KIPS Trans. Comp. and Comm. Sys.
Vol.11, No.8 pp.259~268
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2022.11.8.259

Improving Efficiency of Encrypted Data Deduplication with SGX

Dongyoung Koo†

ABSTRACT

With prosperous usage of cloud services to improve management efficiency due to the explosive increase in data volume, various cryptographic techniques are being applied in order to preserve data privacy. In spite of the vast computing resources of cloud systems, decrease in storage efficiency caused by redundancy of data outsourced from multiple users acts as a factor that significantly reduces service efficiency. Among several approaches on privacy-preserving data deduplication over encrypted data, in this paper, the research results for improving efficiency of encrypted data deduplication using trusted execution environment (TEE) published in the recent USENIX ATC are analysed in terms of security and efficiency of the participating entities. We present a way to improve the stability of a key-managing server by integrating it with individual clients, resulting in secure deduplication without independent key servers. The experimental results show that the communication efficiency of the proposed approach can be improved by about 30% with the effect of a distributed key server while providing robust security guarantees as the same level of the previous research.

Keywords : Trusted Execution Environment, Privacy, Encryption, Deduplication, Efficiency

SGX를 활용한 암호화된 데이터 중복제거의 효율성 개선

구 동 영†

요 약

데이터 양의 폭발적 증가에 따른 관리 효율성 제고를 위한 클라우드 서비스 활용이 일상으로 자리잡고 있는 현재, 데이터 프라이버시 보존을 위한 다양한 암호화 기법이 적용되고 있다. 클라우드 시스템의 방대한 컴퓨팅 자원에도 불구하고 다수 사용자로부터 아웃소싱된 데이터의 중복으로 인한 저장 효율성의 저하는 서비스 효율을 현저히 감소시키는 요인으로 작용하면서, 프라이버시가 보장된 암호문에 대한 데이터 중복제거에서의 효율성 향상을 위한 다양한 연구가 진행되고 있다. 본 연구에서는 최신 USENIX ATC에 발표된 Ren et al.의 신뢰실행환경을 활용한 암호문에 대한 중복제거의 효율성 개선을 위한 연구결과를 분석하고 서비스에 참여하는 키 관리 서버를 사용자에 통합함으로써 제3의 독립적인 키 관리 서버의 필요성을 제거하면서도 키 관리의 안정성 개선 방법을 제시한다. 실험을 통하여 제안 기법에서 약 30%의 통신 효율 개선 효과를 얻을 수 있음을 확인하였다.

키워드 : 신뢰실행환경, 프라이버시, 암호화, 중복제거, 효율성

1. 서 론

정보통신기술의 급속한 발전과 데이터 범람으로 효율적 자원 관리와 비용 절감의 중요성이 더욱 커진 현대 사회에서, 클라우드 컴퓨팅 서비스는 유연한 자원 확보 및 유지관리 비용 절감 등의 장점에 힘입어 널리 이용되고 있다[1]. 하지만 클라우드를 비롯한 원격지 저장소에서 관리되는 데이터의 프라이버시 보존을 위하여 수행하는 아웃소싱되는 데이터의 암호화는 원격지 저장소에서의 중복식별 및 관리를 불가능하게 함으로써 서비스 제공자의 효율적 자원 관리를 어렵게 한다[2].

암호화 기법을 적용하여 프라이버시가 보존된 상태에서 암호화된 데이터 중복을 식별하고 제거 및 관리할 수 있는 다양한 연구가 진행되고 있으나[3], 응용 계층의 복잡한 암호학적 연산은 필연적으로 심각한 성능 저하를 초래한다. 신뢰실행환경(Trusted Execution Environment, TEE)은 이러한 소프트웨어 기반의 암호학적 연산 효율성을 개선하면서도 보안 기능을 적용하지 않은 수준의 성능을 유지하기 위하여 하드웨어의 지원을 통한 안전한 연산 및 데이터 관리를 가능하게 한다[4].

근래 프라이버시 보존 데이터 중복제거의 효율성 제고를

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2021R1F1A1064256).
† 종신회원 : 한성대학교 전자정보공학과 조교수
Manuscript Received : April 15, 2022
First Revision : April 26, 2022
Accepted : May 3, 2022
* Corresponding Author : Dongyoung Koo(dykoo@hansung.ac.kr)

위하여 신뢰실행환경인 Intel SGX를 활용한 연구가 활발히 수행되고 있으며[5, 6], 최근 Ren et al.은 Douceur et al.의 수렴 암호(convergent encryption, CE)[7]를 일반화한 Bellare et al.의 MLE(Message-Locked Encryption)[8]와 Keelveedhi et al.에 의해 제안된 DupLESS[9]를 SGX 환경에 적용하여 성능을 개선한 SGXDedup[6]을 제안하였다.

[6]에서는 원격지 저장소인 클라우드와 데이터 아웃소싱의 주체인 사용자 외에도 별도의 키 관리 서버를 두어 단일 장애 지점(single point of failure)의 위험이 존재하며 키 관리 서버와의 통신 오버헤드 증가를 피할 수 없다. 따라서, 본 연구에서는 SGX의 인클레이브가 동일 시스템에서 다른 응용에 영향을 받지 않고 독립적으로 기능할 수 있는 특징을 활용한다. 별도의 키 관리 서버를 두지 않고 [6]과 동등한 수준의 안전성을 유지하면서 동일 기능을 수행하는 방법을 고안하며, 이로부터 얻을 수 있는 효과는 다음과 같다:

1) 키 관리 서버를 각 사용자 응용에 내포함으로써, 독립적인 키 관리 서버로부터 발생 가능한 단일 장애 지점의 위협을 감소시킨다.
2) 키 관리 서버와 사용자 응용의 통신이 동일 시스템 내에서 이루어지도록 함으로써, 키 관리 서버와의 원격 통신으로 인한 오버헤드를 감소시킨다.

2. 관련 연구

본 연구에서 암호화된 데이터에 대한 중복식별 및 제거는 다중 사용자 환경, 블록 단위의 데이터 중복제거, 사용자 측에서의 중복제거를 대상으로 하며, 다수의 선행연구와 주요 관련 기술 중에서 신뢰실행환경의 활용 및 키 관리 서버의 단일 장애 지점에 따른 위험 극복을 주요 목표로 한다.

2.1 Intel SGX

신뢰실행환경은 일반적으로 실행되는 응용이 동작하는 일반 영역(normal area)과 별개로 프로세서에 의하여 응용이 직접 관리되는 신뢰 영역(secure area)을 제공하고 두 영역 사이에서의 정보 교환을 제어함으로써 외부와 고립된 환경(isolated environment)을 제공하는 플랫폼(platform)을 의미한다. 신뢰실행환경에서는 신뢰 영역에서의 연산이 외부의 접근 및 공격으로부터 하드웨어 지원을 통해 보호되기 때문에 복잡한 암호학적 기법을 적용하지 않고 안전한 연산 및 데이터 관리가 가능하다.

신뢰실행환경을 제공하는 플랫폼으로는 ARM TrustZone, AMD SEV(Secure Encrypted Virtualization) 등이 있으며, 특히 Intel SGX(Software Guard Extensions)는 데스크톱 PC를 비롯한 서버 등 상용화되어 널리 활용 가능한 신뢰실행환경으로 시스템에 대한 탈취 및 공격에 의한 프라이버시 침해의 우려가 있는 경우라 하더라도 사용자 수준(ring-3) 응용이 CPU의 관리하에 기밀성 및 무결성이 침해되지 않는 상태에서 독립적으로 수행되는 인클레이브(enclave)라는 신뢰 영역을 제공한다[10]. 인클레이브의 무결성을 보장하기 위하여 동일 시스템 내의 다수 인클레이브 간 검증을 위한 지역 검증(local attestation, LA)과 이종 시스템에 존재하는 인클레이브를 검증하기 위한 원격 검증(remote attestation, RA) 메커니즘이 제공된다. 또한, 메모리 영역에 존재하는 신뢰 영역인 인클레이브에서 처리된 데이터의 영구 보관을 위하여 인클레이브 외부의 일반 영역에 암호화하여 저장함으로써 기밀성을 보장하는 봉인(sealing) 기법이 제공된다.

2.2 Intel SGX를 활용한 암호문 중복제거 성능 향상

신뢰실행환경의 도입에 따라, 2021년 Ren et al.[6]은 암호문에 대한 중복식별 및 제거 과정에서의 암호학적 연산 복잡도에 따른 성능 저하 개선을 위해 Intel SGX를 활용한 암호문 중복제거 기법을 제시하였다. Fig. 1의 시스템 구성과 같이 원격지 저장소에서의 평문에 대한 중복식별을 위한 과정을 간소화하는데, 독립 키 서버로부터 전달받은 키를 이용하여 개별 사용자가 동일 평문에 대하여 동일한 암호문을 생성할 수 있도록 함으로써 암호문 자체에 대한 중복확인만으

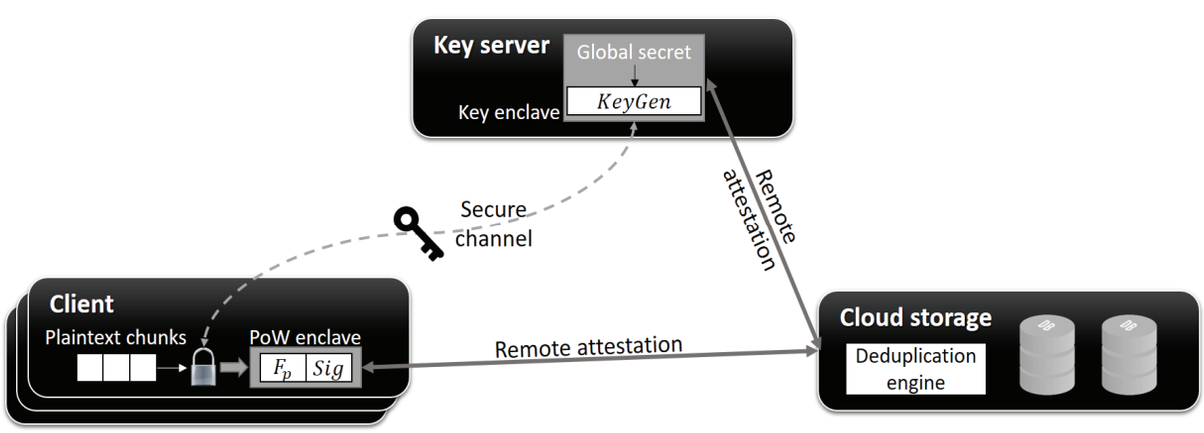

Fig. 1. Ren et al.[6]'s Secure Deduplication Architecture

로 원격지 저장소의 중복제거 용이성을 확보하고 저장 효율성을 개선하였다. 또한, 독립실행공간인 인클레이브에 대한 원격 검증(RA)을 수행함으로써 프로토콜에 참여하는 키 관리 서버 및 사용자의 신뢰성을 확보하도록 설계하였다.

Ren et al.은 Intel SGX를 활용하여 외부로부터의 공격내성을 보장하면서도 선행 연구의 성능을 개선 시켰지만, 데이터 아웃소싱 과정에 필수적인 참여 기관인 사용자와 원격지 저장소 외에 독립적인 단일 키 서버를 추가로 필요로 한다. 이는 다수 키 서버 사이의 동기화에 대한 고려가 이루어지지 않아 지속 가능한 서비스 운용에 있어 키 관리 서버의 단일 장애 지점 위협이 존재하며 안정적인 서비스 제공을 위한 분산 키 서버 환경에 대한 고려가 필요하다.

2.3 별도 키 서버를 필요로 하지 않는 암호문 중복제거

Keelveedhi et al.이 제시한 DupLESS[8]에서 발생 가능한 단일 키 서버의 안정성 개선을 위하여 Liu et al.은 별도의 키 서버가 수행하는 역할을 이미 업로드를 완료한 사용자가 담당하도록 함으로써 키 서버를 필요로 하지 않는 암호문에 대한 중복제거 기법을 제시하였고[11], Duan은 분산 키 관리 모델을 제시하여 단일 장애 지점에 따른 위협을 해소하고자 하였으며 랜덤 오라클(random oracle) 모델을 이용하여 안전성을 증명하였다[12].

Liu et al.의 연구는 이미 업로드를 수행한 사용자가 여전히 온라인 상태로 머물러야 하는 제약이 있으며, Duan의 분산 키 관리 시스템은 비밀분산법(secret sharing scheme)의 응용으로 다수 키 서버 사이에서의 복잡한 연산 및 통신 오버헤드를 야기한다.

본 연구에서는 Intel SGX의 독립실행환경인 인클레이브의 특징을 최대한 활용함으로써 물리적인 별도 키 관리 서버의 필요성을 제거하면서도 SGXDedup[6]의 성능을 보다 개선시킬 수 있는 방안을 모색한다.

3. 개선 목표 및 해결 방안

Ren et al.[6]이 제안한 SGX 기반의 암호문 중복제거는 신뢰실행환경을 활용하여 안전성 저하 없이 효율성 개선을 도모하고 있으나, 독립 키 관리 서버의 존재를 가정하고 있기 때문에 Liu et al.[11]의 연구에서 지적한 바와 같이 단일 장애 지점으로서의 위험을 여전히 내포하고 있다. 본 연구에서는 독립적으로 존재하는 키 관리 서버의 역할을 사용자가 수행하도록 함으로써 별도 키 관리 서버를 필요로 하지 않는 암호문 중복제거 기법을 제안한다.

3.1 개선 목표

1) 데이터 중복제거 절차

DupLESS[9]에 기반하여 Ren et al.이 제안한 SGXDedup[6]의 데이터 중복제거 절차는 Fig. 2와 같다.

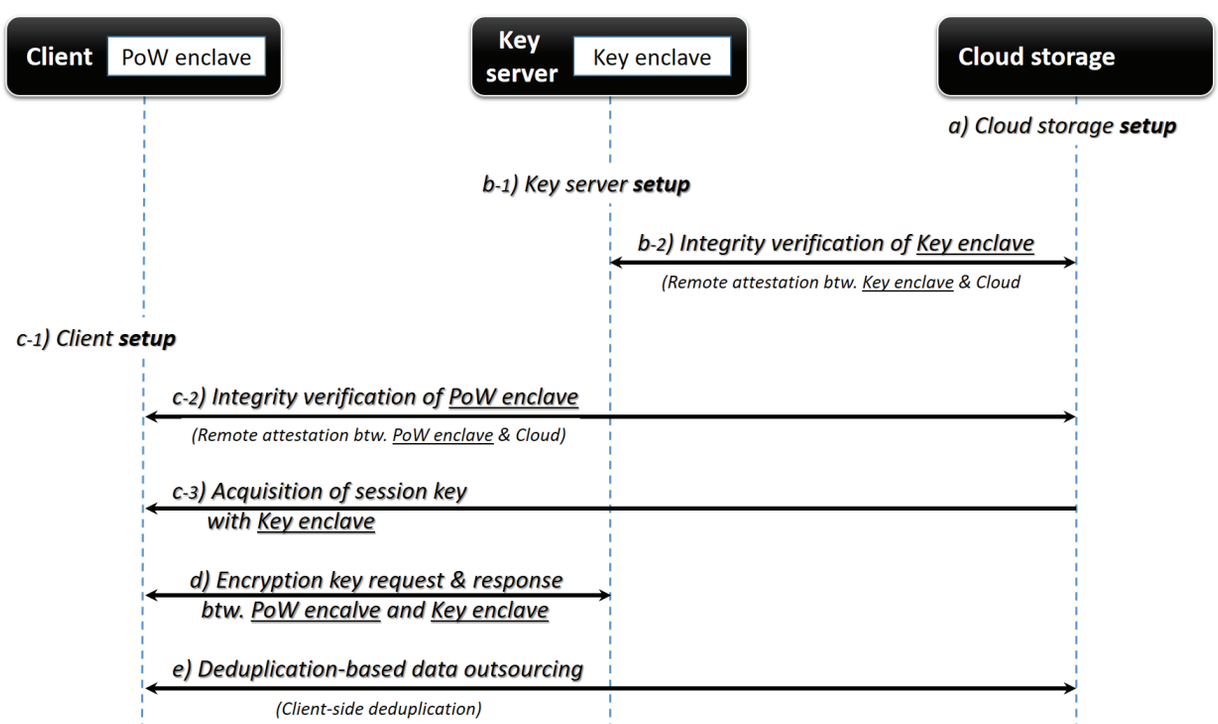

Fig. 2. Deduplication Procedure of Ren et al.[6]'s Approach

a) 원격지 저장소(클라우드) 시스템 구동
b) 키 관리 서버의 구동 및 클라우드 시스템으로부터의 원격 검증을 통한 무결성 확인
c) 사용자 시스템의 구동 및 클라우드 시스템으로부터의 원격 검증을 통한 무결성 확인 절차를 거쳐 키 관리 서버와의 (암호통신용) 세션 키 획득
d) 사용자 시스템의 암호화 키 요청 및 획득을 위한 키 서버와의 통신
e) 사용자 시스템의 중복확인 태그 생성 및 클라우드 시스템 전송을 통한 중복제거와 중복되지 않은 데이터의 아웃소싱

2) 키 관리 서버의 단일 장애 지점 및 통신 복잡도 개선

앞서 기술한 데이터 중복제거 절차의 b) 단계에서는 단일 키 관리 서버가 독립 기관에 의하여 관리된다는 가정이 존재한다. 시스템이 동작하는 동안 안정적인 운영이 보장될 필요가 있는데, 유일한 키 관리 서버의 운영에 따른 단일 장애 지점 위협이 존재한다. 사용자 시스템에 단순 키 관리 기능만을 포함하게 되면 분산 키 관리 서버의 기능을 수행할 수 있으나 무결성 검증을 위한 원격 검증에 수십여 초의 시간이 소요되어 사용자 경험(user experience, UX)의 심각한 훼손을 방지하기 위한 대책이 필요하다.

데이터 중복제거 절차의 c) 단계에서는 클라우드로부터의 무결성 검증을 마친 사용자는 키 관리 서버의 통신 비밀키를 획득하여 별도의 키 관리 서버와 통신이 이루어진다. 사용자 시스템 내부에 키 관리 서버의 기능을 내포하면, 키 관리 서버와의 통신 비밀키 공유를 위한 별도 통신이 필요하지 않으며 원거리에 존재하는 단일 키 관리 서버와의 통신에 따른 효율성 저하를 방지할 수 있다.

3.2 키 관리 기능의 사용자 분산

SGXDedup[6]의 단일 키 관리 서버는 시스템의 모든 사용자에 대한 정보를 관리하고, 키 서버의 마스터 비밀(master secret)키로부터 사용자 요청에 대응하는 암호화 키를 보안 채널을 통해 전달하게 된다. Intel SGX의 인클레이브는 외부로부터 독립된 실행 환경을 제공하여 시스템의 탈취 및 공격 등으로부터 기밀성과 무결성이 보장된 상태에서의 동작이 가능하므로, 동일 시스템에서 관리자 권한을 가진 악의적 행동으로부터 독립적인 기능을 수행하는 개체로 간주할 수 있다.

키 관리 서버의 기능을 각 사용자 시스템에 분산시키는 경우, 원격 검증을 대체할 수 있는 무결성 보장을 위한 별도의 메커니즘이 제공되어야 하는데 이는 지역 검증(LA)을 통하여 해결할 수 있다.

3.3 키 관리 서버와 사용자 간 통신 오버헤드 개선

단독 키 관리 서버와 다수 사용자에 의한 키 요청은 키 관리 서버 부하와 더불어 보안 통신으로 인한 지연을 커지게 한다. 키 관리 서버를 사용자 응용에 내포하여 키 관리 서버와의 통신 효율성을 개선할 수 있다. 나아가 동일 시스템 내에서의 인클레이브 간 메시지 전달을 활용함으로써 원격 시스템 간 소켓 통신 오버헤드를 감소시킬 수 있다.

4. 키 관리 기능을 내포한 사용자 기반 중복제거

본 연구에서 제시하는 데이터 중복제거 시스템에서는 사용자 응용이 키 관리 기능과 중복확인 기능을 내포하여, 독립적인 키 관리 서버의 운영이 필요하지 않다.

4.1 시스템 구성

SGX를 활용한 암호문 아웃소싱 및 중복제거 시스템은 Fig. 3과 같이 크게 클라우드 서비스 제공자 S와 사용자 C로 구성되며, 사용자는 다수의 독립적인 개체로 이루어질 수 있다. 다만, 별도의 물리적 키 관리 서버가 존재하지 않고 C에서 구동되는 하나의 응용이 키 서버 역할과 중복확인을 위한 기능을 수행하는데, Ren et al.[6]의 가정과 같이 초기 시스템 설정 단계에서 신뢰할 수 있는 기관에 의하여 생성된 인클레이브가 동적 라이브러리 형태로 S에 전달되며 S가 제공하는 서비스를 이용하는 각 사용자 C에게 배포된다고 가정한다. 원격 저장 서비스를 이용하고자 하는 C는 S로부터 키 관리(key management)와 중복제거(proof-of-ownership, PoW)를 위한 두 인클레이브를 다운로드 받아 자신이 소유한 데이터를 아웃소싱하는데, [6]에서와 같이 동적 라이브러리에

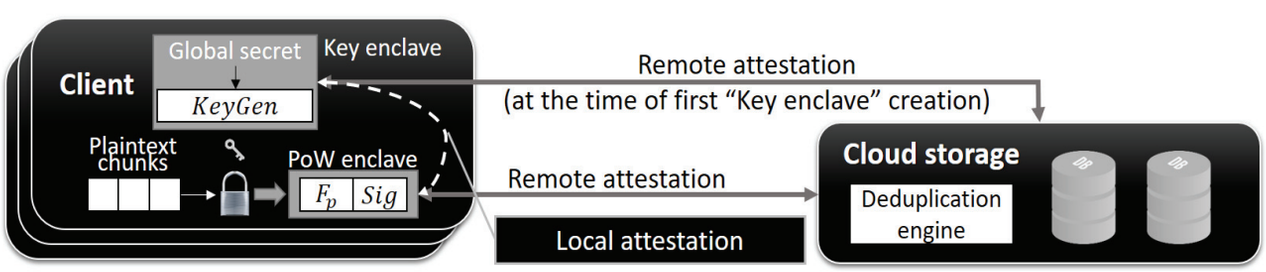

Fig. 3. System Architecture of the Proposed Scheme

대한 역공학(reverse engineering)을 통한 키 복원은 불가능한 것으로 가정한다.

1) 키 관리 인클레이브($Key\ enclave$)

DupLESS[9]의 키 서버는 사용자의 데이터를 아웃소싱하기 전에 암호화 키에 대한 서명을 생성하는 역할을 한다. RSA 기반 은닉 서명(blind signature)을 위하여 사용자는 데이터 $plain$에 대한 해시값 $h(plain)$을 생성한 후 임의값 x를 곱하여 $D = x \cdot h(plain)$를 키 서버에 전달하고, D를 전달받은 키 서버는 자신의 서명키 $sign$을 이용하여 서명값 $\sigma(D) = D^{sign}$를 계산하여 사용자에게 되돌려준다. 사용자는 키 서버로부터 받은 서명으로부터 임의값을 제거하기 위하여 키 서버의 검증 키 $verify$를 이용하여 데이터에 대한 서명 $\sigma(plain) = h(plain)^{sign}$을 아래와 같이 계산한다:

$$\begin{aligned} x^{verify} \cdot \sigma(D) &= x^{verify} \cdot (x \cdot h(plain))^{sign} \\ &= x^{verify} \cdot x^{sign} \cdot h(plain)^{sign} \\ &= 1 \cdot h(plain)^{sign} \\ &= h(plain)^{sign} = \sigma(plain). \end{aligned}$$

사용자는 아웃소싱할 데이터에 대한 해시값의 서명을 암호화 키로 사용하여 데이터를 암호화하고 원격지 저장소에 업로드한다.

제안 기법의 키 관리 인클레이브($Key\ enclave$)는 [6]과 같이 인터넷 등 공개 채널을 통하여 암호화 키를 그대로 주고받는 대신, SSL/TLS 암호통신을 이용한다. 사용자가 아웃소싱하고자 하는 데이터의 해시값을 전달받은 $Key\ enclave$는 서명한 값을 암호 채널의 세션키로 암호화하여 전달한다. 따라서 인클레이브 내부에 저장된 서명키를 통한 서명이 이루어지므로 사용자는 서명키를 알지 못하는 상태에서도 아웃소싱하고자 하는 데이터에 대한 서명을 전달받아 암호화를 수행할 수 있다.

2) 중복확인 인클레이브($PoW\ enclave$)

Halevi et al.에 의하여 제안된 사용자 측에서의 데이터 중복제거를 위한 소유권 증명(proof-of-ownership, PoW)은 데이터 소유자인 사용자가 원격지 저장소에 업로드를 수행하기 전에 동일 데이터가 원격지 저장소에 존재하는지 미리 확인함으로써 중복 데이터 전송에 따른 통신 효율성 저하를 방지하는 기법을 제시하였다[13]. Halevi et al.은 중복확인 대상을 평문으로 가정하였으며, 아웃소싱하는 전체 데이터의 해시값 등 매우 작은 값에 대한 중복확인의 위험성을 지적하며 머클 해시 트리(Merkle hash tree)를 이용하여 시도-응답(challenge-response) 인증을 거침으로써 데이터의 일부 정보로부터 전체 소유권을 획득하거나 정보를 유추하려는 악의적 공격에 대한 대응 방안을 제공하였다.

제안 기법에서는 [6]과 같이 $PoW\ enclave$와 S가 SSL/TLS 암호통신 세션을 생성하여 중복확인 태그(tag)를 전송하기 때문에, 동일 데이터를 소유한 다른 사용자 C'으로부터의 도청 등에 따른 태그의 기밀성 침해 위험을 방지한다.

4.2 제안 기법의 중복제거 절차

키 서버 인클레이브(Key enclave) 내부에는 클라우드 및 키 서버 각각의 고유한 초기 비밀값이 동적 라이브러리 내부에 포함되어 있으며, 중복확인 인클레이브($PoW\ enclave$)는 별도의 비밀 값 없이 키 관리 인클레이브($Key\ enclave$)와 클라이언트 시스템(C) 내부에서 기밀성과 무결성이 보장된 상태에서 인클레이브 간 메시지 통신을 수행하고 생성한 암호문에 대한 중복을 S로부터 확인한다. 중복되지 않은 암호문의 업로드 요청이 있으면 S는 C에게 암호문의 업로드를 요청한다. 제안 기법의 동작은 Fig. 4 및 Table 1과 같다.

a) 원격지 저장소(클라우드, S) 시스템 구동 (라인 1): S는 사용자의 $PoW\ enclave$ 및 $Key\ enclave$로부터 원격 검증을 수행하기 위한 보안 채널을 생성하고, 사용자의 $PoW\ enclave$와 $Key\ enclave$가 이전에 S로부터 원격 검증받은 기록을 기억장치로부터 읽어온다.

b) 사용자 시스템(C)의 구동 (라인 2) 및 $PoW\ enclave$의 원격 검증 (라인 3-5): C는 데이터 아웃소싱을 위한 $PoW\ enclave$ 및 $Key\ enclave$를 기억장치에서 읽어와 생성한다. S로부터 $PoW\ enclave$에 대한 원격 검증을 수행하여 인증이 실패한($\perp$이 반환된) 경우에는 응용을 종료한다.

c) $Key\ enclave$의 검증 (라인 6-16): C가 최초로 $PoW\ enclave$ 및 $Key\ enclave$를 생성한 경우에는 C에 의한 $Key\ enclave$의 자체 검증이 불가능하므로 S는 단계 b)에서 C에게 K_{RA}응답을 보내 원격 검증을 요청 (라인 6-10)하고, 검증이 실패한 경우 C는 응용을 종료한다. $Key\ enclave$가 S로부터 이미 원격 검증을 받은 경우에는 동일 $Key\ enclave$에 대한 검증을 S 대신 C의 $PoW\ enclave$에 의한 지역 검증(LA)을 수행함으로써 $Key\ enclave$의 무결성을 보장할 수 있다 (라인 11-15). 지역 검증에서 $Key\ enclave$의 무결성 검증이 실패하면 C는 응용을 종료한다.

d) 암호문 생성을 위한 키 요청 및 획득 (라인 12): SGXDedup[6]과 달리, 제안 기법에서는 사용자 시스템(C)의 응용 내에 존재하는 $PoW\ enclave$ 및 $Key\ enclave$ 간의 암호화 키 요청 및 응답이 이루어질 수 있으므로, 원격지 소켓 통신을 대체하여 인클레이브 간 호출 및 메시지 전달을 이용하여 효율적 키 요청 및 획득이 가능하다.

e) 암호문 생성 (라인 20) 및 데이터 중복제거 (라인 21): C에 의한 데이터 중복확인 태그의 클라우드 시스템 전송을 통하여 중복여부를 확인하고 중복되지 않은 데이터의 아웃소싱을 수행한다.

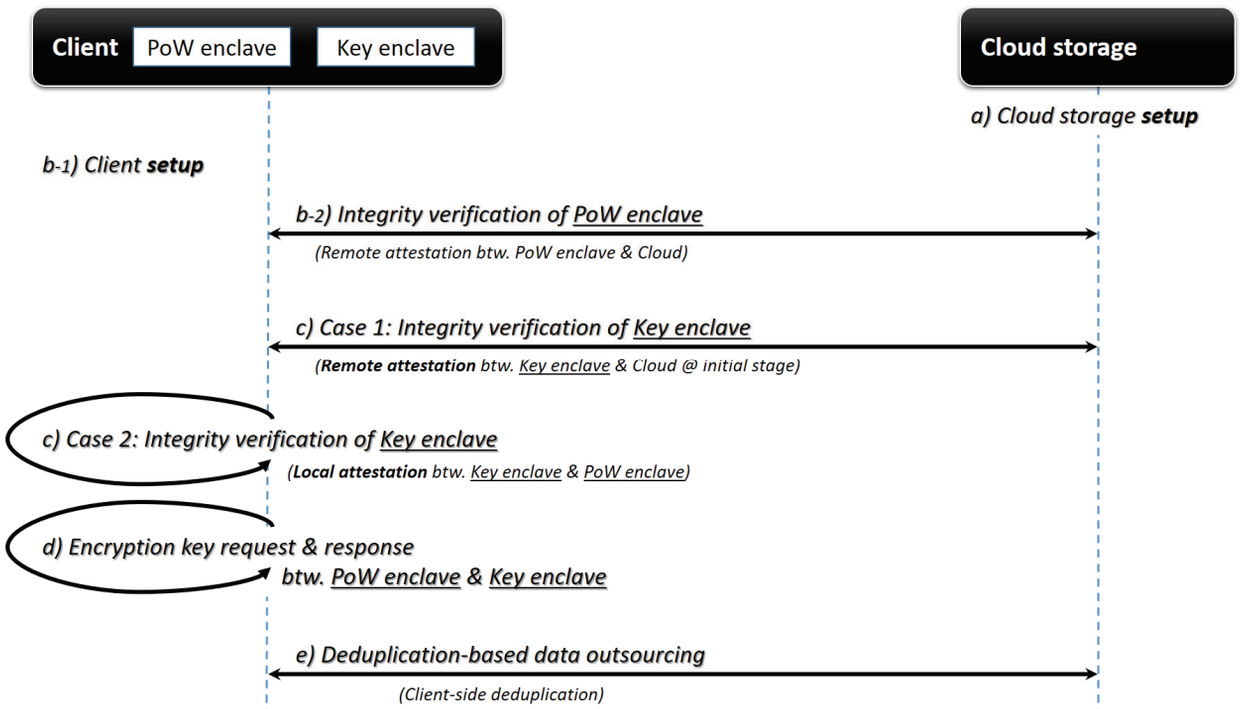

Fig. 4. Deduplication Procedure of the Proposed Scheme

Table 1. Pseudocode of the Proposed Scheme

$Proposed(S, C)$ // overall system operation

1: $Setup_S()$
2: $(Key\ enclave; PoW\ enclave) \leftarrow Setup_C()$
3: $res_{PoW} \leftarrow RemoteAttestation_{PoW\ enclave \leftrightarrow S}()$:
4: if $pow\_res = \perp$: // RA of 'PoW enclave' failure
5: $exit()$ // terminate the program
6: else if $res_{PoW} = K_{RA}$: // RA of 'Key enclave'required
7: $res_{Key\_RA} \leftarrow RemoteAttestation_{Key\ enclave \leftrightarrow S}()$
8: if $res_{Key\_RA} = \perp$: // RA of'Key enclave'failure
9: $exit()$ // terminate the program
10: endif
11: else: // authorized 'PoW enclave' & 'Key enclave'
12: $res_{Key\_LA} \leftarrow LocalAttestation_{Key\ enclave \leftrightarrow PoW\ enclave}()$
13: if $res_{Key\_LA} = \perp$: // LA of 'Key enclave' failure
14: $exit()$ // terminate the program
15: endif
16: endif // completion of Remote (or Local) Attestation
17: while there is data to be outsourced:
18: do
19: $enc\_key \leftarrow KeyReq_{PoW\ enclave \leftrightarrow Key\ enclave}(plain)$
20: $(cipher, tag) \leftarrow Encrypt_{PoW\ enclave}(enc\_key, plain)$
21: $PoW_{PoW\ enclave \leftrightarrow S}(cipher, tag)$
22: endwhile

5. 실험 및 분석

제안 기법의 효율성 개선을 확인하기 위하여 Ren et al.이 제시한 SGXDedup[6]과 제안 기법을 구현하여 저장 및 통신, 연산 오버헤드에 따른 효율성을 비교 검증한다. 실험 환경은 리눅스 (Ubuntu 18.04.6 LTS, 64bits)에서 Intel SGX 구동을 위한 라이브러리인 driver (v2.6.0_4f5bb53), SDK (v2.7.100.4), SGX SSL (version lin_2.5_1.1.1d), OpenSSL (v1.1.1d)을 설치하여 Intel i7-10710U(1.10-4.70GHz) 시스템에서 동일 연산을 100회 반복하여 얻은 평균을 비교 및 분석하였다.

5.1 통신 오버헤드

1) $PoW\ encalve$ 및 $Key\ enclave$ 인증

Intel SGX에서 신뢰 영역인 인클레이브의 무결성 검증을 위한 원격 검증(RA)과 지역 검증(LA) 메커니즘 중, Ren et al.의 SGXDedup [9]은 원격 검증(RA)만을 이용하여 각 참여자의 인클레이브에 대한 무결성을 보장한다.

제안 기법에서는 사용자 시스템(C) 응용이 두 개의 인클레이브를 운영하므로, 원격 검증(RA) 외에도 지역 검증(LA)을 활용한다. 클라우드 스토리지 서비스(S)에 처음 로그인하는 사용자(C)는 $PoW\ enclave$와 $Key\ enclave$에 대한 원격 검증을 수행한 이후, 반복되는 로그인에서는 검증받은 $Key\ enclave$에 대하여 지역 검증을 수행한다. $PoW\ enclave$에 대한 검증은 [6]과 동일한 방법으로 인클레이브에서 저장한 정보(sealed

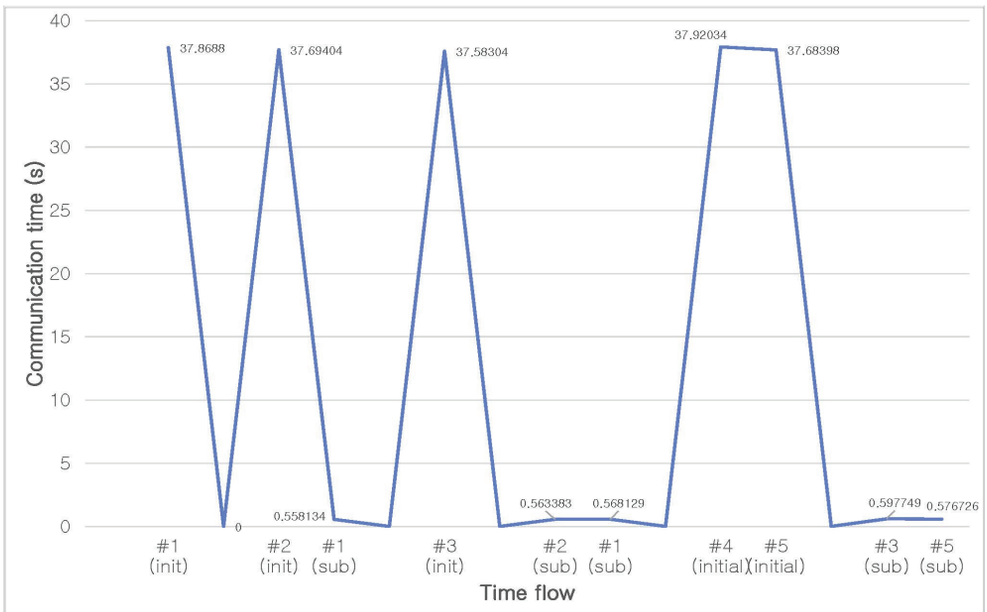

Fig. 5. Comparison of Communication (log-in) Overhead

information)를 재활용한 원격 검증을 수행한다.

*Key enclave*에 대한 원격 검증은 인텔 검증 서비스(Intel attestation service, IAS) 서버를 이용하기 때문에 높은 지연(평균 35.3903초)을 수반하므로 지속적인 원격 검증은 사용자 경험을 현저히 해친다. 이를 개선하기 위하여 제안 기법에서는 최초 로그인 이후 반복되는 로그인 시도에 대하여, 동일 시스템 내부에서 수행되어 보다 빠른 서비스 이용이 가능한 지역 검증을 활용(평균 0.04232초)하여 사용자 경험을 개선한다. Fig. 5는 시간 경과에 따라 5개의 클라이언트 시스템에서의 최초 로그인(initial log-in, init) 및 반복 로그인(subsequent log-in, sub)에 따른 지연 시간을 표시한다.

2) 암호화 키 생성을 위한 *Key enclave*와의 통신

[6]에서 사용자의 *PoW enclave*와 키 관리 서버에 존재하는 *Key enclave* 사이의 통신량 증가는 서비스 지연을 유발한다. 특히 다수 사용자 환경에서는 모든 사용자의 요청을 단일 키 관리 서버가 처리해야 하므로 생성해야 하는 키의 수에 비례하여 병목 현상이 발생할 가능성이 증가한다. 또한, Intel SGX에서는 인클레이브에 할당될 수 있는 메모리 영역이 최대 128MB로 제한되기 때문에 확장성(scalability)의 제약이 따른다.

암호화하고자 하는 데이터 블록(chunk)의 키 생성을 위한 *PoW enclave*의 요청을 전달받은 *Key enclave*는 생성한 키를 회신하는데, SHA-256을 적용한 블록 해시 32B와 메시지 유형을 나타내는 메타 정보를 포함(총 블록 수x32+32B)한 서명된 암호화 키를 주고받는다. 실험에서는 가변 길이 블록 구분(variable-sized chunking) 기법을 적용하였으며, 한 블록은 최소 4,096B에서 최대 16,384B로 평균 8,192B가 되도록 설정하였다. 1MB 데이터로부터 평균 124개의 키가 생성되었으며, 250MB 데이터로부터는 평균 24,111개의 키가 생성되었다.

제안 기법은 *PoW enclave*와 *Key enclave*가 동일 시스템에 존재하여 인클레이브 간 메시지 전달이 일정한 속도를 유지하지만, [6]은 원격지에 존재하는 시스템 사이의 통신 환경에 큰 영향을 받게 된다. Fig. 6은 네트워크 환경 및 데이터 블록의 수에 따른 평균 키 송수신 소요시간이다. Wikipedia[14]의 2020년 5월 기준 국가별 인터넷 평균 속도에 따르면, 대한민국의 인터넷 다운로드 속도는 세계 2위로 유선 환경에서는 평균 40.8~241.58Mbps이고 모바일 환경에서는 평균 59.0Mbps이다. 네트워크 전송 속도가 낮을수록 동일 크기의 데이터 전송 소요시간이 증가하고, 데이터 크기가 클수록 블록의 개수 또한 비례하여 늘어나기 때문에 필요한 키의 개수가 증가하게 된다. 100MB 데이터에 대한 제안 기법에서의 평균 키 전송 시간은 0.00232초로, [6]의 네트워크 환경이 100Mbps인 경우인 0.00310초와 비교하여 33.62%의 속도 향상을 확인할 수 있다.

5.2 저장 오버헤드

SGXDedup[6]과 제안 기법의 사용자 수(n_C)에 따른 각 구성원의 저장 오버헤드는 Table 2와 같다. s_{ID}는 사용자 식별자를 위한 저장 공간이며, s_{PW}는 패스워드(또는 대응되는 해시값)에 할당된 저장 공간이다.

S에서 암호문 저장을 위해 사용되는 공간(s_{DATA})은 두 기법 모두 동일한 암호화 기법을 적용하기 때문에 같은데, 제안 기법은 사용자와 키 관리 서버의 통합에 따라 사용자마다 S에 의한 *PoW enclave* 및 *Key enclave*의 원격 검증 수행여

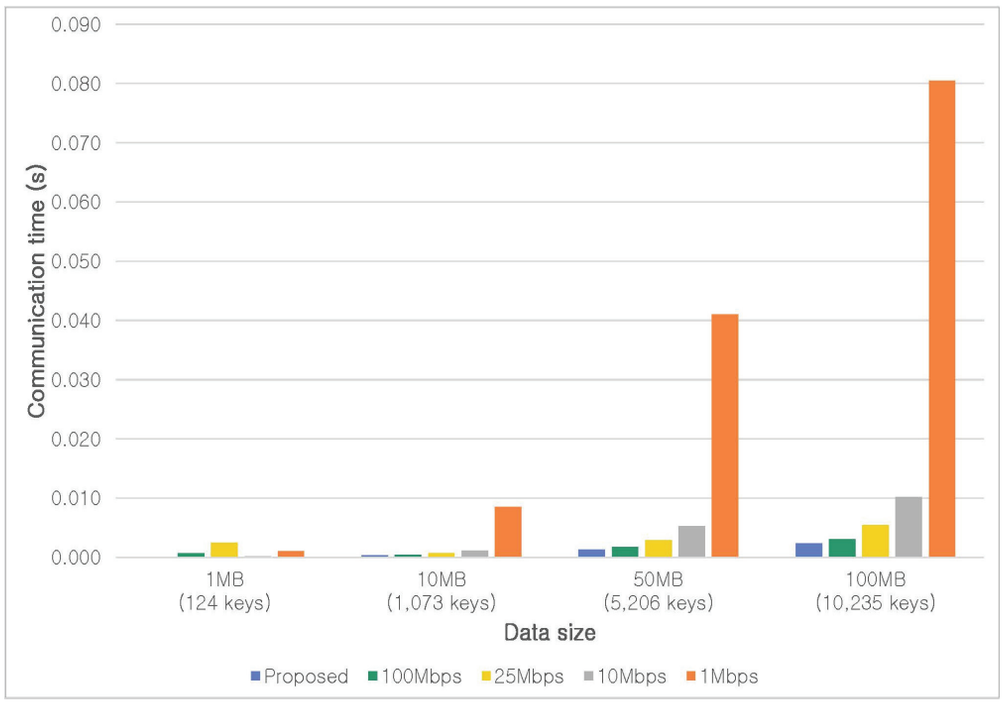

Fig. 6. Comparison of Communication Overhead According to the Size of Data (Number of Data Chunks)

Table 2. Comparison of Storage Overhead

Entity \ Scheme	Ren et al.'s [9]	Proposed scheme
Cloud storage (S)	$n_C \cdot (s_{ID}+s_{PW})+s_{DATA}$	$n_C \cdot (s_{ID}+s_{PW}+s_{AUTH})+s_{DATA}$
Key server ($Key\ enclave$)	$n_C \cdot s_{NONCE}$	$s_{ID}+s_{PW}$
Client($C, PoW\ enclave$)	$s_{ID}+s_{PW}$	

부 기록을 위한 추가 공간(s_{AUTH})을 필요로 한다. 이는 각 사용자에 대한 비트맵(bitmap) 형식으로 저장할 수 있으며, $s_{ID}=4$, $s_{PW}=32$인 경우 1% 미만의 공간을 차지한다. 따라서, s_{AUTH}의 추가 저장 공간으로 인한 제안 기법에서 전체 클라우드 시스템에 미치는 저장 효율성 저하는 크지 않음을 알 수 있다.

단일 키 관리 서버가 모든 사용자에게 서로 다른 임의값(nonce)을 관리하는 [6]에서는 $Key\ enclave$가 임의값 인덱스(nonce index, s_{NONCE})를 위한 추가 저장공간이 필요하지만, 제안 기법에서는 각 사용자 C가 $Key\ enclave$를 내포하고 있으므로 다른 사용자의 임의값에 대한 중복을 확인하지 않고 C 내부의 $PoW\ enclave$에 의한 키 요청 및 응답을 수행하여 별도 인덱스 테이블이 필요하지 않다. 다만, 두 기법 모두 인클레이브 간 통신에서의 기밀성 보장을 위하여 송수신되는 메시지는 암호화된다. 제안 기법에서의 중복제거 과정은 [6]의 메커니즘을 따르기 때문에 $PoW\ enclave$에서의 저장 효율성 차이는 존재하지 않는다.

5.3 연산 오버헤드

SGXDedup[6] 및 제안 기법의 암호문에 대한 데이터 아웃소싱 및 중복제거 과정은 각각 Fig. 2 및 Fig. 4의 절차를 따르며, 중복제거 각 단계에서의 소요시간은 Fig. 7과 같다. [6] 및 제안 기법에서 S 초기화에 각각 0.04577초 및 0.047764초가 소요되었다. C의 $PoW\ enclave$ 초기화에는 각각 0.0018초 및 0.0017초가 소요되었는데, S로부터 사전 인증을 받아 로컬 인증 정보를 활용하는 경우(sealed log in, 0.00156초 및 0.0016초)와 사용자 정보를 이용한 S의 원격 인증(RA)을 받는 경우(0.0018초 및 0.0017초)에 있어서도 연산 시간에는 큰 차이를 보이지 않았다. $Key\ enclave$ 생성은 $PoW\ encalve$와 유사한 시간(0.00156초 및 0.0018초)이 소요되었으며, 1MB 데이터로부터 124개의 키를 생성하는 경우 0.032405초 및 0.030396초가 소요되었다. 또한, 아웃소싱된 암호문의 중복식별에는 0.043386초 및 0.042766초가 소요되었다. [6]과 제안 기법에서 $PoW\ enclave$와 $Key\ enclave$를 생성하는 순서에는 변화가 있으나, 두 기법 모두 동일 연

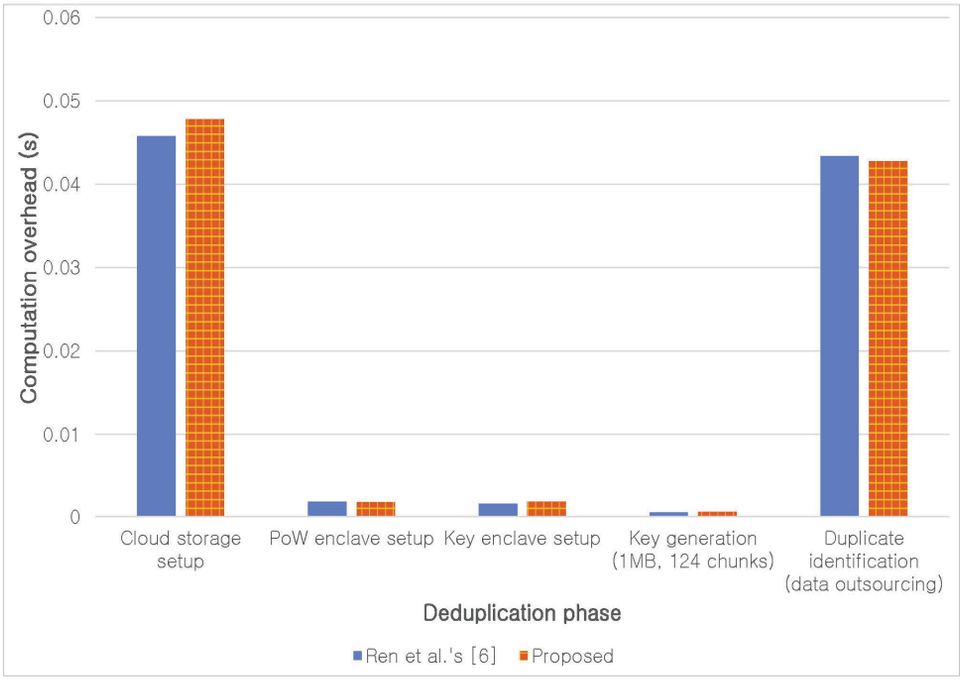

Fig. 7. Comparison of Computational Overhead

산을 수행하므로 연산 소요시간은 실행 환경에 따른 오차 범위 내에서 동등한 수준의 성능을 보임을 알 수 있다.

5.4 키 관리 기능의 사용자 분산에 따른 안전성

제안 기법에서 원격 저장소(S)는 $Key\ enclave$와의 통신을 위한 세션 키를 사용자에게 전송할 필요가 없으며, 사용자가 최초로 $Key\ enclave$를 생성하여 무결성 보장을 위한 원격 검증 이후에는 C에 의한 지역 검증을 수행하므로 추가적인 통신이 필요하지 않다. 다수 사용자에 의하여 생성된 각 $Key\ enclave$는 S의 초기화 단계에서 한 번씩의 원격 검증이 필요하여 초기 통신량은 상대적으로 증가하지만, 이는 분산 키 서버에 따른 안정성 개선과 상호보완적 특징(trade off)을 가지는 것으로 볼 수 있다.

SGXDedup[6]은 독립적으로 존재하는 키 관리 서버에 의한 임의 키 선택이 가능하며, 단독 키 서버에 의한 에스크로(escrow) 위험을 제한하기 위하여 원격지 저장소(S)에 의하여 제공된 비밀정보를 결합하여 마스터 키(master secret)를 생성하도록 한다. 제안 기법에서는 시스템 초기화 단계에서 신뢰할 수 있는 기관으로부터 생성된 비밀정보를 인클레이브에 내포하고 원격지 저장소(S)로부터의 비밀값을 전달받아 결합하도록 함으로써 각 사용자(C)에 분산된 여러 $Key\ enclave$가 동일한 마스터 키를 관리할 수 있도록 한다.

6. 결 론

본 연구에서는 신뢰실행환경을 적용하여 암호화된 데이터를 원격지 저장소에 아웃소싱하는 과정의 중복제거 효율성을 개선한 Ren et al.의 연구를 분석하고, 독립된 키 생성 기관을 사용자 응용에 내포함으로써 단일 장애 지점의 위험성 극복 방안을 제시하였다. 독립적으로 동작하는 단일 키 관리 서버의 기능을 각 사용자 응용에 내포함으로써 단일 장애 지점에 따른 공격 내성을 증가시킬 수 있음을 확인하였다. 실험을 통하여 다중 사용자 환경에서도 유연한 확장이 가능하면서도 25Mbps 통신 환경에서 통신 지연을 약 30% 개선할 수 있음을 확인하였다.

SGXDedup[6]에서는 주기적인 키 갱신을 통한 키 노출 위협에 대한 대응이 고려되었으나, 키 갱신에 따라 이전에 아웃소싱된 데이터에 대한 중복확인은 불가능해진다. 제안 기법에서도 이와 같은 제약사항은 존재하지만 향후 분산된 각 사용자의 키 동기화를 위한 관리 기법 개선 등을 통하여 키 갱신에 대해서도 이전 데이터의 중복확인이 가능한 방안 모색 등 지속 가능한 서비스 제공 방안을 연구할 계획이다.

References

[1] M. Armbrust, et al., "A View of Cloud Computing," in *Communications of the ACM*, Vol.53, No.4, pp.50-58, 2010.

[2] Y. Fan, X. Lin, W. Liang, G. Tan, and P. Nanda, "A secure privacy preserving deduplication scheme for cloud computing," in *Future Generation Computer Systems*, Vol.101, pp.127-135, 2019.

[3] Y. Shin, D. Koo, and J. Hur, "A survey of secure data deduplication schemes for cloud storage systems," in *ACM Computing Surveys*, Vol.49, No.74, pp.1-38, 2017.

[4] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *Proceedings of IEEE Trustcom/BigDataSE/ISPA*, pp.57-64, 2015.
[5] M. Miranda, T. Esteves, B. Portela, and J. Paulo, "S2Dedup: SGX-enabled secure deduplication," in *Proceedings of ACM International Conference on Systems and Storage (SYSTOR)*, pp.1-12, 2021.
[6] Y. Ren, J. Li, P. P. C. Lee, and X. Zhang, "Accelerating encrypted deduplication via SGX," in *Proceedings of USENIX Annual Technical Conference (USENIX ATC)*, pp.303-316, 2021.
[7] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simin, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," *Technical Report MSR-TR-2002-30, Microsoft Research*, pp.1-14, 2002.
[8] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pp.296-312, 2013.
[9] S. Keelveedhi, M. Bellare, and T. Ristenpart, "DupLESS: Server-Aided Encryption for Deduplicated Storage," *in Proceedings of USENIX Security Symposium (USENIX Security)*, pp.179-194, 2013.
[10] V. Costan and S. Devadas, "Intel SGX explained," *Cryptology ePrint Archive, pp.1-118*, 2016.
[11] J. Liu, N. Asokan, and B. Pinkas, "Secure deduplication of encrypted data without additional independent servers," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security (ACM CCS)*, pp.874-885, 2015.
[12] Y. Duan, "Distributed key generation for encrypted deduplication: Achieving the strongest privacy," in *Proceedings of ACM Workshop on Cloud Computing Security (CCSW)*, pp.57-68, 2014.
[13] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proceedings of ACM Conference on Computer and Communications Security(CCS)*, pp.491-500, 2011.
[14] Wikipedia, List of countries by Internet connection speeds, [Internet] https://en.wikipedia.org/wiki/List_of_countries_by_Internet_connection_speeds.

구 동 영

https://orcid.org/0000-0003-3283-5494
e-mail : dykoo@hansung.ac.kr
2009년 연세대학교 컴퓨터산업공학(학사)
2012년 한한국과학기술원 전산학(석사)
2016년 한국과학기술원 전산학(박사)
2016년 ~ 2017년 고려대학교 컴퓨터학과 연구교수
2017년 ~ 현 재 한성대학교 전자정보공학과 조교수
관심분야 : Information Security, Applied Cryptography, Network Security, Cloud/Fog/Edge Computing Security