Journal of the Korea Academia-Industrial cooperation Society
Vol. 24, No. 2 pp. 625-634, 2023

https://doi.org/10.5762/KAIS.2023.24.2.625
ISSN 1975-4701 / eISSN 2288-4688

신원 확인 공격에 안전한 클라우드 데이터 중복 제거 기술

홍성우, 이영주, 하재철*
호서대학교 컴퓨터공학부

Secure Cloud Data Deduplication Resistant to Identification Attacks

Seongwoo Hong, Youngju Lee, Jaecheol Ha*
Division of Computer Engineering, Hoseo University

요 약 저장해야 할 데이터의 양이 급격하게 커지고 있는 클라우드 환경에서 데이터의 저장공간을 효율적으로 관리하는 기술이 필요하다. 중복 제거 기술(deduplication)은 클라우드에 데이터가 중복적으로 저장되는 것을 방지함으로써 저장 공간을 효율적으로 관리하는 기술이다. 그 동안 연구된 많은 중복 제거 기술들은 저장하고자 하는 평문에서 암호화 키를 유도하여 업로드 시 기밀성을 유지하는 방법을 사용하고 있으나 전수 조사 공격, 데이터 오염 공격, 소유자 신원 확인 공격 등에 의해 보안상 취약한 특성을 보이고 있다. 특히, K. Park 등은 전수 조사 공격에 안전한 DupLESS를 클라이언트측 중복 제거 기법으로 실현하여 네트워크 트래픽을 효율적으로 사용하고자 하였으나 이 기법은 신원 확인 공격에 취약한 단점이 있다. 본 논문에서는 K. Park 등이 제안한 기법에 더해 신원 확인 공격에 안전한 중복 제거 기술을 제안하고자 한다. 제안하는 기법은 업로드 과정에서 검증용 파라미터를 서버에 저장하고 이를 다운로드 과정에서 사용하여 클라이언트를 검증하므로 신원 확인 공격에 안전하다. 또한, 검증용 파라미터를 이용하여 클라이언트의 요청에 따라서 불필요해진 데이터를 삭제하는 메커니즘을 설계함으로써 클라우드 상에서 안전하고 효율적인 데이터 저장 기술을 제안한다.

Abstract In the cloud environment, where the amount of data to be stored is growing rapidly, a technology for efficiently managing data storage space is needed. Deduplication is a technology that efficiently manages storage space by preventing data from being stored repeatedly in the cloud. The many deduplication schemes studied thus far used a method of deriving an encryption key from the plaintext to be stored and maintaining confidentiality during upload but have some vulnerabilities, such as brute-force attacks, poison attacks, and identification attacks. In particular, K. Park et al. proposed a deduplication scheme to efficiently use network traffic by realizing DupLESS, which is secure against brute-force attacks as a client-side scheme, but this has a disadvantage in that it is vulnerable to the identification attack. In addition to the technique proposed by K. Park et al., this paper proposes a secure deduplication scheme to resist the identification attack in a cloud server system. The proposed method is secure against the identification attack as it stores verification parameters in the cloud server during the upload process and uses them during the download process to verify the client. In addition, a secure and efficient data storage scheme was presented by designing a mechanism to delete unnecessary data using the verification parameters used in the proposed deduplication scheme.

Keywords : Cloud Storage, Client-Side Deduplication, Identification Attack, Data Deletion Process, Cloud Security

본 성과물은 중소벤처기업부에서 지원하는 2022년도 스마트 제조혁신 기술개발(R&D) (No. RS-2022-00140535)의 연구수행으로 인한 결과물임을 밝힙니다.
*Corresponding Author : Jaecheol Ha(Hoseo Univ.)
email: jcha@hoseo.edu
Received December 13, 2022 Revised January 11, 2023
Accepted February 3, 2023 Published February 28, 2023

1. 서론

저장되는 데이터의 양이 크게 증가하고 있는 클라우드 환경하에서 클라우드 서비스 제공자에게는 저장공간을 효율적으로 사용하기 위한 기술이 필수적이다[1]. 대용량 데이터를 효율적으로 관리하기 위해 등장한 중복 제거 기술(deduplication)은 클라우드에 저장되는 데이터 중에서 중복되는 내용을 제거하고 하나의 값만을 저장하여 저장공간을 효율적으로 관리하는 기술이다.

그러나 중복 제거 기술은 클라우드 스토리지의 저장공간을 효과적으로 관리할 수 있는 기술이지만 여러 가지 보안 문제를 발생시킨다. 특히, 민감한 정보를 다루는 경우 클라이언트는 데이터의 기밀성을 제공하기 위해 데이터를 암호화하여 전송하여야 한다. 하지만 전통적인 암호기법을 사용한다면 클라이언트마다 가지는 암호화 키가 다르기 때문에 같은 데이터를 암호화하여도 각기 다른 암호문이 생성된다[2].

이러한 문제를 해결하기 위해서 J. Douceur 등은 같은 평문에 대해서 같은 암호문이 생성되도록 하는 수렴 암호화(Convergent Encryption) 개념을 처음으로 도입하였다[3]. 이후 M. Bellare 등은 클라우드상에서 데이터 중복 제거를 위한 MLE(Message-Locked Encryption) 기법을 제안하였다[4]. 그러나 수렴 암호화의 경우 낮은 엔트로피를 갖는 데이터에 대해서는 전수 조사 공격(brute-force attack)에 의해 안정성을 보장하지 못한다.

그 후 클라우드의 데이터 중복 제거 기법에 대해 전수 조사 공격을 방어할 목적으로 DupLESS라는 별도의 키 서버로부터 암호화 키를 생성하는 기법을 제안하기도 하였다[5]. 이 기법에서 사용자는 키 서버와의 RSA 블라인드 서명(RSA blind signature)을 통해서 안전하게 데이터 기반의 암호화 키 발급을 수행한다. 하지만 DupLESS는 서버측 중복 제거(server-side deduplication) 기법으로서 데이터 송·수신 시 네트워크 대역폭이 커지는 단점이 있다.

한편, K. Park 등은 키 서버를 통해 암호화 키를 생성하면서 데이터의 중복성을 클라이언트가 확인하는 클라이언트측 중복 제거(client-side deduplication) 기술을 제안하였다[6]. 그러나 이 기법은 데이터를 소유하고 있는 공격자가 타인이 동일한 데이터의 소유하고 있는지 여부를 알 수 있는 신원 확인 공격(identification attack)에 취약한 특성을 가지고 있다.

본 논문에서는 클라우드 시스템에서 전수 조사 공격이나 오염 공격(poison attack) 등과 같은 고전적인 공격뿐만 아니라 신원 확인 공격에도 안전한 중복 제거 기술을 제안하고자 한다. 제안된 중복 제거 기법에서는 데이터 저장 과정에서 데이터 검증용 파라미터를 추가로 생성하고 이를 서버에 저장해 둔 후, 다운로드 과정에서 정당한 클라이언트를 식별하게 하도록 구현하였다. 또한, 사용된 검증 파라미터를 이용하여 불필요해진 저장 데이터를 삭제하는 메커니즘을 설계함으로써 클라우드상에서 안전하고 효율적으로 데이터를 저장하는 기술을 제안하였다.

본 논문의 2장에서는 배경 지식인 중복 제거 기술의 분류와 공격 모델에 대해서 설명하고 3장에서는 안전한 중복 제거 기술을 위한 기존에 연구된 기법을 소개한다. 4장에서는 신원 확인 공격의 취약점을 분석하고 5장에서 신원 확인 공격에 대응하는 방법과 불필요한 데이터 삭제 메커니즘을 제안한다. 마지막으로 6장에서 결론을 맺는다.

2. 중복 제거 기술 분류 및 공격

2.1 중복 제거 기술 분류

중복 제거 기술은 중복 제거를 수행하는 주체에 따라서 서버측 중복 제거와 클라이언트측 중복 제거로 분류한다[7]. 먼저 서버측 중복 제거 기술은 클라우드 서버가 클라이언트에게 데이터를 전달받은 이후에 데이터의 중복 제거를 수행한다. 반면에 클라이언트측 중복 제거 기술은 클라이언트가 서버로 데이터를 보내기 이전에 서버로부터 해당하는 데이터가 스토리지 상에 존재하는지 중복 검사를 수행한다. 데이터의 중복이 확인되면 데이터를 전송하지 않고 데이터 소유자 정보만 저장 테이블에 기록한다.

따라서 서버측 중복 제거 기술은 중복 제거를 위해 매번 데이터 업로드가 수행되기 때문에 많은 네트워크 트래픽이 발생한다. 반면에 클라이언트측 중복 제거 기술은 데이터 업로드 이전에 중복을 확인하기 때문에 네트워크 트래픽을 최소화할 수 있다. 하지만 클라이언트에서 데이터의 중복성 검사를 위한 연산이 필요하기 때문에 추가적인 컴퓨팅 자원을 소모하기도 한다.

2.2 공격 모델

2.2.1 전수 조사 공격

클라우드 시스템상에서 전송 데이터를 암호화하기 위해서는 암호용 키가 필요하고 이 암호용 키를 데이터 원본으로부터 유도하기 위해 수렴 암호화 기법을 사용한다. 그런데 데이터에서 유도되는 암호화 키의 경우 평문 데이터의 엔트로피에 의해 암호화의 안정성이 영향을 받는다. 즉, 클라우드에 낮은 엔트로피의 평문 데이터를 저장하고자 할 때 암호화 키의 유추가 가능해진다. 따라서 전수 조사 공격이나 사전 공격(dictionary attack) 등을 통해 엔트로피가 낮은 데이터의 암호용 키를 찾아낼 수 있다[8-10].

2.2.2 오염 공격

중복 제거 기술이 적용된 클라우드에서는 같은 데이터에 대해서 한 사용자의 값만 저장하고 나머지 사용자들의 값은 저장하지 않는다. 따라서 서버에 저장된 데이터가 훼손된 데이터일 경우에 이후의 다른 정당한 클라이언트 데이터를 잃을 수 있다[9].

클라이언트측 중복 제거의 경우에 데이터를 대변할 수 있는 작은 태그 값으로 중복을 확인한다. 악의적인 공격자가 최초 업로드를 수행할 때 태그 생성 과정에서 사용한 데이터가 아닌 훼손된 데이터를 업로드한다면 이후에 해당하는 태그의 데이터를 저장하는 정당한 클라이언트들의 데이터 저장을 방해할 수 있다.

결국, 클라이언트는 다운로드 시에 훼손된 데이터를 다운로드할 수밖에 없다. 이러한 공격을 데이터 오염 공격이라고 한다[11].

2.2.3 신원 확인 공격

신원 확인 공격 혹은 CoF(Confirmation of File) 공격은 클라이언트측 중복 제거 기법에서 발생하는 위협이다. 해당 공격은 중복 확인을 하는 과정에서 데이터를 대신하는 짧은 길이의 태그 값을 사용하는 경우에 발생할 수 있다. 공격자는 자신이 가진 데이터의 태그 값을 생성하고 이 태그 값과 공격 대상자의 식별자를 서버에 보내게 된다.

이 경우 서버는 공격자와 동일한 데이터를 가진 사람을 확인해서 알려주게 되므로 공격 대상자가 자신과 동일 데이터를 가지고 있음을 알게 된다. 즉, 신원 확인 공격은 공격자와 동일한 데이터를 가진 사용자를 구별해 낼 수 있는 개인정보 침해 공격 중 하나이다. 비슷한 공격으로 태그를 이용하여 데이터의 내용을 추측하는 공격인 온라인 추측 공격(online-guessing attack)도 있다.

2.2.4 소유권 검증 및 프라이버시 침해 공격

클라이언트측 중복 제거에서 서버가 클라이언트가 보낸 태그에 해당하는 데이터를 소유하고 있는지 검증하는 소유권 검증을 수행하지 않으면 데이터를 가지지 않은 클라이언트가 태그를 이용하여 데이터의 소유권을 획득할 수 있다.

따라서 클라이언트측 중복 제거 기법의 업로드 과정에서는 데이터의 소유권을 검증하는 과정이 필수적이다. 또한, 클라우드에 암호화되지 않은 데이터를 저장하는 경우에 특정 사용자가 보유한 데이터에 대한 프라이버시가 침해될 수 있다[6].

결국, 클라우드 시스템에서의 중복 제거 기능을 구현하게 되면 상기한 보안 위협들이 발생하게 되며 이 위협들의 취약 요소와 대응 방안을 정리한 것이 Table 1이다.

Table 1. Cyber threats in cloude deduplication system

Threats	Weakness	Countermeasure
Brute-force attack	Low entropy of plain data	Long encryption key
Poison attack	Incorrect tag according to plain data	Tag confirmation according to plain data
Identification attack	Identification of correct client	Client authentication
Data confidentaility	Unverified ownership and raw data storage	Ownership verification and data encryption

3. 데이터 중복 제거 기법 분석

3.1 MLE 기법

암호화가 적용된 클라우드 시스템의 경우 클라우드 서버는 데이터의 평문을 확인할 수는 없다. 그러나 기존의 암호화 기법을 적용하여 클라이언트마다 각기 다른 암호화 키를 사용하여 암호화를 수행하면 서버는 데이터의 중복 확인이 불가능하게 된다[4].

그러므로 MLE에서는 평문에서 유도한 암호화 키를 사용하여 암호화를 수행한다. 따라서, 같은 평문에서 같은 암호화 키가 유도되고 결과적으로 같은 암호문이 생성된다. 또한 암호문에서 태그를 생성하고 해당 태그를 이용하여 데이터의 중복을 확인한다. 결과적으로 서버는 암호화된 데이터를 복호화하지 않고도 데이터의 중복을 확인할 수 있게 된다.

하지만 MLE는 서버측에서 수행하는 중복 제거 기술

로 동일한 데이터가 반복적으로 업로드되어 네트워크 효율성이 떨어지고 서버는 중복 여부 판단을 위해서 모든 데이터를 해시(hash)하여 태그를 생성해야 한다. 또한 평문의 엔트로피가 작을수록 전수조사 공격이나 사전공격에 취약해진다는 문제점을 가지고 있다.

3.2 DupLESS 기법

DupLESS[5]는 MLE 기법이 가지는 문제인 낮은 엔트로피에 대해서 전수조사나 사전공격에 취약하다는 점을 해결하기 위해서 제안된 서버측 중복 제거 기법이다. DupLESS에서는 별도의 키 서버를 구성하고 키 서버에서 암호화에서 사용될 암호화 키를 생성한다.

암호화 키 생성 과정에서는 RSA 블라인드 서명을 통해 키 생성이 이루어지기 때문에 키 서버의 개인 키를 알 수 없는 상태에서는 암호화 키를 찾아내는 오프라인 전수조사 공격에 안전하다. 또한 DupLESS는 온라인 전수조사 공격에 대응하기 위해서 키 서버에 대한 암호화 키 생성 요청 횟수도 제한하고 있다.

3.3 블라인드 서명 기반 클라이언트측 중복 제거 기법

블라인드 서명에 기반한 DupLESS는 서버측 중복 제거 기법인 점을 고려하여 K. Park 등은 이를 클라이언트 측에서 중복성을 제거하는 방법으로 설계하였다[6]. 기존의 DupLESS에서 클라이언트는 키 서버로부터 생성한 암호화 키로 평문을 암호화하여 클라우드 서버로 전송하는 방식을 택하였다. 그러나 K. Park 등의 기법에서는 클라이언트가 암호문 자체를 전송하는 것이 아닌 암호용 키를 통해 태그값 T를 생성하고 이를 전송한다. 그리고 서버는 전송받은 T를 이용해서 데이터의 중복을 확인한다. 클라이언트측 중복 제거 기술은 서버에 중복되는 데이터가 존재하지 않는 경우에만 업로드가 이루어지므로 서버측 중복 제거 기술에 비해 네트워크 운영 측면에서 효율적이다. 또한 클라이언트는 업로드하기 이전에 서버가 저장하고 있는 데이터가 자신이 업로드하는 데이터와 실제로 같은지를 확인한다. 따라서 서버가 저장하고 있는 데이터가 오염 공격에 의해서 훼손된 데이터인지를 확인할 수 있다.

4. 중복 제거 기법 취약점 분석 및 개선

본 장에서는 K. Park 등이 제안한 블라인드 서명 기반 클라이언트측 중복 제거 기법의 보안 취약점을 찾아내고 이를 해결하기 위한 메커니즘을 제안하고자 한다. 다음 Table 2는 본 논문에서 사용하는 표기법과 그에 대한 설명을 나타낸 것이다.

Table 2. Notations used in the proposed scheme

Notations	Description
H, G	Hash functions
H_1	Cryptographic hash function
pk_{KS}	Public key of the key server
sk_{KS}	Private key of the key server
$Enc_k()$	Symmetric encryption by using the secret key k
ID_c	Identifier of the client
sk_c	Secret key of the client
f	Data
sk_f	Secret key of the data f
T	Tag value by hashing sk_f
C	Ciphertext of the data f by using sk_f
C'	Ciphertext stored in the cloud server realted to the tag value T
K_c	Encrypted value of sk_f by using sk_c
h	Poison attack verifier generated by the cloud server
h'	Data ownership value generated by the client
i	Identification attack verifier generated by the client
s, k, r	Random value, $s, k, r \in \{0,1\}^*$
V_c	Identity verification value generated by hashing ID_c and sk_c

4.1 클라이언트 중복 제거 기술

K. Park 등이 제안한 중복 제거 기술은 클라이언트가 인증 서버로부터 비밀 키를 받는 단계, 클라이언트가 데이터를 암호화하고 태그 값을 만들어 업로드하는 단계 그리고 서버에 저장된 데이터를 다운로드 받는 단계로 구성된다. 단, 여기서 주목할 점은 서버에 저장되었던 데이터를 삭제하는 메커니즘은 아직 제안된 적이 없다는 것이다.

먼저 비밀 키 생성을 담당하는 키 서버에는 RSA 블라인드 서명을 위한 공개 키 $pk=(e,\ N)$와 비밀 키 $sk=(N,\ d)$가 있다고 가정한다.

① 클라이언트는 데이터 f의 해시값 $h=H(f)$을 계산하고 $x=r^e\ \cdot\ h\ \bmod N$을 계산하여 키 서버에 전송한다. 여기서 r은 N보다 작은 난수이다.

② 키 서버는 비밀 키를 이용하여 $y=x^{\ d}\ \bmod\ N$을 계산하여 클라이언트에게 전송한다.

③ 클라이언트는 $z=y\ \cdot\ r^{\ -1}\ \bmod N$을 계산하고 데이터 f의 비밀 키 $sk_f=\ G(z)$를 생성한다.

이후에는 데이터 업로드 과정을 거치게 되는데 Fig. 1은 데이터의 태그 값이 존재하지 않아 처음 업로드를 하는 경우를 도시한 것이고, Fig. 2는 이미 데이터가 저장된 경우를 나타낸 것이다.

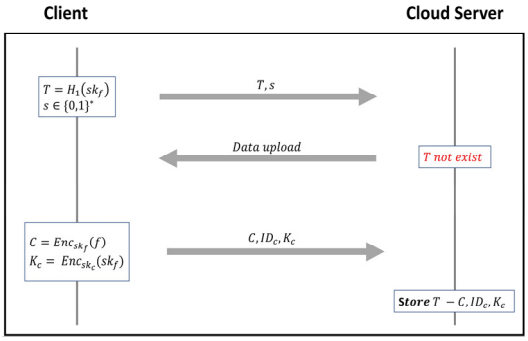


Fig. 1. Data upload process when the data was not stored

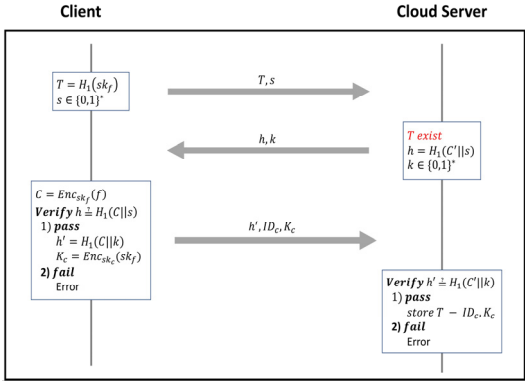


Fig. 2. Data upload process when the data was already stored

4.2 중복 제거 기술의 취약점

다음 Fig. 3은 K. Park 등이 제안한 중복 제거 기법에서 사용하는 데이터 다운로드 단계를 요약하여 나타낸 것이다. 기본적으로 클라우드 서버에는 태그-암호문 $T-C$, 클라이언트의 ID_c, 클라이언트측에서 sk_f를 암호화한 키 K_c를 저장하고 있다.

① 클라이언트가 데이터를 다운로드 받기 위해서는 먼저 암호화 키를 해시한 태그 값 T와 클라이언트 식별자인 ID_c를 클라우드 서버에 전송한다.

② 클라우드 서버는 데이터의 소유권을 검사한 뒤, 검증이 완료되면 암호문 C와 암호화 키 K_c를 클라이언트에게 전송한다.

③ 클라이언트는 서버로부터 받은 암호화 키 K_c를 자신의 비밀 키 sk_c로 복호화하여 sk_f를 복원하고, 이를 이용하여 다시 암호문 C를 복호화하여 최종적으로 데이터 f를 얻게 된다.

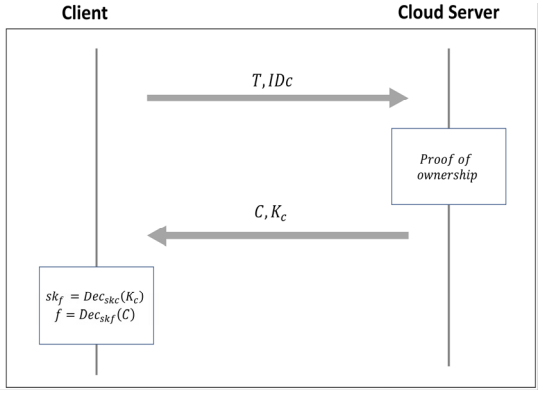


Fig. 3. Data download process

K. Park 등의 기법에서는 비밀 키 sk_c가 없으면암호화 키 K_c를 복구할 수 없고 정상적으로 데이터를 생성할 수 없다. 또한 T를 생성할 때 평문 데이터 f나 암호문 C를 사용하지 않고 키 서버를 통해 생성한 암호화 키인 sk_f를 사용한다. 이렇게 함으로써 태그를 이용하여 데이터의 내용을 직접 추측하지 못하도록 하여 온라인 추측 공격을 방어하였다.

그러나 이 기법에서는 T와 ID_c 두 가지의 값만 가지고 데이터 다운로드 단계를 수행하기 때문에 사용자 신원 확인 공격에 취약하다. 즉, 공격자가 데이터를 소유하

고 있고 T값을 생성할 수 있는 경우, 다운로드 단계에서 공격자는 T와 특정 클라이언트의 식별자를 이용하여 해당 클라이언트가 데이터를 소유하고 있는지 확인할 수 있게 된다. 즉, 공격자는 공격 대상 x의 식별자 값 ID_x를 T와 같이 전송함으로써 데이터 다운로드를 요청한다. 이 경우 서버는 소유권 검사를 한 뒤 소유권이 있으면 정상적인 다운로드 프로세스를 진행하지만, 소유권이 없는 경우에는 거절을 하게 된다. 따라서 공격자는 클라우드 서버의 응답에 따라 공격자가 보유한 데이터를 다른 사용자가 저장하였는지를 식별할 수 있어 민감한 개인정보가 노출되게 된다.

4.3 개선 방법

본 논문에서는 K. Park 등이 제안한 블라인드 서명 기반 클라이언트측 중복 제거 기법에서 신원확인 공격에 대응하기 위한 메커니즘을 제안하고자 한다. 제안 기법에서는 클라우드에 요청을 하는 사람이 정당한 클라이언트 ID_c라는 것을 확인하기 위해 추가적인 검증 파라미터 V_c를 사용하고자 한다. 이 검증 파라미터는 ID_c와 클라이언트의 비밀 정보를 조합하여 구성한 후 서버에 저장해 두고 데이터 다운로드 시 검증을 위해 사용한다.

또한, 기존의 중복 제거 기법에서는 별도의 권한 수정의 알고리즘이 존재하지 않기 때문에 클라우드 스토리지에서 데이터를 업로드한 이후에 데이터를 지우고자 하는 클라이언트가 존재할 경우, 여러 문제점이 발생할 수 있다. 즉, 데이터 다운로드 단계와 마찬가지로 태그와 클라이언트 식별자만 가지고 데이터의 삭제가 진행될 경우, 악의적인 공격자에 의해서 다운로드 시 도청한 T와 ID_c를 그대로 전송하여 삭제에 활용할 수 있다. 본 논문에서는 이러한 문제점들을 해결하기 위해서 클라이언트 식별자 이외의 클라이언트를 인증할 수 있는 파라미터를 생성하고 다운로드나 삭제 과정에서 추가적인 인증 과정을 거치는 중복 제거 기법을 제안한다.

5. 중복 제거 기법 제안 및 안전성 분석

제안하는 중복 제거 기법이 적용된 클라우드 시스템의 전체적인 구성을 나타낸 것이 Fig. 4이다. 제안 기법을 수행하는 객체는 클라이언트, 클라우드 서버 그리고 키 서버이다. 안전한 중복 제거 클라우드 시스템을 운영하기 위해서는 클라이언트는 먼저 데이터에 대한 비밀 키를 인증 서버로부터 받고 데이터를 업로드하는 과정을 거친다. 그리고 필요 시 데이터를 다운로드 받거나 불필요한 데이터에 대한 삭제 기능을 수행할 수 있다.

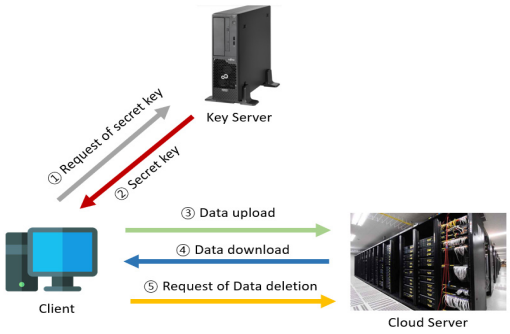


Fig. 4. Overall architecture of proposed scheme

상기한 신원 확인 공격에 대응하기 위해서는 데이터 다운로드 과정에서 별도의 식별자 검증 과정이 추가되어야 한다. 이를 위해서 데이터를 업로드하는 과정에서 검증 파라미터를 생성하고 다운로드 단계나 데이터 삭제 단계에 활용하는 방법을 제안한다.

5.1 업로드 단계

제안하는 신원 확인 공격에 안전한 중복 제거 기법에서는 신원 확인 과정에서 사용될 파라미터 V_c를 클라이언트가 업로드 단계에서 생성하여 서버로 전송한다. 이 검증용 파라미터는 클라이언트의 ID_c와 비밀 키 sk_c를 해시한 것으로 식별자의 비밀 키를 노출하지 않고 신원 검증을 수행하는 데 사용된다.

5.1.1 데이터가 서버에 저장되어 있지 않은 경우

만약 클라이언트가 저장하고자 하는 데이터가 서버에 없을 경우에는 다음과 같은 메커니즘으로 데이터가 업로드 된다.

① 클라이언트는 키 서버를 통해서 생성한 비밀 키 sk_f를 이용하여 태그 $T = H_1(sk_f)$를 생성한다. 그리고 난수 $s \in \{0,1\}^*$를 생성한 다음 클라우드 서버에게 T, s를 전송하여 업로드를 요청한다.

② 클라우드 서버는 데이터베이스를 검색하여 클라이언트가 보낸 T와 일치하는 값이 없음을 확인하고 데이터 업로드 요청을 승인한다.

③ 클라이언트는 데이터 f를 비밀 키 sk_f로 암호화하여

C를 생성한다. 추가적으로 $K_c = Enc_{sk_c}(sk_f)$와 $V_c = H_1(ID_c \| sk_c)$를 생성하여 ID_c와 함께 서버로 전송한다.

④ 서버는 클라이언트가 전송한 ID_c, K_c 그리고 V_c를 T, C와 함께 저장한다.

5.1.2 데이터가 서버에 이미 저장된 경우

다음 Fig. 5는 데이터 업로드 단계에서 데이터가 이미 서버에 존재하는 경우의 제안 메커니즘을 나타낸 것이다.

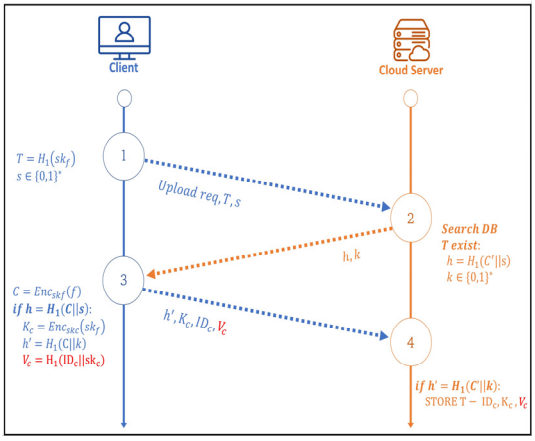


Fig. 5. Proposed data upload process when the data was already stored.

① 클라이언트는 비밀 키 sk_f를 이용하여 태그 $T = H_1(sk_f)$를 생성한다. 그리고 난수 $s \in \{0,1\}^*$를 생성한 후 서버에게 T, s를 전송하여 업로드를 요청한다.

② 서버는 데이터베이스를 검색하여 클라이언트가 보낸 T와 일치하는 값이 이미 존재한다면 $h = H_1(C'\|s)$를 계산하고 난수 $k \in \{0,1\}^*$를 생성하여 클라이언트에게 보낸다. 여기서 C'은 T 값에 해당하는 데이터의 암호문을 의미한다.

③ h와 k를 전송받은 클라이언트는 데이터 f를 비밀 키 sk_f로 암호화하여 C를 생성한다. 이후 생성된 C를 이용하여 서버가 보낸 h가 $h = H_1(C\|s)$를 만족하는지 확인하여 서버에 암호문이 저장되어 있는지 검증한다. 검증을 통과한 뒤 $K_c = Enc_{sk_c}(sk_f)$와 $h' = H_1(C\|k)$그리고 $V_c = H_1(ID_c\|sk_c)$를 생성하여 ID_c와 함께 서버로 전송한다.

④ 마지막으로 서버는 클라이언트가 정말로 데이터를 소유하고 있는지 확인하기 위해 $h' = H_1(C'\|k)$이 성립하는지 계산한다. 소유권이 확인되었다면 클라이언트가 전송한 ID_c, K_c 그리고 V_c를 T와 함께 저장한다.

5.2 다운로드 단계

기존의 K. Park 등의 기법에서는 다운로드 요청 시에 서버는 소유권 검증만 수행한 뒤에 암호화된 데이터와 암호화된 키를 바로 클라이언트에게 전송하였다. 이와 같은 이유로 신원 확인 공격에 취약한 특성을 보이고 있다. 본 연구에서는 이전의 업로드 단계에서 분배한 검증용 파라미터 V_c를 사용하여 정당한 클라이언트를 검증하는 방식으로 메커니즘을 구성하고자 한다. 다음 Fig. 6은 본 논문에서 제안하는 다운로드 절차이다.

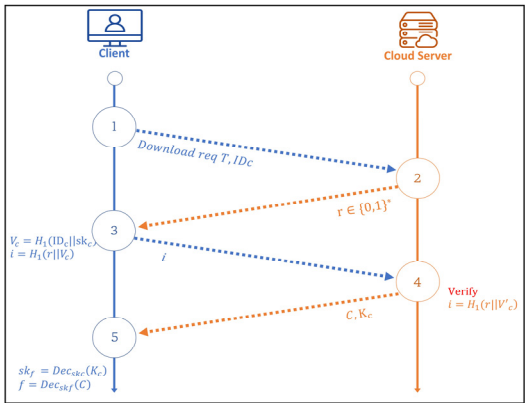


Fig. 6. Proposed data download process

① 클라이언트는 클라우드 서버로 다운로드 요청과 함께 태그 T와 식별자 ID_c를 전송한다.

② T와 ID_c를 전송받은 서버는 검증을 위한 난수 r을 생성하여 클라이언트에게 전송한다.

③ r을 수신한 클라이언트는 $V_c = H_1(ID_c\|sk_c)$를 계산하고 $i = H_1(r\|V_c)$를 생성하여 서버로 전송한다.

④ 서버는 T와 함께 저장되어 있는 ID_c에 해당하는 V_c'을 이용하여 $i = H_1(r\|V_c')$를 확인한다. 검증을 통과했다면 소유권 검증을 한 후 암호문 C와 K_c를 클라이언트에게 전송한다.

⑤ 클라이언트는 전송받은 K_c를 이용하여 데이터를 복호화하는데 사용할 키인 sk_f를 생성하고, sk_f를 이용하여 C를 복호화하여 최종적으로 데이터 f를 생성한다.

5.3 삭제 단계

중복 제거 기법을 사용하는 클라우드 환경에서 데이터 삭제는 두 가지의 경우를 가진다. 첫 번째로 데이터를 삭제하려고 하는 클라이언트가 데이터의 유일한 소유자일 경우에는 클라이언트의 권한과 함께 데이터 또한 삭제되어야 한다. 두 번째로 해당 데이터를 가진 다른 사용자가 존재할 경우에는 클라우드 서버에 태그 T와 함께 저장되어 있는 ID_c, V_c, K_c만 지워야 한다.

중복 제거 기법을 사용하는 클라우드 환경에서도 데이터 공간의 효율적 사용을 위해 데이터의 불필요한 삭제가 필요하지만, 지금까지의 연구에서는 데이터 삭제 메커니즘을 제시하지는 않고 있다. 만약 클라이언트가 ID_c와 T만 가지고 데이터를 삭제를 요청할 수 있다면 데이터의 태그를 생성할 수 있는 공격자는 식별자를 위조하여 다른 사용자의 데이터를 지울 수 있다. 따라서 데이터의 삭제 단계에서도 삭제를 요청한 클라이언트가 정당한 사람인지를 확인한 후 데이터를 삭제하는 메커니즘이 필요하다. 다음 Fig. 7은 클라우드의 데이터를 삭제하는 과정을 나타내는 그림이다.

① 데이터를 삭제하고자 하는 클라이언트는 데이터 삭제 요청과 함께 태그 값 T와 식별자 ID_c를 서버로 전송한다.

② 삭제 요청을 받은 클라우드 서버는 랜덤한 난수 r을 생성하여 클라이언트에게 전송한다.

③ 서버로부터 난수 r을 받은 클라이언트는 $V_c = H_1(ID_c \| sk_c)$를 계산하고 $i = H_1(r \| V_c)$를 생성하여 서버로 전송한다.

④ 서버는 T와 함께 저장되어 있는 ID_c에 해당하는 V_c'을 이용하여 $i = H_1(r \| V_c')$를 확인하여 검증을 수행한다.

⑤ 검증을 통과하였다면 해당 데이터를 가진 다른 사용자가 존재하는지 판단하고, 만약 존재한다면 T와 함께 저장되어 있는 V_c', K_c, ID_c만 삭제한다. 그러나 데이터를 가진 다른 사용자가 존재하지 않는다면 V_c', K_c, ID_c, C, T를 모두 삭제한다.

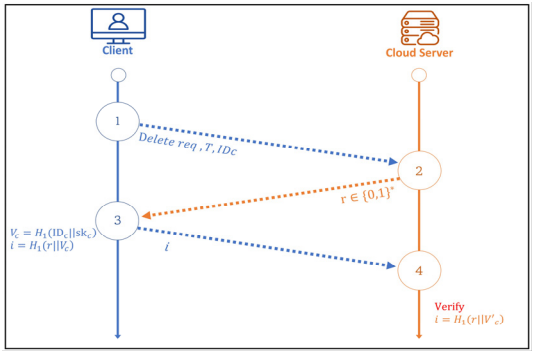


Fig. 7. Data deletion process

5.4 안전성 및 효율성 비교

본 논문에서는 클라우드 환경에서 공격자가 다른 사용자의 데이터 보유 여부를 확인할 수 있는 신원 확인 공격에 대응하기 위한 방법과 서버 저장공간의 효율성 증대를 위해 불필요한 데이터를 안전하게 삭제하는 방법을 제안하였다. 다음 Table 3은 기존의 클라우드 중복 제거 방식을 안전성 측면에서 비교한 것이다.

표에서 보는 바와 같이 MLE나 DupLESS는 서버측 중복 제거 기술로서 실제 암호화된 데이터가 먼저 서버로 전송되는 구조를 가지고 있어 보안 측면에는 비교적 안전하지만 네트워크 트래픽에 대한 효율성이 낮다는 단점이 있다.

Table 3. Comparison with other deduplication schemes

Schemes	S/C	K/T	[C]	[B]	[P]	[I]	[D]
MLE[4]	S	N/A	O	X	O	O	N/A
DupLESS[5]	S	K	O	O	O	O	N/A
T. Youn[12]	C	T	O	O	O	O	N/A
K. Park[6]	C	K	O	O	O	X	N/A
Proposed	C	K	O	O	O	O	O
	S/C : Server/Client-side deduplication K/T : Key Server/Time Server [C] : Data confidentiality [B] : Brute-force attack [P] : Pioson attack [I] : Identification attack [D] : Secure data delition						

T. Youn 등이 제안한 클라이언트측 중복 제거 기술[12]은 특별히 신원 인증 공격에 대응하기 위해 시간-잠금 중복 제거 기술(time-locked deduplication)을 제안하였으나 이 방식은 별도의 타임 서버(time server)를 필요로 하고 중복 제거 프로토콜이 복잡하다는 단점을

갖는다. 이에 반해 K. Park 등의 방식은 성능과 효율성 면에서 우수하지만 신원 확인 공격에 취약한 특성이 있었다. 본 논문에서는 K. Park 등의 취약성을 개선하여 신원 확인 공격을 방어하면서 불필요한 데이터를 안전하게 삭제할 수 있는 기법을 제안하였다.

6. 결론

클라우드 환경에서 클라이언트측 중복 제거 기법은 서버의 데이터 저장 공간의 효율화를 위해 사용되는 기술이지만 다수의 사용자가 사용하게 되는 점을 고려하면 구현 시 많은 보안 사항을 검토하여야 한다. 특히, 기존의 중복 제거 방법 중에는 태그와 식별자만 이용하여 데이터 다운로드를 요청하게 되면, 특정 사용자의 데이터 소유 여부에 관한 개인정보가 노출되는 신원 확인 공격에 취약한 경우가 있다.

본 논문에서는 별도의 키 서버를 이용한 클라이언트측 중복 제거 기법을 분석하여 신원 확인 공격에 강인한 데이터 업로드 기법을 제안하였다. 제안 방식의 원리는 사용자의 신원을 동적으로 확인할 수 있는 파라미터를 사전에 클라우드 서버와 공유하고 이를 이용하여 데이터 업로드 시 정당한 사용자를 식별하도록 한 것이다. 또한 클라이언트 요청에 따라서 불필요하게 저장된 데이터를 안전하게 삭제하는 메커니즘을 설계하여 데이터 저장공간을 효율적으로 사용할 수 있도록 구현하였다. 향후에는 제안 클라우드 중복 제거 기술을 적용 시 필수적인 키 서버의 효율적 운영과 클라이언트와 서버의 통신 프로토콜의 간소화를 위한 연구가 필요하다.

References

[1] M. Dutch., and B. William, "A study of practical deduplication", *ACM Transactions on Storage (ToS)*, Vol. 7, No. 4, pp.1-20, 2012.
DOI: https://doi.org/10.1145/2078861.2078864

[2] M. Storer, K. Greenan, D. Long, and E. Miller, "Secure data deduplication", *Proceedings of the 4th ACM international workshop on Storage security and survivability*, pp. 1-10, 2008.
DOI: https://doi.org/10.1145/1456469.1456471

[3] J. Douceur, A. Adya, W. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system", *Proc. of the 22nd International Conference on Distributed Computing Systems*, pp. 617-624, 2002.
DOI: https://doi.org/10.1109/ICDCS.2002.1022312

[4] M. Bellare., S. Keelveedhi., and T. Ristenpart, "Message-locked encryption and secure deduplication", *EUROCRYPT 2013, LNCS 7881*, pp. 296-312, 2013.
DOI: https://doi.org/10.1007/978-3-642-38348-9_18

[5] M. Bellare, S. Keelveedhi, and T. Ristenpart, "DupLESS: Server-Aided Encryption for Deduplicated Storage", *Proc. of the 22nd USENIX conference on Security*, pp. 179-194, 2013.
DOI: https://dl.acm.org/doi/10.5555/2534766.2534782

[6] K. Park, J. Eom, J. Park, D. Lee, "Secure and Efficient Client-side Deduplication for Cloud Storage", *Journal of The Korea Institute of Information Security & Cryptology*, pp 83-94, 2015.
DOI: https://doi.org/10.13089/JKIISC.2015.25.1.83

[7] D. Koo, Y. Shin, J. Yun, and J. Hur, "A hybrid deduplication for secure and efficient data outsourcing in fog computing", *IEEE International Conference on Cloud Computing Technology and Science*, pp. 285-293, 2016.
DOI: https://doi.org/10.1109/cloudcom.2016.0054

[8] Y. Lu, Y. Qi, S. Qi, F. Zhang, W. Wei, X. Yang, J. Zhang, and X. Dong, "Secure Deduplication-Based Storage Systems With Resistance to Side-Channel Attacks via Fog Computing", I*EEE Sensors Journal, Vol. 22, Issue 18,* pp. 17529-17541, 2022.
DOI: https://doi.org/10.1109/JSEN.2021.3052782

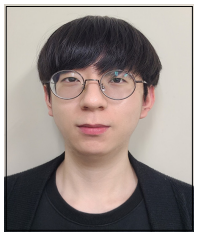
[9] J. Xu, E. Chang, and J. Zhou, "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage", *ACM Symposium on Information, Computer and Communications Security*, pp. 195-206, 2013.
DOI: https://doi.org/10.1145/2484313.2484340

[10] K. Kim, T Youn, N. Cho, K. Chang, "Client-Side Deduplication to Enhance Security and Reduce Communication Costs", *ETRI Journal Vol. 39, Issue 1*, pp. 116-123, 2017.
DOI: https://doi.org/10.4218/etrij.17.0116.0039

[11] F. Rashid, A. Miri, and I. Woungang, "Secure enterprise data deduplication in the cloud", *IEEE International Conference on Cloud Computing,* pp. 367-374, 2013.
DOI: https://doi.org/10.1109/cloud.2013.123

[12] T. Youn, N. Jho, K. Kim, K. Chang and K. Park, "Locked Deduplication of Encrypted Data to Counter Identification Attacks in Cloud Storage Platforms", *MDPI Energies*, Vol. 13, No. 11, pp. 2742, 2020.
DOI: https://doi.org/10.3390/en13112742

홍 성 우(Seongwoo Hong) [준회원]



• 2017년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 학부과정

〈관심분야〉
부채널 공격, 암호학, 정보보호, 인공지능 보안

이 영 주(Youngju Lee) [준회원]

• 2018년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 학부과정

〈관심분야〉
인공지능 보안, 부채널 공격, 양자 내성 암호

하 재 철(Jaecheol Ha) [종신회원]

• 1989년 2월 : 경북대학교 전자공학과 (학사)
• 1993년 8월 : 경북대학교 전자공학과 (석사)
• 1998년 2월 : 경북대학교 전자공학과 (박사)
• 1998년 3월 ~ 2007년 2월 : 나사렛대학교 정보통신학과 교수
• 2007년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 교수
• 2013년 1월 ~ 현재 : 한국정보보호학회 수석부회장
• 2009년 1월 ~ 현재 : 한국산학기술학회 이사

〈관심분야〉
암호학, 네트워크 보안, 부채널 공격, 머신러닝