파일 특성 기반 효율적인 부분 블록 암호화 알고리즘 설계

이남욱*, 양승수*, 박석천**
*가천대학교 IT융합공학과
**가천대학교 컴퓨터공학과
e-mail:scpark@gachon.ac.kr

Design of Efficient Partial Block Encryption Algorithm Based on File Type

Nam-Uk Lee*, Seung-Su Yang*, Seok-Cheon Park**
*Dept. of IT Convergence Engineering, Gachon University
**Dept. of Computer Engineering, Gachon University(Corresponding Author)

요 약

최근 클라우드 기반의 스토리지 서비스가 지속적으로 성장하고 있다. 클라우드 스토리지 서비스는 데이터의 가용성 및 기밀성을 보장하기 위하여 다양한 암호화 방식을 사용한다. 그러나 기존 암호화 방식은 대용량 데이터 전송 시 처리 및 전송 속도가 저하되는 문제점이 발생한다. 이를 개선하기 위하여 파일 특성을 고려한 효율적인 부분 암호화 알고리즘을 설계하였다.

1. 서론

최근 기업 및 이용자에게 축적되는 데이터의 용량이 증가함에 따라 클라우드 기반 스토리지 서비스 시장이 성장하고 있다[1].

클라우드 스토리지 서비스는 데이터의 가용성 및 기밀성을 보장하기 위하여, 데이터 전송 시 다양한 암호화 방식을 적용한다.

그러나 일반적인 암호화 방식은 데이터 크기에 비례하여 암·복호화 시간이 증가하며, 이로 인해 대용량 데이터 전송 시 처리 및 전송 속도가 저하되는 문제점이 있다[2].

따라서 본 논문에서는 부분 블록 암호화 및 데이터 셔플링(Data Shuffling)을 기반으로 하여, 대상 파일 속성을 고려한 효율적인 부분 암호화 알고리즘을 설계하였다.

본 논문은 1장 서론에 이어 2장 관련연구로 LEA (Lightweight Encryption Algorithm), 부분 암호화, 데이터 셔플링에 대해 조사하였다. 3장에서는 파일 특성 기반 효율적인 부분 블록 암호화 알고리즘을 설계하고, 마지막으로 4장에서 결론을 기술하였다.

2. 관련 연구

2.1 LEA

LEA는 빅데이터, 클라우드 등 고속 환경 및 모바일기기 등 경량 환경에서 기밀성을 제공하기 위해 국내에서 개발된 대표적인 128비트 블록암호 알고리즘이다[3].

ARX(Addition, Rotation, Xor) 기반 GFN(Generalized Feistel Network) 구조이며, 다양한 SW 환경에서 국제 표준암호 AES 대비 1.5배 ~ 2배 성능을 보인다. 표 1은 범용 CPU 별 소프트웨어 구현 효율성 비교 결과이다[3].

<표 1> 범용 CPU 별 소프트웨어 구현 효율성 비교 결과

CPU	LEA-128	AES-128
Intel Core 2 Quad Q6600	9.29	12.2
AMD Phenom II X4 965	8.85	10.35

2.2 부분 암호화

부분 암호화는 파일을 일정한 크기로 나누어, 일부분만 암호화하여 연산 과정을 축소시킨 기법이다. 이를 통해 처리 성능을 향상시킬 수 있으나, 암호화되지 않은 평문으로부터 데이터를 유추할 수 있는 문제점을 가진다[4].

2.3 데이터 셔플링

데이터 셔플링은 데이터를 섞는 과정을 의미한다. 평문 공격에 취약하여, 랜덤 셔플링 테이블을 지속적으로 변경해 주어야 한다. 일반적으로 셔플링 기법의 보안 강도는 셔플링 블록의 개수가 n일 때, n!의 값을 가지게 된다[2].

3. 제안 알고리즘 설계

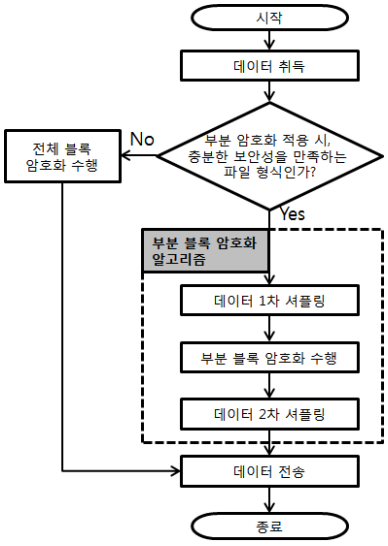
3.1 파일 특성 기반 효율적인 부분 블록 암호화 알고리즘 개요

클라우드 스토리지 서비스에서 파일 전송 시 다양한 암호화 기법을 적용하여 보안성을 보장한다.

그러나 기존 암호화 방식은 대용량 데이터 전송 시 처리 및 전송 속도가 저하되는 문제점이 발생한다.

이를 개선하기 위하여 부분 암호화를 적용할 경우 암호화되지 않은 평문으로부터 데이터를 유추할 수 있는 문제점이 발생하게 된다.

따라서 본 논문에서는 대상 파일의 특성을 고려하여 평문 노출로 인한 데이터 유추의 가능성이 높은 파일은 전체 암호화하고, 데이터 유추 가능성이 낮은 파일은 데이터 셔플링 및 부분 블록 암호화를 통해 처리 및 전송 속도를 향상시켰다. 그림 1은 제안 알고리즘의 전체 개요도이다.



(그림 1) 제안 알고리즘 전체 개요도

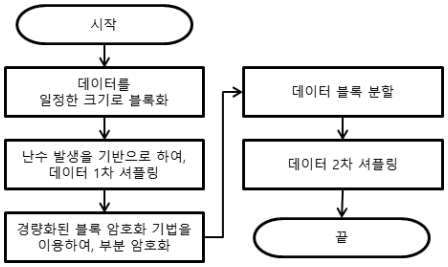
제안 알고리즘은 시스템이 데이터를 취득하고 부분 암호화 적용 시 충분한 보안성을 만족하는 파일 형태인지 확인한다.

텍스트 파일 등 평문 노출로 인해 보안성이 충족되지 않는 파일은 전체 블록 암호화를 수행하여 데이터를 전송하게 된다.

영상, 음원 등, 부분 암호화 적용 시 충분한 보안성을 만족하는 파일 형식은 2회에 걸친 데이터 셔플링 및 부분 블록 암호화를 수행하여 데이터를 전송한다.

3.2 부분 블록 암호화 알고리즘 설계

제안 알고리즘은 부분 암호화 적용 시 충분한 보안성을 만족하는 파일 형식의 경우 부분 블록 암호화 알고리즘을 수행하게 된다. 그림 2는 부분 블록 암호화 알고리즘이다.



(그림 2) 부분 블록 암호화 알고리즘

부분 블록 암호화 알고리즘은 우선적으로 데이터를 일정한 크기로 블록화 한다.

블록화 된 데이터에 난수 발생을 기반으로 한 셔플링을 1회 수행한다. 이후 셔플링 된 데이터 블록에 경량 블록 암호화 기법을 적용하여 부분 암호화를 수행한다.

부분 암호화가 수행된 데이터 블록과 평문 블록을 분할하고 2차 셔플링을 수행한 후 데이터를 전송한다.

수신 측에서는 송신의 역순으로 데이터를 복호화하며 사용된 경량 블록 암호화 기법의 복호화 방법을 사용한다.

본 논문에서 제안한 알고리즘은 파일 형식을 고려한 부분 암호화를 통해 처리 속도를 개선할 수 있으며, 2회에 걸친 데이터 셔플링은 빠른 속도로 처리되어 소요 시간에는 영향을 거의 미치지 않는다.

따라서 제안 알고리즘은 기존 암호화 기법에 비해 처리 시간은 큰 차이가 없으나, 전송 속도 및 보안성은 크게 향상될 것으로 사료된다.

4. 결론

최근 기업 및 이용자에게 축적되는 데이터 용량이 증가하면서 클라우드 스토리지 서비스 시장이 성장하고 있다.

클라우드 스토리지 서비스는 데이터 전송 단계에서의 가용성 및 기밀성을 보장하기 위하여 다양한 암호화 방식을 적용한다.

그러나 기존의 암호화 방식은 대용량 데이터 전송 시 처리 및 전송 속도가 저하되는 문제점이 발생한다.

이에 본 논문에서는 부분 블록 암호화 및 데이터 셔플링을 기반으로 하여 대상 파일 속성을 고려한 효율적인 부분 암호화 알고리즘을 설계하였다.

제안 알고리즘은 데이터의 일부만 암호화하여 처리 속도를 개선하였으며, 2회에 걸친 데이터 셔플링을 통해 보안성을 보장하였다.

따라서 제안 알고리즘은 기존 암호화 방식 대비, 처리 시간은 큰 차이가 없으나 전송 속도 및 보안성은 크게 향상될 것으로 사료된다.

향후 연구로는 효율적인 데이터 셔플링 함수 개발 및 전체 블록 암호화 시 효율성을 향상시키기 위한 방안을 수립하고, 구현 및 평가를 진행하고자 한다.

참고문헌

[1] 유요셉, 김기천, “클라우드 스토리지 전송단계에서의 보안성 강화를 위한 LPES 제안”, 한국통신학회 동계종합학술발표회, 2017

[2] 김현호, 황선태, “USB 메모리의 효율적인 부분 암호화 기법 적용에 관한 연구”, 한국멀티미디어학회 학술발표논문집, 2009

[3] Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, Dong-Geon Lee, “LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors”, Proc. of WISA 2013, vol. 8269, 2014

[4] 유경인, 김민재, 이진영, 조성제, 김준모, “모바일 콘텐츠의 안전한 부분암호화 방법에 대한 연구”, 한국정보과학회 학술발표논문집, Vol. 35, 2008