한국시뮬레이션학회 논문지
Vol. 25, No. 3, pp. 77-83 (2016. 9)
http://dx.doi.org/10.9709/JKSS.2016.25.3.077
ISSN 1225-5904

부분 암호화를 위한 해쉬 트리 체인 기반 키 생성 및 관리 알고리즘*

김경민 · 손규식†† · 남승엽†

Key Generation and Management Scheme for Partial Encryption Based on Hash Tree Chain

Kyoung Min Kim · Kyu-Seek Sohn†† · Seung Yeob Nam†

ABSTRACT

A new key generation scheme is proposed to support partial encryption and partial decryption of data in cloud computing environment with a minimal key-related traffic overhead. Our proposed scheme employs a concept of hash tree chain to reduce the number of keys that need to be delivered to the decryption node. The performance of the proposed scheme is evaluated through simulation.

Keyword : Cloud, Partial encryption, Partial decryption, Data protection, Hash Tree Chain

요 약

본 논문에서는 부분 암호화를 위한 키 생성 및 관리 알고리즘의 연구한다. 제안하는 알고리즘에서는 해쉬 트리 구조를 이용하여 적은 양의 해쉬키로부터 다양한 암호화 키를 생성하는 기능을 제공하고 있다. 본 논문에서는 새로운 키 생성 알고리즘을 이용하여 부분 암호화 및 부분 복호화하는 방법을 제시하고 시뮬레이션을 통하여 그 성능을 분석한다.

주요어: 클라우드, 부분 암호화, 부분 복호화, 데이터 보호, 해쉬 트리 체인

1. 서론

통신 및 컴퓨팅 리소스의 발달로 인해 클라우드 컴퓨팅 서비스는 우리에게 많은 영향을 주고 있다. 하지만 네트워크를 이용해서 컴퓨팅 자원을 제공받게 되는 클라우드 컴퓨팅은 그로 인해서 많은 보안성 문제를 가지게 된다(Takabi et al., 2010). 클라우드 환경을 이용하는 가장 큰 장점은 네트워크를 이용해서 어디서나 필요한 자료를 읽고 수정할 수 있는 있다는 것이다. 하지만 이것은 네트워크를 통해 언제나 다른 누군가에게 정보가 노출될 수 있음을 의미한다.

국내와 해외 모두 다양한 종류의 데이터 유출 사태가 발생하고 있다. 국내에서 발생한 대표적인 사례로는 2013년 3월 20일에 발생한 언론 및 금융기관 해킹사고가 있다(Lee, 2014). 악성코드에 의해서 PC내부에 저장되어 있던 개인정보와 같은 다양한 정보가 유출된 사건이다. 해외에서 발생한 네트워크 상의 데이터가 유출된 대표적 사례로는 MS의 기업용 클라우드 서비스의 데이터 유출 사건이 있다(Kim, 2010). 이 사건은 보안 정책의 문제로 클라우드 서비스를 사용하는 기업 직원들이 보유한 업무용 연락처 정보가 엉뚱한 사용자에게 전달된 경우다. 또 다른 사례로는 미국에서 의료 서비스를 제공하는 CareFirst사의 건강관리정보 유출 사건이 있다(Kuranda, 2015). 해커들의 취약점 공격으로 서비스 이용자 약 백십만명의 개인정보가 유출되었다.

위와 같은 데이터 유출 사고 사례들은 저렴한 비용에도 불구하고 기업들이 클라우드 컴퓨팅 서비스 사용를 사용한 인프라 구축을 주저하는 원인이 되고 있다. 이런

* 이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2015R1D1A1A01058595).

Received: 10 June 2016, **Revised**: 18 July 2016, **Accepted**: 21 July 2016

† **Corresponding Author**: Seung Yeob Nam
E-mail: synam@ynu.ac.kr
Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Korea

†† Department of Hacking and Security
Hanyang Cyber University, Seoul, Korea

보안적 문제를 해결하는 대표적인 방법 중 하나는 암호화 알고리즘이다.

일반적으로 매우 중요한 자료라면 전체 내용을 암호화하면 된다. 전체 내용을 암호화하는 경우에는 자료의 내부 검색이 번거로울 뿐만 아니라 많은 시간이 걸리게 된다. 왜냐하면 전체 암호화된 문서에 대해 키워드 검색을 하기 위해서는 문서를 다시 완전히 복호화 해야 하기 때문이다. 이 때문에 Boneh 등(Boneh et al., 2004; Boneh, 2007)이 검색이 가능한 암호화 알고리즘들을 최근에 제안하였다. 하지만 널리 사용되는 자료의 경우 전체 내용을 암호화 및 복호화 하는 것은 많은 오버헤드가 발생한다. 또한 개발되고 있는 검색 가능한 암호화 알고리즘은 대부분 클라우드 서버에서 해당 알고리즘의 설치를 위한 업그레이드를 필요로 한다.

Kim등은 이러한 문제들을 해결하기 위하여 부분 암호화 알고리즘을 개발하였다(Kim et al., 2016). 기존의 클라우드 서버 환경에서 특별한 업그레이드 과정 없이 암호화 되지 않는 부분에 대한 검색기능을 제공하는 알고리즘으로서 Kim 등이 제시한 부분 암호화 알고리즘은 암호화된 데이터의 부분적인 복호화가 가능하다는 장점을 가지고 있다. 해당 알고리즘은 부분 복호화가 필요한 데이터 블록의 키를 전달하는 과정을 필요로 한다. 하지만 부분 복호화 하는 데이터 블록의 개수가 많아진다면 전달해야하는 키의 개수도 점점 증가하게 되어서 많은 네트워크 오버헤드가 발생하는 문제를 가지고 있다. 이러한 문제는 네트워크상에 많은 키가 전송됨으로써 외부에 암호화 키들이 노출될 기회 또한 증가시키게 된다. 따라서 본 논문에서는 이러한 문제를 해결하기 위하여 새로운 방식의 키 생성 및 관리 알고리즘을 연구한다.

본 논문에서는 Kim 등이 이전에 제안한 부분 암호화 알고리즘에 적용할 수 있는 키 생성 및 관리 알고리즘을 제안한다. 이 알고리즘은 해쉬 트리 구조를 기반으로 하여 구성되어 있다. 본 논문에서 기여하는 바는 기존에 제시된 부분 암호화 알고리즘의 성능을 네트워크 오버헤드 관점에서 개선하는 것이다.

각 섹션은 다음과 같은 순서로 진행된다. 먼저 관련연구 섹션에서는 기존에 제시된 검색 가능한 알고리즘 및 부분 암호화 알고리즘을 간략하게 소개하고, 최근 제시된 부분 암호화 알고리즘을 소개한다. 3장에서는 본 논문에서 제안한 알고리즘의 이론적 배경을 소개하고 실제 사용방법의 예시를 제안한다. 4장에서는 네트워크 오버헤드와 복잡도의 두 가지 관점에서 제안하는 방식의 성능을 분석한다. 특히 제안하는 방식과 이전에 제안된 부분 암호화 방식을 비교 평가한다. 그리고 5장에서는 전체적인 정리 및 결론을 맺는다.

2. 관련연구

이 섹션에서는 기존에 제시된 검색 가능한 알고리즘 및 부분 암호화 알고리즘을 간략하게 소개한다.

Song 등(Song et al., 2000)은 암호화된 데이터에 대한 검색 가능한 알고리즘을 개발하였다. 이 알고리즘은 스트림 사이퍼 알고리즘을 기반으로 하고 있다. 검색 가능한 기능을 제공하지만 암호화 단계에서 검색 가능한 키워드를 미리 지정할 필요가 있어 미리 지정되지 않은 키워드에 대해서는 검색이 어려운 한계가 있다.

Shi 등(Shi et al., 2007)은 암호화된 데이터에 대한 검색기능을 제공하고 있다. 이 알고리즘은 검색 알고리즘의 키워드 적용을 다차원으로 적용하는 것이다. 이 알고리즘의 장점은 검색에서 더 많은 키워드가 적용된 우선순위에 따라 검색결과를 출력해 낼 수 있는 것이다. 하지만 이 알고리즘 역시 기존의 검색가능 알고리즘과 마찬가지로 검색가능한 키워드를 암호화 단계에서 설정할 필요가 있다.

Droogenbroeck(Droogenbroeck et al., 2002; Droogenbroeck, 2004)는 이미지에 대한 부분 암호화 알고리즘을 개발했다. 이 알고리즘은 이미지에 대한 부분적인 암호화 기능을 제공하고 있지만 모든 데이터 블록이 동일한 암호화 키를 사용하고 있다. 이 경우 하나의 데이터 블록을 복호화하는 키를 가진 노드는 암호화된 다른 모든 부분을 열수 있게 되어 부분 복호화 기능이 제공되지 않는다.

Panduranga 등(Panduranga et al., 2012)은 Hill Cipher를 이용한 부분 암호화 알고리즘을 개발하였다. 이 알고리즘은 Self-Invertible matrix를 이용하여 부분 암호화를 적용하는 알고리즘으로 이 방식 또한 부분적인 복호화 기능을 제공하지 못한다.

Kim 등(Kim et al., 2016)이 제시한 부분 암호화 알고리즘은 다음과 같은 과정으로 구성되어 있다. 메타 데이터는 주어진 자료의 데이터 블록별로 암호화 여부를 저장하고 있다. 그리고 각 데이터 블록의 암호키를 다르게 설정하고 관리함으로써 부분적인 복호화가 가능하다는 장점을 가지고 있다. 하지만 이 알고리즘은 부분적인 복호화를 할 때 연산하는 데이터 블록의 숫자만큼 키를 전송해 줘야한다. 이것은 네트워크 오버헤드 관점에서 큰 부담이 될 수 있다.

3. 제안하는 부분 암호화 및 복호화 방식

본 논문에서 제안하는 알고리즘의 이론적 배경 및 구체적인 동작을 예시와 함께 설명한다.

먼저 메타 데이터를 이용하여 암호화된 부분의 위치 정보를 저장한다. 평문 데이터 유닛의 전체 개수를 N으로 나타낸다. 문제를 단순화하기 위해서 N은 2의 거듭제곱형태로 표현된다고 가정한다. 각각의 메타 데이터는 비트 단위로 구성되어 있으며, i번째 데이터 블록의 메타 데이터는 b_i로 나타낸다. 메타데이터의 b_i의 값이 0인 경우는 i번째 데이터 블록은 평문인 경우이고, b_i의 값이 1인 경우는 i번째 데이터 블록이 암호화된 것을 나타낸다.

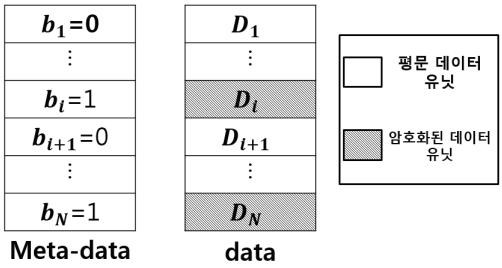


Fig. 1. Meta data diagram

N개의 데이터 블록을 위해서는 N개의 암호키를 생성해야한다. 본 논문에서는 N개의 암호키를 효율적으로 생성 및 관리하기 위하여, 특히, 암호키를 전달해야 하는 경우 암호키 관련 전송 트래픽 오버헤드의 감소를 위하여 해쉬 트리 체인에 기반해서 암호키를 생성하는 방식을 제안한다.

먼저 해쉬 트리의 루트에 해당하는 $h_{0,0}$의 값을 임의로 선택한다 예를 들어서 암호화 하는 노드만이 알고 있는 비밀키 K가 있는 경우에 T를 현재 시간이라고 하면 $h_{0,0} = H(K\|T)$의 형태로 생성이 될 수 있다. 그리고 $h_{0,0}$로부터 그 다음 계층의 해쉬값들을 생성한다. 각 계층의 해쉬값들은 다음과 같은 수식에 따라서 다음 계층의 해쉬값을 생성하는데 사용한다.

$$h_{i+1,2j} = H(h_{i,j}\|0) \quad (1)$$

$$h_{i+1,2j+1} = H(h_{i,j}\|1) \quad (2)$$

H()는 cryptographic 해쉬 함수를 나타낸다. 데이터 유닛의 전체 개수가 N인 경우 가장 아래 계층은 계층 $\log_2 N$이 된다. 식 (1)과 (2)를 반복하여 가장 아래 계층까지 해쉬 트리를 완성시킨 후 가장 아래 계층의 해쉬값을 각 데이터 블록의 암호키로 사용하게 된다. Fig. 2는 N=8일 때 완성된 해쉬 트리를 도식화한 것이다.

부분 복호화 알고리즘을 적용하려면 암호화를 할 때는 각각의 데이터 블록의 암호키가 서로 달라야한다. i번째 데이터 블록의 암호키를 k_i라고 가정하자. i번째 데이터 블록을 암호화하게 될 때, 암호화하는 사용자는 해쉬 트리로부터 암호키를 가져오게된다. N개의 데이터 블록을 가진 해쉬 트리에서 k_i는 $h_{\log_2 N,i}$를 사용한다. k_i를 이용하여 데이터 블록의 평문을 암호화 하게 된다. 암호화에 사용하는 모듈의 경우는 DES나 AES같은 상용화된 암호화 알고리즘을 사용할 수 있다. Fig. 3는 암호화 과정을 도식화한 것이다.

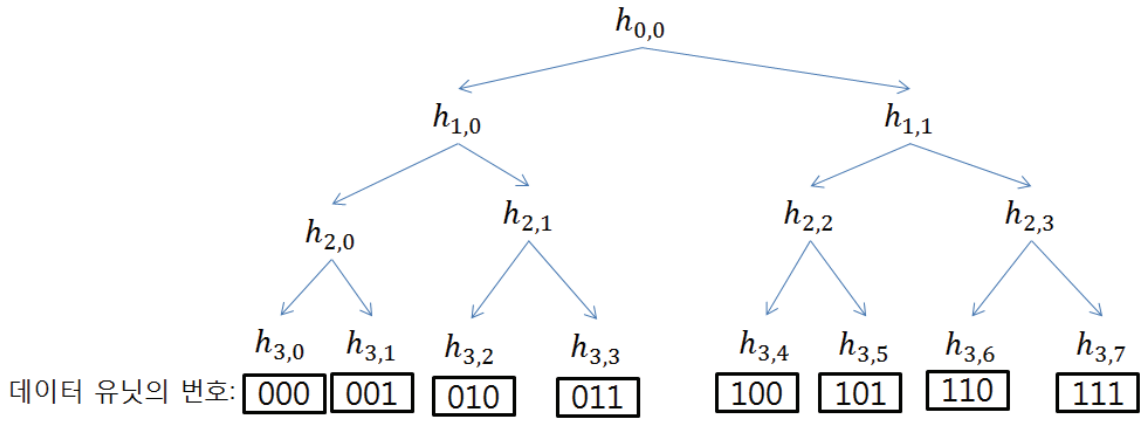


Fig. 2. Binary hash tree for partial encryption for N=8

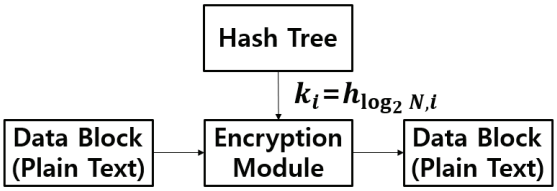


Fig. 3. Encryption diagram

부분적으로 복호화가 필요한 경우 사용자는 복호화가 필요한 데이터 블록의 암호키를 암호화 노드에게 요청한다. 암호화 노드는 전달받은 데이터 블록의 번호를 확인하고 인접한 데이터 블록 간에 상위 계층의 해쉬값이 동일하고 복호화 대상에 포함되지 않는 데이터 블록의 해쉬값을 계산할 수 있는 경우를 피할 수 있는 가장 상위 계층의 해쉬값을 전송한다. 상위 계층의 해쉬값으로부터 하위 계층의 해쉬값을 계산할 수 있다. 그리고 사용자는 이러한 절차에 따라 암호화 노드로부터 전송받은 해쉬값으로부터 암호키를 계산한다. 암호화가 필요한 데이터 블록의 양보다 적은 양의 해쉬값으로부터 암호화에 필요한 데이터 블록의 암호키를 생성할 수 있다. 따라서, 전달이 필요한 암호키의 양을 줄일 수 있다.

데이터 블록의 개수인 N이 8인 경우를 고려해 보자. 만약 한 사용자가 데이터 블록 0, 1, 7번을 부분적으로 복호화 한다면, 기존의 부분 암호화 알고리즘에서는 암호화 노드가 0, 1, 7번의 각각의 암호키들을 전달해야 했다. 하지만 본 논문에서 제안한 알고리즘을 적용할 경우는 전달해야하는 키의 개수를 줄일 수 있다. 데이터 블록 0번과 1번의 암호키는 $h_{3,0}$과 $h_{3,1}$이고 데이터 블록 7번의 암호키는 $h_{3,7}$이다. 암호화 노드는 사용자에게 $h_{2,0}$과 $h_{3,7}$을 전달한다. 사용자는 $h_{2,0}$으로부터 $h_{3,0}$과 $h_{3,1}$을 계산할 수 있다. 또 다른 경우로 사용자가 데이터 블록 1~8번을 복호화 하는 경우를 생각해보자. 본 논문에서 제안한 알고리즘을 이용할 경우 $h_{3,1}$, $h_{2,1}$과 $h_{1,1}$을 암호화 노드가 사용자에게 전송한다. 사용자는 $h_{2,1}$로부터 $h_{3,2}$과 $h_{3,3}$을 계산하고, $h_{1,1}$로부터 $h_{3,4}$~$h_{3,7}$을 계산할 수 있다. Fig. 4는 데이터 블록 1~7을 부분 복호화 하는 경우의 예시를 나타낸다.

전체 복호화에 경우에는 해쉬 트리의 루트에 해당하는 $h_{0,0}$을 관리자가 전송하면 사용자는 그로부터 각각의 데이터 블록의 키를 계산해서 복호화를 진행하면 된다.

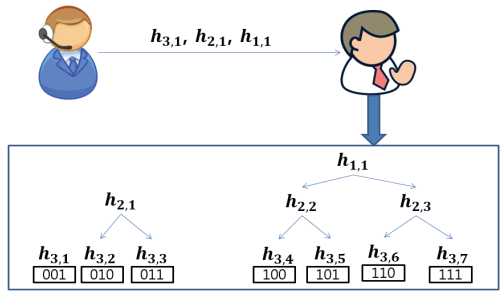


Fig. 4. Example about partial decryption for data block No. 1~7

4. 성능 분석

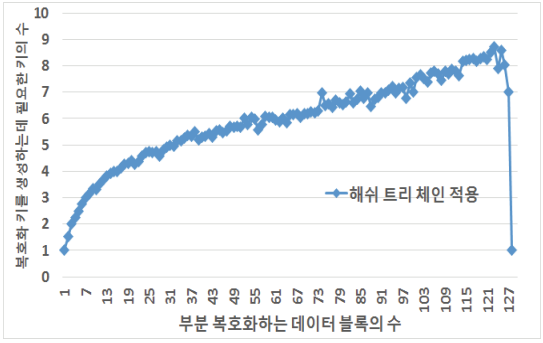
3장에서는 기존 부분 암호화 방식이 가지는 키 관련 트래픽 오버헤드 문제를 효율적으로 해결할 수 있는 새로운 부분 암호화 키 생성 방식 및 그에 기반한 부분 암호화 및 복호화 방식을 논의하였다. 4장에서는 기존의 알고리즘과 본 논문에서 제안한 알고리즘을 적용한 경우의 성능을 분석하게 된다. 모든 실험은 Intel Core i5-3470 CPU, RAM 4GB, Windows 7 32bit환경의 동일한 장치에서 진행되었다. 암호화 알고리즘은 AES를 선택하고, 데이터 블록 및 암호키의 크기는 128bit로 지정하였다. 키 생성을 위해서 사용한 해쉬 함수는 SHA 256이고, 해쉬값의 상위 128bit를 취하여 사용하였다.

4. 1. 네트워크 오버헤드 관점 분석

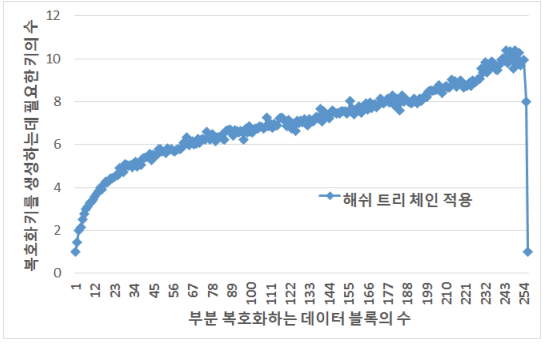
Kim 등이 제안한 부분 암호화 방식(Kim et al., 2016)은 부분 복호화에 데이터블록의 수만큼 복호키를 네트워크를 통해서 전송해야 한다는 문제점을 가지고 있었다. 지금부터는 본 논문에서 제안한 방식이 네트워크 트래픽 오버헤드 측면에서 기존 방식을 단점을 어느 정도 개선하는지 분석한다.

먼저 데이터 블록의 수에 따라 사용자가 필요로 하는 복호화 키의 수를 실험을 통해서 확인하였다. 실험은 다음과 같은 순서로 진행되었다. 먼저 128개의 데이터를 부분 복호화 하는 경우를 가정하였다. 복호화하는 데이터 블록을 그 중 임의로 선택한다. 그리고 데이터 블록을 복호화 하는 키를 생성하기 위해 필요한 키의 최소 개수를 계산하고 기록한다. 그리고 그 과정을 반복해서 평균을 기록한다. 이후 복호화 하는 데이터 블록을 개수를 증가시키며 실험을 반복한다. Fig. 5와 Fig. 6은 데이터 블록 중 N개의 데이터 블록을 부분적으로 복호화 하기 위하여 필요한 최소한의 키의 개수를

실험을 통해서 확인한 그래프이다. 실험은 N=128와 N=256의 두가지 경우에서 진행되었다. 그래프의 가로축은 복호화하는 데이터 블록의 수를 나타내고 세로축은 사용자가 부분 복호화를 위하여 전송받아야하는 최소한의 키의 수를 나타낸다. 제안된 알고리즘을 적용할 경우, 데이터 블록의 수가 증가함에 따라 필요한 최소한의 키의 수는 로그형 함수의 형태를 나타낸다. Fig. 5의 실험 결과에 따르면 본 논문에서 제안하는 키 생성 방식을 사용하는 경우 N=128일 때 전송해야 하는 해쉬값의 개수는 평균적으로 10을 넘지 않으며, N=256일 때 12를 넘지 않는다.



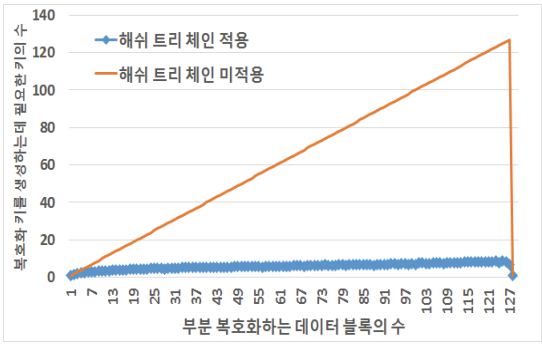
(a) N=128



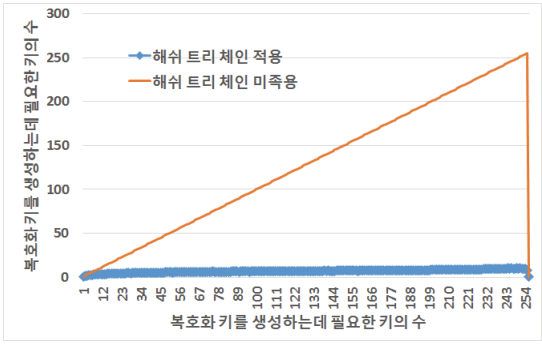
(b) N=256

Fig. 5. Number of keys required for partial encryption in the proposed scheme

그리고 실험을 통해서 기존의 알고리즘을 적용한 경우와 비교를 진행했다. 기존의 부분 암호화 알고리즘은 필요한 키의 수가 선형함수 형태로 증가한다면 제안된 알고리즘을 적용한 경우 로그형 함수의 그래프이므로 증가율이 매우 작은 것을 확인할 수 있다. Fig. 6은 두가지 알고리즘을 적용한 결과를 비교한 실험의 그래프이다.



(a) N=128



(b) N=256

Fig. 6. Number of keys required for partial encryption in the proposed scheme and the existing schme (Kim et al., 2016)

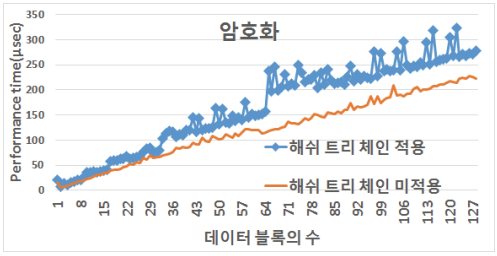
기존의 알고리즘의 경우 N개의 데이터 블록을 부분 복호화하는 경우 N개의 해쉬값 또는 복호키를 필요로 한다. 해쉬 트리 구조를 이용하여 키를 관리할 경우 복호화에 필요한 해쉬값의 수가 훨씬 감소하게 된다. 해쉬 트리를 이용하여 사용자가 키를 생성하기 때문에 상부 계층의 해쉬키로부터 하부 계층의 해쉬키를 생성한다. 전체 복호화 경우에는 하나의 해쉬값만 전송받아 전체 해쉬키 트리를 생성하면 된다. 따라서 전체적으로는 적은 키만 전송받으면 됨으로써 기존의 알고리즘보다 네트워크 오버헤드가 감소하게 된다.

4. 2. 복잡도 관점 분석

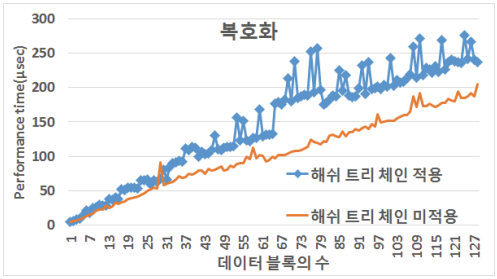
이 섹션에서는 해쉬 트리 구조를 적용한 경우와 적용하지 않은 경우의 복잡도를 실험을 통해서 확인한다. 해쉬 트리 구조를 적용시킬 경우 N개의 데이터 블록이 존재하는 경우 해쉬 트리 구조 상에 최대 2N-1개의 노드가 존재한다. 그리고 암호화 및 복호키를 생성하기 위해서는 상부 계층의 해쉬키가 생성된 후 하부 계층의 해쉬키가

생성된다. 그리고 가장 마지막 계층의 해쉬키가 각각의 데이터 블록의 암호화 및 복호화 키가 된다. 따라서 필요한 암호화 및 복호화의 경우 기존의 알고리즘과 비교하여 많은 연산이 수행된다.

실험을 통해서 두 알고리즘 간의 복잡도를 비교한다. 먼저 128개의 데이터 블록이 있다는 것을 가정하고 그 중 임의의 데이터 블록을 암호화 및 복호화 하는 경우 걸린 시간을 측정한다. 이 과정을 반복하여 평균 시간을 측정하였다. 그리고 암호화 및 복호화 하는 데이터 블록의 수를 증가시키며 실험을 진행하였다. Fig. 7. (a)와 (b)는 해쉬 트리 구조를 적용시킨 경우와 기존의 알고리즘의 경우 걸린 시간을 실험을 통해서 측정한 그래프이다.



(a) Encryption



(b) Decryption

Fig. 7. Comparison of partial encryption time and partial decryption time between the proposed scheme and the existing schme (Kim et al., 2016)

Fig. 7을 통하여 우리는 실제 두 알고리즘간에 복잡도의 차이를 확인할 수 있다. 실제 연산량에서 해쉬 트리 구조를 적용한 경우 더 많은 연산이 발생하여 더 복잡도가 높은 것을 확인할 수 있으나, 두 방식 모두 계산 시간이 대체로 부분 암호화하거나 부분 복호화 하는 블록의 개수에 비례하며 그 차이가 크지 않음을 확인할 수 있다.

5. 결론

통신 인프라의 발달에 따라서 클라우드 서비스가 다양하게 사용되고 있다. 따라서 클라우드 환경에서 중요한 이슈 중 하나인 보안 문제, 특히, 기밀성 문제를 해결하기 위하여 부분 암호화 알고리즘을 연구하였다. 기존에 존재하던 부분 암호화 알고리즘의 경우 부분적인 암호화나 복호화를 위해서는 데이터 블록의 개수만큼의 암호화 키를 전달해야하는 문제가 있었다. 이로 인해서 네트워크 오버헤드 문제가 발생하는 것을 고려하여 해쉬 트리 체인을 이용하여 보다 적은 양의 키 또는 해쉬값을 전달하는 것으로도 동일한 수준의 키를 생성 및 관리할 수 있는 알고리즘을 연구하였다. 본 논문에서 제안하는 알고리즘은 암호화 노드가 작은 개수의 키를 전달함으로써 효과적으로 사용자의 데이터에 대한 접근권한을 제어할 수 있으며 기존의 방식에 비해서 네트워크 오버헤드 측면에서 우수한 성능을 보여주었다. 그리고, 계산 복잡도 측면에서는 기존의 방식과 큰 차이가 나지 않는 성능을 보여주었다.

References

Boneh. D., G.D. Crescenzo, R. Ostrovsky, "Public key encryption with keyword search", *Adcances in Cryptology – EUROCRYPT*, Interlaken, Switzerland, 2004, 506-522.

Boneh, D., B. Waters, "Conjunctive, subset, and range queries on encrypted data“, *Theory of Cryptography*. Amsterdam, Netherlands. 2007, 535-554.

Droogenbroeck. M.V., R. Benedett, “Techniques for a selective encryption of uncompressed and compressed images”, *Advanced Concepts for Intelligent Vision Systems*, Ghent, Belgium, 2002, 90-97.

Droogenbroeck. M.V., “Partial encryption of images for real-time applications”, *Fourth IEEE Benelux Signal Processing*, Belgium, 2004, 11-15.

Kim, H.Y., “MS, BPOS 클라우드 데이터 유출”, 2010, http://www.zdnet.co.kr/news/news_view.asp?artice_id=20101224105449 (Accessed June 09).

Kim, K.M., S.Y. Nam, “Partial encryption algorithm based on block cipher algorithm”, *Proceedings of the 33th KSII Spring Conference*, Kyoungsan, Korea, 2016, 119-120

Kuranda, S., “The 10 Biggest Data Breaches of 2015”, 2015, http://www.crn.com/slide-shows/security/300077563/the-10-biggest-data-breaches-of-2015-so-far

.htm (Accessed June 09).

Lee, D.Y., “2013년 최악의 데이터 유출 사고”, 2014, www.itworld.co.kr/slideshow/86276 (Accessed June 09).

Pandurange, H.T., S.K. Kumar, “Advanced partial image encryption using two-stage hill cipher technique” *International Journal of Computer Application*, Vol.60, No.16, 2012, 14-19.

Shi, E., H. Bethencourt, T.H. Chan, D. Song, A. Perrig, “Multi-dimensional range query over encrypted data”, 2007 *IEEE Symposium on Security and Privacy*, Berkeley, USA, 2007, 350-364.

Song, D., D. Wagner, A. Perrig, “Practical techniques for searches on encrypted data”, *Security and Privacy*, Berkeley, USA, 2000, 44-55.

Takabi, H., J.B.D. Joshi, G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments", *IEEE SECURITY & PRIVACY*, Vol.8 No.6, 2010, 24-31.

김 경 민 (pantom02@naver.com)

2014 영남대학교 정보통신공학과 학사
2014~영남대학교 정보통신공학과 석사과정

손 규 식 (kssohn@gmail.com)

1982 한양대학교 전자공학과 학사
1984 한양대학교 전자통신공학과 석사
1984~2002 LG전선 광통신연구소 책임연구원
2003 KAIST 전자전산학과 박사
2004~현재 한양사이버대학교 해킹보안학과 부교수

관심분야 : 분산컴퓨팅, 인증 프로토콜, 키 관리

남 승 엽 (synam@ynu.ac.kr)

1997 한국과학기술원(KAIST) 전기 및 전자공학과 학사
1999 한국과학기술원(KAIST) 전기 및 전자공학과 석사
2004 한국과학기술원(KAIST) 전자전산학과 박사
2012 Carneige Mellon University 방문 교수
2007～현재 영남대학교 공과대학 정보통신공학과 교수

관심분야 : Network Security, Network Architecture, Wireless Network