\left[Data(F_j)\right]_{K_O}$는 소유자 비밀키($K_O$)로 암호화된 데이터이다. CS는 수신한 파일을 자신의 스토리지에 저장하고, 파일 기반 인덱스에 해당 식별자를 추가한다.

[절차2] $Owner \rightarrow TA$:

$$\left[H(F_j),\ |F_j|,\ \left\{id_{keyword},\ w'_{j,h},\ f_{w_{j,h}}\right\}\right]_{K_{O,TA}}$$
$$,h = 1,\ldots,s$$

$id_{keyword}$는 키워드의 id로서 실제 키워드($w_{j,h}$)에 대한 해쉬 값($H(w_{j,h})$)이다. $w'_{j,h}$는 3.1절에서 설명한 것과 같은 것으로서 아래 Equation (3)에 의해서 계산된다.

TA는 수신한 메타데이터 내의 키워드 id를 기반으로 하는 키워드 인덱스를 구성하여 관리한다.

$$w'_{j,h} = \left(g^{k_{j,w} \cdot r_o \cdot H(w_{j,h})},\ g^{k_{j,w} \cdot r_o}\right) \tag{3}$$

여기에서 $k_{j,w}$는 소유자의 개인키이고 r_o는 소유자가 생성한 난수이다.

2) 검색 단계

사용자($User$)는 원하는 데이터를 다운로드하기 위해 트랩도어를 생성한 후에 [절차3]과 같이 TA에게 전송한다.

[절차3] $User \rightarrow TA$: T'_{w_u}

여기에서 T'_{w_u}는 3.1절에서 설명한 것과 같은 것으로써 아래 Equation (4)에 의해서 계산한다.

$$T'_{w_u} = \left(g^{H(w_u) \cdot r_u},\ g^{r_u}\right),\ \ u = 1,\ldots,d \tag{4}$$

w_u는 사용자가 찾고자 하는 파일의 키워드이고 r_u는 사용자가 임시로 생성한 난수이다.

TA는 트랩도어 T'_{w_u}($T_{u1} = g^{H(w_u) \cdot r_u}$, $T_{u2} = g^{r_u}$)를 수신하면 사용자가 접근권한이 있는 정당한 사용자가 아닌 경우는 무시하고, 그렇지 않는 경우에는 트랩도어와 키워드 인덱스를 검색하여 Equation (5)를 만족하는 파일들의 식별자를 찾는다. 그리고 이들 식별자들 중에서 Equation (1)을 적용하여 가장 관련성 점수가 높은 최대 k개의 파일 id들로 구성된 새로운 질의를 생성한 후에 [절차4]와 같이 CS에게 안전한 채널을 통해 전송한다.

[절차4] $TA \rightarrow CS$: $\left[H(F_j)\right]_{K_{TA,CS}}$, $j = 1,\ldots,k$

$$\begin{aligned} e(E_{o1},\ T_{u2}) &= e\left(g^{k_{i,w} \cdot r_o \cdot H(w_{i,h})},\ g^{r_u}\right) \\ &= e(g,\ g)^{k_{i,w} \cdot r_o \cdot H(w_{i,h}) \cdot r_u} \\ &= e(E_{o2},\ T_{u1}) \end{aligned} \tag{5}$$

TA는 CS에게 파일 식별자들을 CS와 공유하는 비밀키로 암호화하여 [절차5]와 같이 $User$에게 전송한다.

[절차5] $CS \rightarrow User$: $\left\{\left[Data(F_j)\right]_{K_O},\ Owner(F_j)\right\}$,
$j = 1,\ldots,k$

CS는 $User$에게 암호화된 파일들(k)을 전송하면 각 소유자로부터 받은 비밀키(K_O)로 복호화해서 내용을 읽는다.

5. 데이터 프라이버시와 성능 분석

5.1 프라이버시 분석

1) 데이터와 키워드

데이터 파일은 아웃소싱되기 전에 비밀키로 암호화되어 있기 때문에 소유자와 정당한 사용자 이외에는 데이터에 접근할 수 없다. 그리고 데이터 파일을 해쉬 함수의 입력으로 한 결과를 파일 id로 사용하기 때문에 불법적으로 데이터를 위/변조하거나 소유자의 정당한 변경 요청에 대한 CS의 불이행 등을 감사(auditing)할 수 있다. 그리고 키워드는 비밀값을 모르는 상태에서 쌍 선형 Diffie-Hellman 문제를 다항시간 내에 계산하는 것은 어렵다고 알려져 있기 때문에 대칭형 암호화 알고리즘보다 강한 비밀성이 보호된다.

2) 관련성 노출

Zhang 등[6]이 제안한 구조에서는 소유자로부터 수신한 암호화된 메타데이터를 중간 노드인 관리 서버가 재암호화한 후에 CS에게 전송하기 때문에 CS는 파일의 실제 크기와 키워드의 빈도수를 알 수 없지만 사용자의 파일 요청 메시지를 처리하는 과정에서 특정 키워드와 관련이 있는 파일들을 알 수 있을 뿐만 아니라 관련된 파일들이 어떤 소유자와 관련이 있는지를 추론할 수 있다. 특히 단일 키워드를 사용해서 관리하는 경우에 관련성 추론은 더 쉬워진다. 하지만 본 논문에서 제안한 구조에서는 소유자가 아웃소싱될 파일에 관련된 메타데이터(파일 id와 파일 크기, 키워드의 id와 암호화된 값, 빈도수)를 안전한 채널을 통해 TA에게 전송하면 TA는 키워드 인덱스를 구성하고 관리하는 구조이다. 따라서 제안된 구조는 키워드 기반 검색을 신뢰할 수 없는 CS가 아닌 신뢰된 TA에서 수행하기 때문에 CS는 키워드들 간의 관련성 추론은 더 어려워진다. 그리고 TA는 파일들 간의 관련성을 간접적으로 추론이 가능하지만 실제 키워드를 알아내기 것은 거의 불가능하다.

5.2 성능 분석

실제 시뮬레이션 환경은 3.0GHz에서 작동하는 듀얼 Intel Xeon CPU 및 4GB 메모리가 장착된 PC에서 구현하였다.

2,000개의 각 파일에 대해 3~7개씩의 키워드를 랜덤하게 생성하는 방식으로 총 10,000개를 생성했다. 여기에서 키워드는 5개 분야의 전문학회지에 게재된 논문에서 추출하였다. 트랩도어에 사용되는 키워드의 수는 1~8개로하고, 옵션 값(k)은 7~10으로 하였다.

측정 항목은 소유자로부터 수신한 메타데이터들에 대한 인덱스를 구성하는 시간과 사용자의 트랩도어의 요청에 대해 키워드를 기반으로 데이터를 검색을 한 후에 그 응답을 받을 때까지 소요되는 시간을 측정하였다.

PRMSM의 구조에서 소유자 데이터 파일들에 대한 메타데이터를 수신한 관리 서버(PRMSM_A)는 키워드들을 재암호화하는데 적은 처리 시간이 소요되고, 서버(PRMSM_S)는 같은 키워드들을 일정한 규칙에 의해 분류할 수 있는 정보가 없기 때문에 부가적인 처리과정 없이 인덱스에 입력되는 순서대로 저장된다. 따라서 키워드의 수가 증가하더라도 인덱스를 구성하는데 소요되는 시간은 일정하다. 하지만 제안된 구조는 소유자들로부터 파일에 대한 메타데이터를 수신한 TA (Ours_TA)가 키워드 기반 인덱스를 일정한 규칙에 의해서 구성되기 때문에 상대적으로 더 많은 처리 시간이 소요된다. Fig. 2는 파일의 수가 2,000개로 고정된 경우 키워드의 수가 증가할수록 키워드 인덱스를 구성하는데 소요되는 시간이 증가함을 알 수 있다. 이러한 결과는 소유자가 데이터를 아웃소싱하기 위한 준비단계에서 발생하는 계산 비용이기 때문에 큰 부담은 되지 않는다고 판단된다.

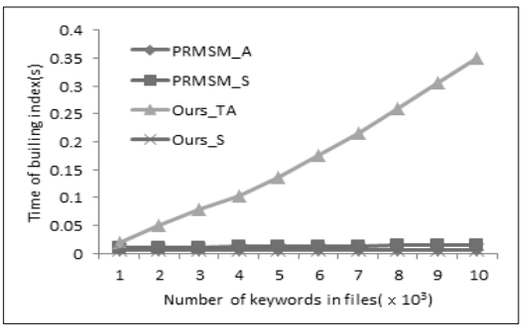

Fig. 2. Time Cost of Keyword Index Generation

Fig. 3은 사용자와 중간 노드에서 트랩도어를 재구성하는데 소요되는 시간을 나타낸 것이다. PRMSM 구조에서 사용자는 원하는 데이터를 다운로드하기 위해 키워드들을 암호화한 후에 관리 서버에게 전송하고, 관리 서버는 다시 재암호화해서 CS에게 전송한다. 반면에 제안된 구조에서는 TA는 재암호화하지 않고 자신이 관리하고 있는 키워드 인덱스에서 트랩도어 내의 키워드들과 관련성이 가장 높은 파일들의 식별자를 찾은 다음에 파일 기반 질의를 생성해서 CS에게 전송한다. 따라서 실제 실험을 통해 중간 노드의 처리 시간을 알아본 결과는 PRMSM 구조가 제안된 구조보다 더 적은 시간이 소요됨이 확인되었다.

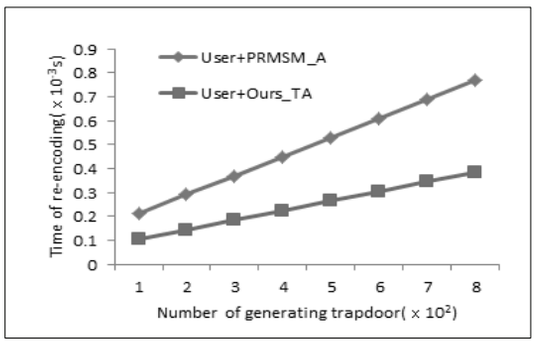

Fig. 3. Time Cost of Trapdoor Generation

Fig. 4는 100명의 데이터 사용자가 800개의 트랩도어를 생성하여 CS에게 키워드 기반 검색을 요청한 경우에 노드별로 소요되는 시간을 나타낸 것이다. PRMSM_S와 Ours_TA에서 비교적 많은 시간이 소요되지만 두 노드 간에 차이가 현저함을 알 수 있다. 이러한 결과를 보인 이유는 PRMSM_S에서 비교적 많은 시간이 소비되는 쌍 선형 페어링 연산을 키워드 인덱스에 저장된 전체 키워드의 수에 비례하여 증가하지만, 제안된 구조에서는 키워드별로 그룹화되어 있어서 TA에서 페어링 연산을 수행하는 횟수가 상대적으로 적기 때문이다. Fig. 5는 PRMSM 구조와 제안된 구조에서 소요되는 전체 처리 시간을 나타낸 것으로 제안된 구조가 훨씬 더 효율적임을 보여 주고 있다.

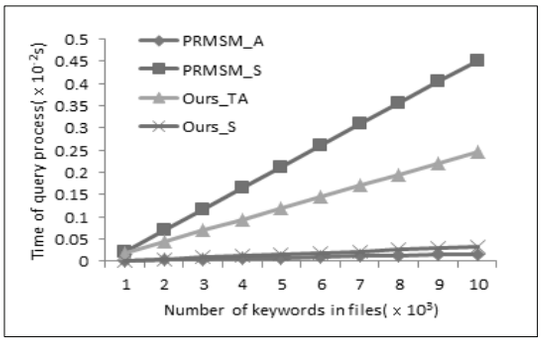

Fig. 4. Processing Time of Queries Per Node

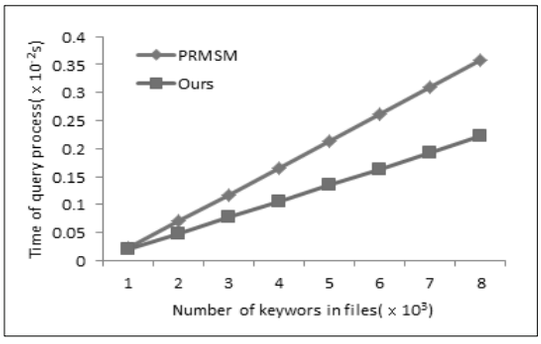

Fig. 5. The Total Time of Processing Queries

6. 결 론

본 논문은 다수의 데이터 소유자와 사용자로 구성된 클라우드 컴퓨팅 환경에서 CS를 전적으로 신뢰할 수 없는 노드라는 가정 하에서 아웃소싱된 데이터의 프라이버시가 보호되는 멀티 키워드 기반 검색 프로토콜을 제안하였다. 제안된 구조에서 데이터 파일은 아웃소싱되기 전에 비밀키로 암호화되어 있기 때문에 정당한 노드이외는 접근할 수 없다. 또한 해쉬 함수를 사용하여 파일 id를 생성함으로써 불법적으로 데이터를 위/변조하거나 소유자의 정당한 변경 요청에 대한 클라우드 서버의 불이행 등을 간단하게 감사할 수 있다. 그리고 신뢰성 없는 CS가 사용자가 요청한 파일을 검색하는 과정에서 불법적으로 시도할 수 있는 파일간의 관련성 추론을 방지하기 위해 TA로 하여금 키워드 기반 검색을 수행하도록 하였다.

그리고 CS는 트랩도어 내의 키워드들이 암호화된 키워드 인덱스에 존재하는지를 검사하기 위해 많은 시간이 소비되는 쌍 선형 페어링 연산을 인덱스의 크기만큼 반복적으로 수행하기 때문에 사용자 요청에 대한 응답이 늦어지는 문제가 발생하였다. 이런 반복적인 연산의 횟수를 줄이기 위해 TA로 하여금 키워드 id를 기반으로 하는 키워드 인덱스를 생성하여 관리하도록 하는 구조를 제안하였다.

지금까지 제안된 프로토콜에 대한 프라이버시의 보호 측면과 성능 측면에서 비교분석해 본 결과를 살펴보았듯이 제안된 구조가 클라우드 컴퓨팅 환경에 더 적합함이 증명되었다. 향후 연구과제는 제안된 알고리즘을 실제 환경에 구현하는 것이다.

References

[1] W. Sun, W. Lou, Y. T. Hou, and H. Li, "Privacy-Preserving Keyword Search Over Encrypted Data in Cloud Computing," *Secure Cloud Computing*, Springer New York, pp.189-212, 2014.

[2] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving Multi-keyword text Search in the Cloud Supporting Similarity-based Ranking," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ACM, pp.71-81, 2013.

[3] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, Vol.23, Issue 8, pp.1467-1479, 2012.

[4] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," *Journal of Computer Security*, Vol.19, No.5, pp.895-934, 2011.

[5] L. Chen, X. Sun, Z. Xia, and Q. Liu, "An Efficient and Privacy Semantic Multi-Keyword Ranked Search over Encrypted Cloud Data," *International Journal of Security and Its Application*, Vol.8, No.2, pp.323-332, 2014.

[6] W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, "Privacy Preserving Ranked Multi-Keyword Search for Multiple Data Owners in Cloud Computing," *IEEE Transactions on Computers*, Vol.65, No.5, pp.1566-1577, 2016.

[7] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," in *Proc. of IEEE Symposium on Security and Privacy 2000*, pp.44-55, 2000.

[8] E. J. Goh, "Secure Indexes," Cryptology ePrint Archive, Report 2003/216, http://eprint.iacr.org/, 2003.

[9] B. Bloom, "Space/time Trade-offs in Hash Coding with Allowable Errors," in *Communications of the ACM*, Vol.13, Issue 7, pp.422-426, 1970.

[10] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Advances in CryptologyASIACERYPT 2001*, Ed., Springer, pp.514-532, 2001.

김 태 연

e-mail : kimcopper@naver.com
1988년 전남대학교 계산통계학과(석사)
1996년 전남대학교 전산통계학과(박사)
1996년~현 재 서남대학교 컴퓨터정보학과 조교수
관심분야 : 클라우드 컴퓨팅, 모바일 컴퓨팅, 네트워크 보안, 센서 네트워크

조 기 환

e-mail : ghcho@chonbuk.ac.kr
1987년 서울대학교 계산통계학과(석사)
1996년 영국 뉴캐슬대학교 전산학과 (박사)
1997년~1999년 목포대학교 컴퓨터과학과 전임강사
1999년~현 재 전북대학교 컴퓨터공학부 부교수
관심분야 : 모바일 컴퓨팅, 센서 네트워크, 네트워크 보안, 분산컴퓨팅시스템

이 영 록

e-mail : dogu@jnu.ac.kr
2003년 전남대학교 전산통계학과(박사)
2003년~현 재 전남대학교
시스템보안연구센터 연구교수
2007년~현 재 전남대학교
정보보안협동과정 초빙교수
관심분야 : 클라우드 컴퓨팅, 네트워크 보안, 빅 데이터,
임베디드 시스템, 웹 보안, 모바일 컴퓨팅