DAS 모델에서 데이터베이스 검색에 적합한 고속 암복호화 메커니즘 설계

송유진*, 이동혁**, 이승민**, 남택용**
*동국대학교 전자상거래학과
**한국전자통신연구원 정보보호연구단
e-mail : jassbop@etri.re.kr

Design of Fast Encryption/Decryption Mechanism for Query in DAS(Database As a Service) Model

You-Jin Song*, Dong-Hyeok Lee**, Seung-Min Lee**, Taek-Yong Nam**
*Dept. of Electronic Commerce, Dongguk University
**Information Security Research Division, ETRI

요 약

데이터베이스를 아웃소싱하는 DAS 모델 환경에서 데이터베이스를 암호화하기 위해 암호화 알고리즘을 적용할 때, 암호화된 데이터의 순서는 평문과 달라, 인덱스를 구축할 수 없기 때문에 암호화된 데이터베이스에 대한 검색 처리상의 효율성 문제가 발생한다. 따라서, 아웃소싱된 데이터베이스 환경에 적합한 효율적인 암호화 메커니즘이 요구된다. 본 논문에서는 평문을 노출시키지 않은 상태에서 범위 검색이 가능한 새로운 고속 메커니즘을 제안하였다. 제안한 메커니즘은 복호화시 평문 데이터당 1 회의 XOR 연산과 버킷당 1 회의 암호화 연산만을 요구하므로 기 제안된 Hacigümüş 의 방식보다 효율적이다.

1. 서론

DAS 모델에서는 데이터를 특정한 하나의 서버에 저장하지 않고 외부에 아웃소싱된 복수의 데이터베이스를 사용한다. 따라서, DAS 모델에서는 민감 정보를 포함한 데이터의 기밀성을 위한 암호화가 필수적이라고 할 수 있다. 그러나 암호화 알고리즘을 아웃소싱된 데이터베이스에 적용할 경우, 암호화된 데이터의 순서는 평문과 달라지므로 인덱스를 구축할 수 없으며 효율성 측면에서 많은 저하를 가져온다. 이러한 방법을 극복하기 위하여 Hacigümüş 는 버킷팅 기법을 제안하였으며[6], Agrawal 등에 의해 순서 유지가 가능한 암호화 기법이 제안되어 암호화된 상태 그대로 인덱싱이 가능하고, 별도의 절차 없이 일치 및 범위 검색이 가능하게 되었다[1][8]. 그러나, 버킷팅 기법은 버킷 ID 에 대한 별도의 관리가 요구되는 번거로움이 있으며, 범위검색시 불필요한 어트리뷰트를 함께 가져오므로 재필터링과정에서 필요하지 않은 데이터에 대한 복호화가 요구된다. 또한, 순서유지 암호화 기법은 평문의 순서를 그대로 나타내게 되어, 결과적으로 추론 공격에 대하여 안전성을 보장할 수 없다. 특히, 순서를 나타내는 서열척도에 대한 암호화는 의미가 없게 된다. 한편, Damiani 는 충돌이 가능한 Hash 기반의 인덱싱을 제안하였으며[2][3], 데이터 빈도수 기반의 추론 공격(Inference Attack)에 대한 안전성을 유지할 수 있다. 그러나, Hash 기반의 인덱싱 기법은 범위검색을 할 수 없으며, 하나의 데이터에 대하여 다량의 동일한 Hash 정보를 가진 다른 어트리뷰트를 가져와야 하므로 버킷팅 기법의 문제를 해결할 수 없다. 동일한 논문에서, Damiani 는 이러한 문제를 해결하기 위해 범위 검색이 가능한 Auxiliary B+-Tree 기법을 제안하였다. 이 방법은, 암호화된 데이터에 대한 범위검색이 가능하며 안전성이 매우 높은 방법이나, 범위검색이 한번에 되지 않고 데이터 검색회수와 노드 수를 합한 만큼의 SQL 검색을 시도해야 하는 단점이 있어 효율성에 큰 문제가 있다.

효율성을 위해서는 순서를 유지하는 것이 바람직하나, 안전성 측면에서는 순서를 유지하게 되면 정보 노출의 위험이 매우 크다. 따라서, 본 논문에서는 버킷팅 기법의 장점을 그대로 유지하면서 복호화 속도를 대폭 향상시킨 새로운 암호화 알고리즘을 제안한다. 제안한 방법은 종래의 AES 암호화 알고리즘의 카운터 모드를 응용하여 버킷 아이디를 생성한 이후, 데이터에 XOR 을 취하는 방식으로, 복호화시 데이터당 1 회의 XOR 연산과 버킷당 1 회의 암호화 연산만 필요하므로, 기존의 Hacigümüş 의 방식에 비해 매우

본 연구는 정보통신부 및 정보통신연구진흥원의 IT 신성장동력핵심기술개발사업의 일환으로 수행하였음.
[2007-S-021-01, 개인정보 DB 를 위한 통합형 보안 기술 개발]

효율적이다. 또한, 제안한 방법의 장점으로, 순서 노출이 되지 않으므로 Agrawal 의 순서유지 방식보다 안전성이 강화되었다.

2. 관련 연구

(1) 버킷 기반 인덱스 방식[6]

Hacigümüş et al.[6] 는 암호화된 데이터를 검색하기 위하여, 버킷을 활용하는 방식을 제안하였다. 이는 어트리뷰트 영역상 버킷의 수의 정의에 기반한다. r_i 가 스키마 $R_i(A_{i1}, A_{i2}, \ldots, A_{in})$ 의 평문 릴레이션이고, r_{ki} 가 그에 대응되는 R_{ki}(Counter, Etuple)상의 암호화된 릴레이션이다. 도메인이 D_{ij} 인 R_i 내에 임의의 평문 어트리뷰트 A_{ij} 가 있을 때, 버킷 기반 인덱싱 기법은 D_{ij} 를 겹치지 않게 나눌수 있다. 이는 각각의 연속된 값을 가진 버킷이라고 칭한다.

(그림 1) 평문에 대한 버킷팅

평문을 일정한 간격으로 분할하는 절차를 버킷팅이라고 하며, 버킷은 항상 같은 사이즈로 생성된다. 각 버킷은 유일한 값으로 연결되어 있으며, 이 값은 인덱스 I_j 와 A_{ij} 의 연결을 위한 도메인이다. r_i 내의 평문 tuple t 가 주어졌을 때, t 에 대한 어트리뷰트 A_{ij} 의 값은 버킷에 속하게 된다. 즉, 암호화된 평문이 어느 버킷에 속하는지를 알수 있으므로, 암호화된 데이터베이스 전체를 복호화하지 않고 어느 범위에 있는지를 판단하여 데이터를 가져올 수 있다. 그러나 이 방식은, 버킷 사이즈가 매우 클 경우 다량의 복호화 연산이 추가로 필요하고, 버킷 아이디에 대한 별도의 관리가 필요하다는 부분에서 한계점이 있다.

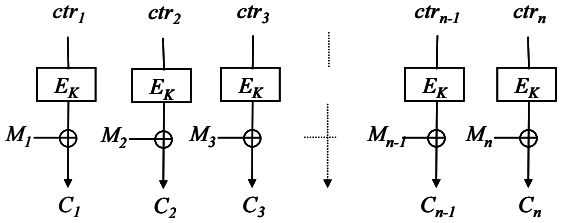
(2) OPES[1]

수치데이터의 경우, 공격자는 암호문 c 에 대하여 정확한 평문 p 를 모르더라도, p 에 매우 근접한 값을 얻을 수 있다면 큰 문제가 될 수 있다. 즉, 통상적인 순서유지 암호화 메커니즘의 경우, 분포를 알고 있으면 추측에 의하여 평문데이터가 노출될 수 있다. 본 논문에서는 ciphertext only attack 에 대해서만 고려하여, 제안한 메커니즘이 estimation exposure 에 대하여 안전함을 보이고 있다.

그러나, 이러한 방식은 순서유지의 특성으로 인하여, 순서가 그대로 노출되며, 알고자 하는 값 p 가 특정 위치를 지정함을 통하여 추론될 수 있다. 순서가 유지된다는 것은 결과적으로, 순위를 그대로 알 수 있다는 것이며, 그 자체만으로도 정보가 일정 부분 노출되었다고 볼 수 있다. 따라서 본 논문에서는, 순서를 유지하지 않는 상태에서 Range 쿼리가 가능한 Hacigümüş 의 버킷팅 기법의 장점을 살린 메커니즘을 제안한다.

(3) AES 카운터 모드[11]

AES 카운터 모드의 동작 과정은 (그림 2) 와 같다. i 번째 카운터 t_i 를 특정 key 로 암호화하고, 이 값을 AES 로 암호화한다. 최종적으로, 평문 M_i 에 대하여 $E_K(ctr_i)$와 XOR 연산을 취하는 것으로, 암호문 C_i 를 생성할 수 있다.



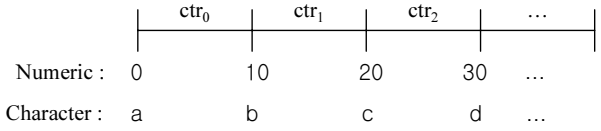
(그림 2) AES 카운터 모드

3. 제안 메커니즘

(1) 암호화 과정

1) 버킷팅 단계

평문을 (그림 3)과 같이 임의의 적당한 간격을 갖는 버킷으로 분할하고, 이에 대한 카운터 값 ctr 을 지정한다.



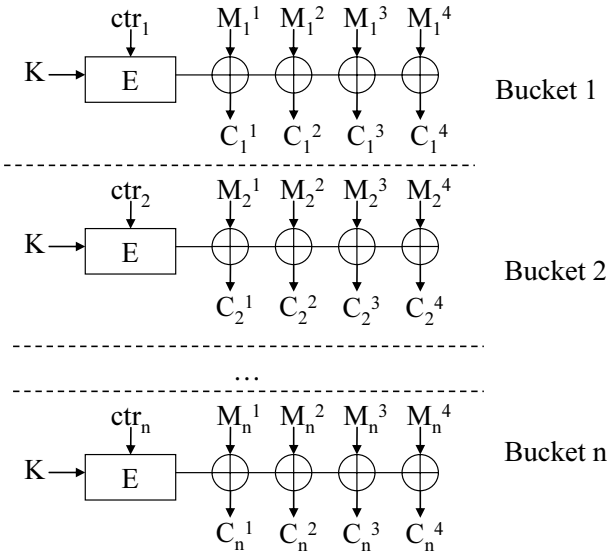
(그림 3) 버킷팅

2) Counter 모드기반 암호화 단계

암호문 C 를 얻는 식은 다음과 같다.

$$C = E_K(ctr_B) \oplus M$$

각 값들은 각 버킷의 카운터 ctr_B 의 암호화된 값과 어트리뷰트와의 XOR 연산을 수행하여 암호화한다. 암호화 과정은 (그림 4)와 같다.



(그림 4) Counter 모드 기반 암호화

3) 버킷 ID 생성단계

각 버킷에 대한 ID 를 다음과 같이 생성한다.

$$I_B = HMAC_K(E_K(ctr_B))$$

생성된 버킷 ID 에 대하여 데이터베이스에 (그림 5) 와 같이 저장한다.

Bucket ID	Ciphertext
$HMAC_K(E_K(ctr_0))$	$E_K(ctr_1) \oplus M$
$HMAC_K(E_K(ctr_1))$	$E_K(ctr_2) \oplus M$
• • •	• • •
$HMAC_K(E_K(ctr_n))$	$E_K(ctr_n) \oplus M$

(그림 5) 저장된 데이터베이스 테이블 형태

각 평문의 값은 암호화되어 저장되며, 이에 대한 버킷 아이디가 각각 연결되어 있다. 한편, 암호화된 카운터에 대한 정보는 HMAC 으로 처리되어 있으므로 공격자가 쉽게 버킷 아이디에 대한 카운터 정보를 암호문 정보로부터 식별할 수 없다.

이후, 버킷 아이디 I_B에 대응하는 B+-Tree 인덱스를 생성함으로써 범위검색 수행시 해당되는 I_B 값을 버킷 아이디로 지정하여 인덱스를 참조한 상태에서 검색할 수 있다.

(2) 복호화 과정

평문 M 은 다음과 같은 식에 의해 구할 수 있다.

$$M = C \oplus E_K(ctr_B)$$

각 버킷은 $E_K(ctr_B)$ 값을 가지고 있으며, 버킷 내 데이터에 대한 복호화 과정은 1 회의 XOR 연산만이 요구된다.

4. 연산량 비교 분석

OPES[1] 순서유지 암호화 방식의 특성에 따라 암호문의 순서는 평문과 완전히 일치한다. 이러한 특성에 따라 순서 정보를 그대로 노출하게 되어 학생의 등수와 같은 서열척도에 대해서는 암호화가 무의미하다. 한편, Mapping Function 에 따라, 암호문은 실수의 형태를 띠게 되며, 이 과정에서 정확도 에러를 발생시킬 수 있다. 예를 들어, 0.333... 과 같은 형태의 암호문이 존재할 경우, 원본값이 정확하게 복원되지 않을 가능성이 있다.

Bucketization[6] 이 방식은 단일한 값인 버킷 ID 값에 대한 별도의 관리가 필요하며, 버킷 ID 값이 노출될 경우, 암복호화시 버킷 내의 데이터 수 만큼의 AES 암복호화 연산이 필요하다. 그러나, 제안한 방식은 버킷당 1 회의 AES 연산만 요구되며, 각 데이터당 1 회, 즉 버킷내 데이터 수 n 회의 XOR 만이 요구되어 Hacigümüş 의 방식에 비해 효율적이다.

B+-Tree based Index[2][3] 이 방식은 한번의 범위검색을 위해 다수의 SQL 이 요구된다. 즉, 범위 내의 데이터가 100 건이면, 실제 SQL 은 건수+노드수로서 100 회 이상 다수의 질의가 필요하다. 이러한 방식은 데이터베이스에 상당한 오버헤드를 가져온다. 또한, 데이터의 입력, 수정,삭제가 반복될 경우, B+-Tree 를 재구성하여야 하며, 이러한 과정에서 전체 데이터의 복호화가 필요하다. 이러한 문제로 인하여 범위검색에 대한 빠른 처리를 할 수 없다.

Hash Based Index[2][3] 이 방식은 빈도수 기반 추론공격을 방지할 수 있으나, 범위 검색이 매우 어렵고, 일치검색에 대하여 과다한 정보를 가져와서 재필터링해야 하는 번거로움이 있다.

Our Scheme 제안한 메커니즘은 순서를 유지하지 않으므로, 순서 노출에 따른 추론 공격을 방지할 수 있다. 또한, 버킷팅의 장점을 그대로 살린 상태에서 암복호화 고속 연산 기능을 추가하였다. 또한 버킷 아이디에 대한 특별한 관리가 필요하지 않으며, 데이터의 Insert 및 Update 시 다른 어트리뷰트의 변경이 필요하지 않다. 한편, 한번의 질의로 암호화된 데이터베이스에 대한 범위검색이 가능하므로, Damiani 의 방식에 비해 매우 효율적이다.

<표 1>은 암복호화시 Hacigümüş 의 방식과 필요 연산량을 비교하고 있다.

<표 1> 암복호화시 필요 연산

	암호화시 필요 연산	복호화시 필요 연산
Hacigümüş [6]	AES : n 회	AES : n 회
제안한 방식	AES : 1 회 XOR : n 회	AES : 1 회 XOR : n 회

n : 버킷내 어트리뷰트 데이터 수

5. 결 론

아웃소싱된 데이터베이스에서의 안전성은 암호학적인 안전성과는 별도의 고려가 필요하다. 특히, Inference Attack, Query Execution 상의 공격, Known-Plaintext Attack 등 아웃소싱된 데이터베이스의 특성에 따른 다양한 공격을 시도할 수 있어, 아웃소싱된 데이터베이스 환경에 적합한 암호화 메커니즘이 요구된다. 본 논문에서는 Hacigümüş 의 장점을 그대로 살리고, 취약점인 암복호화 속도 문제를 해결하여 고속으로 범위 검색이 가능한 새로운 암호화 메커니즘을 제안하였다. 제안한 방법은, 기 제안된 Agrawal 의 순서 유지 방법보다 더욱 강건하며, Hacigümüş 및 Damiani 의 방식보다 더욱 효율적으로 데이터를 보호할 수 있을 것으로 기대된다.

향후 과제로서, 성능 평가를 위한 simulated experiment 를 수행하여 결과를 비교하고, exposure measure 에 대한 정량적인 평가를 수행할 것이다.

참고문헌

[1] Agrawal, R. et al., Order preserving encryption for numeric data. In Weikum, G., K¨onig, A., and Deßloch, S., Eds., Proc. of the ACM SIGMOD 2004, Paris, France, 2004.

[2] E. Damiani, S. D. C. di Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Balancing confidentiality and efficiency in untrusted relational dbmss. In Proc. of the 10th ACM Conf. on Computer and Communications Security (CCS), October 2003.

[3] Damiani, E. et al., Implementation of a storage mechanism for untrusted DBMSs. In Proc. of the Second International IEEE Security in Storage Workshop, Washington DC, USA. IEEE Computer Society, 2003.

[4] J. Domingo i Ferror. A new privacy homomorphism and applications. Information Processing Letters, 60(5):277–282, 1996.

[5] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In IEEE Symp. on Security and Privacy, Oakland, California, 2000.

[6] H. Hacigümüş, B. R. Iyer, C. Li, and S. Mehrotra. Executing SQL over encrypted data in the database-service-provider model. In Proc. of the ACM SIGMOD Conf. on Management of Data, Madison,Wisconsin, June 2002.

[7] H Hacigumus, B Iyer, S Mehrotra, Providing database as a service, Proc. 18th International Data Engineering, 2002.

[8] Iyer, B. et al., A framework for efficient storage security in RDBMS. In Bertino, E. et al., Eds., Proc. of the International Conference on Extending Database Technology (EDBT 2004), volume 2992 of Lecture Notes in Computer Science, Crete, Greece. Springer, 2004.

[9] Sun S. Chung, Gultekin Ozsoyoglu, Anti-Tamper Databases: Processing Aggregate Queries over Encrypted Databases, EECS Department, Case Western Reserve University, Cleveland Ohio, U.S.A., ICDEW'06, 2006.

[10] Aggarwal, G. et al.. Two can keep a secret: a distributed architecture for secure database services. In Proc. of the Second Biennial Conference on Innovative Data Systems Research (CIDR 2005), Asilomar, CA, 2005.

[11] H Lipmaa, P Rogaway, D Wagner, CTR-Mode Encryption, First NIST Workshop on Modes of Operation, 2000.