준동형 암호를 이용한 안전한 데이터 관리 시스템 설계

차현종* · 양호경** · 최강임*** · 유황빈*** · 신효영****

요 약

기업체에서는 정보를 암호화 후 저장하는 것을 법적으로 의무화하고 있다. 하지만, 실제로 정보를 암호화하여 저장하면 검색 또는 수정 시 서버에서 사전에 반드시 복호화 과정을 수행해야만 한다. 그러므로 처리지연 시간이 발생하고, 효율성이 떨어진다. 이러한 작업은 서버에 부담을 주게 되므로 서버를 관리하는 업체나 관리자는 정보를 암호화하여 저장하지 않는다. 본 논문에서는 네트워크 환경에서 정보의 수집과 빠른 의사결정을 복호화 과정 없이 암호문을 수정할 수 있는 준동형 연산을 이용하여 안전성이 보장되고 빠른 처리가 가능한 효과적인 보안 데이터 관리 시스템을 설계하고 구현한다. 구현된 시스템은 보안성의 보장을 위해 기존의 암호화 알고리즘을 사용할 수 있다. 검색 시에는 키워드 검색 방식을 사용한다. 추가로 트랩도어를 사용함으로써 키워드가 노출되지 않고 검색 시마다 변경되어 키워드에 대한 정보가 노출되지 않는다.

Design of the secure data management system using homomorphic encryption

Hyun-Jong Cha* · Ho-Kyung Yang** · Kang-Im Choi*** · Hwang-Bin Ryou*** · Hyo-Young Shin****

ABSTRACT

General companies consider saving the information after enciphering as law. However, if the actual information is saved as enciphered, the decoding process must be conducted when the information is searched or edited in the ser ver. Therefore, process delay time occurs and is less efficient. This kind of work gives burden to the server, so the companies or managers handling the server do not save the information after enciphering. In this paper, the Networ k constructs and realizes an efficient security data management system that ensures safety and haste in operating u sing the homomorphic encryption technology, which collects information and decides quickly, and enables editing the encryption without a decoding process. To ensure the security of the embodied system, the existing encryption algo rithm can be used. Search method to use the keyword search. Additionally, by using a trapdoor, the keyword is not expose and it is changed whenever it is searched, and the formation of the keyword does not get exposed.

Key words : Homomorphism, Secure Data management

접수일(2015년 6월 3일), 수정일(1차: 2015년 6월 30일),
게재확정일(2015년 6월 30일)

* 광운대학교 방위사업학과
** 선문대학교 IT교육학부
*** 광운대학교 컴퓨터과학과
**** 경복대학교 IT보안과(교신저자)

1. 서 론

네트워크 기술의 발전과 대중화되어 기업의 운영방식도 변화하고 있다. 네트워크를 이용하여 정보를 수집하고, 상황의 인식과 빠른 의사결정으로 정보의 우위를 달성하여 기업의 경쟁력을 높이는 것에 주로 사용된다. 그러므로 기업들은 데이터 관리 시스템에서 정보를 안전하게 관리하고 빠르게 처리할 수 있어야 한다. 또한 고객의 정보를 안전하게 관리할 시스템이 필요하다. 일반 기업체에서는 정보를 암호화 후 저장하는 것을 법적으로도 의무화 시키고 있다. 하지만, 실제 정보를 암호화하여 저장하면 검색 또는 수정 시 서버에서 사전에 반드시 복호화 과정을 수행해야만 한다. 그러므로 처리 지연 시간이 발생하고, 효율성이 떨어진다. 이러한 작업은 서버에 부담을 주게 되므로 서버를 관리하는 업체나 관리자는 정보를 암호화하여 저장하지 않는다[1].

이런 환경에서 최근 복호화 과정 없이 암호문 상태에서 연산이 가능한 준동형 암호 기술과 검색 가능 암호 기술이 주목을 받고 있다[2, 3].

최근에 많은 연구가 진행되고 있지만 암호 알고리즘에 대한 수학적인 증명만 이루어지고 안전성이나 효율성을 만족하지는 못해 실제시스템에 적용하는 경우가 드물다.

본 논문에서는 네트워크 환경에서 정보의 수집과 빠른 의사결정을 복호화 과정 없이 암호문을 수정할 수 있는 준동형 연산을 이용하여 안전성이 보장되고 빠른 처리가 가능한 효과적인 보안 데이터 관리 시스템(이하 보안 데이터 관리 시스템)을 설계하고 구현한다. 본 논문은 효과적인 데이터 관리 시스템을 구현하고 효율성을 실험하였다. 검색 시에는 키워드 검색 방식을 사용하며 트랩도어를 사용함으로써 키워드가 노출되지 않고 검색 시마다 변경되어 키워드에 대한 정보가 노출되지 않는다.

2. 관련연구

2.1 준동형 암호 시스템

준동형 암호는 1978년 Rivest의 소개로 'Privacy Homomorphism'이 소개되었는데 기본 방식은 RSA 방식과 유사하다. 이 방식은 안전성을 인정받지 못했다[2].

준동형성(homomorphism)이란, 수학에서 정의된 두 집합 사이에서 정의된 연산을 보존할 수 있는 매핑을 말한다[4].

(그림 1) 준동형 암호 기법의 개요도

(그림 1)은 준동형 암호 기법을 간단히 나타내고 있다. 준동형 암호란 암호화 함수 중에서 평문 공간과 암호문 공간에 정의된 연산을 보존하는 암호화 함수이다. 준동형 암호는 기본적으로는 대표적인 산술 연산인 덧셈 연산과 곱셈 연산을 암호문에 적용하여 평문에 대한 연산을 수행할 수 있도록 하는 암호화 기법이다.

일반적인 암호화 알고리즘을 이용하여 암호문을 수정할 경우에는 원문에서 복호화 하여 수정하고 다시 재 암호화하여 암호문을 보관한다. 반면 준동형 암호는 원문의 복호화 과정 없이 암호문 상태에서 특수연산을 통하여 원하는 값으로 수정이 가능하다.

2.2 검색가능 암호 시스템

검색 가능 암호 시스템은 기존의 암호 기술과 같이 암호화된 데이터에 대한 안전성을 보장하면서 동시에 특정 키워드를 포함하는 데이터를 검색할 수 있도록 고안된 암호 기술이다. 데이터베이스(DB : Database)에서 제공되는 다양한 기능 중 많은 경우가 특정 키워드를 포함하는 데이터에 대한 검색을 바탕으로 이루어진다[5].

검색 가능 암호 시스템에서 특정 키워드로 데이터를 검색할 때 외부로 키워드가 노출되지 않도록 키워드를 변환하여 검색을 진행한다. 이때 변환된 키워드는 검색을 요청한 사용자가 생성하고, 요청을 받은 서버가 풀어볼 수 있어야 한다.

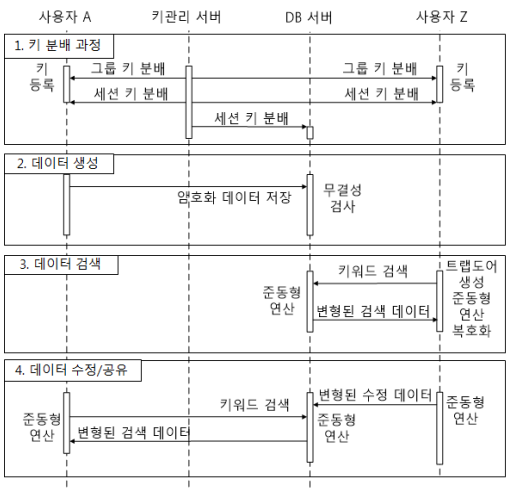
3. 보안 데이터 관리 시스템

본 장에서는 시스템의 세부 처리절차에 대해서 기술한다. 먼저, 시스템의 개요 및 구성, 데이터의 생애주기를 기준으로 각 단계를 설계한다. 각 단계는 데이터의 생성에서부터 검색, 수정으로 구성되었으며 각 단계에 대해 설명한다.

3.1 시스템의 통신절차

보안 데이터 관리 시스템에서 인증 서버와 키 관리 서버에 대해서는 기존에 연구되고 사용되는 기술로 별도로 거론하지 않는다. 이외에 통신의 주체가 되는 서버와 클라이언트에 대해서만 설계하고 구현한다. 다만 DB서버에 암호화데이터는 대칭키 암호기법을 사용한다. 클라이언트는 저장된 데이터별로 이용할 수 있는 사용자와 그룹으로 나뉘며, 사용자는 복수의 그룹에 소속될 수 있다[6].

(그림 2)는 보안 데이터 관리 시스템의 처리 절차를 전체적으로 나타낸 것이다.



(그림 2) 시스템의 처리 절차

보안 데이터 관리 시스템은 암호화된 데이터를 저장하는 환경에서 데이터의 생성/저장/수정/검색 시에 안전하게 데이터를 관리하기 위해 DB 서버와 사용자 간의 통신절차를 5단계로 구성하였다.

첫째, 사전 등록 절차는 사용자와 DB 서버간의 통신하기 전에 수행되며, 키 관리 서버에 자신들의 정보를 등록하고 키를 등록하는 과정이다.

둘째, 키 분배 과정은 그룹 키와 세션 키를 분배한다.

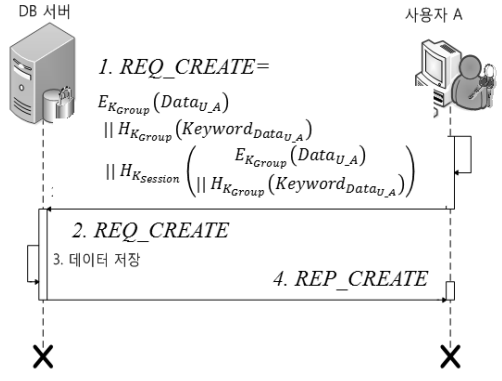
셋째, 데이터 생성 과정은 사용자가 DB 서버에 암호화된 데이터를 저장한다. 사용자는 데이터를 생성해서 저장할 권한을 갖고 있어야만 자신이 생성한 데이터를 암호화하여 DB 서버에 저장한다.

넷째, 데이터 검색 과정은 사용자가 생성한 데이터를 다른 사용자가 검색하고자 할 때 사용된다.

마지막 처리절차는 DB 서버에 암호화 상태로 저장되어 있는 데이터를 수정하고, 다른 사용자가 검색하여 데이터를 공유하는 과정이다.

3.2 데이터의 생성

보안 데이터 관리 시스템의 DB서버는 암호문을 저장한다. 클라이언트에서 생성한 데이터를 암호화하여 DB서버에 전송한다. DB 서버는 전송된 암호문의 무결성 검사를 하고 이상이 없다면 데이터베이스에 저장한다. 만약 이상이 있다면 재전송을 요청한다.



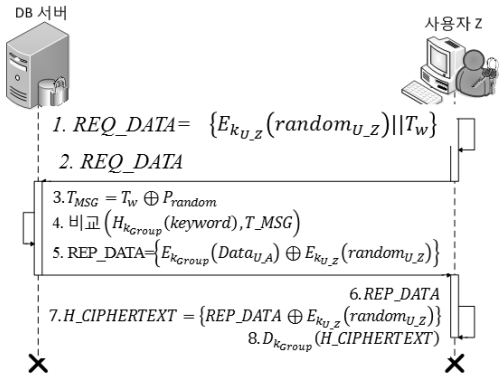
(그림 3) 데이터 생성 단계

데이터의 생성은 (그림 3)와 같이 서버와 사용자간의 암호화된 데이터의 생성을 위해 4단계의 절차를 수행하게 된다.

각 단계에서 암호화 과정은 서버에서 수행되지 않

고 클라이언트에서만 수행된다. 때문에 클라이언트에서 데이터를 암호화한 암호문과 전달된 데이터의 무결성 확보를 위해 암호문에 대해 해쉬 알고리즘을 통해 생성한 해쉬 값과 같이 전달한다. 이후에 데이터의 검색이나 수정을 위해 암호문을 찾을 수 있는 근거가 필요하다. 때문에 암호문에 대한 키워드를 생성하여 암호문과 같이 DB서버에게 전달한다. 키워드의 경우 데이터의 검색이나 수정 과정에서 복호화 하여 검증할 필요가 없기 때문에 암호화하지 않고 해쉬 값을 생성하여 사용한다. 같은 그룹에 있는 사용자만이 데이터를 이용할 수 있도록 데이터를 암호화하거나 키워드의 해쉬 값을 생성할 때 그룹 키를 사용한다. 서버에서는 암호화 과정 없이 클라이언트로부터 전달받은 데이터의 무결성을 검사하고 함께 전달받은 암호화된 키워드를 저장하게 된다.

3.3 데이터의 검색



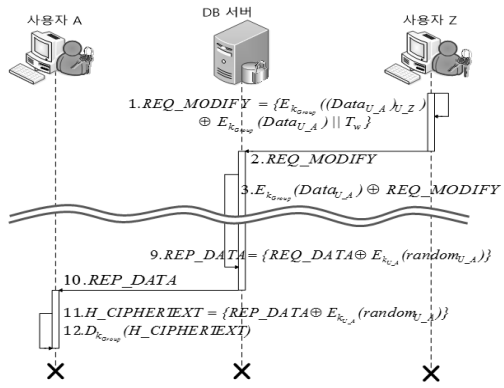
(그림 4) 데이터 검색 단계

보안 데이터 관리 시스템은 암호화 데이터를 저장한 서버에서 복호화 과정 없이 사용자가 원하는 데이터를 검색하기 위해서 데이터에 대한 단일 키워드 기법을 사용한다. 일반적인 시스템에서 사용자는 원하는 데이터를 검색하기 위해 키워드를 서버에게 전송하여 검색하도록 한다. 하지만, 이 과정에서 악의적인 공격자는 모니터링을 통해 키워드와 그에 대한 응답으로 전해지는 암호문과의 관계를 파악할 수 있고, 암호문의 내용을 유추할 수 있다. 이렇게 되면 공격자가 필요로 하는 문서를 걸러낼 수 있는 빌미를 줄 수 있다. 때문에 보안 데이터 관리 시스템에서는 암호화 데이터의 검색을 위해 트랩도어를 사용한다.

(그림 4)는 보안 데이터 관리 시스템에서의 준동형 연산과 트랩도어를 사용하여 검색하는 과정을 나타내고 있다.

DB 서버에서 저장하고 있는 암호화 데이터를 검색하기 위해서 기존에 Song이 2000년에 제안한 single keyword search 기법을 개선한다[7].

3.4 데이터의 수정



(그림 5) 데이터 수정과 공유 절차

(그림 5)은 보안 데이터 관리 시스템의 암호화되어 저장된 데이터의 수정을 위한 처리절차를 나타냈다. 상위 1단계부터 3단계까지는 암호화되어 저장된 데이터를 수정하는 단계이다. 이후의 12단계까지는 데이터의 공유 단계로 수정된 데이터를 수정요청을 하지 않은 사용자가 해당 데이터를 검색하는 과정으로 앞 절의 데이터 검색 단계와 같은 과정을 수행한다.

4. 실 험

4.1 실험 환경

시스템의 구성요소 중 가장 중요한 비중을 차지하는 준동형 연산에 대한 연산 처리비용에 대해서 측정하였다. 준동형 연산중에서도 수행되는 수정부분을 추

출하는 과정이 성능비용이 발생할 것이므로 평가항목으로 설정하여 실험하였다. 성능 평가를 위한 실험 환경은 (표 1)과 같다.

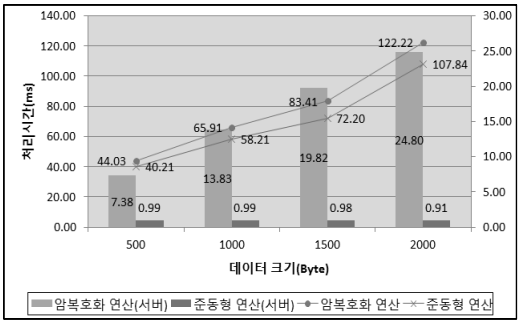
<표 1> 실험 환경

구분	설명
시스템	프로세서 : Intel Core i5 3.40GHz 메모리 : 8GByte 운영체제 : Windows 8 사용도구 : Visual Studio 2012, MySQL
실험환경	실험 횟수 : 10,000회 데이터 크기 : 500, 1000, 1500, 2000Byte

성능 평가의 신뢰도를 높이기 위해 실험횟수는 각 항목별 10,000회씩 수행하였으며, 실험 결과는 10,000회의 실험 결과 값에서 평균값으로 분석하였다.

4.2 실험 분석

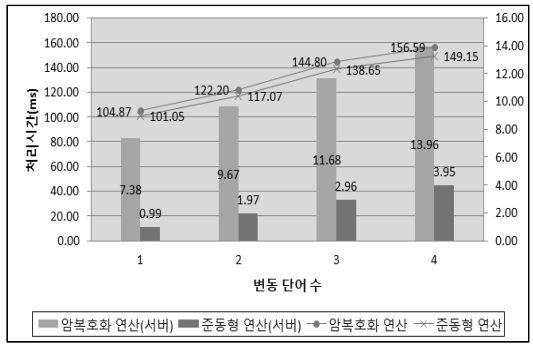
준동형 연산에 대한 성능 평가를 위해 일반적인 암호화 시스템과 준동형 연산을 하는 시스템의 성능을 비교하고자 한다. 일반적인 암호화 시스템의 경우 암호문을 변경하려면, 복호화 이후에 내용을 변경하고 다시 암호화를 해야 한다. 하지만 데이터 관리 시스템의 경우 복호화 과정 없이 암호문상태에서 연산하기 때문에 서버에서는 연산시간을 줄일 수 있다. 준동형 연산에 대하여 다른 준동형 시스템과의 비교가 필요하지만, 아직 준동형 암호 방식의 연구로 인하여 비교 가능한 시스템이 없다. 때문에 일반적인 암호화 시스템과 비교한다.



(그림 6) 암호문 수정에 대한 데이터 크기별 처리시간 비교

(그림 6)은 암호문의 수정에 대해서 일반적인 암호 시스템과 데이터 관리 시스템의 데이터 크기별 처리시간을 비교하여 나타내고 있다. 데이터의 크기는 500 Byte, 1000Byte, 1500Byte, 2000Byte로 증가시키며 각각 10,000번의 실험을 하여 평균값을 구한 것이다.

데이터의 크기가 증가함에 따라 처리속도가 두 시스템 모두 증가되는 것은 같으나 증가폭에서 차이가 난다. 이는 일반 시스템의 경우 클라이언트에서 전송된 수정정보를 복호화하고, 검색된 암호문을 복호화하여 암호문을 수정한 후 다시 암호화의 과정을 거친다. 하지만 제안시스템은 클라이언트에서 보내온 수정정보를 복호화하여 검색된 암호문과 그대로 연산을 하기 때문이다.



(그림 7) 암호문 수정에 대한 변동 단어 수별 처리시간 비교

(그림 7)은 암호문의 수정에 대해서 일반적인 암호 시스템과 데이터 관리 시스템의 변동 단어 수 별 처리시간을 비교하여 나타내고 있다. 2000Byte의 데이터에 수정한 단어를 1개에서 4개까지 증가하면서 실험하였다. 실험은 각각 10,000번씩 실행하여 평균값을 나타내었다. 전체 처리시간의 차이는 변동되는 단어 수가 증가함에 따라서 변동 위치 검색과정의 처리시간이 큰 영향을 미치기 때문이다.

데이터의 크기별 처리시간에서는 단어의 수가 고정되었기 때문에 준동형 연산의 시간이 일정하였다. 하지만, 단어의 수가 증가되면 클라이언트에서 전송되는 수정정보의 데이터크기도 커져서 복호화 시간이 증가되고, 준동형 연산할 데이터도 많아지기 때문이다.

5. 결 론

모든 분야에서 네트워크를 통해 정보를 공유하기 때문에 정보를 안전하고 빠르게 처리해야만 한다. 이를 위해서는 반드시 정보를 암호화하여 저장하고 관리해야만 한다. 하지만 암호화된 정보를 수시로 검색, 수정, 전달하는 과정에서 서버의 과부하와 의사결정 지연 등의 문제가 발생할 수 있다.

본 논문에서는 네트워크 환경에서 정보의 수집과 빠른 의사결정을 고려해 처리하고자 하는 정보를 복호화 과정 없이 수정할 수 있는 준동형 연산을 이용하여 안전성이 보장되고 빠른 처리가 가능한 효과적인 데이터 관리 시스템을 설계 및 구축하였다.

향후에는 시스템의 안정성 검증이 필요하다. 그리고, 네트워크가 중심이 되는 산업사회에서의 이점을 최대화하기 위해 이동성을 갖는 사용자를 위한 클라이언트의 연산 복잡도가 낮은 기법에 대한 연구가 필요하다. 특히, 준동형 연산에 필요한 변동부분 검색에 대한 처리시간을 감소시키기 위한 기법의 연구가 필요하다.

참고문헌

[1] 송유진, 박광용, “데이터베이스 아웃소싱을 위한 준동형 암호기술”, 한국정보과학회 논문지, 제19호, 제3권, pp.80-89, 2009.

[2] R. Rivest, L. Adleman, M. Dertouzos, “On data bank and privacy homomorphisms,” in Proceedings of the 9th Annual Symposium on Foundations of Secure Computation(FSC 1978), pp.169-180, 1978.

[3] 송영진, 이석환, 권기룡, 권혁철, “Paillier 암호 방식을 이용한 EPC 정보 검색 기법”, 대한전자공학회 추계학술대회 논문집, 제 32권 제2호, pp.423-424, 2009.

[4] N. S. Jho, K. Y. Jang, “Trend and Issue on Homomorphic Encryption,” Weekly IT Brief., Vol.1522, pp.15-25, 2011.

[5] 조남수, 홍도원, “검색 가능 암호 시스템 기술 동향”, 전자통신동향분석, 제23권, 제4호, 2008.

[6] Hodges, J. and R. Morgan, “Lightweight Directory Access Protocol(v3):Technical Specification,” RFC 3377, 2002.

[7] D. Song, D. Wagner, and A. Perrig, “Practical Techniques for Searching on Encrypted Data,” in Proceeding of IEEE Symposium on Security and Privacy, pp.44-55, 2000.

[저 자 소 개]

차 현 종 (Hyun-Jong Cha)

2005년 광운대학교 컴퓨터소프트웨어학과 공학사
2008년 광운대학교 컴퓨터과학과 공학석사
2011년 광운대학교 방위사업학과 공학석사
2014년 광운대학교 방위사업학과 공학박사

email : chj826@kw.ac.kr

양 호 경 (Ho-Kyung Yang)

2005년 광운대학교 컴퓨터소프트웨어학과 공학사
2007년 광운대학교 컴퓨터과학과 공학석사
2010년 광운대학교 방위사업학과 공학석사
2013년 광운대학교 방위사업학과 공학박사

email : porori2000@kw.ac.kr

최 강 임 (Kang-im Choi)

2001년 광운대학교 전자계산학과 이학사
2003년 광운대학교 컴퓨터과학과 공학석사
2014년~현재 광운대학교 컴퓨터과학과 박사과정

email : kichoi96@kw.ac.kr

유 황 빈 (Hwang-Bin Ryou)

1968년 인하대학교 전자공학과 학사
1975년 연세대학교 전자공학과 공학석사
1984년 경희대학교 전자공학과 공학박사
1981년~현재 광운대학교 컴퓨터소프트웨어학과 교수

email : ryou@kw.ac.kr

신 효 영 (Hyo-Young Shin)

1986년 광운대학교 전자계산학과 이학사
1988년 광운대학교 전자계산학과 이학석사
1998년 광운대학교 전자계산학과 이학박사
1988년~1993년 (주)LG소프트 연구원
1994년~현재 경복대학교 IT보안과 교수

email : hyshin@kbu.ac.kr