암호화 데이터베이스 상에서의 효율적인 영역 질의처리 알고리즘

최문철*, 김형일*, 장재우**†
*전북대학교 컴퓨터공학과, **전북대학교 IT정보공학과
e-mail:jwchang@jbnu.ac.kr

An Efficient Range Query Processing Algorithm on Encrypted Databases

Mun-Chul Choi*, Hyeong-Il Kim*, Jae-Woo Chang**†
*Dept of Computer Engineering, Chonbuk University
**Dept of Information Technology and Engineering, Chonbuk University

요 약

최근 클라우드 컴퓨팅에 대한 관심이 고조됨에 따라, 아웃소싱된 암호화 데이터베이스 상에서의 영역 질의처리 연구가 활발히 진행되고 있다. 그러나 기존 연구 중 데이터 접근 패턴 보호를 지원하는 연구는 전무하다. 따라서 본 논문에서는 데이터 보호, 사용자 질의 보호 및 데이터 접근 패턴 보호를 모두 지원하는 암호화 데이터베이스 상에서의 영역 질의처리 알고리즘을 제안한다. 성능평가를 통해, 제안하는 기법이 정보보호를 지원하는 동시에 효율적인 처리 성능을 제공함을 보인다.

1. 서론

최근 클라우드 컴퓨팅에 대한 연구가 활성화됨에 따라 데이터베이스 아웃소싱에 대한 관심이 고조되고 있다. 데이터베이스 아웃소싱이란 데이터 소유자가 데이터베이스 관리를 전문적으로 수행하는 클라우드에게 자신의 데이터베이스를 아웃소싱 하는 것을 말한다. 클라우드는 아웃소싱된 데이터베이스에 대한 저장 및 관리를 수행할 뿐 아니라, 데이터 소유자를 대신하여 인증된 사용자에게 해당 데이터베이스를 기반으로 질의처리 서비스를 제공한다. 데이터 소유자는 데이터베이스를 클라우드에 아웃소싱 함으로써 클라우드에서 제공하는 고성능의 컴퓨팅 리소스를 필요한 만큼 유연하게 활용 가능한 장점이 존재한다.

그러나 데이터베이스는 데이터 소유자에게 있어 소중한 자산이며 신상 정보 등 민감한 정보를 포함할 수 있기 때문에, 데이터베이스를 가공 없이 클라우드에게 아웃소싱할 경우 이를 악용하는 사례가 발생할 수 있다. 예를 들어, 부동산 관련 데이터베이스가 가공 없이 아웃소싱될 경우, 클라우드가 해당 데이터를 상업적으로 이용할 수 있다. 따라서 데이터베이스를 암호화 알고리즘을 활용하여 은닉한 후, 이를 클라우드에 아웃소싱하는 연구가 진행되었다. 해당 연구들은 암호화된 데이터베이스 및 질의를 기반으로 서비스를 제공하기 때문에, 데이터 보호 및 사용자 질의 보호를 지원할 수 있다.

한편, 영역 질의는 영역 혹은 범위 내에 존재하는 모든 데이터를 탐색하는 질의로써, 데이터 마이닝, 위치 기반 서비스 등 다양한 분야에서 폭넓게 활용된다. 한편, 영역 질의 결과는 사용자의 선호도 등 개인정보와 높은 연관성을 갖기 때문에, 정보 보호를 지원하기 위한 암호화 데이터베이스 상에서의 영역 질의처리 연구가 진행되어 왔다[1-8]. 그러나 기존 암호화 데이터베이스 상에서의 영역 질의처리 알고리즘 중 데이터 접근 패턴 보호를 지원하는 연구는 전무하다. 즉, 질의 처리 시 탐색된 노드 및 데이터의 식별자가 노출됨으로써, 클라우드는 사용자와 연관성이 높은 데이터를 판단할 수 있다.

이러한 문제점을 해결하기 위해서 본 논문에서는 데이터 보호, 사용자 질의 보호 및 데이터 접근 패턴 보호를 모두 지원하는 암호화 데이터베이스 상에서의 영역 질의처리 알고리즘을 제안한다. 아울러, 본 연구에서는 데이터 접근 패턴 노출없이 데이터 필터링을 수행하는 암호화 인덱스 탐색 기법을 통해 높은 질의처리 효율을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 암호화된 데이터베이스 상에서의 영역 질의처리 연구를 소개한다. 3장에서는 전체적인 시스템 구조 및 암호화 연산 프로토콜에 대해서 기술한 후, 제안하는 암호화 데이터베이스 상에서의 영역 질의처리 알고리즘에 대해 설명한다. 4장에서는 제안하는 질의처리 알고리즘의 성능평가를 수행한다. 마지막으로, 5장에서는 결론 및 향후 연구를 제시한다.

2. 관련 연구

기존 대표적인 암호화 데이터베이스 상에서의 영역 질의처리 연구는 다음과 같다. A. Boldyreva et al.의 연구[1]는 MOPE(Modular Order-Preserving Encryption) 암호화 기법 및 이를 활용한 영역 질의처리 수행 기법을 제시

† Corresponding author
이 논문은 2014년 교육부와 한국연구재단의 지역혁신창의인력양성사업의 지원을 받아 수행된 연구임(NRF-2014H1C1A1065816)

하였다. 그러나 해당 기법은 데이터의 순서 정보가 노출되는 문제점을 보인다.Y. Lu의 연구[2]는 단일 차원 및 다차원에서의 영역 질의처리 기법을 각각 제안하였다. 그러나 다차원 영역 질의처리 시, 토큰을 각 단일 차원으로 분해함으로써 클라우드는 각 차원에서의 데이터 값에 대한 정보를 유추할 수 있는 문제점을 보인다.

한편, 그룹화된 데이터 기반의 인덱스를 활용하여 데이터의 순서 정보 보호 및 질의 효율성을 지원하는 연구들이 제안되었다[3-9]. M. Yiu et al.의 연구[3]는 AES 기반의 암호화된 R-tree를 활용한 영역 질의처리 알고리즘을 제안하였다. H. Hu et al.의 연구[4]는 암호화된 데이터 상에서 다양한 연산을 지원하는 암호화 기법 및 R-트리 기반의 암호화 인덱스를 활용한 영역 질의처리 알고리즘을 제안하였다. B. Hore et al.의 연구[5]는 버켓(bucket) 단위로 그룹화된 데이터 기반의 인덱스를 활용하여 영역 질의처리를 수행한다. P. Wang et al.의 연구[6] 및 B. Wang et al.의 연구[7]는 트리 기반의 인덱스를 활용한 영역 질의처리 알고리즘을 제안하였다. 마지막으로, H. Kim et al.의 연구[8]는 힐버트 커브 기반의 암호화 인덱스를 활용한 영역 질의처리 알고리즘을 제안하였다. 그러나 [3-6, 8]는 데이터 그룹 단위로 질의 결과가 반환됨에 따라 다수의 false-positive 결과가 포함되는 문제점을 보인다. 이는 사용자에게 반환되지 않아야 하는 데이터가 추가적으로 노출되는 것이기 때문에, 정보 보호 측면에서 매우 취약하다고 할 수 있다. 또한, 는 데이터 소유자 혹은 사용자가 인덱스의 저장 및 탐색을 담당하기 때문에, 아웃소싱 데이터베이스의 목적에 위배되는 문제점을 보인다[3-5, 8].

특히, 기존 암호화 데이터베이스 상에서의 영역 질의처리 알고리즘 연구 중 데이터 접근 패턴 보호를 지원하는 연구는 전무하다. 즉, 질의 처리 과정에서 노드 및 데이터의 식별자가 클라우드에게 노출되는 문제점을 보인다.

3. 암호화 데이터베이스 상에서의 영역 질의처리 알고리즘

본 장에서는 먼저 전체적인 시스템 구조 및 암호화 연산 프로토콜을 제시한다. 다음으로, 이를 기반으로 암호화 데이터베이스 상에서의 효율적인 영역 질의처리 알고리즘을 제안한다.

3.1 시스템 구조

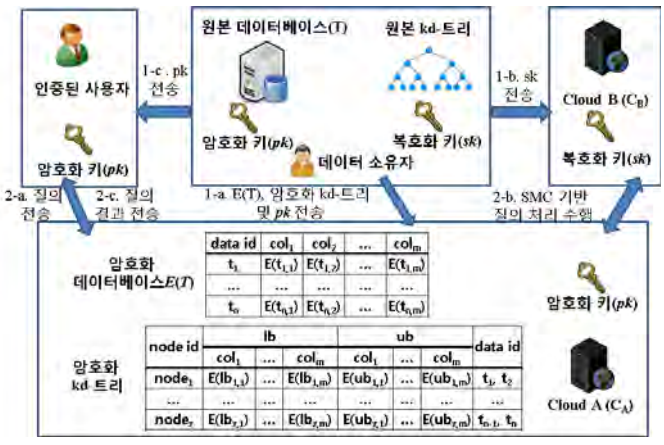
<그림 1>은 본 논문의 시스템 구조를 나타낸다. 시스템은 데이터 소유자, 두 개의 클라우드 (C_A, C_B), 인증된 사용자로 구성된다. 데이터 소유자는 n개의 레코드로 구성된 원본 데이터베이스(T)를 보유하고 있다. 각 레코드는 m개의 속성(attribute)으로 구성되며, i번째 레코드의 j번째 속성은 $t_{i,j}$와 같이 표기한다. 데이터 소유자는 해당 데이터베이스에 대한 색인을 지원하기 위해 kd-트리 기반의 데이터 분할을 수행한다. kd-트리의 말단 노드는 해당 노드가 담당하는 각 속성 별 영역 정보($lb_{z,j}$, $ub_{z,j}|1\le z\le 2^{h-1}$, $1\le j\le m$) 및 해당 말단 노드 영역 내에 포함된 데이터의 id를 저장한다. 여기서 lbz,m과 ubz,m은 말단 노드가 담당하는 영역의 각 속성별 하계점(lower bound) 및 상계점(upper bound)을 의미하며, h는 구축된 kd-트리의 레벨을 의미한다. 데이터베이스 암호화는 Paillier 암호화 시스템[9]을 기반으로 수행한다. Paillier 암호화 시스템은 다음과 같은 특성을 기반으로 다양한 암호화 연산 및 정보 보호를 지원 가능하다.

- 준동형 암호화 덧셈 : 두 암호화 데이터 E(a), E(b)의 곱 E(a)×E(b)은 a+b의 암호화 값 E(a+b)과 같다.
- 준동형 암호화 곱셈 : 암호화 데이터 E(a)의 b 제곱 $E(a)^b$은 a×b의 암호화 값 E(a×b)과 같다.
- 의미적 보안 지원 : 동일한 데이터에 대해 다양한 암호화 값이 존재한다.

이를 위해 데이터 소유자는 암호화 공개키(pk) 및 복호화 비밀키(sk)를 생성하고, *pk*를 이용하여 각 레코드의 속성 단위($E(t_{i,j})|1\le i\le n,\ 1\le j\le m$)로 암호화를 수행한다. 또한, 클라우드가 암호화 데이터베이스 상에서 효율적으로 질의처리를 수행할 수 있도록 지원하기 위해, 구축된 kd-트리의 말단 노드를 속성 별로 암호화 한다.

해당 시스템에는 서로 결탁하지 않는 두 개의 클라우드 C_A, C_B가 존재하며, C_A와 C_B는 모두 semi-honest 하다고 가정한다. 즉, C_A 및 C_B는 질의 처리를 위해 자신이 담당해야 하는 프로토콜은 정직하게 수행하지만, 질의 처리 과정 중에 노출되는 정보를 바탕으로 데이터 소유자 및 질의 요청자에 대한 추가적인 정보를 획득하기 위한 시도를 수행할 수 있다. 데이터 소유자는 암호화 데이터베이스 및 암호화 kd-트리를 암호화 키(pk)와 함께 클라우드 C_A에 아웃소싱 한다. 이 때, 암호화 kd-트리의 각 노드 영역에 포함되는 데이터 id를 평문의 형태로 함께 아웃소싱한다. 한편, 데이터 소유자는 클라우드 C_B에게는 복호화 키(sk)를 전송한다. 또한, 데이터 소유자는 질의 처리 과정을 위해 인증된 사용자에게 암호화 키를 전송한다.

인증된 사용자는 질의 요청 시, 질의 영역을 구성하는



(그림 1) 전체 시스템 구조

하계점 및 상계점($q.lb_j$, $q.ub_j|1\leq j\leq m$)을 암호화하여 C_A에게 전송한다. C_A는 C_B와 함께 다자간 계산(SMC : Secure Multiparty Computation)을 통해 안전하게 영역 질의처리를 수행한 후, 질의 결과를 사용자에게 전송한다.

3.2 암호화 연산 프로토콜

제안하는 암호화 데이터 상에서의 영역 질의처리 알고리즘은 다양한 암호화 연산 프로토콜을 기반으로 수행된다. 본 절에서는 기 제안된 암호화 연산 프로토콜에 대해서 간략히 소개하고 [10, 11], 데이터 접근 패턴 보호를 위해 새로운 세 가지 암호화 연산 프로토콜을 제안한다.

SM(Secure Multiplication) 프로토콜은 암호화 된 두 데이터 E(a)와 E(b)가 주어졌을 때, a*b의 암호화 값 E(a*b)를 계산한다. SSED(Secure Squared Euclidean Distance) 프로토콜은 암호화 된 두 벡터 E(X)와 E(Y)가 주어졌을 때, 두 벡터 간 거리의 제곱 $E(|X-Y|^2)$을 계산한다. SBOR(Secure Bit-OR) 프로토콜은 두 개의 1 비트 암호화 데이터 $E(o_1)$과 $E(o_2)$가 주어졌을 경우, 두 암호화 비트에 대한 OR 연산을 수행한다. 한편, SBD(Secure Bit Decomposition) 프로토콜은 암호화 데이터 E(a)가 주어졌을 때, a의 암호화 이진수 값 [a]를 계산한다.

본 논문에서 새로이 제안하는 암호화 연산 프로토콜을 제시한다. 첫째, SBN(Secure Bit-Not) 프로토콜은 1 비트의 암호화 데이터 E(a)가 주어졌을 때, Not 연산을 수행한다. SBN의 수행과정은 다음과 같다. i) C_A는 준동형 암호화의 특성을 이용하여 $E(a)^{-1} = E(-a)$을 계산한다. ii) C_A는 $E(-a)\times E(1)$을 통해 $E(-a+1)$을 계산함으로써, E(a)에 대한 NOT 연산 결과를 계산할 수 있다.

둘째, SCMP (Secure Compare) 프로토콜은 두 개의 암호화 이진수 [u], [v]가 주어졌을 때, $u\leq v$를 만족하는 경우 E(1)을 반환하고, $u>v$인 경우 E(0)을 반환한다. SCMP의 수행과정은 다음과 같다. i) C_A는 u, v의 최하위 비트에 각각 E(0), E(1)을 첨가한다. ii) C_A는 두 개의 $F(F_0{:}u>v,\ F_1{:}v>u)$ 중 임의로 하나를 선택한 후, SM을 이용하여 $E(u_i*v_i)$를 계산한다. 단, F_0과 F_1중 무엇이 선택되었는지는 C_B에게 공개되지 않는다. iii) C_A는 선택된 F에 따라 다음과 같이 W_i를 계산한다.

- F_0 : $u > v$ 가 선택된 경우,

$$W_i = E(u_i)*E(u_i*v_i)^{N-1} = E(u_i*(1-v_i))$$

- F_1 : $v > u$ 가 선택된 경우,

$$W_i = E(v_i)*E(v_i*u_i)^{N-1} = E(v_i*(1-u_i))$$

iv) C_A는 $E(u_i)$와 $E(v_i)$의 XOR 결과를 G_i에 저장하고, $H_i=(H_i-1)r_i*G_i$ 및 $\Phi i=E(-1)*H_i$를 계산한다. 여기서 r_i는 랜덤 값을 의미하고, $H_0=E(0)$으로 설정한다. 이를 통해, G_i에서 E(1)이 처음 등장하는 위치(j), 즉 u와 v의 대수 판별이 가능한 위치에 $H_j=E(1)$ 및 $\Phi_j=E(0)$이 저장된다. v) C_A는 $L_i = W_ir_i*\Phi_ir_i$를 계산함으로써 u와 v의 대수 판별이 가능한 위치에서 $u\geq v$의 결과에 대한 정보를 L_i에 저장한다. vi) C_A는 임의의 순서 변경 함수 π_1를 생성하여 L의 순서를 변경하고, 이를 C_B에게 전송한다. vii) C_B는 L′를 복호화한 후, 0의 존재 여부를 확인한다. 만약 0이 존재한다면 E(α)=E(0), 그렇지 않다면 E(α)=E(1)으로 설정한 후, E(α)를 C_A에게 전송한다. viii) C_A는 F_0한해, SBN(E(α))를 수행한다. ix) 최종적으로 계산된 E(α)를 반환하고 프로토콜을 종료한다.

셋째, SRO (Secure Range Overlapping) 프로토콜은 하계점(lb) 및 상계점(ub)으로 표현된 두 개의 암호화 영역 정보 range1, range2가 주어졌을 때, 두 영역이 겹치는 경우 E(1)을, 겹치지 않는 경우 E(0)을 반환한다. SRO의 수행과정은 다음과 같다. i) C_A는 E(α) 값을 E(1)으로 초기화 한다. ii) C_A는 range1의 하계점과 range2의 상계점의 각 차원 데이터를 기반으로 SCMP를 수행한 후, SCMP의 반환 결과와 E(α)를 이용해 SM을 수행한다. iii) CA는 range2의 하계점과 range1의 상계점의 각 차원 데이터를 기반으로 SCMP를 수행한 후, SCMP의 반환 결과와 E(α)를 이용해 SM을 수행한다. 여기서, SCMP에서 반환된 값이 모두 E(1)인 경우에만, α의 값이 E(1)로 유지된다. iv) 최종적으로 계산된 E(α)를 반환하고 프로토콜을 종료한다.

3.3 암호화 데이터 기반 영역 질의처리 알고리즘

본 절에서는 제안하는 암호화 데이터 상에서의 영역 질의처리 알고리즘에 대해 설명한다. 제안하는 알고리즘은 크게 암호화 인덱스 탐색 단계와 데이터 탐색 단계로 구성된다. 먼저 암호화 인덱스 탐색 단계의 수행 과정은 다음과 같다. i) C_A는 SBD를 이용하여 사용자로부터 전송받은 암호화된 질의 영역을 암호화 이진수 [q]로 변환한다. ii) C_A는 각 노드의 영역 정보를 SBD를 이용하여 암호화 이진수 [node]로 변환하고, [q] 및 [node]를 기반으로 SRO를 수행한다. iii) C_A는 임의의 순서 변경 함수 π_1를 생성하여 SRO에서 반환된 E(α)의 순서를 변경하여 E(α′)를 생성하고, 이를 C_B에게 전송한다. iv) C_B는 E(α′)를 복호화하여 1의 개수(c)를 확인하고, c개의 노드 그룹을 생성한다. C_B는 각 노드 그룹에 α′=1인 노드 한 개와 α′=0인 노드 $num_{node}/c-1$개를 할당한다. 아울러, 각 노드 그룹에 할당된 노드의 순서를 임으로 변경한 후, 이를 C_A에게 전송한다. v) C_A는 역변경 함수 π_1^{-1}를 이용하여 각 노드 그룹에 속한 노드들의 식별 번호를 역변경한다. vi) C_A는 노드 그룹 별 노드에 저장된 데이터 및 SRO를 통해 반환된 각 노드의 E(α)를 이용해 SM을 수행한다. 또한, SM을 통해 반환된 결과를 준동형 암호화 특성을 이용하여 모두 더한 후 E(cand)에 저장한다. 이를 통해, 질의 영역과 겹치는 노드가 저장하고 있는 모든 데이터가 데이터 접근 패턴 노출없이 E(cand)에 저장된다. vii) 마지막으로 E(cand)를 반환하고 프로토콜을 종료한다.

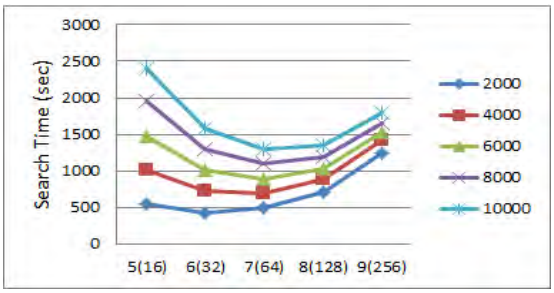
한편 데이터 탐색 단계의 수행 과정은 다음과 같다. i) C_A는 SBD를 이용해 E(cand)에 저장된 데이터를 암호화 이진수로 [cand]로 변환한다. ii) C_A는 [cand] 및 [q]를 기반으로 SRO를 수행한다. 단, [cand]는 하계점 및 상계점

이 동일한 영역으로 간주하며, SRO의 반환값인 E(α)는 C_B에 의해 복호화되어 C_A에게 전송된다. iii) SRO 수행 결과 1이 반환된 경우, C_A는 해당 데이터를 최종 결과 E(result)에 저장한다. 이 때, [cand]에 저장된 각 데이터는 어느 노드로부터 추출된 것인지 알 수 없기 때문에, C_A 및 C_B는 반환되는 질의 결과의 실제 데이터 식별자를 알지 못한다. iv) C_A는 E(result)에 랜덤 값 r을 더한 후, E(result+r)은 C_B에게 전송하고, r은 사용자에게 전송한다. v) C_B는 전송받은 데이터를 복호화하여 result+r을 획득하고, 이를 사용자에게 전송한다. vi) 마지막으로 사용자는 C_A 및 C_B에게 전송받은 데이터를 가산하여, 최종 질의결과인 result를 획득한다.

4. 성능평가

본 장에서는 암호화 데이터베이스 상에서의 영역 질의처리 알고리즘에 대한 성능평가를 수행한다. 기존 기법 중 데이터 보호, 사용자 질의 보호 및 데이터 접근 패턴 보호를 모두 지원하는 연구는 존재하지 않기 때문에, 제안하는 기법의 자체 성능을 측정하였다. 제안하는 기법은 C++로 구현하였으며, 임의로 설정한 사각형의 영역을 질의로 설정하였다. 성능 평가는 Linux ubuntu 14.04.2의 환경에서 Intel Xeon E3-1220v3 4-Core 3.10GHz, 32GB(8GB × 4개) DDR3 UDIMM 1600MHz를 기반으로 수행하였다. 또한, 성능평가를 위해 2k~10k개의 임의(Random) 데이터를 생성하였으며, 데이터의 차원은 6차원, 데이터 도메인의 크기는 12로 설정하였다. 또한, kd-트리 레벨을 5~9로 변경하며 성능평가를 수행하였다.

<그림 2>은 kd-트리의 레벨 변화에 따른 영역 질의처리 알고리즘의 성능을 나타낸다. 전체적으로 총 데이터 수가 증가함에 따라 질의 처리 시간이 증가함을 확인할 수 있다. 이는 데이터 수가 증가할수록 탐색 노드 및 데이터의 수가 증가하기 때문이다. 한편, 전체적으로 kd-트리의 레벨이 5~7인 경우에는 트리 레벨이 증가함에 따라 질의처리 시간이 감소하고, kd-트리의 레벨이 7~9인 경우에는 트리 레벨이 증가함에 따라 질의처리 시간이 증가한다. 이는 kd-트리의 레벨 변화에 따라 다음과 같은 특성이 존재하기 때문이다. 첫째, kd-트리의 레벨이 높아질수록 말단 노드가 저장하고 있는 데이터의 수가 감소한다. 따라서 선택된 노드 내 데이터 중 실제 질의 영역에 포함된 데이터를 탐색하기 위한 비용이 감소한다. 둘째, kd-트리의 레벨이 높아질수록 말단 노드의 수가 증가한다. 따라서 질의를 포함하는 노드를 찾기 위한 SRO 프로토콜의 수행 시간이 증가하기 때문이다.



(그림 2) 각 데이터 크기에 대한
kd-트리 레벨에 따른 영역 질의 처리 성능

5. 결론 및 향후 연구

암호화 데이터베이스 상에서의 영역 질의처리 연구가 활발히 수행되고 있다. 그러나 기존 연구 중 데이터 접근 패턴 보호를 지원하는 연구는 전무하다. 따라서 본 논문에서는 데이터 보호, 사용자 질의 보호 및 데이터 접근 패턴 보호를 모두 지원하는 암호화 데이터베이스 상에서의 영역 질의처리 알고리즘을 제안하였다. 제안하는 기법은 데이터 접근 패턴 보호를 지원하는 암호화 인덱스 탐색 기법을 기반으로 영역 질의처리를 수행한다. 성능평가를 통해, 제안하는 기법이 정보보호를 지원하는 동시에 효율적인 질의처리 성능을 보임을 검증하였다.

향후 연구는 다양한 파라미터를 이용하여 제안하는 알고리즘의 성능 분석을 수행하는 것이다.

참고문헌

[1] A. Boldyreva et al. "Order-preserving encryption revisited: Improved security analysis and alternative solutions." Proceedings of the CRYPTO (2011): 578-595.
[2] Y. Lu et al. "Privacy-preserving Logarithmic-time Search on Encrypted Data in Cloud." In Proc. of NDSS. 2012.
[3] M. Yiu et al. "Enabling search services on outsourced private spatial data." In Proc. of VLDB Journal 19.3 (2010): 363-384.
[4] H. Hu et al. "Processing private queries over untrusted data cloud through privacy homomorphism." In Proc. of ICDE, 2011.
[5] B. Hore et al. "Secure multidimensional range queries over outsourced data." In Proc. of VLDB Journal 21.3 (2012): 333-358.
[6] P. Wang et al. "Secure and efficient range queries on outsourced databases using rp-trees." In Proc. of ICDE, 2013.
[7] B. Wang et al. "Maple: scalable multi-dimensional range search over encrypted cloud data with tree-based index." In Proc. of ASIACCS, 2014.
[8] H. Kim et al. "Hilbert-curve based cryptographic transformation scheme for protecting data privacy on outsourced private spatial data." In Proc. of BigComp, 2014.
[9] P. Paillier et al. "Public-key cryptosystems based on composite degree residuosity classes." In Proc. of EUROCRYPT, 1999.
[10] Y. Elmehdwi et al. "Secure k-nearest neighbor query over encrypted data in outsourced environments." In Proc. of ICDE, 2014.
[11] B. Samanthula et al. "An efficient and probabilistic secure bit-decomposition." In Proc. of ASIACCS, 2013.