KIPS Trans. Comp. and Comm. Sys.
Vol.10, No.9 pp.243~250
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2021.10.9.243

# Privacy-Preserving Parallel Range Query Processing Algorithm Based on Data Filtering in Cloud Computing

Hyeong Jin Kim† · Jae-Woo Chang††

## ABSTRACT

Recently, with the development of cloud computing, interest in database outsourcing is increasing. However, when the database is outsourced, there is a problem in that the information of the data owner is exposed to internal and external attackers. Therefore, in this paper, we propose a parallel range query processing algorithm that supports privacy protection. The proposed algorithm uses the Paillier encryption system to support data protection, query protection, and access pattern protection. To reduce the operation cost of a checking protocol (SRO) for overlapping regions in the existing algorithm, the efficiency of the SRO protocol is improved through a garbled circuit. The proposed parallel range query processing algorithm is largely composed of two steps. It consists of a parallel kd-tree search step that searches the kd-tree in parallel and safely extracts the data of the leaf node including the query, and a parallel data search step through multiple threads for retrieving the data included in the query area. On the other hand, the proposed algorithm provides high query processing performance through parallelization of secure protocols and index search. We show that the performance of the proposed parallel range query processing algorithm increases in proportion to the number of threads and the proposed algorithm shows performance improvement by about 5 times compared with the existing algorithm.

Keywords : Privacy-preserving, Cloud Computing, Range Query Processing, Paillier Cryptosystem, Parallelism, Garbled Circuit

# 클라우드 컴퓨팅에서 프라이버시 보호를 지원하는 데이터 필터링 기반 병렬 영역 질의 처리 알고리즘

김 형 진† · 장 재 우††

## 요 약

최근 클라우드 컴퓨팅이 발전함에 따라 데이터베이스 아웃소싱에 대한 관심이 증가하고 있다. 그러나 데이터베이스를 아웃소싱하는 경우, 데이터 소유자의 정보가 내부 및 외부 공격자에게 노출되는 문제점을 지닌다. 따라서 본 논문에서는 프라이버시 보호를 지원하는 병렬 영역 질의처리 알고리즘을 제안한다. 제안하는 알고리즘은 Paillier 암호화 시스템을 사용하여 데이터 보호, 질의 보호, 접근 패턴 보호를 지원한다. 또한 기존 알고리즘에서 영역 겹침을 확인하는 프로토콜(SRO)의 연산 비용을 줄이기 위해 garbled 서킷(circuit) 을 통해 SRO 프로토콜의 효율성을 향상시킨다. 제안하는 병렬 영역 질의 처리 알고리즘은 크게 2단계로 구성된다. 이는 kd-트리를 병렬적으로 탐색하고 질의를 포함하는 단말 노드의 데이터를 안전하게 추출하는 병렬 kd-트리 탐색 단계와 다수의 thread를 통해 질의 영역에 포함된 데이터를 병렬 탐색하는 병렬 데이터 탐색 단계로 구성된다. 한편, 제안하는 알고리즘은 암호화 연산 프로토콜과 인덱스 탐색의 병렬화를 통해 우수한 질의 처리 성능을 제공한다. 제안하는 병렬 영역 질의 처리 알고리즘은 thread 수에 비례하여 성능이 향상됨을 알 수 있고 10 thread 상에서 기존 기법은 38초, 제안하는 기법은 11초로 약 3.4배의 성능 향상이 있음을 보인다.

키워드 : 프라이버시 보호, 클라우드 컴퓨팅, 영역 질의 처리, Paillier 암호화 시스템, 병렬화, Garbled Circuit

## 1. 서 론

최근 클라우드 컴퓨팅의 시장 점유율이 높아짐에 따라 데이터베이스 아웃소싱에 대한 관심이 증가하고 있다. 데이터베이스 아웃소싱이란 데이터 소유자가 데이터베이스를 전문적으로 관리하는 기업 및 클라우드에게 자신의 데이터를 위탁하는 것을 의미한다. 이때 위탁받은 기업은 데이터 소유자의 데이터를 저장 및 관리할 뿐만 아니라 다양한 질의처리 서비스를 제공한다. 데이터 소유자는 자신이 사용한 컴퓨팅 리소스에 대해서만 비용을 지불하기 때문에, 컴퓨팅 리소스를 구축하기 위한 초기 비용을 절약할 수 있다는 장점을 지닌다. 이를 통해, 데이터 소유자는 컴퓨팅 자원을 위한 인력 및 자원을 절약할 수 있기 때문에 다양한 스타트업 및 기업에서 클라우드 컴퓨팅 모델을 비즈니스 모델로 선택하고 사용중이다

※ 이 논문은 2019년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 임(No. NRF-2019R1I1A3A01058375).
※ 이 논문은 2021년 한국정보처리학회 춘계학술발표대회의 우수논문으로 "클라우드 상에서 정보 보호를 지원하는 garbled circuit 기반 병렬 영역 질의처리 알고리즘"의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 전북대학교 컴퓨터공학부 박사과정
†† 종신회원 : 전북대학교 IT정보공학과 교수
Manuscript Received : June 28, 2021
First Revision : July 27, 2021
Accepted : July 28, 2021
* Corresponding Author : Jae-Woo Chang(jwchang@jbnu.ac.kr)

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/ licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

[1]. 그러나 데이터베이스를 아웃소싱하는 경우, 데이터 소유자의 민감한 정보가 위탁 기업 및 클라우드에 그대로 노출되는 문제점이 존재한다. 데이터 소유자의 민감한 정보가 클라우드에 노출되는 경우, 클라우드 내부의 악의적인 공격자가 언제든지 데이터를 유출할 수 있다. 데이터 소유자의 민감한 정보가 유출되면, 데이터 소유자는 금전적 및 정신적 피해를 입게 되고 클라우드를 관리하는 기업의 신뢰도에 악영향을 미친다. 따라서 클라우드 컴퓨팅 상에서 프라이버시 보호를 위한 연구는 필수적이다[2,3].

한편, 프라이버시 보호를 지원하는 질의 처리 알고리즘에 대한 연구가 활발하게 수행되었다[4-9]. 그 가운데 영역 질의는 데이터베이스에서 자주 수행되는 대표적인 질의이다. 영역 질의는 주어진 범위 내에 존재하는 모든 데이터를 탐색하는 질의를 의미하며, 대표적으로 공간 데이터베이스에서 영역 모니터링, 위치 기반 서비스를 제공할 때 사용되며, 그 외에도 금융, 보험, 은행에서 제공하는 기본적인 탐색 질의에 사용된다[10]. 따라서 클라우드 컴퓨팅 상에서 프라이버시 보호를 지원하는 영역 질의처리 알고리즘의 연구가 필요하다.

데이터베이스 아웃소싱 환경에서 프라이버시 보호를 지원하는 영역 질의처리 알고리즘이 고려할 사항은 다음과 같다. 첫째, 데이터 보호 및 사용자 질의 보호를 지원해야 한다. 데이터 및 사용자 질의는 데이터 소유자와 사용자의 민감한 데이터를 포함할 수 있기 때문에 원본 데이터 및 질의를 보호하는 것은 프라이버시 보호에 필수적이다. 둘째, 데이터 접근 패턴 은닉을 지원해야 한다. 데이터 접근 패턴이 노출되는 경우, 공격자는 접근 패턴으로 원본 데이터 및 질의와 관련된 데이터를 유추 가능하다. 따라서 추가적인 정보 노출을 방지하기 위해서는 접근 패턴 보호가 필수적이다. 셋째, 사용자에게 정확한 질의 결과를 반환함으로써, 사용자 QoS를 지원해야 한다. 마지막으로, 효율적인 질의처리 성능을 보장해야 한다. 따라서 이러한 네 가지 고려 사항을 만족하는 안전한 영역 질의처리 알고리즘에 대한 연구가 필수적이다.

첫째, P. Wang와 C. V. Ravishankar의 연구는 프라이버시 보호를 지원하는 영역질의 처리 알고리즘을 제안하였다[11]. 그러나 두 번째 고려사항인 데이터 접근 패턴을 보호하지 못하는 문제점을 지닌다. 둘째, B. Wang et al.의 연구는 인덱스 탐색 기법을 이용한 영역 질의처리 알고리즘을 제안하였다[12]. 그러나 첫 번째와 두 번째 고려사항인 사용자 질의 및 접근 패턴을 보호를 지원하지 못하는 문제점을 지닌다. 셋째, H. Kim et al.의 연구는 힐버트 커브(Hilbert-curve) 기반의 암호화 인덱스 및 영역 질의처리 알고리즘을 제안하였다[13]. 그러나 두 번째 고려사항인 데이터 접근 패턴을 보호하지 못하는 문제점을 지닌다. 마지막으로 H. Kim et al.의 연구는 준동형 암호화 기반의 영역 질의 처리 알고리즘을 제안하였다[14]. 데이터 보호, 질의 보호, 접근 패턴을 보호하는 유일한 연구이다. 그러나 암호화 이진 배열 기반의 암호화 연산 프로토콜을 사용함으로써 높은 질의 처리 비용이 요구되기 때문에 네 번째 고려사항을 만족하지 못하는 문제점을 지닌다.

따라서 본 논문에서의 클라우드 컴퓨팅 상에서 위의 4가지 요구 사항을 모두 만족하는 병렬 영역 질의 처리 알고리즘을 제안한다. 제안하는 알고리즘의 contribution은 다음과 같다.

- 준동형 암호화 기법을 통한 데이터 보호, 질의 보호, 접근 패턴 보호
- kd-트리 암호화 인덱스를 통한 데이터 필터링 기법 제공
- garbled 서킷을 통한 암호화 프로토콜 최적화
- thread_pool을 이용한 병렬 질의 처리 지원

본 논문의 구성은 다음과 같다. 2장에서는 배경 및 관련 연구로써 기존에 제안된 프라이버시 보호를 지원하는 영역 질의 처리 알고리즘을 소개한다. 3장에서는 본 논문에서 사용하고 있는 전체 시스템의 구조를 설명한다. 4장에서는 제안하는 프라이버시 보호를 지원하는 병렬 영역 질의 처리 알고리즘에 대해 기술하고, 5장에서 제안하는 기법의 보안 분석을 기술한다. 6장에서는 성능 평가를 통해 제안하는 알고리즘의 성능을 분석한다. 마지막으로 7장에서는 본 논문의 결론 및 향후 연구를 제시한다.

## 2. 배경 및 관련 연구

### 2.1 배경

본 논문에서 데이터 보호 및 질의 보호를 위해 사용하는 암호화 시스템은 Paillier 암호화 시스템(Paillier crypto system)[15]이다. 이는 대표적인 덧셈 준동형 암호화 기법으로써, 동일한 값이라 할지라도 암호화 할 때마다 다른 암호문이 생성되는 확률적 암호체계이다. Paillier 암호화 시스템에서 암호화 키 pk는 (N, g)와 같이 주어지며, N은 두 개의 큰 소수 간의 곱셈 값이고, g는 $\mathbb{Z}^*N^2$에서의 값이다. 한편, Paillier 암호화 시스템에서 복호화 키 sk는 (p, q)로 주어진다. Paillier 암호화 시스템은 복호화 과정 없이 암호문 간의 연산을 통해 평문 간의 덧셈에 대응되는 암호문을 계산 가능한 특징을 보인다. Paillier 암호화 시스템의 암호화 함수를 E(·), 복호화 함수를 D(·)라고 할 때, Paillier 암호화 시스템은 다음과 같은 특성을 보인다. 이를 위한 기호(notation)는 Table 1과 같다.

첫째, 두 암호화 데이터 E(a), E(b)의 곱 E(a)×E(b)은 평문 a+b의 암호화 값인 E(a+b)와 같다. Equation (1)은 이를 나타낸다.

$$E(a+b) = E(a) \times E(b) \bmod N^2 \quad (1)$$

둘째, 암호화 데이터 E(a)의 b 제곱인 E(a)b은 평문 a×b의 암호화 값인 E(a×b)와 같다. Equation (2)는 이를 나타낸다.

$$E(a \times b) = E(a)^b \bmod N^2 \quad (2)$$

Table 1. Equation Notation

| | |
|---|---|
| a, b | a and b is integer (0≤a, b≤N) |
| p, q | big prime |
| N | p×q |
| E(·) | encrypt the value |
| D(·) | decrypt the value |

셋째, Paillier 암호화 시스템은 의미적 보안(semantic security)[16]을 지원한다. 즉, 하나의 평문에 대해 다양한 암호문이 존재할 수 있기 때문에, 암호문을 기반으로 평문을 유추하는 것이 불가능하다. 따라서 paillier 암호화 시스템은 선택 평문 공격으로부터 안전하다. 클라우드 컴퓨팅 상에서 데이터 보호, 질의 보호를 지원하기 위해서는 암호화된 상태에서 특정 연산을 수행되는 것이 필수적인데, paillier 암호화 시스템은 이를 지원하기 위한 대표적인 방법으로 널리 사용된다[5,14]. 따라서 본 논문에서는 데이터 보호 및 질의 보호를 위해 paillier 암호화 시스템을 사용한다.

한편 아웃소싱 데이터베이스에 기반한 클라우드 컴퓨팅 환경에서 고려할 수 있는 대표적인 공격 모델은 semi-honest 공격 모델 및 malicious 공격 모델이다[17]. semi-honest (혹은 honest-but-curious) 공격 모델은 클라우드가 자신이 맡은 프로토콜을 정직하게 수행하지만, 프로토콜 수행 중 획득한 정보를 바탕으로 데이터 소유자 및 질의 요청자에 대한 추가적인 정보를 획득하기 위한 시도를 할 수 있는 공격 모델을 의미한다. malicious 공격 모델[18]은 클라우드가 주어진 프로토콜에서 벗어나 악의적인 의도로 정보 획득을 시도하는 공격 모델을 의미한다. 따라서 특정 프로토콜 혹은 알고리즘이 malicious 공격 모델에 대해 안전함을 검증할 경우, 해당 프로토콜이 다른 공격 모델에 대해서도 안전함을 입증할 수 있다. 그러나 malicious 공격 모델에 대해서 안전한 프로토콜의 경우, 구현 및 활용 측면에서 매우 높은 비용이 요구되기 때문에 실제 환경에 적용하기 힘든 문제점을 지닌다. 반면, semi-honest 공격 모델에 대해서 안전한 프로토콜의 경우, 실제 환경에 적용 가능할 뿐만 아니라 malicious 공격 모델에 대해서도 안전한 프로토콜을 설계하기 위한 기본으로 활용 가능하다. 따라서 본 논문에서는 기존 연구[14]와 같이 semi-honest 공격 모델을 고려한 연구를 수행한다.

### 2.2 관련 연구

일반적으로 영역 질의는 영역으로 주어진 질의 내에 존재하는 모든 데이터를 탐색하는 질의로써, 데이터 마이닝, 위치 기반 서비스 등 다양한 분야에서 활용된다. 따라서 프라이버시 보호를 지원하는 영역 질의처리 알고리즘의 연구가 활발히 수행되었다.

첫째, P. Wang와 C. V. Ravishankar의 연구는 R-트리 기반의 인덱스를 활용한 영역 질의처리 알고리즘을 제안하였다[11]. 해당 기법은 데이터의 순서 정보를 은닉하기 위해, 트리의 노드 단위로 질의처리를 수행한다. 그러나 해당 기법은 질의와 겹치는 노드에 존재하는 모든 데이터를 질의 결과로 반환하기 때문에, 거짓양성(false positive) 결과가 전 체 질의 결과의 20~40%를 차지하는 문제점을 보인다. 또한, 클라우드가 질의 영역과 겹치는 R-트리의 단말 노드를 확인 가능하기 때문에, 데이터 접근 패턴이 노출되는 문제점을 보인다. 아울러, 데이터 레벨에서의 순서 정보는 은닉 가능하지만, 노드 레벨에서의 순서 정보가 노출되는 문제점을 나타낸다.

둘째, B. Wang et al.는 트리 기반 인덱스를 기반으로 정확한 질의결과를 보장하는 영역 질의처리 알고리즘을 제안하였다[12]. 해당 기법은 HVE 암호화 기법을 확장하여 단일 차원에서의 추가적인 정보노출을 방지하는 인덱스 탐색 기법을 제시하였으며, 이를 통해 정보보호 수준을 향상시켰다. 그러나 해당 기법에서 확장한 HVE 암호화 기법은 사용자 질의 보호를 지원하지 못한다. 아울러, 클라우드가 질의 영역과 겹치는 R-트리의 노드 및 최종 질의 결과의 식별자를 확인 가능하기 때문에, 데이터 접근 패턴이 노출되는 문제점을 보인다.

셋째, H. Kim et al.는 힐버트 커브(Hilbert-curve) 기반의 암호화 인덱스 및 영역 질의처리 알고리즘을 제안하였다[13]. 해당 기법은 힐버트 커브를 이용하여 데이터 그룹을 생성하고, 해당 데이터 그룹을 기반으로 인덱스를 구축한다. 사용자는 질의 처리 시 힐버트 커브를 통해 질의 영역이 속하는 데이터 그룹을 찾고, 해당 데이터 그룹의 식별자를 통해 질의 처리를 수행함으로써 질의 보안이 지원된다. 그러나 해당 기법은 인덱스 탐색으로 인해 질의처리 비용이 높은 문제점이 존재한다. 또한, 해당 기법은 데이터 그룹을 기반으로 질의처리를 수행하기 때문에, 거짓양성 결과가 포함될 수 있다. 아울러, 질의 결과와 관련된 데이터 그룹의 식별자가 클라우드에게 노출되기 때문에, 데이터 접근 패턴 노출이 발생하는 문제점이 존재한다.

마지막으로 H. Kim et al.의 연구는 kd-트리 기반의 암호화 인덱스 및 영역 질의처리 알고리즘을 제안하였다[14]. 해당 기법은 준동형 암호화 기법을 사용하여 데이터 보호, 질의 보호, 접근 패턴의 보호가 가능하다. 또한 암호화 kd-트리 기반 필터링을 지원함으로써 효율적인 질의처리를 지원한다. 그러나 해당 기법은 비트 배열 기반의 암호문을 사용함으로써 높은 암호화 프로토콜 연산 비용이 요구된다.

Table 2는 프라이버시 보호를 지원하는 영역 질의 처리 알고리즘의 직접적인 관련 연구와의 비교표이다. 아울러 표에서 제안하는 연구와 H. Kim의 연구만이 데이터 보호, 질의 보호, 접근 패턴 보호를 지원한다.

Table 2. Comparison of Related Works

| | Data and Query Protection | Hiding Data Access Patterns | Minimize Query Processing Cost on the user Side | Query Correctness | Efficient Query Processing |
|---|---|---|---|---|---|
| P. Wang and C. V. Ravishankar.'s work [11] | ○ | × | ○ | ○ | × |
| B. Wang et al.'s work [12] | × | × | ○ | ○ | ○ |
| H. Kim et al.'s work [13] | ○ | × | × | ○ | × |
| H. Kim et al.'s work [14] | ○ | ○ | ○ | ○ | × |
| The proposed | ○ | ○ | ○ | ○ | ○ |

○ : Support × : Not support

## 3. 전체 시스템 구조

Fig. 1은 제안하는 영역 질의처리 알고리즘의 전체 시스템 구조를 나타낸다. 데이터 소유자는 $n$개의 레코드 $t_i(1 \le i \le n)$로 구성된 원본 데이터베이스($T$)를 보유하고 있다. 각 레코드는 $m$개의 속성(attribute)으로 구성되며, $i$번째 레코드의 $j$번째 속성은 $t_{i,j}(1 \le i \le n,\ 1 \le j \le m)$와 같이 표기한다. 데이터 소유자는 해당 데이터베이스에 대한 색인을 지원하기 위해 kd-트리 기반의 데이터 분할을 수행한다. 한편, 본 논문에서는 질의 처리 시 질의 요청자의 데이터 접근 패턴이 노출되는 것을 방지하기 위해, 생성된 kd-트리의 단말 노드만을 고려한다. 구축된 kd-트리의 레벨이 $h$, 단말 노드의 총 수는 $2^{h-1}$, 한 노드가 저장할 수 있는 최대 데이터 수(FanOut)는 $F$라고 가정한다. kd-트리의 단말 노드는 해당 노드가 담당하는 각 속성 별 영역 정보 $lb_{z,j}$, $ub_{z,j}(1 \le z \le 2^{h-1},\ 1 \le j \le m)$ 및 해당 단말 노드의 영역 내에 포함된 데이터 $id$를 저장한다. 여기서 $lb_{z,j}$과 $ub_{z,j}$은 단말 노드가 담당하는 영역의 각 속성별 하한점(lower bound) 및 상한점(upper bound)을 의미한다.

데이터베이스 암호화는 Paillier 암호화 시스템[9]을 기반으로 수행하며, 이를 위해 데이터 소유자는 암호화 공개키($pk$) 및 복호화 비밀키($sk$)를 생성한다. 데이터베이스 아웃소싱으로 인한 데이터 노출을 방지하기 위해, 데이터 소유자는 암호화 키를 이용하여 데이터베이스 암호화를 수행한다. 이 때, 데이터베이스의 암호화는 각 레코드의 속성 단위로 수행되며, 이를 통해 $E(t_{i,j})(1 \le i \le n,\ 1 \le j \le m)$을 생성한다. 또한, 클라우드가 암호화 데이터베이스 상에서 효율적으로 질의처리를 수행할 수 있도록 지원하기 위해, 구축된 kd-트리의 단말 노드를 암호화 한다. 이 때, kd-트리의 각 단말 노드의 영역 정보를 속성 별로 암호화하여, $E(lb_{z,j})$, $E(ub_{z,j})(1 \le z \le 2^{h-1},\ 1 \le j \le m)$를 생성한다. 해당 시스템에는 서로 결탁하지 않는(non-colluding) 두 개의 클라우드 $C_A$, $C_B$가 존재하며, $C_A$와 $C_B$는 모두 semi-honest 하다고 가정한다. 즉, $C_A$ 및 $C_B$는 질의 처리를 위해 자신이 담당하는 프로토콜을 정직하게 수행하지만, 질의 처리 과정 중에 획득한 정보를 바탕으로 데이터 소유자 및 질의 요청자에 대한 추가적인 정보를 획득하기 위한 시도를 수행할 수 있다. 하지만 추가적인 정보를 획득하기 위해 다른 클라우드와 결탁하여 데이터 및 정보를 주고받지는 않는다.

한편, 암호화된 데이터베이스를 기반으로 다양한 질의처리를 지원하기 위해서는 $C_A$와 $C_B$ 간의 안전한 다자간 계산(SMC : Secure Multiparty Computation)[19]이 요구된다. 안전한 다자간 계산이란 자신이 보유하고 있는 데이터를 노출하지 않은 채, 다른 개체의 도움을 통해 프로토콜 및 연산을 수행하는 기법을 의미한다. 이를 위해, 데이터 소유자는 암호화 데이터베이스 및 복호화 비밀키를 각기 다른 클라우드에게 전송한다. 즉, 데이터 소유자는 암호화 데이터베이스 및 암호화 kd-트리를 암호화 공개키와 함께 클라우드 $C_A$에 아웃소싱 한다. 이 때, 암호화 kd-트리의 각 노드 영역에 포함되는 데이터 id를 평문의 형태로 함께 아웃소싱한다. 한편 데이터 소유자는 $C_B$에게 복호화 비밀키를 전송한다. 또한, 질의 처리 과정에서, 데이터 소유자는 인증된 사용자에게 암호화 공개키를 전송한다. $C_A$에게 아웃소싱된 암호화 데이터베이스와 암호화 인덱스, $C_B$에게 전송된 복호화 비밀키를 바탕으로, $C_A$와 $C_B$는 안전한 다자간 계산을 통해 영역 질의를 지원한다.

Yao의 garbled circuit(이하 garbled 서킷)[20]은 수행하고자 하는 특정 함수의 입력 값을 각각 garbled 서킷 생성자

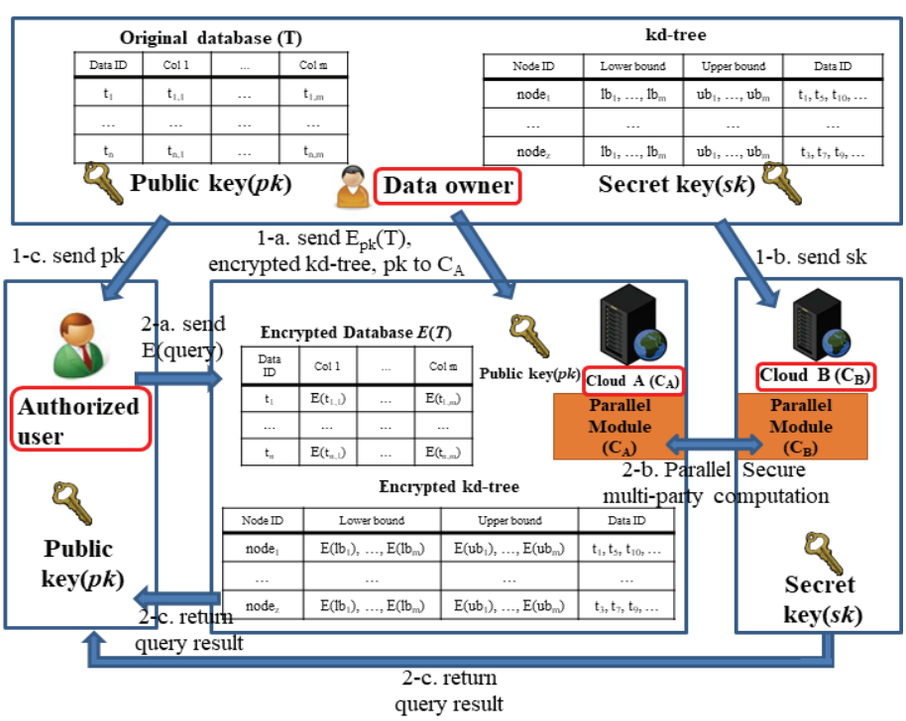


Fig. 1. Overall System Architecture

(generator)와 garbled 서킷 평가자(evaluator)가 지니고 있는 환경에서, 함수의 결과 값을 안전하게 계산하는 기법이다. Garbled 서킷은 함수의 결과를 계산하는 과정 중에 garbled 서킷 생성자 및 평가자가 보유한 입력 값이 상대방에게 노출되지 않을 뿐만 아니라, 함수 수행 도중에 생성되는 중간 데이터의 값도 서로에게 공개되지 않는다. 따라서 garbled 서킷은 정보 노출 없이 함수의 결과 값을 계산하는 것이 가능하다. Garbled 서킷 생성자는 계산하고자 하는 함수를 위한 암호화된 버전의 서킷을 생성하는 역할을 수행한다. Garbled 서킷 평가자는 서킷 수행 도중 생성되는 중간 데이터의 값을 알지 못한 채, 주어진 함수의 결과 값을 계산한다.

## 4. 프라이버시 보호를 지원하는 병렬 영역 질의 처리 알고리즘

본 절에서는 클라우드 컴퓨팅 상에서 프라이버시 보호를 지원하는 병렬 영역 질의처리 알고리즘을 제안한다. 제안하는 알고리즘은 준동형 암호화 기법을 사용하여 고수준의 프라이버시 보호를 제공할 뿐만 아니라, 비트 배열 기반의 암호화 연산 프로토콜을 garbled 서킷 기반의 암호화 연산 프로토콜로 변환하여 연산 속도를 향상시킨다. 또한 kd-트리 인덱스 탐색의 병렬화 및 영역 데이터 검색의 병렬화를 수행함으로써 효율적인 질의처리를 지원한다. 제안하는 병렬 범위 알고리즘은 병렬 인덱스 탐색 단계와 질의 영역 내 데이터 병렬 탐색 단계로 구성되며, 이는 RangePI(Range query processing algorithm using Parallelism and Index)로 명명한다.

□ 수행단계 1 : 병렬 인덱스 탐색 단계

Algorithm 1은 병렬 인덱스 탐색 단계의 수행 과정을 나타낸다. 병렬 인덱스 탐색 단계에서는 사용자가 전송한 질의 영역(Q)과 겹치는 kd-트리의 단말 노드를 모두 탐색하고, 해당 노드 내에 존재하는 데이터를 안전하게 추출한다. 이때, kd-트리의 단말 노드를 탐색하는 과정 및 노드 내에 존재하는 데이터를 추출하는 과정을 병렬적으로 처리함으로써 효율적인 질의처리가 가능하다.

첫째, $C_A$는 병렬 처리를 위해 thread pool을 생성하고 E($Q$)와 E($node_z$)간에 GSRO 프로토콜[14]을 작업 단위로 병렬 처리한다(line 1~3). 둘째, $C_A$는 순서 변경 함수 $\pi$를 생성하여 E($\alpha$)의 순서를 변경하고(e.g., E($\alpha'$)←$\pi$(E($\alpha$))), E($\alpha'$)를 $C_B$에게 전송한다(line 4). 셋째, $C_B$는 E($\alpha'$)를 복호화하여 $\alpha'$를 획득한 후, 1의 개수, 즉 질의 영역과 겹치는 노드의 수(c)를 확인한 후, c개의 노드 그룹 Group을 생성한다. $C_B$는 각 노드 그룹에 $\alpha'$=1 인 노드 한 개와 $\alpha'$=0 인 노드 ($num_{node}/c$)-1개를 할당한다. 이 때, 각 노드 그룹 별로 균등한 수의 노드가 할당되도록 한다. 아울러, 각 노드 그룹에 할당된 노드의 순서를 랜덤하게 변환한 후, 이를 $C_A$에게 전송한다(line 5~12). 넷째, $C_A$는 자신이 생성한 순서 변경 함수의 역변경 함수 $\pi$-1을 이용하여 각 노드 그룹에 속한 노드들의 식별 번호를 변경한다(line 13~15). 다섯째, $C_A$는 노드 그룹에 할당된 각 노드를 차례로 방문하고, 각 데이터에 대한 GSRO 프로토콜 및 SM 프로토콜을 하나의 작업 단위로 thread_pool에 삽입한다. 각 thread는 순차적으로 thread_pool에 존재하는 작업을 수행하며, SM 프로토콜 수행을 통해 반환된 결과 값을 E($cand$)로 반환한다. 마지막으로 암호화 인덱스 탐색 알고리즘은 질의 영역과 겹치는 노드 내에 존재하는 모든 데이터가 저장된 E($cand$)를 반환하고 수행단계 1을 종료한다.

**Algorithm 1.** pIndexSearch(parallel Index Search)

**Input : E($q$), E($node$)**
**Output : E($cand$)** // all the data inside nodes related to a query

$C_A$ :
01. generate *thread_pool* // create a thread and wait in the pool until a task is given
02. for 1 ≤ $z$ ≤ $num_{node}$
03. call thread_pool_push(GSRO($E$(Q), $E(node_z)$), $E(\alpha_z)$)
04. E($\alpha'$) = $\pi$(E($\alpha$)); send E($\alpha'$) to $C_B$

$C_B$ :
05. $\alpha'$ ← D(E($\alpha'$))
06. $c$ ← the number of '1' in $\alpha'$
07. create $c$ number of *Group* // *Group* : node group
08. for each *Group*
09. assign a node with $\alpha'$=1
10. assign ($num_{node}/c$)-1 nodes with $\alpha'$=0
11. shuffle the sequence of nodes
12. send *Group* to $C_A$

$C_A$ :
13. $cnt$ ← 0
14. for each *Group*
15. permute node IDs using $\pi^{-1}$
16. for each *Group*
17. for 1 ≤ $z$ ≤ $num$
18. for 1 ≤ $s$ ≤ $F$
19. assign task $T_s$ to threads in the thread pool
20. for each $T_s$
21. for 1≤$j$≤$m$
22. E($t'_{z,j}$) ← SM($node_z.t_{s,j}$, E($\alpha_z$))
23. E($cand_{cnt+s,j}$) ← $E(cant_{cnt+s,j})$×$E(t'_{z,j})$
24. $cnt$ ← $cnt$ + F
25. return E($cand$)

**End Algorithm**

□ 수행단계 2 : 질의 영역 내 데이터 병렬 탐색 단계

질의 영역 내 데이터 탐색 단계에서는 암호화 인덱스 탐색 단계에서 추출한 데이터를 기반으로 암호화 질의 영역에 실제로 포함되는 모든 데이터를 탐색한다. Algorithm 2는 질의 영역 내 데이터 병렬 탐색 단계의 수행 과정을 나타낸다.

첫째, $C_A$는 수행단계 1에서 추출된 질의 결과 후보인 E($cand$)와 E($Q$)를 기반으로 GSRO 프로토콜을 수행하여함으로써, 질의 영역에 속하는 데이터를 병렬 탐색한다(line 1~4). 이때, 각 암호화 프로토콜은 thread_pool에 삽입되고, thread는 순차적으로 pool에서 추출되어 처리한다. GSRO 수행 결과 반환된 E($\alpha$)의 값이 E(1)인 데이터는 질의 영역 내에 존재하는 데이터이다. 한편, Paillier 암호화 시스템은 의미적 보안을 지원하기 때문에, $C_A$ 및 $C_B$는 어느 데이터가 질

**Algorithm 2.** pDataRetrieval(parallel Data Retrieval)

**Input : E(*q*), E(*cand*)**
**Output : E(*result*)** // all the data inside the query region
$C_A$ :
01. for $1 \le i \le cnt$
02. call thread_pool_push(GSRO($E$(Q), $E$(cand$_i$)), $E(\alpha_2)$)
03. for $1 \le i \le cnt$
04. assign task $T_i$ to threads in the thread pool
05. for each $T_i$
06. for $1 \le j \le m$
07. $E(\gamma_{i,j}) \leftarrow E(cand_{i,j}) \times E(r_{i,j})$
08. $E(\alpha') \leftarrow \pi(E(\alpha))$; $E(\gamma') \leftarrow \pi(E(\gamma))$
09. $r' \leftarrow \pi(r)$
10. send $E(\alpha')$, $E(\gamma')$ to $C_B$ and $r'$ to user
$C_B$:
11. for $1 \le i \le cnt$
12. $\alpha'_i \leftarrow D(E(\alpha'_i))$
13. for $1 \le j \le m$
14. $\gamma'_{i,j} \leftarrow D(E(\gamma'_{i,j}))$
15. send $\alpha'$, $\gamma'$ to user
AU:
16. for $1 \le i \le cnt$
17. for $1 \le j \le m$
18. $result_{i,j} \leftarrow \gamma'_{i,j} - r'_{i,j}$
**End Algorithm**

의 영역 내에 존재하는지 알지 못한다.

둘째, GSRO의 수행 결과가 E(1)인 데이터는 영역 질의의 최종 질의 결과이기 때문에, 사용자에게 전송되어야 한다. 이때, 사용자 측에서의 질의처리 비용을 최소화하기 위해사용자에게 복호화된 질의 결과를 전송해야 한다. 그러나 질의 결과 자체를 복호화 할 경우 클라우드에게 질의 결과가 노출되기 때문에, 이를 방지하는 것이 필요하다. 아울러, 사용자에게 거짓양성 결과가 전송되는 것을 방지해야 한다. 이를 위해, $C_A$는 난수 $r_{i,j}$를 생성한 후, E($cand_{i,j}$)와 E($r_{i,j}$)를 기반으로 E($cand_{i,j}$)×E($r_{i,j}$)($1 \le i \le cnt$, $1 \le j \le m$) 연산을 수행한다. 해당 연산 결과는 E($\gamma_{i,j}$)에 저장된다. 아울러, $C_A$는 임의의 순서 변경 함수 $\pi$를 생성하여 E($\alpha_i$), E($\gamma_i$), $r_i$($1 \le i \le cnt$)의 순서를 데이터 단위로 변경한 후, 그 결과를 각각 E($\alpha'$), E($\gamma'$), $r'$에 저장한다(line 5~9). 마지막으로, $C_A$는 E($\alpha'$)와 E($\gamma'$)를 $C_B$에게 전송하고, $r'$를 사용자(AU)에게 전송한다(line 10~18).

## 5. 보안 분석

제안하는 병렬 영역 질의처리 알고리즘의 안전함을 증명하기 위해, $C_B$ 측에서의 보안 분석을 수행한다. $C_B$측에서의 제안하는 기법 수행 이미지는 $\Pi C_B$(RangePI) = {⟨E($\alpha'$), $\alpha'$⟩}와 같다. 여기서 E($\alpha'$)는 $C_A$로부터 전송받은 데이터이며, $\alpha'$는 $C_B$가 E($\alpha'$)의 복호화를 통해 획득하는 데이터이다. $C_B$측에서의 제안하는 기법의 시뮬레이션 수행 이미지를 $\Pi C_{Bs}$(RangePI) = {⟨E($\beta'$), $\beta'$⟩}라 가정한다. 여기서 E($\beta'$)는 $\mathbb{Z}_{N^2}$에서 생성된 난수이고, $\beta'$는 c개의 1 값과 $num_{node}$-c개의 0 값으로 구성된 벡터를 의미한다. 본 논문에서 고려한 암호화 함수는 의미적 보안을 지원하며, $N^2$ 도메인에서의 값을 반환한다. 이로 인해, E($\alpha'$)는 E($\beta'$)으로부터 계산적으로 구별 불가능하다. 또한, $C_A$가 임의 생성한 $\pi$는 $C_B$에게 공개되지 않기 때문에, $\alpha'$는 $\beta'$로부터 계산적으로 구별 불가능하다. 한편 $C_B$는 $\alpha'$에 대한 복호화를 수행할 경우, 질의 영역과 겹치는 kd-트리의 단말 노드 수(c)를 알 수 있다. 그러나 해당 정보를 통해 추가적인 정보 유출은 불가능하다. 위의 사항을 종합할 때, 제안하는 기법이 $C_B$에서 semi-honest 공격 모델 상에서 안전함을 보장할 수 있다.

## 6. 성능 평가

본 절에서는 클라우드 상에서 프라이버시 보호를 지원하는 병렬 영역 질의처리 알고리즘에 대한 성능평가를 수행한다. 관련 연구 비교에서 H. Kim et al.의 연구[14]만이 데이터 보호, 사용자 질의 보호 및 데이터 접근 패턴 보호를 모두 지원하기 때문에 성능평가를 위한 비교대상으로 선정하였다. 따라서 본 논문에서 제안한 기법(RangePI)을 기존 연구(RangeI)와 성능비교를 수행한다. 이를 통해, 제안하는 기법이 병렬적으로 처리됨에 따른 질의 처리 효율을 측정하였다. 성능 평가는 Linux ubuntu 14.04.2의 환경에서 Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz, 64GB(8GB × 4개) DDR4 DIMM 2400MHz를 기반으로 수행하였다.

해당 기법을 C++로 구현하였으며, Paillier 암호화 시스템의 구현을 위해 gmp 라이브러리를 사용하였다[21]. 성능평가에 사용된 데이터는 정규 분포에 따라 생성된 임의의 데이터이며 데이터 도메인은 0~512이다. 영역 질의가 자주 쓰이는 공간 데이터베이스를 위해 해당 논문에서는 2차원의 데이터를 사용하였다. 한편, 기존 연구에서 8k 데이터를 사용하여 질의처리 성능평가를 수행하였기 때문에[14], 본 논문에서는 2~10k 크기의 데이터를 사용한다. kd-트리의 높이(h)는 7로 생성하여 사용하였다. 질의 영역의 크기는 전체 도메인 크기의 0.1%로 설정하여 수행하였다. Table 3은 성능평가를 위한 실험 매개변수를 나타낸다. 표에서 공개키는 공개키 암호화 512, 1024를 많이 사용한다[17]. 그 이상은 너무 시간이 많이 걸리기 때문에 사용하지 않는다. 따라서 공개키 크기가 512여도 안정성 측면에서 안전하다. 보다 안전함을 추구하기 위해서는 1024bit 를 사용할 수 있으나, 이 경우에는 처리 시간이 많이 걸리는 오버헤드를 감수해야한다. 따라서 대부분의 기법은 512와 1024 사이의 trade off 측면에서 키 사이즈를 선택한다[17].

Fig. 2은 병렬화를 수행하지 않은 1 thread 상에서 데이

Table 3. Experimental Parameter

| Parameters | Values | Default Value |
|---|---|---|
| Total number of data(n) | 2k, 4k, 6k, 8k, 10k | 10k |
| Level of kd-트리(h) | 7 | 7 |
| # of attributes(m) | 2 | 2 |
| Encryption key size(K) | 512, 1024 | 512 |
| # of thread | 2, 4, 6, 8, 10 | 10 |

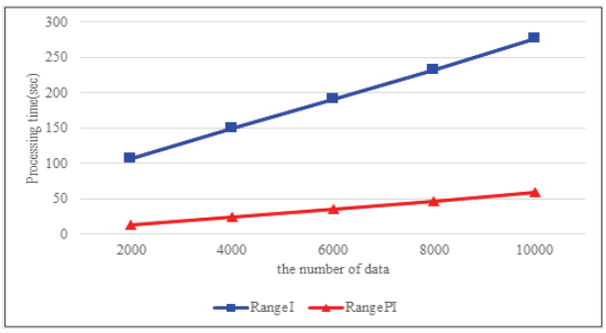


Fig. 2. Performance Evaluation According to the Change in the Number of Data

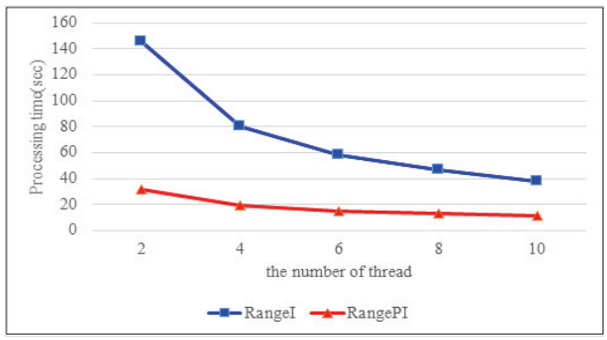


Fig. 3. Performance Evaluation According to the Change in the Number of Threads

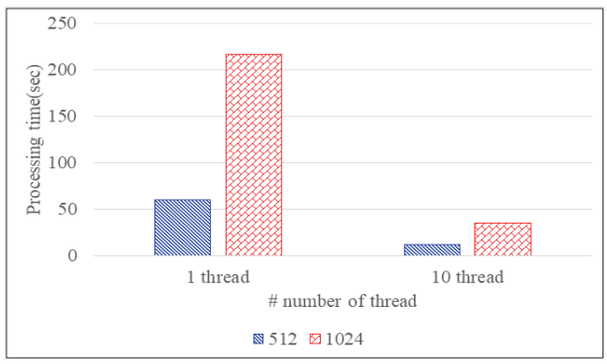


Fig 4. Performance Evaluation According to the Change in Key Size (512 and 1024)

터 수(n) 변화에 따른 제안하는 알고리즘(RangePI)과 기존 기법(RangeI)의 질의처리 성능을 나타낸다. 데이터 수가 증가함에 따라 질의처리 시간도 선형적으로 증가함을 볼 수 있다. 1 thread 상에서 제안하는 알고리즘(RangePI)은 기존 RangeI [14] 기법보다 평균 5.7배 성능 향상을 보인다. 이는 garbled 서킷을 통해 암호화 연산 프로토콜을 효율적으로 처리하기 때문이다.

Fig. 3은 병렬 영역 질의 처리 알고리즘에서 thread 수 변화에 따른 기존 알고리즘과 제안하는 알고리즘의 성능평가를 수행한다. 이를 위해 기존 알고리즘(RangeI)을 병렬 처리 환경으로 확장하였다. 아울러, 성능평가에 사용한 데이터는 (그림 2)에서 사용한 2차원의 공간 데이터를 사용하였다. 알고리즘의 병렬화를 통한 성능 향상을 측정하기 위해 thread를 2에서 10으로 증가시키면서 성능평가를 수행하였다. 제안하는 알고리즘은 thread 수에 비례하여 질의처리 성능이 향상됨을 알 수 있다. 2 thread의 경우, 제안하는 알고리즘(RangePI)은 기존 기법(RangeI)에 비해 약 4.6배 성능 향상이 있음을 보인다. 또한 10 thread의 경우 제안하는 알고리즘(RangePI)은 기존 기법에 비해 약 3.4배 성능 향상이 있음을 보인다. 한편, 10 thread 상에서 제안하는 알고리즘(RangePI)의 thread 당 처리 시간은 약 11초, 기존 기법(RangeI)의 thread 당 처리 시간은 약 38초이다. 제안하는 알고리즘이 thread 당 처리 성능이 우수한 이유는 garbled 서킷 기반의 암호화 연산 프로토콜을 사용하기 때문이다. garbled 서킷은 하드웨어 기반의 비교 연산을 수행하기 때문에 기존 비교 연산 프로토콜보다 빠른 연산 처리가 가능하다.

Fig. 4은 병렬 영역 질의 처리 알고리즘에서 key size 변화에 따른 제안하는 알고리즘의 성능평가를 수행한다. 아울러, 성능평가에 사용한 데이터는 Fig. 2에서 사용한 2차원의 공간 데이터를 사용하였다. key size 변화에 따른 알고리즘의 성능을 측정하기 위해, 512bit와 1024bit에서 1 thread 및 10 thread를 통해 성능평가를 수행하였다. key size를 512bit와 1024bit로 선택한 이유는 paillier 암호화 시스템에서 가장 널리 사용되는 key size이기 때문이다[5,17]. 1 thread 상에서 key size가 512인 경우 질의 처리 시간이 56초, 1024인 경우 질의 처리 시간이 220초가 소요된다. 10 thread 상에서 key size가 512인 경우 17초, 1024인 경우 37초가 소요된다. 따라서 1 thread 및 10 thread 상에서 1024 key size는 512 key size에 비해 약 3배 느린 성능을 보인다. 이를 통해 key size가 증가하면 프라이버시 보호 측면에서 안정성이 높아지지만 질의 처리 성능이 저하되는 특징을 보인다. 따라서 클라우드 컴퓨팅 환경에서 프라이버시 보호 측면과 질의 처리 성능 측면에서 가장 적절한 key size를 선택하는 것이 필요하다.

## 7. 결론 및 향후 연구

아웃소싱 데이터베이스에 기반한 클라우드 컴퓨팅 상에서 프라이버시 보호를 지원하는 영역 질의처리 연구가 활발히 수행되고 있다. 그러나 기존 연구는 높은 데이터 보호, 질의 보호, 접근 패턴 보호를 지원하지만 처리 속도가 늦다는 단점을 지닌다. 따라서 본 논문에서는 garbled 서킷 및 병렬 처리를 수행하여 높은 질의처리 성능을 지원하는 병렬 영역 질의처리 알고리즘을 제안하였다. 또한 제안하는 알고리즘이 정보 보호 측면에서의 보안 분석을 통해 높은 보안 수준을 제공함을 증명하였다. 마지막으로 성능평가를 통해, 제안하는 알고리즘이 10 thread 상에서 기존 기법에 비해 약 24배의 질의처리 성능 향상이 있음을 보였다.

향후 연구로는 다양한 질의(예를 들면 k-NN 질의)를 위한 프라이버시 보호를 지원하는 병렬 질의처리 알고리즘을 연구하는 것이다.

## References


[1] B. Hayes, "Cloud computing," pp.9-11, 2008.

[2] C. Bing, "Atos, IT provider for winter olympics, hacked months before opening ceremony cyberattack," 2018. [Internet], https://www.cyberscoop.com/atos-olympics-hackolympic-destroyer-malware-peyongchang/.

[3] 김재광, "개인정보보호법에 관한 새로운 법적 문제," *강원법학*, Vol.36, pp.95-120, 2012.

[4] W. Wu, Wei, X. Ming, P. Udaya, and L. Bin, "Efficient privacy-preserving frequent itemset query over semantically secure encrypted cloud database," *World Wide Web*, Vol.24, No.2, pp.607-629, 2021.

[5] W. Wu, P. Udaya, L. Jian, and X. Ming, "Privacy preserving k-nearest neighbor classification over encrypted database in outsourced cloud environments," *World Wide Web*, Vol.22, No.1, pp.101-123. 2019.

[6] H. Dai, J. Yan, Y. Geng, H. Haiping, and Y. Xun, "A privacy-preserving multi-keyword ranked search over encrypted data in hybrid clouds," *IEEE Access*, Vol.8, pp.4895-4907, 2019.

[7] C. S. H. Eom, C. C. Lee, W. Lee, and C. K. Leung, "Effective privacy preserving data publishing by vectorization," *Information Sciences*, Vol.527, pp.311-328, 2020.

[8] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "Accountable privacy preserving attribute based framework for authenticated encrypted access in clouds," *Journal of Parallel and Distributed Computing*, Vol.135, pp.1-20, 2020.

[9] L. Xu, C. Y. Weng, L. P. Yuan, M. E. Wu, R. Tso, and H. M. Sun, "A shareable keyword search over encrypted data in cloud computing," *The Journal of Supercomputing*, Vol.74, No.3, pp.1001-1023, 2018.

[10] B. U. Pagel, H. W. Six, H. Toben, and P. Widmayer, "Towards an analysis of range query performance in spatial data structures," *Proceedings of the twelfth ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems*, pp.214-221, 1993.

[11] P. Wang and C. V. Ravishankar, "Secure and efficient range queries on outsourced databases using R-trees," In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp.314-325, 2013.

[12] B. Wang, Y. Hou, M. Li, H. Wang, and H. Li, "Maple: Scalable multi-dimensional range search over encrypted cloud data with tree-based index," *ACM symposium on Information, Computer and Communications Security*, pp.111-122, 2014.

[13] H. I. Kim, S. T. Hong, and J. W. Chang. "Hilbert curve-based cryptographic transformation scheme for spatial query processing on outsourced private data," *Data & Knowledge Engineering*, Vol.104, pp.32-44, 2016.

[14] H. J. Kim, H. I. Kim, J. W. Chang, "Secure Range Query Processing Algorithm on Outsourced Database Environment," *The Journal of Korean Institute of Next Generation Computing*, Vol.12, No.4, 2016, pp.71-88.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," In *International Conference on the Theory and Applications of Cryptographic Techniques*, pp.223-238, 1999.

[16] Y. Watanabe, J. Shikata, and H. Imai, "Equivalence between semantic security and indistinguishability against chosen ciphertext attacks," *International Workshop on Public Key Cryptography*, Springer, Berlin, Heidelberg, 2003.

[17] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "*K*-nearest neighbor classification over semantically secure encrypted relational data," *IEEE Transactions on Knowledge and Data Engineering*, Vol.27, No.5, pp.1261-1273, 2014.

[18] C. M. Schneider, A. A. Moreira, J. S. Andrade, S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proceedings of the National Academy of Sciences*, Vol.108, No.10, pp.3838-3841, 2011.

[19] R. Cramer and I. B. Damgård, "Secure multiparty computation," Cambridge University Press, 2015.

[20] A. C. Yao, "How to Generate and Exchange Secrets," In *27th Annual Symposium on Foundations of Computer Science*, IEEE pp.162-167, 1986.

[21] "GMP «Arithmetic without limitations»" The GNU Multiple Precision Arithmetic Library [Internet], https://gmplib.org/

**김 형 진**

https://orcid.org/0000-0002-9363-3960
e-mail : yeon_hui4@jbnu.ac.kr
2015년 전북대학교 IT정보공학부(학사)
2017년 전북대학교 컴퓨터공학부(학사)
2017년 ~ 현 재 전북대학교 컴퓨터공학부 박사과정
관심분야 : 데이터베이스, 데이터 마이닝, 딥러닝, 트랜잭셔널 메모리

**장 재 우**

https://orcid.org/0000-0002-0037-6812
e-mail : jwchang@jbnu.ac.kr
1984년 서울대학교 전자계산기공학과(학사)
1986년 한국과학기술원 전산학과(석・박사)
1991년 ~ 현 재 전북대학교 IT정보공학과 교수
관심분야 : 데이터베이스, 하부 저장구조, 데이터 마이닝, 데이터베이스 아웃소싱