Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.6, Dec. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.6.1077

트랩도어 센터로부터 보호받는 순위 검색 가능한 암호화 다중 지원 클라우드 컴퓨팅 보안 모델*

김 예 은,[1†] 오 희 국[2‡]
[1,2]한양대학교 (대학원생, 교수)

Protecting Multi Ranked Searchable Encryption in Cloud Computing from Honest-but-Curious Trapdoor Generating Center*

YeEun Kim,[1†] Heekuck Oh[2‡]
[1,2]Hanyang University (Graduate student, Professor)

요 약

검색 가능한 암호화 모델은 원격 서버에 저장한 암호화된 데이터를 선택적으로 검색할 수 있는 모델이다. 현실 시나리오에 적용했을 때 다중 검색 키워드, 다중 데이터 소유자(업로더)와 다중 데이터 사용자(다운로더)를 지원할 수 있어야 하며, 검색이 이루어졌을 때 요청한 내용과 가장 관련성이 높은 순으로 결과를 낼 수 있어야 한다. 본 논문에서는 이러한 모델을 순위 검색이 가능한 암호화 다중 지원 모델이라고 칭한다. 그러나 본 논문이 작성된 시점까지 제안된 모델은 완전 신뢰 가능한(fully-trusted) 트랩도어 센터를 암호 생성에 사용하고 있으며, 일부는 데이터 다운로더와 트랩도어 센터 간의 연결이 안전하다고 가정한다. 하지만 실생활에서 이러한 가정이 지켜질 확률은 낮다. 따라서 본 논문은 이러한 검색 가능한 암호화 모델의 실용성과 보안성을 개선하기 위하여, 새로운 순위 검색이 가능한 암호화 다중 지원 모델을 제안한다. 해당 모델은 기존 연구의 가정이 없이도, 데이터 다운로더가 요청한 검색어를 외부 공격자와 정직하지만 동시에 호기심이 있는(honest-but-curious) 트랩도어 센터로부터 보호한다. 공격자는 서로 다른 두 검색 요청이 같은 검색어를 포함하고 있는지 구별할 수 없다. 또한, 보호 과정을 추가함으로써 발생하는 오버헤드를 고려하더라도 제안하는 모델이 합리적인 성능을 달성함을, 실험을 통해 증명한다.

ABSTRACT

The searchable encryption model allows to selectively search for encrypted data stored on a remote server. In a real-world scenarios, the model must be able to support multiple search keywords, multiple data owners/users. In this paper, these models are referred to as Multi Ranked Searchable Encryption model. However, at the time this paper was written, the proposed models use fully-trusted trapdoor centers, some of which assume that the connection between the user and the trapdoor center is secure, which is unlikely that such assumptions will be kept in real life. In order to improve the practicality and security of these searchable encryption models, this paper proposes a new Multi Ranked Searchable Encryption model which uses random keywords to protect search words requested by the data downloader from an honest-but-curious trapdoor center with an external attacker without the assumptions. The attacker cannot distinguish whether two different search requests contain the same search keywords. In addition, experiments demonstrate that the proposed model achieves reasonable performance, even considering the overhead caused by adding this protection process.

Keywords: Searchable encryption, Multi-keyword search, Multiple data owners/users, Enhanced security

Received(10. 18. 2023), Modified(11. 28. 2023), Accepted(11. 28. 2023)

* 본 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임. (NRF-2022R1A2C2007255).

† 주저자, gikosei@hanyang.ac.kr
‡ 교신저자, hkoh@hanyang.ac.kr(Corresponding author)

I. 서 론

클라우드 저장소 서비스가 널리 보급됨에 따라, 이러한 원격 서버에 개인 이메일, 금융 데이터 또는 개인 건강 기록과 같은 민감한 정보를 관리하고자 하는 사용자에게 데이터 프라이버시와 같은 문제는 매우 중요한 사안이다. 데이터 업로더는 자신이 사용하는 클라우드 서버가 외부 공격자뿐만 아니라 클라우드 서버 자체로부터 안전하도록 보장받고자 하는데, 데이터가 아웃소싱되면 데이터 업로더로부터 클라우드 서버로 직접적인 제어권이 이전되기 때문이다. 이 문제를 해결하기 위해 데이터 업로더는 클라우드 서버에 저장하기 전에 민감한 정보를 암호화하여 데이터 프라이버시를 보호할 수 있다[1-3].

그러나, 이러한 접근법은 키워드 검색과 같은 대부분의 서비스가 평문 시나리오로 제공되기 때문에 데이터 사용성 측면에서 큰 비용을 초래한다. 따라서, 암호화된 데이터에 대한 키워드 검색을 수행하기 위한 검색 가능한 암호화, SE(Searchable Encryption) 기법이 제안되었다. 그 후, 이 기법은 많은 다른 연구자들에 의해 더욱 발전되었는데, 기능을 추가한다거나, 효율성을 높인다거나, 보안성을 증가하는 연구들이 현재도 진행되고 있다.

이러한 SE 모델 중 실제 상황에 가장 가까운 모델은 다중 키워드로 검색이 가능하고, 요청된 검색 키워드에 가장 인접한 결과를 인접도 순으로 반환하며 다중 데이터 업로더 및 다중 데이터 다운로더를 지원할 수 있는 모델일 것이다. 본 논문에서는 이러한 모델을 순위 검색이 가능한 암호화 다중 지원 모델이라고 칭한다. 그러나 현재 시점에서 안전한 순위 검색이 가능한 암호화 다중 지원 모델에 대한 연구는 거의 수행되지 않았다.

키워드 검색을 요청하기 위해, 데이터 다운로더는 먼저 클라우드 서버가 암호화된 데이터 검색을 위해 사용할 트랩도어를 생성해야 한다. 만일 데이터 업로더와 데이터 다운로더가 각각 한 명이라면, 다운로더가 업로더에게 직접 트랩도어를 요청할 수 있으나 다중 지원 모델에서는 다운로더가 관계된 모든 업로더에게 트랩도어를 요청해야 하기 때문에, 막대한 통신 오버헤드가 발생하며 접근법의 현실성이 떨어지게 된다.

따라서 기존의 다중 지원 모델은 시스템 모델 내의 다른 개체를 이용하여 트랩도어를 생성하였다. 이 개체를 트랩도어 센터라고 하며, 이는 시스템에서 완전 신뢰 가능한(fully-trusted) 개체로 가정된다. 또한 사용자와 트랩도어 센터의 연결은 주로 안전하기 때문에 외부 공격자가 그 내용을 도청할 수 없다고 가정된다. 즉, 완전 신뢰할 수 있는 트랩도어 센터만이 검색 요청 내의 키워드에 관한 정보를 학습할 수 있지만 자신에게 맡겨진 임무 외의 다른 쪽으로 이 정보를 사용하려고 시도하지는 않을 것이라고 가정하는 것이다[4-6]. 그러나 실제로 이 가정은 성립하기 어렵다[7]. 따라서 트랩도어 센터로부터도 검색 키워드 프라이버시를 확보할 수 있는 다중 지원 모델에 대한 연구가 필요하다.

본 연구에서는 트랩도어 생성 센터 자체를 포함한 공격자들로부터 검색 키워드 프라이버시를 보호할 수 있는, 정직하지만 동시에 호기심이 많은(honest-but-curious) 트랩도어 센터를 이용한 새로운 순위 검색이 가능한 암호화 다중 지원 모델을 제안한다. 해당 모델에는 공격자들로부터 검색 키워드 프라이버시를 보호하기 위한 추가 프로세스가 포함되어 있으며, 이로 인해 검색 수행 중 오버헤드가 증가한다. 과도한 오버헤드는 실용성을 감소시키므로, 이러한 오버헤드를 가능한 한 줄이는 것이 본 연구의 목표이다.

II. 문제 정의

2.1 시스템 모델

시스템 모델의 구조는 중앙 권한 에이전시(Authority Agency), 데이터 업로더(Data Owner), 데이터 다운로더(Data User), 두 개의 프록시 서버 PS1과 PS2, 클라우드 서버 등 6개의 엔티티로 이루어져 있다. Fig. 1은 시스템 모델을

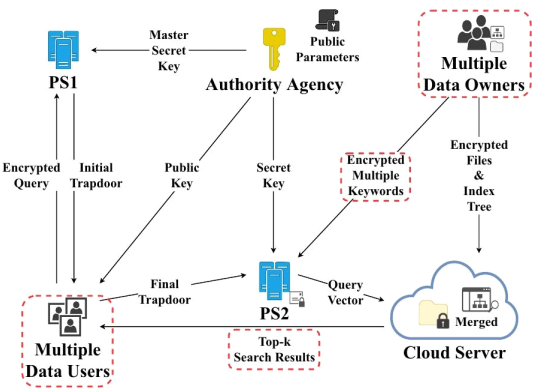


Fig. 1. System model.

그림으로 표현한 것이다.

- **중앙 권한 에이전시**는 완전히 신뢰할 수 있는 엔티티 역할을 하며 다른 엔티티에서 사용하는 공개 파라미터와 공개/비공개 키 쌍을 생성한다.
- 각 **데이터 업로더**는 개인의 데이터 및 데이터 내용에 관련된 키워드 암호화 작업을 수행하고, GBB-Tree(Grouped Keyword-Balanced Binary Tree)를 구성하며, 암호화된 데이터 및 GBB-Tree/암호화된 키워드를 클라우드 서버/PS2로 각각 전송한다.
- **클라우드 서버**는 모든 데이터 소유자로부터 각각의 GBB-트리를 수집하고 그들을 병합해 하나의 GBB-트리를 구축한다. 이 GBB-트리를 이용하여 클라우드 서버는 검색을 수행할 때 상위 k위의 검색 결과를 다운로더에게 반환할 수 있다.
- PS1은 초기 트랩도어의 제작을 총괄하는 프록시 서버다. PS1은 데이터 다운로더로부터 검색할, 암호화된 키워드를 수신한 후, 각 키워드에 대한 초기 트랩도어 세트를 생성하고, 그 세트를 다운로더에게 반환한다.
- PS2는 검색 벡터의 제작을 담당하는 또 다른 프록시 서버다. 데이터 다운로더가 자신이 제작한 최종 트랩도어를 보내면 PS2는 해당 트랩도어를 검색 벡터로 변환해 클라우드 서버로 전송한다.
- **데이터 다운로더**는 다중 키워드 쿼리를 생성하고 클라우드 서버에서 검색 결과를 가져온다. 다운로더는 먼저 자신이 검색할 키워드를 암호화해 PS1에 전송하고, 반환된 결과를 이용해 최종 트랩도어를 생성하고 PS2에 제출한다.
-

본 모델은 시스템 검색 키워드 모음집 W가 고정되어 있으며, 이미 모든 데이터 업로더와 사전에 공유되었다고 가정한다.

2.2 공격 모델

본 논문에서는 1) 중앙 권한 에이전시가 어떤 공격자와도 공모하지 않으며 2) 데이터 업로더가 자신의 데이터를 정직하게 공유하는 것으로 가정한다. 또한 중앙 에이전시와 마찬가지로 그 어떤 서버도 서로를 포함한 그 어떤 공격자와도 결탁하지 않는 것으로 간주한다. 하지만 이들은 정직하지만 동시에 호기심이 있기 때문에, 주어진 일을 정직하게 수행하면서도 자신들이 받은 검색 키워드와 트랩도어에 대한 정보를 빼내고 싶어 한다고 가정한다.

클라우드 서버는 known ciphertext model로 가정하는데, 이는 저장된 데이터에 대한 내용 키워드 빈도 통계와 같이 데이터 업로더/다운로더가 제공하는 지식 외에 다른 지식이 없다는 것을 의미한다.

데이터 업로더는 유효한 키워드만 요청한다고 가정한다.

외부 공격자는 본 시스템 내의 유효한 멤버(데이터 업로더/다운로더)가 아닌 공격자로 정의하며, 유효한 다운로더의 키워드 검색 절차 중에 키워드 쿼리, 초기/최종 트랩도어 또는 검색 벡터를 캡처할 수 있다고 가정한다. 마지막으로, 모든 공격자는 어려운 계산 문제를 다항(polynomial) 시간 내에 해결할 수 없는 제한된 계산 능력을 가지고 있다고 가정한다. 우리는 다양한 엔티티의 관점에서 공격 모델을 다음과 같이 고려한다:

- PS1: PS1은 데이터 다운로더들 및 그들의 비밀 키들을 사용해 암호화된 키워드들이 주어지면, 키워드의 평문을 복구하려고 시도하거나 또는 두 개의 상이한 키워드들이 동일한 내용을 포함하는지 학습을 시도할 수 있다.
- PS2: PS2는 데이터 다운로더의 트랩도어와 데이터 업로더의 암호화된 키워드가 주어지면, 서로 다른 두 암호문/트랩도어가 동일한 키워드를 포함하고 있는지를 밝혀내기 위해 자신이 받은 트랩도어에 키워드 추측 공격 또는 동등성(equivalence) 테스트 공격을 시도할 수 있다. 또한, PS2는 데이터 다운로더로부터 해당 트랩도어를 획득하지 않은 상태로 어떤 암호문이 어떤 키워드를 암호화했는지에 대한 학습을 스스로 시도할 수 있다.
- **외부 공격자**: 외부 공격자는 자신이 캡처한 정보를 이용하여 서버의 비밀 키, 키워드 평문 또는 키워드 접근 패턴을 추측하려고 시도할 수 있다.

2.3 시스템 프레임워크

본 모델의 시스템은 설정 알고리즘 ***Setup***, PS2 서버 키 생성 알고리즘 ***p2KeyGen***, 키워드 암호화 알고리즘 ***KeywordEnc***, 키워드 인덱스 구축 알고리즘 ***IndexBuild***, 다중 인덱스 병합 알고리즘 ***IndexMerge***, 트랩도어 생성 알고리즘 ***Trapdoor***, 검색 알고리즘 ***Search***의 7가지 알고리즘으로 구성된다.

1. ***Setup*** $(1^{l}\rightarrow(msk,pub))$: 중앙 권한 에이전시가 수행한다. 시스템 보안 파라미터 l를 입력받아 마스터 비밀 키 ***msk***와 공개 파라미터 ***pub***를 출력한다.

2. ***p2KeyGen*** $(pub\rightarrow(pk_{p2},sk_{p2}))$: 중앙 에이전시는 ***pub***을 사용하여 PS2의 공개/개인 키 쌍 pk_{p2},sk_{p2}를 생성하고 출력한다.

3. ***KeywordEnc*** $((pub,W,W_i)\rightarrow\overline{W_i})$: i번째 데이터 업로더 DO_i가 수행한다. 공개 파라미터 ***pub***, 시스템 키워드 모음집 W, 그리고 데이터 업로더의 키워드 모음집을 입력으로 받은 알고리즘은 암호화된 업로더의 키워드 모음집을 출력한다.

4. ***IndexBuild*** $((F_i,ID_i)\rightarrow I_i)$: 각 데이터 업로더 DO_i는 해당 알고리즘을 수행하여 자신의 GBB-트리를 구축한다. 알고리즘은 데이터 업로더의 노드 리스트 $NodeList_i$와 인증정보 ID_i에 따라 데이터 업로더의 키워드 인덱스 트리 I_i를 구축하고 출력한다.

5. ***IndexMerge*** $((I,ID_s)\rightarrow I_{merged})$: 클라우드 서버가 사용한다. 알고리즘은 모든 데이터 업로더들이 각각 구축한 인덱스 트리를 받고, 자신의 인증정보 I_s를 이용해 하나의 통합 인덱스 트리 I_{merged}를 구축한다.

6. ***Trapdoor*** $((msk,pub,pk_{p2},Q)\rightarrow T)$: PS1과 데이터 다운로더가 같이 진행하는 알고리즘이다. 알고리즘은 다운로더가 검색할 키워드 세트 Q에 대한 트랩도어 T를 생성한다.

7. ***Search*** $((pub,\overline{W},T,sk_{p2},k)\rightarrow R)$: PS2와 클라우드 서버 둘이서 같이 진행하는 알고리즘이다. 알고리즘은 검색 요청으로 들어온 키워드 내용과 가장 근접한 k개의 결과 R을 출력한다.

2.4 보안 목표

제안된 모델은 다음과 같은 목표를 달성해야 한다:

- **키워드 보안**: 공격자가 암호문이나 트랩도어에 내포된 키워드에 대한 정보를 학습하는 것을 방지해야 하며, 암호화된 키워드와 트랩도어에 숨겨진 동일한 키워드를 구분할 수 없도록 해야 한다. 즉, 이 방식은 선택 키워드(chosen keyword) 공격과 동등성 테스트 공격 모두에 안전해야 한다.

III. 검색 과정

3.1 Setup

중앙 권한 에이전시는 주어진 보안 파라미터 l을 사용하여 공용 파라미터 ***pub***와 마스터 비밀키 ***msk***를 생성한다, 알고리즘은 쌍선형군(Bilinear Group) (G,G_T)를 선정하고 이들의 소수 차수 p, 생성자 g, 쌍선형군 상의 함수 e를 고른다.

그 후 G,G_T 상의 랜덤한 숫자 u, v, G,G_T 상의 해쉬 함수 H_p,H_T, 마지막으로 p보다 작은 정수 중 랜덤한 수인 ϕ,k_1,k_2,k_3,k_4를 이용하여 ***msk***와 ***pub***를 생성한다. 둘을 수식으로 표현하면 다음과 같다.

$$pub\leftarrow[H_p,H_T,\Phi=e(g,g)^{\phi k_1 k_2},g,u,v,\ p1=g^{k1},p2=g^{k2},p3=g^{k3},p4=g^{k4}] \quad (1)$$

$$msk\leftarrow[\phi,k_1,k_2,k_3,k_4] \quad (2)$$

이때 ***msk***는 PS1에게, ***pub***은 시스템 전체의 엔티티들에게 공개된다.

3.2 p2KeyGen

중앙 에이전시는 이어 PS2를 위한 공개/비밀키를 생성한다. p보다 작은 양의 정수 κ를 선정해, 생성되는 (pk_{p2},sk_{p2})는 다음과 같다.

$$(pk_{p2},sk_{p2})=(g^{\kappa},\kappa) \quad (3)$$

생성된 공개/비밀키는 PS2에게 전달된다.

3.3 KeywordEnc

각 데이터 업로더 i는 시스템 공통 키워드 모음집 W를 암호화하여 PS2에게 전송한다. 이때 W의 a번째 키워드가 업로더의 키워드 모음집 W_i에도 존재할 경우, 암호화된 a번째 키워드 $\overline{w_{i,a}}$의 값은 다음과 같이 결정된다.

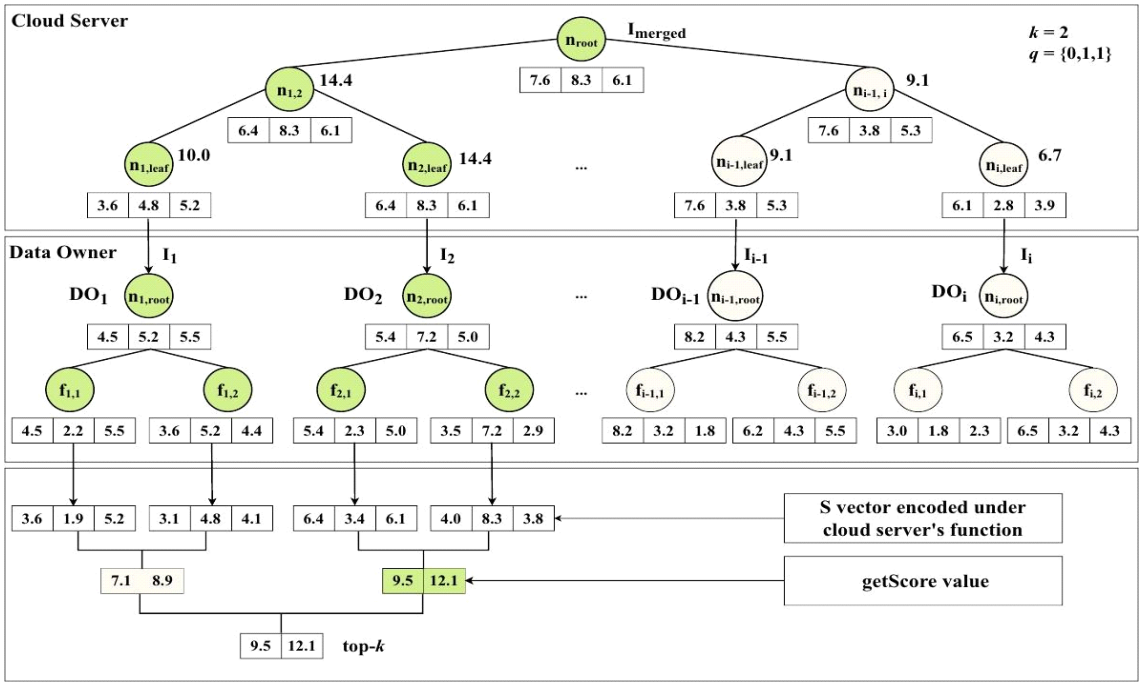


Fig. 2. I_merged GBB-Tree and calculating the top-k result.

$$\overline{w_{i,a}} = [C_i', C_{i,0}, C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}] \leftarrow [\Phi^{-o_i}, (uv^{H_p(w_{i,a})})^{o_i}, p_1^{o_i - oi,1}, p_2^{o_{i,1}}, p_3^{o_i - o_{i,2}}, p_4^{o_{i,2}}] \quad (4)$$

만일 $w_{i,a}$가 W에 존재하지 않는다면, W에 존재하지 않는 랜덤한 값 $w'_{i,a}$를 선정해 $\overline{w'_{i,a}}$로 둔다.

3.4 IndexBuild

각 데이터 업로더 i는 자신의 인증정보 ID_i를 사용하여, 자신이 업로드할 t번째 파일 $F_{i,t}$에 대한 GBB-트리 노드를 생성한다. 이때 노드에 저장되는 정보는 W에 대해 F가 가지는 TF-IDF(Term Frequency - Inverse Document Frequency) 가중치 점수이다. TF-IDF 가중치는 더하기 연산에 대해 프라이버시 정보가 보존되는 함수인 AOPPF 함수로 암호화되어 노드에 저장된다[5]. 노드들은 인접한 값을 가진 노드들끼리 그룹화되고 엮여 자신들의 부모 노드를 만들고, 최종적으로 만들어진 루트 노드를 알고리즘은 출력하게 된다. 이렇게 만들어진 자신의 인덱스 트리의 루트 노드를, 각 데이터 업로더는 클라우드 서버에게 전송한다.

3.5 IndexMerge

클라우드 서버는 각 데이터 업로더가 제출한 GBB-트리의 루트 노드를 모아, 그것을 자신이 구축할 통합 GBB-트리의 리프 노드로 삼는다. 그리고 자신의 인증정보 ID_s를 사용하여 같은 방식으로 AOPPF 암호화하고, 통합 GBB-트리 I_{merged}로 병합한다. 알고리즘의 출력물은 병합된 GBB-트리의 루트 노드이다. I_{merged}의 모습 예는 Fig. 2에서 확인할 수 있다.

3.6 Trapdoor

데이터 다운로더가 검색을 원할 때, 암호화된 문서 검색을 위해 사용될 트랩도어는 2가지 과정에 걸쳐 생성된다. 먼저 PS1이 다운로더가 제출한 암호화된 키워드를 이용해 임시 트랩도어를 계산하고 반환하면, 다운로더는 그 임시 트랩도어에 추가적인 계산을 진행하고 최종 트랩도어를 만들어 PS2에게 제출한다.

다음은 차례대로 다운로더가 암호화한 검색 키워드, 임시 트랩도어, 다운로더가 임시 트랩도어에 가하는 추가 과정, 최종 트랩도어를 표현한 수식이다.

$$\overline{Q_{w_b}} = (uv^{H_p(\widetilde{w_b})})^{\epsilon} \quad (5)$$

$$\begin{aligned} \mathrm{IT}_{\widetilde{W_b}} &= [\mathrm{T_{j,0}}, \mathrm{T_{j,1}}, \mathrm{T_{j,2}}, \mathrm{T_{j,3}}, \mathrm{T_{j,4}}, \mathrm{t_{j,1}}, \mathrm{t_{j,2}}] \\ &\leftarrow [g^{r_{j,1}k_1k_2 + r_{j,2}k_3k_4}, g^{-\phi k_2} \cdot (\overline{Q_{w_b}})^{-r_{j,1}k_2}, \\ &\mathrm{g}^{-\phi \mathrm{k}_1} \cdot (\overline{\mathrm{Q}_{\widetilde{\mathrm{W}}_\mathrm{b}}})^{-\mathrm{r}_{\mathrm{j},1}\mathrm{k}_1}, (\overline{\mathrm{Q}_{\widetilde{\mathrm{W}}_\mathrm{b}}})^{-\mathrm{r}_{\mathrm{j},2}\mathrm{k}_4}, \\ &(\overline{\mathrm{Q}_{\widetilde{\mathrm{W}}_\mathrm{b}}})^{-\mathrm{r}_{\mathrm{j},2}\mathrm{k}_3}, \mathrm{g}^{-\phi \mathrm{k}_2}, \mathrm{g}^{-\phi \mathrm{k}_1}] \end{aligned} \quad (6)$$

$$\begin{aligned} T'_{j,1} &= (T_{j,1}/t_{j,1})^{\epsilon-1} \cdot t_{j,1} \\ T'_{j,2} &= (T_{j,2}/t_{j,2})^{\epsilon-1} \cdot t_{j,2} \\ T'_{j,3} &= (T_{j,3})^{\epsilon-1} \\ T'_{j,4} &= (T_{j,4})^{\epsilon-1} \end{aligned} \quad (7)$$

$$\begin{aligned} T_{w_b} &= [T, T', T'_{j,0}, T'_{j,1}, T'_{j,2}, T'_{j,3}, T'_{j,4}] \\ &\leftarrow [\mathrm{g}^{\gamma}, \mathrm{g}^{\gamma'}, \mathrm{H_T(e(pk_{p2}, g^{\gamma'})^{\gamma})}, \\ &\mathrm{T'_{j,1}, T'_{j,2}, T'_{j,3}, T'_{j,4}}] \end{aligned} \quad (8)$$

여기서 새로이 등장한 기호들은 모두 p보다 작은 양의 정수를 무작위로 뽑은 것이다.

이때, 공개키 기반 SE 방식에서는 동등성 테스트 공격(또는 키워드 추측 공격)이 발생할 수 있는데, 공격자는 이를 통해 유효한 암호화된 키워드/트랩도어/쿼리를 저장하고 동일한 키워드가 주어진 암호문에 내포되어 있는지 여부를 판단할 수 있다[9]. 그러나 본 논문에서는 선형 분할(linear splitting) 기법을 사용하여 쌍선형군 상의 요소들을 넷으로 분할하고 그 분할된 조각 각각으로 키워드를 암호화한다[10]. 이 경우, 공격자는 간단한 연산만으로 주어진 시간 내에 내포된 키워드의 동등성을 파악할 수 없게 된다. 따라서 외부 공격자와 내부 공격자 모두로부터 키워드 프라이버시는 보호된다.

3.7 Search

검색 알고리즘은 데이터 다운로더, PS2와 클라우드 서버가 같이 참여하여 진행된다. 전 단계에서 최종 트랩도어를 완성한 다운로더는 PS2에게 그 트랩도어를 제출하고, PS2는 그 트랩도어와 자신의 비밀키, 그리고 모든 데이터 업로더가 제출한 암호화된 키워드 모음집 $\overline{W_i}$를 가지고 트랩도어에 어떤 단어들이 내포되어 있는지를 파악한다. 이는 검색 벡터를 계산하며 밝혀지는데, 이진 벡터인 검색 벡터의 a번째 값을 계산하는 법은 다음과 같다.

$$\begin{aligned} C'_i = e(C_{i,0}, \frac{T'_{j,0}}{H_T(e(T,T'))^{sk_{p2}}}) \cdot e(C_{i,1}, T'_{j,1}) \cdot \\ e(C_{j,2}, T'_{j,2}) \cdot e(C_{i,3}, T'_{j,3}) \cdot e(C_{i,4}, T'_{j,4}) \end{aligned}$$

만일 트랩도어의 b번째 원소가 키워드 모음집의 a번째 원소와 동일한 키워드로부터 계산되었다면, 해당 등식이 성립한다. 이 경우 i번째 업로더에 대한 검색 벡터 $\overline{q_i}$의 a번째 원소는 1이 된다. 등식이 성립하지 않을 경우, 값은 0이 된다. 모든 업로더에 대해 그들의 $\overline{q_i}$를 구하고 모두에 대해 합 연산을 거치고 나면 최종 검색 벡터 q가 된다. 그러면, PS2는 이 벡터를 클라우드 서버에 넘겨 GBB-트리와의 연산을 거쳐 상위 k개의 검색 결과가 다운로더에게 반환되도록 한다. 연관성 계산은 검색 벡터와 트리 노드들의 내적 연산으로 계산된다. Fig. 2에서 그 예시를 확인할 수 있다.

PS2는 검색 벡터를 계산하면서 어떤 키워드가 트랩도어 내에 내포되어 있는지를 파악할 수 있지만, PS2 자신에게는 시스템 키워드 모음집이 존재하지 않기 때문에 평문의 키워드를 알 수 없다. 클라우드 서버는 검색 벡터와 GBB-트리와의 연산을 통해 검색 결과를 알 수 있지만, 역시 시스템 키워드 모음집이 존재하지 않기 때문에 검색된 키워드를 알 수 없다. 따라서 검색 키워드의 프라이버시는 보호된다.

IV. 보안성 분석

데이터 다운로더 U_j는 자신이 원하는 검색 키워드에 대응하는 트랩도어를 만들기 위해 다음과 같은 과정을 거친다.

먼저 U_j는 암호화된 키워드 쿼리 $\overline{Q}$를 계산하여 임시 트랩도어를 계산해줄 PS1에게 제출한다. 이때 쿼리 $\overline{Q}$가 포함하는 키워드 집합의 b번째 키워드를 수식으로 표현하면 다음과 같다.

$$\overline{Q_{\widetilde{w_b}}} = (uv^{H_p(\widetilde{w_b})})^{\epsilon}$$

이때, ϵ은 무작위로 선정된 값이다. 이 단계에서, 트랩도어를 노리는 공격자들을 해당 연구는 다음과 같이 방어할 수 있다.

- **PS1**: PS1은 제출된 $\overline{Q}$를 가지고 임시 트랩도어를 계산한다. 이 단계에서, $\overline{Q}$는 그저 계산의 입력값으로 사용될 뿐이다. PS1은 마스터 비밀

키 msk를 가지고 있지만, $\overline{Q}$로부터 사용자가 무작위로 선정한 값 ϵ를 밝혀내는 일은 이산 로그 문제(Discrete Logarithm problem)로 귀결되므로 PS1은 다항식으로 표현할 수 있는 시간 내에 해당 문제를 해결할 수 없다. 또한 ϵ는 데이터 다운로더가 $\overline{Q}$를 계산할 때마다 그 값이 달라지므로, PS1은 서로 다른 $\overline{Q}$를 가지고 있을 때 둘이 서로 같은 값에서 파생되었는지를 알 수 없다.

- **외부 공격자**: 외부 공격자는 이 단계에서 $\overline{Q}$와 임시 트랩도어를 획득할 수 있다. 그러나, PS1과 달리 외부 공격자는 마스터 비밀키 msk의 상세값을 알지 못한다. 또한, 외부 공격자가 키워드를 알아내기 위해 밝혀내야 하는 무작위 선정값은 PS1가 선정한 2개($r_{j,1}$,$r_{j,2}$)가 더 존재한다. 따라서, 외부 공격자가 트랩도어나 쿼리 안에 내포된 키워드를 알기 위한 난이도는 PS1보다 더 더욱 높다.

공개키를 사용한 검색 가능한 암호화 모델은 동등성 테스트 공격을 받을 가능성이 있다. 이 공격에서 공격자는 이미 한번 사용된 키워드, 트랩도어, 또는 쿼리를 저장해 두었다가 새로운 암호문을 받았을 때 두 암호문이 서로 같은 키워드를 내포하고 있는지 판단할 수 있다[9].

그러나 본 논문에서 사용한 선형 분할기법은 쌍선형군 상의 요소들을 넷으로 분할하고 그 분할된 조각 각각으로 키워드를 암호화한다[10]. 이 경우, 공격자는 다항식으로 표현 가능한 시간 내에 내포된 키워드의 동등성을 파악할 수 없게 된다. 따라서 외부 공격자와 내부 공격자 모두로부터 키워드 프라이버시는 보호된다.

V. 실 험

성능 평가와 비교를 위해, 본 논문은 2020년에 발표된 Sun et al.의 연구를 인용한다. 오직 한 연구와 비교하는 이유는 논문 작성 시점에서 순위 검색이 가능한 다중 지원 SE 모델에 대한 연구가 많이 진행되지 않은 상태였기 때문이다.

Sun et al.의 연구와의 비교를 위해, 본 연구는 발표된 논문의 내용을 토대로 해당 연구에서 제시하는 모델을 본 논문의 모델과 동일한 환경 아래에서 직접 구현하여 동일한 조건 내에 실험을 진행하였다. 논문 작성 시점까지 Sun et al.의 연구팀은 해당 모델을 공개하지 않았다. 따라서 본 연구는 해당 모델을 직접 구현하는 방식을 선택하였다.

5.1 성능 평가

Table. 1은 Sun et al.의 모델과 본 논문 모델과의 계산 복잡도를 비교해둔 것이다[6]. 다른 알고리즘의 계산 복잡도는 거의 동일하나 IndexBuild와 IndexMerge 단계에서 본 모델이 조금 더 높은 복잡도를 보이는데, 이는 비교군 연구에서 사용한 트리와 달리 GBB-트리가 생성 과정에서 노드를 그룹화하는 추가 과정을 거치기 때문이다. 그 외로, 두 연구 다 시스템 키워드 모음집의 크기에 따라 선형적으로 복잡도가 증가하는 것을 Fig. 3, 4에서 확인할

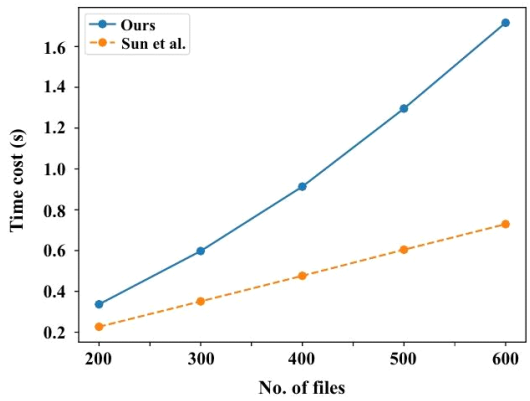


Fig. 3. Time cost of IndexBuild for different number of files(f).

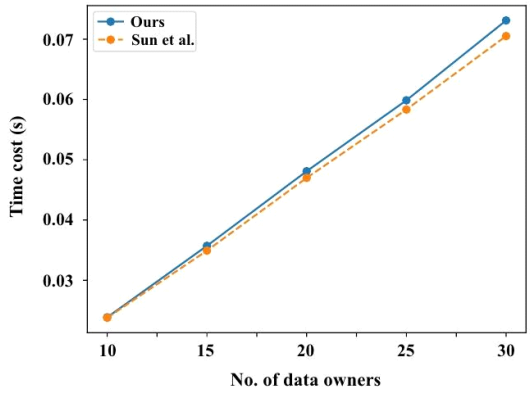


Fig. 4. Time cost of IndexMerge for different number of data owners(d).

Table 1. Complexity Comparison.

	IndexBuild	IndexMerge	Trapdoor Translate	Tree Search
Sun et al.	$O(wf)$	$O(wd)$	$O(qdw)$	$O(cw(\log(d)+\log(f)))$
Ours	$O(wf^{2})$	$O(wd^{2})$	$O(qdw)$	$O(cw(\log(d)+\log(f)))$

수 있다.

IndexBuild와 IndexMerge 단계의 시간 증가는 자칫 단점으로 보일 수 있으나, 해당 알고리즘들은 시스템이 구축되거나 시스템 키워드의 변경이 있을 때에만 드물게 실행되므로 전체적으로 보았을 때 그렇게 큰 영향을 준다고 볼 수 없다.

5.2 실험 평가

실험을 위해 본 연구는 TEXTFILES.com의 짧은 단편 소설 모음집을 데이터로 사용하였으며, 대략 760개의 영어단어 모음집을 시스템 키워드 모음집으로 삼았다[11]. 영단어 모음집을 추리는 과정은 Natural Language Toolkit을 사용하였다. 또한 TF-IDF 벡터 추출을 위해 scikit-learn 라이브러리와, nltk_data의 영단어 전처리 과정을 응용하였다[12]. 모델 구현으로는 Python 3의 pypbc 라이브러리를 사용하여 쌍선형군 페어링 암호를 구현하였다[13]. 모든 실험은 리눅스 운영체제의 가상환경에서 실행되었으며, 사용된 CPU는 Intel core i7-8700, 6 코어 3.19GHz이다. RAM 사이즈는 4GB를 사용하였다.

Fig. 5,6은 각각 검색 요청 내 키워드 수가 증가할 때, 시스템 키워드 모음집의 키워드 수가 증가할

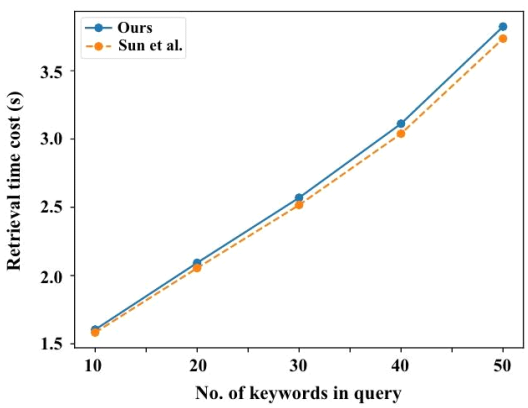


Fig. 5. Search time cost for different number of keywords in query.

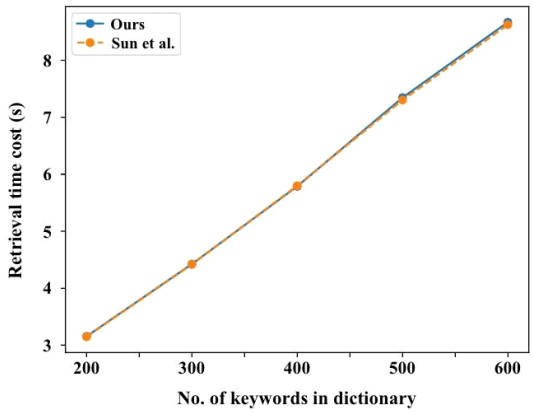


Fig. 6. Search time cost for different number of keywords in system dictionary.

때의 검색 실험 결과를 그래프로 나타낸 것이다. 그 결과, 표 1에서 분석한 대로 두 모델의 검색 시간에는 그렇게 큰 차이가 없음을 확인할 수 있다. Sun et al.의 모델은 트랩도어 생성 과정에서 키워드 보호를 고려하고 있지 않으므로, 본 연구는 더 낮은 보안 가정에서도 큰 성능의 등가 교환 없이 성능을 유지할 수 있음을 확인할 수 있다.

VI. 기존 연구

Zhang et al.과 Li et al.은 최초로 다중 키워드 순위 검색이 가능한 SE 모델을, 다중 업로더 모델로 확장하는 기법을 제안하였다[14-15]. 또한, Zhang et al.은 AOPPF 함수를 이용하여 클라우드 서버에서 암호화된 문서와 검색 요청 키워드가 실제로 얼마나 근접한지에 대한 값을 숨기며 순위 검색을 가능하도록 만든 공개키 SE 모델을 개발하였다[5].

Peng et al.은 효율적인 검색을 위해 KBB-Tree 인덱스 구조를 활용했으며, 이어서 Guo et al.과 Sun et al.은 해당 체계에 기능과 효율성을 추가하고 확장하였다[5-6,8]. 구체적으로, Guo et al.는 검색 프로세스의 정확성과 효율성을 향상시키기 위해 문서 품질 평가와 GBB-Tree를 포

함한 방안을 제안하였다[5]. Sun et al.은 Zhang et al.과 Peng et al.의 연구에서 발견된 취약점을 보완하는 방법을 제시하였다[6].

그러나 현재까지 시스템이 완전히 신뢰할 수 있는 개체나 트랩도어 생성 중 안전한 연결에 의존하지 않는 안전한 순위 검색이 가능한 다중 지원 SE 모델에 대한 연구는 수행되지 않았다.

VII. 결 론

본 논문에서는 클라우드 컴퓨팅에서 순위 검색이 가능한 암호화 다중 지원 모델의, 트랩도어 센터를 완전 신뢰 가능한 것에서 정직하지만 호기심이 많은 엔티티로 변경함으로써 보안 과정을 낮춰, 실생활의 경우에 가장 가까운 모델을 제안하였다.

무작위적인 요소를 이용하여 다운로더의 검색 요청 키워드가 마스터 비밀키가 없는 외부 공격자나 키의 일부를 가지고 있는 트랩도어 센터에게 노출되지 않으며, 선형 분할 기법으로 이들은 동등성 테스트 공격에 내성을 가짐으로써 공격자가 두 개의 서로 다른 트랩도어/검색 요청 간의 연결 가능성을 밝히는 것을 성공적으로 방지한다. 또한 2개의 프록시 서버를 사용함으로써 클라우드 서버로부터 트랩도어와 검색 결과 간의 직접적인 상관관계를 공격자들로부터 숨길 수 있다.

실험을 통해 본 논문은 해당 접근 방식이 완전 신뢰 가능한 트랩도어 센터를 사용하는 것에서 정직하지만 호기심이 있는 트랩도어 센터를 사용하는 방향으로 보안성을 높이는 과정에서, 이를 위해 발생하는 추가 계산 과정에도 불구하고 합리적인 시간 비용을 가지고 있음을 입증하였으며, 이는 실생활에서 해당 모델의 실현성을 높인다고 볼 수 있다.

향후 작업에서 본 연구는 더욱 강력한 보안성과 더욱 높은 효율성을 목표로 할 것이다.

References

[1] Q. Zhang, L. Cheng, R. Boutaba, "Cloud computing: State-of-the-art and research challenges, Journal of Internet Services and Applications", vol. 1, no. 1, pp. 7 - 18. Apr. 2010

[2] M. Ali, S. U. Khan, A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges", Information Sciences vol. 305, pp. 357 - 383, Jun. 2015

[3] R. Latif, H. Abbas, S. Assar, Q. Ali, "Cloud computing risk assessment: A systematic literature review", Future Information Technology, pp. 285-295, 2014

[4] W. Zhang, Y. Lin, S. Xiao, J. Wu, 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, Jan. 2015

[5] Z. Guo, H. Zhang, C. Sun, Q. Wen, W. Li, "Secure multi-keyword ranked search over encrypted cloud data for multiple data owners", Journal of Systems and Software, vol. 137 pp. 380 - 395, Mar. 2018

[6] J. Sun, S. Hu, X. Nie, J. Walker, "Efficient Ranked Multi-Keyword Retrieval with Privacy Protection for Multiple Data Owners in Cloud Computing", IEEE Systems Journal, vol. 14, no. 2, pp. 1728 - 1739, Aug. 2019

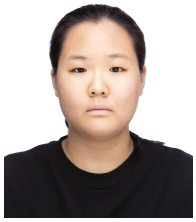
[7] L.Chen, H, Löhr, M. Manulis, A. Sadeghi, "Improving privacy of Property-Based Attestation without a Trusted Third Party", Proceedings of the 2011 7th International Conference on Computational Intelligence and Security, pp. 559 - 563, Sep. 2011

[8] T. Peng, Y. Lin, X. Yao, W. Zhang, "An Efficient Ranked Multi-Keyword Search for Multiple Data Owners over Encrypted Cloud Data", IEEE Access vol. 6, pp. 21924 - 21933, Apr. 2018

[9] A. Kiayias, O. Oksuz, A. Russell, Q. Tang, B. Wang, "Efficient encrypted keyword search for multi-user data sharing, Proceedings of the European

Symposium on Research in Computer Security", pp. 173 - 195. Sep. 2016
〔10〕 X. Boyen, B. Waters, "Anonymous hierarchical identity-based encryption (Without random oracles)", Proceedings of the Annual International Cryptology Conference, pp. 290 - 307. Jun. 2006
〔11〕 Textfiles.com, "textfiles", http://textfiles.com, (accessed 12 May 2022).
〔12〕 Natural language toolkit, "NLTK", http://www.nltk.org/nltk_data, (accessed 12 May 2022)
〔13〕 pypbc, "pypbc", https://github.com/debatem1/pypbc, (accessed 12 May 2022).
〔14〕 W. Zhang, S. Xiao, Y. Lin, T. Zhou, S. Zhou, "Secure ranked multi-keyword search for multiple data owners in cloud computing", International Conference on Dependable Systems and Networks, pp. 276 - 286, Jun. 2014
〔15〕 J. Li, Y. Lin, M. Wen, C. Gu, B. Yin, "Secure and verifiable multi-owner ranked-keyword search in cloud computing", Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications, vol. 9204, pp. 325 - 334, Aug. 2015

〈저자소개〉



김 예 은 (YeEun Kim) 학생회원
2019년 2월: 한양대학교 ERICA 컴퓨터공학과 졸업
2019년 3월~현재: 한양대학교 바이오인공지능융합과 석박사통합과정
〈관심분야〉 정보보호, 암호학

오 희 국 (Heekuck Oh) 종신회원
1982년: 한양대학교 전자공학과 졸업
1989년: 아이오와주립대학 전자계산학과 석사
1992년: 아이오와주립대학 전자계산학과 박사
1993년~1994년: 한국전자통신연구원 선임연구원
1995년 3월~현재: 한양대학교 ERICA 소프트웨어융합대학 교수
〈관심분야〉 암호기술응용, 시스템보안