암호화된 DB에서 대칭키 기반 검색기법 구현

정민경*, 송희정*, 신승수*, 한군희**
*동명대학교 정보보호학과
**백석대학교 정보통신학부
e-mail:shinss@tu.ac.kr

Database with Keyword Based on Symmetric-Key Cipher

Min-Kyoung Jeong*, Hee-Jeong Song*, Seung-Soo Shin*,
Kun-Hee Han**
*Dept of Information Security, TongMyoung University
**Division of Information & Communication Engineering,
Baekseok University

요 약

최근 개인정보유출사건으로 프라이버시에 대한 관심이 급증하면서, 데이터베이스의 내용을 암호화할 필요성이 요구된다. 초기에는 문서 전체의 복호화를 통해서만 검색이 가능하기 때문에 효율성이 떨어져 암호화기법이 거의 사용되지 않았다. 최근에는 복호화 하지 않고 암호화된 데이터로부터 특정 키워드를 포함하는 정보를 효율적으로 검색하고자 하는 연구가 시작되었다. Song의 연구를 시작으로 점차 효율적인 검색 기법이 제안되어졌다. 본 논문에서는 데이터베이스내의 암호화된 데이터를 검색하는 기법에 대한 설계 및 구현하고, 그에 따른 정확도 및 오류율을 분석한다.

1. 서론

최근 IT서비스가 다양화되면서, 이메일, 카페, 블로그, 미니홈피 등과 같은 데이터베이스의 내부 자원을 공유하는 서비스가 증가하고 있다. 대부분의 기업들은 고객의 개인정보 및 데이터를 모두 데이터베이스에 저장한다. 데이터베이스에 불법적으로 접근하여 데이터들을 빼내는 사례와 같은 고객정보 유출 사건이 일어나고 있다[1,7]. 개인 정보유출 문제를 해결하기 위한 방법으로 데이터베이스를 암호화하는 기법을 사용할 수 있지만[6], 데이터베이스를 암호화할 경우 사용자가 원하는 데이터를 찾기 위해서는 모든 데이터를 복호화 해야 하기 때문에 계산적 비용으로 인해 서버과부하를 유발할 수 있다. 최근에는 복호화 하지 않고, 암호화된 데이터로부터 특정 키워드를 포함하는 정보를 효율적으로 검색하고자 하는 연구가 시작되었다.

Song[2]이 대칭키 기반에서 암호화된 데이터에서의 키워드 검색이 가능한 기법을 2000년에 처음으로 제안하였다. Song 기법에서는 문서의 양과 키워드의 수에 따라 데이터베이스의 저장 공간이 크게 증가하게 되는 문제점이 있다. 이를 보완하고자, Goh[3]가 블룸필터를 사용하는 검색 기법을 제안하였고, Chang[4] 등은 원격으로 서버에 정보를 노출시키지 않으면서 서버로부터 검색을 할 수 있는 모바일환경의 검색기법을 제안하였다. 대칭키 기반의 검색기법은 저장량 또는 계산 속도의 효율성을 높이는 방향으로 연구가 진행되고 있다.

본 논문에서는 암호화된 데이터베이스 검색기법에 대한 구현가능성과 검색결과의 정확도 및 오류율에 대해 분석한다. 본 논문의 구성은 다음과 같다. 2장에서는 기존의 관련연구를 살펴본다. 3장에서는 설계 및 구현을 하고, 4장에서 실험결과를 분석하고, 5장에서 결론을 맺는다.

2. 관련연구

이 장에서는 암호화된 데이터베이스에서 대칭키 기반으로 데이터를 검색하는 기법으로 Chang[4]기법에 대해서 알아본다. Chang기법은 설정단계, 검색단계로 구성된다. 본 논문에서는 다음과 같은 표기법을 사용한다.

[표 1] 표기법

표기	설명
t	안전성 파라미터
[n]	{1,2,…,n}, 만약 i∈[n] ⇔1≤i≤n
I[n]	인덱스 I의 i번째 비트
$K\in\{0,1\}^t$	의사난수 집합
$P_k(x)$	$\{0,1\}^d\rightarrow\{0,1\}^d$ (의사난수 치환함수)
$F_k(x)$	$\{0,1\}^d\rightarrow\{0,1\}^t$ (의사난수 함수)
$G_k(x)$	$\{0,1\}^d\rightarrow\{0,1\}$ (의사난수 비트생성기)
Dic	키워드 인덱스 쌍 - (i,W_i), for $i\in[2^d]$
E()	암호화 연산

2.1 설정단계

Step 1. s, $r\in_R\{0,1\}^t$, $P(\ )$, $F(\ )$, $G(\ )$, Dic=(i, w_i) with $i\in[2^d]$

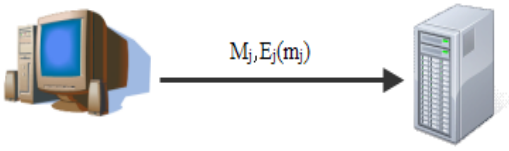
사용자는 s, r을 $\{0,1\}^t$에서 비밀키로 선택한다.

Step 2. 각각의 파일 m_j에 대해 사용자는 2^d비트인 인덱스 스트림 I_j를 준비한다. $1\le j\le n$, $1\le i\le 2^d$.

$$I_j[P_S(i)]=\begin{cases}1 \text{ if } w_i\ni m_j\\ 0 \text{ if } w_i\not\ni m_j\end{cases}$$

m_j가 키워드 w_i를 갖고 있다면 사용자는 i번째 키워드 w_i의 인덱스 i를 치환시킨 값 $P_S(i)$가 j번째 문서에 존재한다면 $I_j[P_S(i)]=1$, 존재하지 않는다면 $I_j[P_S(i)]=0$으로 정의한다.

Step 3. 사용자는 j번째 문서가 i번째 키워드를 포함하고 있는지 없는지를 나타내는 $M_j[i]=I_j[i]\oplus Gr_i(j)$를 계산한다. 여기서 r_i는 w_i의 인덱스 i를 의사난수 함수 F에 대응시켜 나온 값 $F_r(i)$이다. 시드(seed)값의 패턴을 최소화하기 위해서 의사난수 함수를 두 번 사용하게 된다.



[그림 1] 전송과정

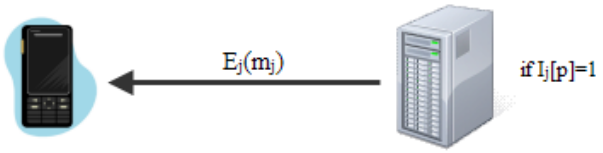
Step 4. 사용자는 서버에게 M_j와 $E_j(m_j)$를 보낸다.

Step 5. 사용자는 비밀키(s, r)와 Dic을 자신의 모바일 디바이스에 복사한다.

2.2 검색단계

Step 1. 키워드 w_λ를 가진 파일을 검색하기 위해서 $p=P_S(\lambda)$와 $f=F_r(p)$를 이용해서 사용자의 Dic으로부터 λ에 대응하는 파일을 찾아서 전송한다.

Step 2. 서버는 $I_j[p]=M_j[p]\oplus G_f(j)$를 각 문서마다 계산하여 $I_j[p]=1$이라면 사용자에게 $E_j(m_j)$를 전송한다.



[그림 2] 암호화된 문서검색

본 논문에서는 대칭키 기반의 암호화된 데이터베이스 검색 알고리즘을 설계하고 구현하고자 한다.

3. 구현 및 설계

이 장에서는 대칭키 기반의 암호화된 데이터베이스 검색 알고리즘에 대한 설계, 구현과 실험에 대해 기술한다.

3.1 구현

암호화된 데이터베이스의 검색 알고리즘의 설계 및 구현환경은 다음과 같다. CPU는 Intel Dual T2370이고, RAM은 1.75GB, OS는 Windows XP SP3이다. 데이터베이스는 MS-SQL이고, Language는 C#

Framework 3.5 SP1, Visual Studio 2008에서 구현하였다. 데이터베이스는 키워드 32개와 문서 약 120개로 구성하여 실험환경을 구축하였다. 그 결과 정확성 측면에서는 일반 검색알고리즘과 유사한 정도로 검색되었고, 성능적인 측면에서도 일반 검색알고리즘에 비해 크게 떨어지지 않았다.

[표 2] 구현환경

CPU	Intel Dual T2370
RAM	1.75GB
OS	Windows XP SP3
Database	MS-SQL 2008 Enterprise
Language	C# Framework 3.5 SP1
Tool	Visual Studio 2008

3.2 설계

암호화된 데이터베이스 검색기법의 개체에 따른 분류로 데이터공급자와 서버, 사용자로 구성된다.

3.2.1 사용자 설계

사용자는 서비스를 이용하기 위해 먼저 서버에 등록을 하여야 한다. 이 때 서버의 내부자공격[5]을 막기 위해서 패스워드의 해시값만 보내게 된다. 등록 및 로그인을 성공하면, 사용자는 문서에 대한 인덱스를 계산하고 암호화한 문서와 계산된 인덱스를 서버에 전송한다. 이 때 인덱스를 계산하기 위해 사용한 Seed값은 사용자가 보관하고 있어야 한다. 사용자가 키워드를 이용해 서버에 저장되어 있는 암호화된 데이터를 검색하고자 할 때, Seed값과 키워드의 인덱스를 서버에 전송한다. 마지막으로 서버로부터 전송받은 문서를 복호화하여 원하는 문서를 얻는다.

```
// 서버에 등록하기 위해, 개인정보를 전송
// 패스워드는 해싱되어 전송
// 서버에 로그인하기 위해, 아이디와 패스워드를 전송
// 파일 저장시, 계산한 인덱스와 암호화된 문서를 전송
// 검색시, Seed, 키워드의 인덱스를 전송
// 받은 암호화된 문서를 복호화
```

[그림 3] 사용자 알고리즘

3.2.2 서버 설계

서버는 등록절차에서 사용자가 보낸 아이디와 패스워드를 데이터베이스에 저장한다. 이 때 저장되는 패스워드는 패스워드를 해시한 값이다. 로그인 단계에서 사용자에 대한 인증을 수행하고, 인증에 성공한 사용자에게 서비스를 제공한다. 저장단계에서는 사용자로부터 암호화된 파일과 인덱스를 전송받아 데이터베이스에 저장한다. 마지막으로 검색단계에서는 사용자가 Seed값과 키워드에 대한 인덱스를 보내면 파일의 인덱스를 계산하여 암호화된 파일을 전송한다.

```
// 클라이언트의 개인정보를 저장
// 아이디와 패스워드 인증
// 인덱스와 암호화된 파일 저장
// 받은 인덱스를 계산하여 암호화된 파일 전송
```

[그림 4] 서버 알고리즘

4. 실험결과 분석

이 장에서는 데이터베이스 검색 시, 검색결과의 정확도 및 오류율에 대해서 분석하고자 한다. 이 실험에서는 32개의 키워드와 여러 키워드가 포함된 100개의 데이터가 사용된다. 검색결과와 검색 시 사용한 키워드가 포함된 문서가 일치하는 지 비교하였다. 정확도는 키워드를 검색하였을 때, 데이터베이스에 저장된 문서들 중에서 검색한 키워드가 포함되어 있는 모든 문서가 검색되는 정도이다. 오류율은 키워드를 검색하였을 때, 그 키워드를 포함하지 않는 문서가 검색이 될 확률이다. 또한 키워드를 포함하였는데도 검색이 되지 않을 확률이다. 예를 들어, '암호'라는 키워드를 검색 하였을 때, 5개의 문서가 검색되었다. 검색된 5개의 문서의 내용을 확인하였더니, '암호'라는 키워드가 포함되어 있었다. 100번의 실험결과를 분석해보면, 검색결과는 정확하게 일치하였고, 오류율도 거의 없었다. 키워드를 n으로 가정하고, 문서의 개수를 m개로 했을 때, 알고리즘의 복잡도는 O(n*m)을 가지게 된다. 현재 인덱스기반으로 암호화된 DB검색기법에서는 DB에 인덱스를 저장해야 되고, 그에 따른 인덱스의 저장용량을 고려할 필요가 있다. 데이터베이스의 저장 공간은 (n/8)*m 이상의 저장 공간이 필요하게 된다. 그러므로 키워드의 개수에 따라 데이터베이스의 크기가 비선형적으로 증가하게 된다.

5. 결론

최근 개인 프라이버시에 대한 관심이 급증하면서,

데이터베이스의 암호화에 대한 연구가 진행되고 있으며, 암호화된 데이터베이스 검색기법에 대한 필요성이 요구된다. 이러한 기법은 이메일, 검색엔진, 개인정보관리 등과 같은 데이터베이스를 사용하는 다양한 분야에서 응용될 수 있다.

본 논문에서는 암호화된 데이터베이스에서 검색기법에 대한 구현가능성과 효율성을 분석하였다. 이 기법의 시간복잡도와 정확성은 일반 검색알고리즘의 성능과 비교해 볼 때, 거의 동일하다. 하지만, 공간복잡도 측면에서는, 키워드의 개수에 따라 데이터베이스 저장 공간이 비선형적으로 증가한다. 따라서 키워드에 따른 데이터베이스 용량증가를 효율적으로 관리 할 수 있는 연구가 필요하다.

참고문헌

[1] 「데이터베이스 보안은 선택이 아닌 필수」, http://www.itdaily.kr/news/articleView.html?idxno=5677, IT데일리, 09,19,2006.

[2] D.song, "*Practical Techniques for Searching on Encrypted Data*", In Proceedings of IEEE Symposium on Security and Privacy, pp.44-55, May, 2000.

[3] Goh, "*Secure Index*", in Cryptology ePrint Archive: Report 2003/216, May, 2004.

[4] Y.Chang and M.Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data", Lecture Notes in Computer Science, 2005.

[5] W. C. Ku, C. M. Chen, and H. L. Lee, "*Cryptanalysis of a variant of Peyravian-Zunic's password authentication scheme*", IEICE Trans. on Consumer, Vol.E86-B, No.5, pp.1682-1684, 2003.

[6] Byunghee Lee, Yunho Lee, Seokhyang Cho, Seungjoo Kim, Dongho Won, "*A Study on the Keyword Search on Encrypted Data using Symmetric Key Encryption*" 한국정보보호학회 하계정보보호학술대회논문집, Vol.16, No.1, 2006.

[7] N.S. Jho, D.W. Hong, "*Technical Trend of the Searchable Encryption System*" 전자통신동향분석,제23권,제4호 8월, 2008.