클라우드 상에서 정보 보호를 지원하는 garbled circuit 기반 병렬 영역 질의처리 알고리즘

김형진*, 장재우**
*전북대학교 컴퓨터공학과
**전북대학교 IT정보공학과, 교신저자
yeon_hui4@jbnu.ac.kr, jwchang@jbnu.ac.kr

Privacy-preserving Parallel Range Query Processing Algorithm based on Garbled Circuit in Cloud Computing

Hyeong-Jin Kim*, Jae-Woo Chang**
*Dept. of Computer Engineering, Chonbuk National University
**Dept. of Information and Engineering, Chonbuk National University

요 약

최근 클라우드 컴퓨팅이 발전함에 따라 데이터베이스 아웃소싱에 대한 관심이 증가하였다. 그러나 데이터베이스를 아웃소싱하는 경우, 데이터 소유자의 정보가 내외부 공격자에게 노출되는 문제점을 지닌다. 따라서 본 논문에서는 정보 보호를 지원하는 병렬 영역 질의처리 알고리즘을 제안한다. 제안하는 알고리즘은 garbled circuit 및 thread pool을 통해 암호화 연산 프로토콜의 효율성을 향상시키고, 알고리즘의 처리과정을 병렬화함으로써 높은 질의 처리 성능을 제공한다. 성능평가를 통해, 제안하는 알고리즘이 고수준의 정보 보호를 지원하는 동시에 기존 알고리즘에 비해 약 20배의 우수한 질의 처리 성능을 보인다.

1. 서론

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·관리할 뿐만 아니라 다양한 질의처리 서비스를 제공한다. 데이터 소유자는 데이터베이스를 위한 전산 자원 및 인력 자원을 절약할 수 있는 장점을 가진다. 그러나 데이터베이스를 아웃소싱하는 경우, 데이터 소유자의 민감한 데이터가 위탁 기업 및 클라우드에 그대로 노출되는 문제점을 지닌다.

한편, 영역 질의는 데이터베이스에서 자주 수행되는 대표적인 질의로써 금융, 보험, 의료 분야 등 다양한 분야에서 사용된다[1]. 그러나 데이터의 보호 없이 영역 질의를 수행하게 될 경우, 데이터 소유자의 민감한 정보가 클라우드에 그대로 노출되어 금전적 손실과 법적 처벌을 받을 수 있다[2]. 따라서 클라우드 컴퓨팅 상에서 정보 보호를 지원하는 영역 질의처리 알고리즘의 연구가 필요하다.

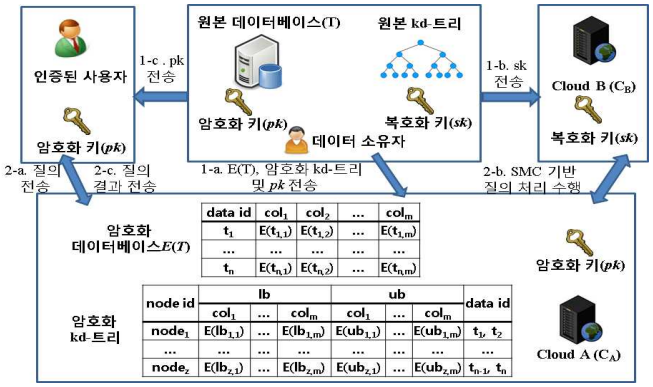
2. 관련 연구

클라우드 상에서 정보 보호를 지원하는 영역 질의 연구는 다음과 같다. 첫째, B. Wang et al.의 연구[3]는 트리 기반 인덱스를 기반으로 정확한 질의 결과를 보장하는 영역 질의처리 알고리즘을 제안하였다. 그러나 해당 연구는 사용자 질의 및 결과가 노출된다는 단점이 존재한다. 둘째, H. Kim et al.의 연구[4]는 힐버트 커브(Hilbert-curve) 기반의 암호화 인덱스 및 영역 질의처리 알고리즘을 제안하였다. 그러나 해당 기법은 인덱스 탐색 등으로 인해 사용자 측에서 질의처리 비용이 높은 문제점이 존재한다. 또한, 해당 기법은 데이터 그룹을 기반으로 질의처리를 수행하기 때문에, 거짓양성(false positive) 결과가 포함될 수 있다. 셋째, H. Kim et al.의 연구[5]는 kd-트리 기반의 암호화 인덱스 및 영역 질의처리 알고리즘을 제안하였다. 해당 기법은 준동형 암호화 기법을 사용하여 데이터 보호, 질의 보호, 접근 패턴의 보호가 가능하다. 그러나 해당 기법은 준동형 암호화 기법을 사용하기 때문에 높은 질의처리 비용이 요구되는 문제점을 지닌다. 따라서 제안하는 알고리즘은 준동형 암호화 기법을 사용하여 고수준의 정보 보호를 제공할 뿐만 아니라, 높은 질의 처리 비용을 해결하기 위해 garbled circuit [6] 및 병렬 처리 기법을 적용하여 효율적인 연산을 수행한다.

3. 전체 시스템 구조

<그림 1>은 제안하는 기법의 시스템 구조를 나

타낸다. 데이터 소유자는 n개의 레코드 $t_i(1\le i\le n)$로 구성된 원본 데이터베이스(T)를 보유하고 있다. 각 레코드는 m개의 속성(attribute)으로 구성되며, i번째 레코드의 j번째 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. 구축된 kd-트리의 레벨이 h, 단말 노드의 총 수는 2^{h-1}, 한 노드가 저장할 수 있는 최대 데이터 수(fan out)는 F이다. 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 id를 저장한다. 데이터베이스 암호화는 Paillier 암호화 시스템(Paillier Crypto System)[7]을 기반으로 수행하며, 이를 위해 데이터 소유자는 암호화 공개키(pk) 및 복호화 비밀키(sk)를 <그림 1>과 같이 분배한다. 데이터 소유자는 암호화 키를 이용하여 데이터베이스 및 kd-트리의 암호화를 수행한다.



<그림 1> 전체 시스템 구조

한편, 해당 시스템에는 서로 결탁하지 않는 두 개의 클라우드 C_A, C_B가 존재하며, C_A와 C_B는 모두 semi-honest[8]하다고 가정한다. 한편, 암호화된 데이터베이스를 기반으로 다양한 질의처리를 지원하기 위해서는 C_A와 C_B 간의 안전한 다자간 계산(SMC : Secure Multiparty Computation)[8]이 요구된다.

4. 정보 보호를 지원하는 병렬 영역 질의 처리 알고리즘

본 절에서는 클라우드 컴퓨팅 상에서 정보 보호를 지원하는 병렬 영역 질의처리 알고리즘을 제안하며, 이는 $Range_{PI}$(Parallel Range query processing algorithm using garbled circuit and Index)로써 크게 병렬 인덱스 탐색 단계와 질의 영역 내 데이터 탐색 단계로 구성된다.

□ 수행단계 1 : 병렬 인덱스 탐색 단계

Algorithm 1은 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 처리가 가능하다.

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 $1 \le z \le num_{node}$
03. call thread_pool_push(GSPO(*E*(Q), *E*($node_z$)), *E*(a_z))
04. E(a') = π(E(a)); send E(a') to C_B

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

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

End Algorithm

첫째, C_A는 병렬 처리를 위해 thread pool을 생성하고 E(Q)와 E($node_z$)간에 GSRO 프로토콜[5, 6]을 작업 단위로 병렬 처리한다(line 1~3). 둘째, C_A는 순서 변경 함수 π를 생성하여 E(α)의 순서를 변경하고(e.g., E(α′)←π(E(α))), E(α′)를 C_B에게 전송한다(line 4). 셋째, C_B는 E(α′)를 복호화하여 α′를 획득한 후, 1의 개수, 즉 질의 영역과 겹치는 노드의 수(c)를 확인한 후, c개의 노드 그룹 Group을 생성한다. C_B는 각 노드 그룹에 α′=1 인 노드 한 개와 α′=0 인 노드 (num_{node}/c) - 1개를 할당한다. 이 때, 각 노드 그룹 별로 균등한 수의 노드가 할당되도록 한다. 아

울러, 각 노드 그룹에 할당된 노드의 순서를 랜덤하게 변환한 후, 이를 C_A에게 전송한다(line 5~12). 넷째, C_A는 자신이 생성한 순서 변경 함수의 역변경 함수 π^{-1}을 이용하여 각 노드 그룹에 속한 노드들의 식별 번호를 변경한다(line 13~15). 다섯째, C_A는 노드 그룹에 할당된 각 노드를 차례로 방문하고, 각 데이터에 대한 GSRO 프로토콜 및 SM 프로토콜[5, 6]을 하나의 작업 단위로 thead pool에 삽입한다. 각 thread는 순차적으로 thread pool에 존재하는 작업을 수행하며, SM 프로토콜 수행을 통해 반환된 결과값을 E(cand)로 반환한다. 마지막으로 암호화 인덱스 탐색 알고리즘은 질의 영역과 겹치는 노드 내에 존재하는 모든 데이터가 저장된 E(cand)를 반환하고 알고리즘을 종료한다.

□ 수행단계 2 : 질의 영역 내 데이터 탐색 단계

질의 영역 내 데이터 탐색 단계는 암호화 인덱스 탐색 단계에서 추출한 데이터를 기반으로 암호화 질의 영역에 실제로 포함되는 모든 데이터를 탐색한다. Algorithm 2는 질의 영역 내 데이터 탐색 단계의 수행 과정을 나타낸다. 첫째, C_A는 E(Q)와 E(cand)간에 GSRO 프로토콜[5, 6]을 작업 단위로 병렬적으로 처리한다. 단, E(cand)는 점 데이터이기 때문에, 상한점 및 하한점이 모두 E(cand)인 데이터로 설정하여 GSRO를 수행한다(line 1~2). 둘째, C_A는 난수 $r_{i,j}$를 생성한 후, thread pool에 $E(cand_i)\times E(r_{i,j})(1\le i\le cnt,\ 1\le j\le m)$을 작업 단위로 병렬 처리를 수행하고 연산 결과를 $E(\gamma_{i,j})$에 저장한다(line 3~7). 이후의 과정은 H. Kim 연구[5]의 과정과 동일하다(line 8~18).

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(GSPO(*E*(Q), *E*($node_z$)), *E*(a_z))
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. $E(a') \leftarrow \pi(E(a))$; $E(\gamma') \leftarrow \pi(E(\gamma))$
09. $r' \leftarrow \pi(r)$
10. send $E(a')$, $E(\gamma')$ to C_B and r' to user

C_B:

11. for $1 \le i \le cnt$
12. $a'_i \leftarrow D(E(a'_j))$
13. for $1 \le j \le m$
14. $\gamma'_{i,j} \leftarrow D(E(\gamma'_{i,j}))$
15. send a', γ' 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

5. 보안 분석

제안하는 병렬 영역 질의처리 알고리즘의 안전함을 증명하기 위해, C_B 측에서의 보안 분석을 수행한다. C_B측에서의 제안하는 기법 수행 이미지는 $\Pi C_B(pRange_{GI}) = \{<E(\alpha'), \alpha'>\}$와 같다. 여기서 $E(\alpha')$는 C_A로부터 전송받은 데이터이며, α'는 C_B가 $E(\alpha')$의 복호화를 통해 획득하는 데이터이다. C_B측에서의 제안하는 기법의 시뮬레이션 수행 이미지를 $\Pi C_{Bs}(pRange_{GI}) = \{<E(\beta'), \beta'>\}$라 하자. 여기서 $E(\beta')$는 $\mathbb{Z}_{N^2}$에서 생성된 난수이고, β'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_A가 임의 생성한 π는 C_B에게 공개되지 않기 때문에, α'는 β'로부터 계산적으로 구별 불가능하다. 한편, C_B는 α'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. 위 사항을 종합할 때, 제안하는 기법이 C_B에서 semi-honest 공격 모델 하에서 안전함을 보장할 수 있다.

6. 성능 평가

본 장에서는 클라우드 상에서 정보 보호를 지원하는 병렬 영역 질의처리 알고리즘에 대한 성능평가를 수행한다. H. Kim et. al.의 연구는 데이터 보호, 사용자 질의 보호 및 데이터 접근 패턴 보호를 모두 지원한다. 따라서 본 논문에서 제안한 기법($PRange_{GI}$)을 기존 H. Kim et. al.의 연구($SRange_I$)와 성능비교를 수행한다. 이를 통해, 제안하는 기법이 병렬적으로 처리됨에 따른 질의 처리 효율을 측정하였다. 해당 기법들은 C++로 구현하였으며, 임의로 설정한 사각형의 영역을 질의로 설정하였다. 또한, 질의 영역의 크기는 전체 도메인 크기의 0.1로 설정하여 수행하였다. 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. 다양한 데이터에서 성능평가를 수행하기 위해 임의 데이터를 생성하여 성능평가를 수행하였으며, 생성한 데이터의 매개변수는 <표 1>과 같다.

<표 1> 실험 매개변수

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

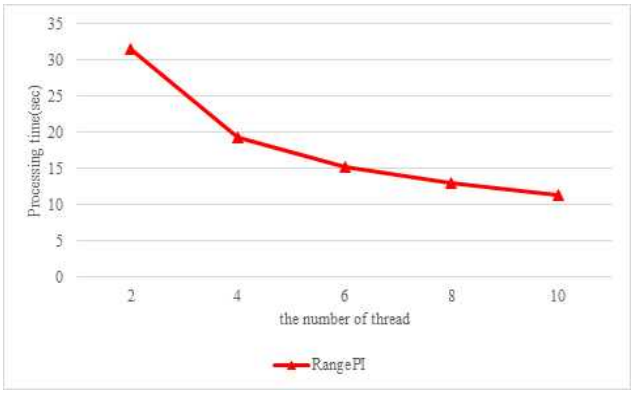
<그림 2>은 데이터 수(n) 변화에 따른 제안하는 기법의 질의처리 성능을 보인다. 데이터 수가 증가함에 따라 질의처리 시간도 증가함을 볼 수 있다. 1 thread 상에서 제안하는 기법($Range_{PI}$)은 기존 $Range_I$ [5]보다 평균 5.7배 좋은 성능을 보인다. 이는 garbled 서킷을 통해 암호화 연산 프로토콜의 효율적으로 처리하기 때문이다. 또한 병렬화를 통한 성능 향상을 측정하기 위해 thread를 2에서 10으로 증가하면서 성능평가를 수행하였다(<그림 3>). 제안하는 기법은 thread 수에 비례하여 성능이 향상됨을 알 수 있다. 아울러 10 thread 상에서 제안하는 기법과 기존 기법을 비교하면, 제안하는 $Range_{PI}$이 기존 기법인 $Range_I$에 비해 24배 성능 향상이 있음을 알 수 있다. 이는 제안하는 기법이 garbled circuit 및 병렬 처리 기법을 적용하여 효율적인 질의 처리가 가능하기 때문이다.

<그림 2> 데이터 수 변화에 따른 성능평가

7. 결론 및 향후 연구

클라우드 상에서의 정보 보호를 지원하는 영역 질의처리 연구가 활발히 수행되고 있다. 그러나 기존 연구는 높은 보호 수준을 제공하지만 처리 속도가 늦다는 단점을 지닌다. 따라서 기존 연구의 병렬처리를 수행하여 높은 성능을 지원하는 클라우드 상에서의 정보 보호를 지원하는 병렬 영역 질의처리 알고리즘을 제안하였다. 성능평가를 통해, 제안하는 기법이 정보 보호를 지원하는 동시에 효율적인 질의처리 성능을 보임을 검증하였다.

향후 연구는 다양한 환경에서의 제안하는 알고리즘의 성능 분석을 수행하는 것이다.



<그림 3> thread 수 변화에 따른 성능평가

Acknowledgement

이 논문은 2019년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 한국연구재단 과제번호 : NRF-2019R1I1A3A01058375)

참고문헌

[1] Pagel, Bernd-Uwe, et al. "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. 1993, p214-221.

[2] 김재광. "개인정보보호법에 관한 새로운 법적 문제." 강원법학 36, 2012, p95-120.

[3] Kim, Hyeong-Il, Seungtae Hong, and Jae-Woo Chang. "Hilbert curve-based cryptographic transformation scheme for spatial query processing on outsourced private data." Data & Knowledge Engineering 104, 2016, p32-44.

[4] 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, 2014, pp. 111-122.

[5] 김형진, 김형일, 장재우. "아웃소싱 데이터베이스 환경에서의 안전한 영역 질의처리 알고리즘." 한국차세대컴퓨팅학회 논문지, 12, 4, 2016, p71-88.

[6] A. C. Yao, "How to Generate and Exchange Secrets", Proc. 27th IEEE Symp. Foundations of Computer Science, 1986, pp. 162-167.

[7] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes", EUROCRYPT. Springer-Verlag, 1999, p223-238.

[8] Samanthula, B. K., Yousef Elmehdwi, and Wei Jiang. "K-nearest neighbor classification over semantically secure encrypted relational data." IEEE transactions on Knowledge and data engineering 27.5 (2014): 1261-1273.