IaaS 신뢰도 향상을 위한 데이터 베이스 보안모델 제안

이해준*, 김남욱**, 조신영**, 최영현**, 정태명***
*성균관대학교 컴퓨터 공학과
**성균관대학교 전자전기컴퓨터공학과
***성균관대학교 정보통신공학부
e-mail : *ghksdosl@nate.com, **{nukim, sycho, yhchoi}@imtl.skku.ac.kr,
***tmchung@ece.skku.ac.kr

A proposal of database security model for trusted IaaS

Hea-Jun Lee*, Nam-Uk kim**, Shin-Young Cho**, Young-Hyun Choi**, Tai-Myoung Chung***
*Dept. of Computer Science, Sungkyunkwan Univ
**Dept. of Electrical and Computer Engineering, Sungkyunkwan Univ
***School of Information Communication Engineering, Sungkyunkwan Univ

요 약

근래 인터넷을 통해 IT 자원을 제공하는 클라우드 컴퓨팅이 중요한 이슈로 부각되고 있다. 이러한 클라우드 컴퓨팅이 신뢰성 있는 서비스로 자리잡기 위해서는 보안 문제가 가장 먼저 해결되어야 한다. 따라서 본 논문에서는 먼저 클라우드 컴퓨팅과 암호화의 기본 개념에 대해 알아본 뒤에 신뢰성 향상을 위해 주파수 도약을 응용한 보안 시스템을 제안하였다. 주파수 도약을 응용한 암호화 방법으로 서버의 부담을 줄이고 확장성을 제공하여 클라우드 서비스에 적합한 데이터 베이스 암호화 모델을 제안하였다.

1. 서론

클라우드 컴퓨팅은 기존에 유사한 서비스를 제공하던 세일즈포스닷컴, Amazon Web Service 의 아마존, AppEngine 및 GoogleApps 의 구글 등 일부 인터넷 기업들만이 서비스를 제공하고 있었다. 2009 년 Gartner 에 의해 10 대 전략기술 중 2 위로 선정되고 Microsoft, IBM, HP, Sun, Oracle 등의 IT 벤더들과 AT&T, NTT, BT 등의 통신사업자들까지 관련 서비스를 제공하기 시작하면서 IT 업계 전체의 주목을 받기 시작했다[1].

효율적인 IT 자원의 사용을 통한 관련 비용의 절약은 클라우드 컴퓨팅의 가장 큰 장점이다. 전세계적인 인터넷 사용자 수의 증가와 그에 따른 인터넷 트래픽의 급증은 기업들의 IT 비용을 증가시키고 있다. 하지만 장기간 지속되고 있는 경제불황으로 IT 비용을 절감하기를 원하는 기업들은 클라우드 컴퓨팅의 이러한 특징에 관심을 가지고 있다. 특히 공간절약과 에너지 절감을 통해 그린 IT 정책의 수단으로도 사용될 수 있어 그 유용성이 더욱 부각되고 있다[2].

하지만 2008 년 9 월의 Google Docs 의 데이터 유출 사고와 같은 보안 문제는 대부분의 기업들이 클라우드 컴퓨팅의 도입을 망설이게 하고 있다[3]. 따라서 본 논문에서는 보안 문제, 그 중에서도 데이터 베이스에 저장되는 정보를 보호하는 정보 보호 기술에 관점을 두었다.

암호화 기술은 데이터 베이스에서 정보가 유출되는 경우에 대비하기 위해서는 필수적인 기술이다. 하지만 기존의 데이터 베이스 암호화 기술은 운용 시에 대부분 데이터 베이스의 성능을 수 배 또는 수십 배 이상 떨어뜨린다. 성능 향상을 위해 만들어진 하드웨어 제품의 경우에도 초당 쿼리 수의 제한이 있어 대용량 데이터 베이스에는 적합하지 않다. 이러한 이유 때문에 기존의 데이터 베이스 암호화 기술들은 클라우드 컴퓨팅에 사용하기에는 적절하지 않다[4].

이 문제의 해결을 위해 본 논문에서는 주파수 도약을 응용한 암호화 방법인 Hopping Storage System(HSS)을 제안하였다. 이 방법은 2 차원 행렬과 같은 형태를 가진 데이터 블록을 사용해서 데이터를 저장하는데, 데이터를 읽어 들이는 방향과 저장하는 방향을 서로 다르게 하는 것이 핵심이다. HSS 는 별도의 복잡한 암호화 코드와 알고리즘을 사용하지 않기 때문에 데이터 암호화 과정에서 데이터 베이스의 성능 저하를 줄일 수 있다.

본 논문에서는 2 장에서 클라우드 컴퓨팅과 보안 기술에 대해 설명하고, 3 장에서는 먼저 HSS 의 암호화 알고리즘에 대해 설명한다. 이어서 HSS 의 암호화 알고리즘이 클라우드 컴퓨팅에 적합한 이유를 설명하고 그 성능에 대해 논한다. 그리고 마지막으로 4 장에서는 결론과 함께 향후 연구 과제를 제안한다.

2. 관련 연구

2.1 클라우드 컴퓨팅 서비스

클라우드 컴퓨팅이란 인터넷 기술을 활용해서 사용자가 '가상화된 IT 자원'을 서비스로 제공받으며, 서비스 부하에 따라서 실시간 확장성을 지원받고 사용한 만큼만 비용을 지불하는 컴퓨팅 서비스를 말한다[5].

클라우드 컴퓨팅에서 제공하는 서비스는 서로 연관되어 있어 완전히 분리될 수 있는 것은 아니다. 하지만 일반적으로 사용자가 제공받는 기능에 따라 SaaS, PaaS, IaaS 를 가장 대표적인 서비스로 분류한다[6].

SaaS 는 클라우드 컴퓨팅 서비스 사업자가 인터넷을 통해 제공한 소프트웨어 서비스에 사용자가 인터넷환경에서 원격 접속하여 해당 소프트웨어를 사용하는 모델이다. 이 서비스는 클라우드 컴퓨팅이 등장하기 이전부터 독립적으로 사용되던 기술로 다른 서비스들에 비해 널리 알려져 있어 인지도가 높다. SaaS 에서는 메일 시스템부터 ERP 까지 다양한 애플리케이션들이 온디맨드 방식으로 제공된다.

PaaS 는 서비스 구성 컴포넌트 및 호환성 제공 서비스를 통해 소프트웨어 개발을 위한 환경을 마련해주는 서비스이다. 사용자들은 데이터 베이스, 애플리케이션 등 미들웨어까지 확장된 자원을 인터넷에 접속할 수 있는 곳이라면 어디에서든지 활용할 수 있다. 구글의 AppsEngine 이 대표적이다.

IaaS 는 서버 인프라를 제공하는 서비스로 저장공간 또는 컴퓨팅능력을 판매하는데, 하드웨어의 '능력'을 판매한다는 것이 특징이다. 즉 하드웨어는 서비스 제공자가 운영하고, 사용자는 그 하드웨어의 일부분에 대한 사용권을 구매하는 것이다. 아마존 웹서비스의 스토리지 서비스인 Elastic Compute Cloud, Simple Storage Service 가 대표적이다.

클라우드 컴퓨팅을 그 구성 형태를 기준으로 분류하면 public cloud, private cloud, hybrid cloud 로 나눌 수 있다.

주로 개인 사용자를 대상으로 하는 public cloud 는 대부분의 인터넷 사용자들에게 열려 있으며 서비스가 대규모로 이루어지는 특징이 있다. 개인 사용자들은 주로 무료 서비스에 관심을 가지고 있고, 가장 관심을 표하는 보안 이슈는 개인정보 노출, 감시 및 상업적 이용이다.

주로 기업 또는 접속권한을 제한하기를 원하는 고객을 대상으로 하는 private cloud 는 폐쇄적인 환경에서 특정 사용자만 사용하는 구조로 이루어져 있다. Private cloud 사용자들의 주된 관심사로는 서비스 중단, 기업 정보 훼손 및 유출, 고객 정보 유출, 법/규제의 준수, 데이터의 복구성 등이 있다. 최근에는 실시간 확장성을 위해 private cloud 와 public cloud 를 결합한 hybrid cloud 모델도 등장하고 있다. 새롭게 등장한 Hybrid colud 모델은 private cloud 보다 외부 접속에 대한 통제가 어렵기 때문에 보안 문제가 더욱 중요해지고 있다[7].

```
'Read from the input file, then
 encrypt and write to the output file.
 While rdlen < totlen
     len = fin.Read(bin, 0, 4096)
     encStream.Write(bin, 0, len)
     rdlen = Convert.ToInt32(rdlen + len /
                 des.BlockSize * des.BlockSize)
   Console.WriteLine("Processed {0} bytes,
           {1} bytes total", len, _ rdlen)
 End While
   encStream.Close()
End Sub
```

(그림 1) MSDN DES Class

2.2 데이터 암호화

(그림 1)은 MSDN 에 기재되어 있는 암호 알고리즘 DES Class 코드 중의 일부분으로 입력을 받아들여서 루프가 끝나는 조건이 만족될 때까지 암호화를 반복하는 코드이다. 일반적인 암호화 알고리즘의 구조를 기준으로 분석해보면 (그림 1)의 코드는 암호화 키를 이용해 평문에 대한 여러 단계의 변환 과정을 수행하는 부분이다. 이 부분은 암호화 알고리즘의 핵심이 되는 부분으로 암호화 알고리즘의 모든 속성은 이 부분의 설계에 의해 결정된다.

암호화 알고리즘은 키를 관리하는 방법에 따라 대칭키 알고리즘과 비대칭키 알고리즘으로 나누어진다.

대칭키 알고리즘은 암호화에 사용되는 암호화 키를 안전하게 보관해야 되는 특성을 가지기 때문에 비밀키 알고리즘이라고도 불린다. 대표적인 대칭키 알고리즘으로는 지금까지도 널리 쓰이고 있는 DES(Data Encryption Standard)가 있다. DES 알고리즘은 암호화 과정에서 내부적으로 암호화를 위한 라운드 함수가 총 16 회 적용된다. 그리고 평문을 입력 받을 때 입력받은 평문에 대해 하나, 암호화 결과물을 출력하기 전 결과물에 대해 하나씩 총 2 개의 함수가 더 사용된다. 데이터 변환 과정을 수행하는 암호화 함수의 호출 회수 증가는 암호화 알고리즘의 수행시간 증가로 이어진다. 대칭키 알고리즘은 암호화에 XOR, MOD 등 비교적 빠른 연산을 사용하지만 암호화에 필요한 연산 회수가 그 이상으로 많기 때문에 데이터베이스에 적용할 경우 성능 저하를 피할 수 없다.

비대칭키 알고리즘은 키 보관에 있어 편의성을 제공하기 때문에 공개키 알고리즘이라고도 불리며 대칭키 알고리즘보다 더 높은 보안성능을 제공한다. 하지만 암호화를 위한 연산의 대부분이 소수 구하기, 2 의 거듭제곱 연산 등이다. 이러한 연산들은 연산에 걸리는 시간이 대칭키 알고리즘이 사용하는 연산들보다 훨씬 더 길기 때문에 비대칭키 알고리즘 또한 데이터베이스에서 사용하기에는 부적절하다[8].

암호화 알고리즘의 사용에 있어 추가적으로 고려해야 할 사항은 암호화 키의 라이프 사이클이다. 하나의 키가 많이 사용될수록 그 키를 사용한 암호화의 보안성은 점점 떨어지게 되어 키의 교체가 필요하다.

키를 교체하게 되면 과거의 키로 암호화된 모든 데이터는 복호화 된 후에 새로운 키로 암호화 하는 과정을 거쳐야 하는데, 이러한 과정은 서버에 많은 부하를 준다[9]. 그런데 클라우드 컴퓨팅은 다수의 사용자 데이터가 함께 보관된다. 따라서 사용자마다 최소 한 개 이상의 암호화 키가 지정되어야 하기 때문에 클라우드 컴퓨팅 환경에서는 키 교체로 인한 오버헤드가 더욱 커지게 된다. 이러한 이유들 때문에 클라우드 컴퓨팅 서비스를 제공하는 업체에서 대용량의 데이터베이스에 암호화 기능을 추가하는 것을 기피하게 되었다.

실례로 아마존의 IaaS 서비스 중 Elastic Compute Cloud, Simple Storage Service, Simple Queue Service 에서도 사용자가 업로드 한 데이터에 대한 암호화를 제공하지 않고 있으며, 단지 사용자 자신이 직접 암호화한 데이터를 업로드 하는 것을 허용하고 있을 뿐이다[10].

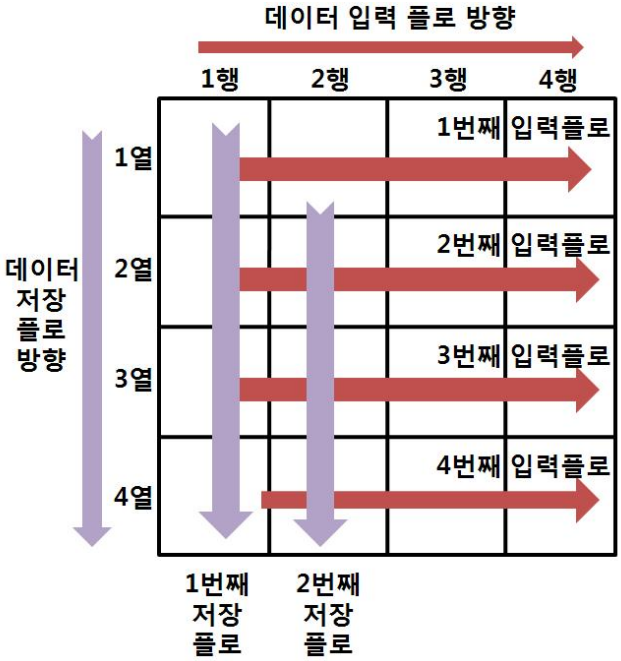
3. 보안 시스템 제안

3.1 Hopping Storage System (HSS)

별도의 암호화 기능을 제공하지 않는 데이터 베이스에서는 인증 코드 및 프로세스가 유출되었을 경우에 데이터의 유출을 막을 수 있는 방법이 없다. 클라우드 컴퓨팅 서비스에 사용되는 데이터 베이스에서도 동일한 위협이 존재한다. 하지만 클라우드 컴퓨팅에서는 데이터 교환과 트래픽이 기존 데이터 베이스에 비해 훨씬 더 많아지기 때문에 암호화 시스템 설계가 어려워 진다. 본 논문에서는 클라우드 컴퓨팅의 이러한 특징들을 고려하여 주파수 도약을 응용한 Hopping Storage System 을 기초로 하는 데이터 베이스 암호화 시스템을 제안한다.

HSS 에서는 데이터를 잠시 저장하는 로딩 블록이 필요하다. 입력되는 데이터를 바로 저장하지 않고 2 차원 행렬의 형태를 가지는 로딩 블록에 먼저 데이터를 읽어 들인 다음, 그 블록 안의 데이터를 저장 공간에 저장한다. 이 때 블록 안에 데이터가 로딩되는 방향과 저장하기 위해 블록 안의 데이터를 읽는 방향에 차이를 두어 암호화를 수행한다.

(그림 2)는 데이터 로딩 블록을 개념화한 그림으로, 설명의 편의를 위해 4x4 의 사이즈로 표현하였다. 먼저 데이터 베이스에 입력으로 들어온 데이터를 로딩 블록 전체의 크기만큼 잘라내어서 행 방향으로 각 블록에 데이터를 로딩한다. 즉, 1 행 1 열, 1 행 2 열, 1 행 3 열, 1 행 4 열, 2 행 1 열, 2 행 2 열... 4 행 4 열의 순서로 데이터를 로드하는 것이다. 그런 다음 로딩 블록 안의 데이터를 행 방향 순서로 저장장치에 저장한다. 다시 말하자면, 로딩 블록 안의 데이터를 저장할 때는 1 행 1 열, 2 행 1 열, 3 행 1 열, 4 행 1 열, 1 행 2 열, 2 행 2 열, 3 행 2 열… 4 행 4 열의 순서로 블록 안의 데이터를 저장장치에 저장하는 것이다. 암호화를 마치고 저장된 데이터는 로딩 블록에 입력으로 들어왔던 데이터와는



(그림 2) 로딩 블록

전혀 다른 형태를 가지게 된다. 복호화를 할 때는 암호화를 할 때와 동일한 방법을 사용한다. 로딩 블록에 데이터를 읽어 들인 후에 방향을 바꾸어서 데이터를 출력해주면 된다. HSS 에서는 데이터 플로우를 바꾸는 것 외에는 별도의 수학적 연산을 수행하지 않는다. 따라서 암호화에 걸리는 시간이 짧고, 필요로 하는 프로세싱 파워를 최소화 할 수 있다.

실제 HHS 에서는 복잡성의 증가를 위해서 데이터 저장을 위해 로딩블록을 읽을 때 랜덤한 순서로 열 데이터를 가져온다. 또 열 방향으로 사용자가 지정한 크기의 더미 데이터를 중간에 삽입한다. 이 두 가지 방법을 추가 적용함으로써 암호화된 데이터의 안전성을 증가시킨다. 더미 데이터의 크기와 위치, 열 데이터를 읽는 순서는 암호화된 해시 함수 안에 테이블로 저장된다. 사용자와 데이터 베이스는 해시 함수에 사용되는 키를 암호화 시켜서 주고 받게 된다.

HSS 의 암호화 방식이 클라우드 컴퓨팅에 적합한 이유는 다음과 같다.

첫째, 암호화 과정의 오버헤드가 적기 때문에 데이터 베이스의 성능 저하가 적다. HSS 에서 수행되는 연산은 정해진 크기의 데이터 덩어리들의 저장 순서만을 바꿔주는 연산이다. 데이터 플로우 방향을 바꿔주고 중간에 더미 데이터를 삽입하는 것 외에 다른 연산은 없다. 그리고 XOR, 소수 구하기, 2 의 거듭제곱과 같은 별도의 연산들이 여러 번 반복되는 과정이 없기 때문에 함수 깊이가 1 또는 2 에 불과하다. 따라서 암호화 과정에 필요한 시간과 프로세싱 파워를 최소화 할 수 있다.

둘째, HSS 는 저장되는 데이터의 타입에 종속되지 않는다. 데이터 베이스에 저장되는 모든 데이터는 최종적으로 0 과 1 로 이루어진 이진 코드로 저장된다. 그리고 HSS 는 단순하게 데이터 블록들의 저장 순서

를 바꾸어 주는 것이 암호화 과정의 전부이다. 따라서 1bit 를 한 개의 블록으로 지정해 준다면 데이터 타입에 관계없이 HSS 로 암호화를 수행할 수 있다.

셋째, 키 값을 유추할 수 있는 특정 패턴이 생성되지 않는다. HSS 에서 키는 열 데이터의 순서, 더미 데이터의 위치 및 크기, 블록 사이즈 정보의 조합이다. 이 정보들은 데이터들의 저장 순서를 바꾸는 것 외에 다른 연산을 수행하지 않는다. 따라서 암호화의 결과물인 암호문에 키 값을 추측할 수 있는 단서가 남지 않기 때문에 HSS 는 추론 공격과 같은 시도들로부터도 높은 안전성을 보장할 수 있다.

3.2 HSS 성능

일반적인 관점에서 암호화 알고리즘을 분석한다는 것은 암호문이 도청되었을 경우 원래의 데이터를 알아내거나 암호화 키를 찾아내는 것이다. 만약 공격자가 암호화 키를 찾아낸다면 암호화 함수의 역함수를 이용해서 원래의 데이터를 알아낼 수 있다. 따라서 공격자의 가장 큰 목적은 암호화 키를 찾는 키 복구 공격이다. 키 복구 공격 중 가장 쉽게 떠올릴 수 있는 것은 모든 가능한 키를 조사하여 키를 찾아내는 전수 조사 공격이다. DES 는 56bit 의 키 K 를 사용하기 때문에 전수 조사 공격을 수행하기 위해서는 2^{56} 번의 연산이 필요하다. 안전성을 평가하는 시점에서의 컴퓨터로 이 연산을 수행하는 데에 걸리는 시간이 해당 암호화 알고리즘의 안전성을 나타낸다.

HSS 의 안전성도 전수 조사 공격을 통해 분석할 수 있다. HSS 의 키 값은 더미 데이터의 위치와 크기, 데이터 저장시의 열 순서를 가지고 만들어 낼 수 있는 조합이다. 따라서 전수 조사 공격으로 이 조합을 밝혀내는데 얼마만큼의 시간이 필요한지가 HSS 의 안전성이 된다. 만약 더미 데이터를 사용하지 않고 100x100 크기의 로딩 블록을 사용한다면 모든 가능한 조합의 숫자는 1 부터 100 까지 숫자의 순서를 정하는 모든 경우의 수가 된다. 이것은 100 의 계승수를 구하는 것과 같으며 그 결과값은 $9.33262154 \times 10^{157}$ 이다.

DES 는 1999 년 DES ChallengeⅢ 프로젝트에 의해 22 시간 만에 키가 복구된 전례가 있다[11]. 이것을 기준으로 DES 의 해독시간을 약 하루로 가정했을 때 동일한 방법으로 HSS 의 해독시간을 계산해보면 다음과 같다. 전수 조사 공격에 필요한 연산 회수를 계산상의 편의를 위해 2 의 누승으로 표현하면 $9.33262154 \times 10^{157} \fallingdotseq 2^3 \times (2^3)^{157} \fallingdotseq 2^{163}$ 이다. 따라서 HSS 는 전수 조사 공격으로 키 값을 알아내려면 2^{107} 일이 필요하다. 즉, 전수 조사 공격으로 HSS 의 키 값을 알아내려면 수천억 년 이상의 시간이 필요하다. 실제 HSS 가 사용될 때는 더미 데이터 열을 삽입하고 블록 크기 정보 또한 보호되기 때문에 HSS 의 안전성은 더 증가된다.

4. 결론 및 향후 연구과제

본 논문에서는 IaaS 의 신뢰도를 높이기 위해서 클라우드 컴퓨팅 환경의 데이터 베이스에서 낮은 오버헤드로 적용할 수 있는 암호화 시스템을 제안하였다.

HSS 는 낮은 오버헤드, 암호화 알고리즘의 안전성, 다양한 데이터 타입의 수용성을 특징으로 한다. 이러한 특징들은 각각 클라우드 컴퓨팅의 가용성, 보안 신뢰성, 호환성 문제의 해결책으로 사용된다.

하지만 클라우드 컴퓨팅은 여러 가지 기술들이 모여서 이루어진 IT 기술들의 복합체이므로 이 외에도 해결해야 할 문제들이 존재한다. 따라서 추후 연구과제로 사용자와 클라우드 시스템간의 안전한 데이터 전송과 서비스 제공자에 대한 보안 솔루션도 함께 제공할 수 있는 방안에 대해 연구한다.

참고문헌

[1] 최성, 클라우드 컴퓨팅 서비스 플랫폼 기술 동향, 주간 기술동향, 통권 1438 호, pp.27-41, 2010.3

[2] 이주영, 클라우드 컴퓨팅의 특징 및 사업자별 제공 서비스 현황, 방송통신정책, 제 22 권 6 호, pp.1-22, 2010.4

[3] 임철수, 클라우드 컴퓨팅 보안 기술, 정보보호학회지, 19 권 3 호, pp.14-17, 2009.6

[4] 이호균, 이승민, 남택용, 데이터 베이스 암호화 기술과 제품 동향, 전자통신동향분석, 22 권 1 호, pp.105-113, 2007.2

[5] 은성경, 클라우드 컴퓨팅 보안 기술 동향, 정보보호학회지, 20 권 2 호, pp.27-31, 2010.4

[6] 민옥기, 김학영, 남궁한, 클라우드 컴퓨팅 기술 동향, 전자통신동향분석, 24 권 4 호, pp.1-13, 2009.8

[7] 은성경, 조남수, 김영호, 최대선, 클라우드 컴퓨팅 보안기술, 전자통신동향분석, 24 권 4 호, pp.79-88, 2009.8

[8] 정기훈, 노삼혁, 암호화 알고리즘이 웹 서버에 미치는 영향에 대한 연구, 한국정보과학회 학술발표논문집, 31 권 1 호, pp853-855, 2004.4

[9] 이호균, 이승민, 남택용, 데이터 베이스 암호화 기술과 제품 동향, 전자통신동향분석, 22 권 1 호, pp.105-113, 2007.2

[10] “Amazon Web Services: Overview of Security Processes”, http://aws.amazon.com/security, 2010.8

[11] 성재철, 대칭키 암호, 물리학과 첨단기술, 16 권 3 호, pp.2-6, 2007.3