Journal of Korea Multimedia Society Vol. 21, No. 8, August 2018(pp. 909-916)
https://doi.org/10.9717/kmms.2018.21.8.909

클라우드 스토리지에서 안전한 중복 제거 기법들에 대한 보안 취약점 분석

박지선†, 신상욱††

Analysis of Security Weakness on Secure Deduplication Schemes in Cloud Storage

Ji Sun Park†, Sang Uk Shin††

ABSTRACT

Cloud storage services have many advantages. As a result, the amount of data stored in the storage of the cloud service provider is increasing rapidly. This increase in demand forces cloud storage providers to apply deduplication technology for efficient use of storages. However, deduplication technology has inherent security and privacy concerns. Several schemes have been proposed to solve these problems, but there are still some vulnerabilities to well-known attacks on deduplication techniques. In this paper, we examine some of the existing schemes and analyze their security weaknesses.

Key words: Secure Deduplication, Brute-Force Attack, Convergent Encryption, Cloud Computing

1. 서 론

스토리지 아웃소싱(Storage Outsourcing)은 낮은 비용과 높은 접근성, 그리고 쉬운 공유 방법을 제공한다. 다양한 스토리지 아웃소싱 기법 중 클라우드 환경에서의 스토리지는 최근 많은 주목을 받고 있고, 네트워크의 다양한 어플리케이션 활용 면에서 필수적인 요소가 되고 있다[1,2].

2020년까지 데이터 센터의 스토리지가 382 EB에서 1.8 ZB까지 5배 증가할 것으로 전망한다[3]. 이와 같은 저장되는 데이터의 양과 스토리지 요구량의 빠른 증가는 클라우드 스토리지에 많은 부담을 주게 된다. 따라서 클라우드 스토리지 서비스 제공자는 폭발적으로 증가하는 스토리지의 요구량 증가를 감당하기 위해서는 스토리지의 효율적인 사용이 요구된다. 이에 대처하기 위한 방안으로 클라우드 스토리지의 확장성을 높이기 위한 분산 스토리지 시스템 구조에 대한 연구도 이루어졌다[4]. 일반적으로 클라우드 스토리지의 효율적 사용을 위해 서비스 제공자에 의해 활용되는 가장 대표적인 기술 중의 하나로 데이터 중복 제거(Data Deduplication) 기술이 있다. 데이터 중복 제거의 목적은 중복된 파일들을 하나의 사본만 저장/보유하고 데이터의 중복된 사본들을 이 사본에 의한 참조(Reference)로 대체함으로써 더 적은 공간으로 더 많은 데이터를 저장하는 기술이다[2].

중복 제거 기법은 중복 제거를 수행하는 주체에 따라 서버 측과 클라이언트 측으로 분류된다. 이 두 기법은 데이터 업로드 시 다른 중복제거 시점을 가진

※ Corresponding Author : Sang Uk Shin, Address: (48513) 45, Yongso-ro, Nam-Gu. Busan, Korea, TEL : +82-51-629-6249, FAX : +82-51-629-6230, E-mail : shinsu@pknu.ac.kr
Receipt date : Jul. 9, 2018, Approval date : Jul. 23, 2018

† Interdisciplinary Program of Information Security, Graduate School, Pukyong National University (E-mail : 201211812@pukyong.ac.kr)
†† Dept. of IT Convergence and Application Eng., Pukyong National University
※ This work was supported by a Research Grant of Pukyong National University(2017 year)

다. 서버 측 중복 제거는 클라이언트의 모든 데이터를 업로드한 후, 서버가 저장된 데이터에 대한 중복을 제거한다. 반면에, 클라이언트 측 중복 제거는 서버에 데이터를 업로드하기 전에 메타 데이터(예, 데이터의 해시 값 등)를 통해서 중복된 데이터가 스토리지에 있는지 확인하여 중복된 데이터는 업로드하지 않는다. 따라서 클라이언트 측 중복 제거는 스토리지의 효율적 사용과 함께 통신 대역폭의 이점도 제공한다. 이러한 장점과 동시에, 데이터 중복 제거 기술에는 보안 취약점과 프라이버시 위협이 함께 존재한다. 예를 들어, 클라이언트 측 중복제거 기법에서는 데이터의 메타 데이터만을 가진 사용자가 파일을 가진 것처럼 행동하여 파일을 다운로드할 수 있다는 보안 위협이 발견되었다[10].

이러한 보안위협과 더불어 암호 데이터에 대한 안전한 중복 제거 기법들은 온라인/오프라인 전수조사(Online/Offline brute-force) 공격, CDN(Content distribution network) 공격, Targeted Collision 공격 등과 같은 보편적인 유형의 공격들이 적용 가능하므로[8], 이에 대한 고려가 필요하다.

최근 제안된 Yan-Ding-Zhu의 기법[5], Hur-Koo-Shin-Kang의 기법[6], Yan-Ding-Yu-Zhu-Deng의 기법[7], XDeup[8]은 사용자 프라이버시 보호를 위해 암호 데이터에 대한 안전한 중복 제거 지원을 목적으로 하는 기법들이지만, 위에서 언급한 중복 제거 기법에 대한 기본적인 공격에 취약한 부분이 존재한다. 본 논문에서는 이러한 기법들의 보안 취약점을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서 안전한 중복 제거 기법들에 대한 보편적인 공격들과 기법들을 간략히 소개한다. 3장에서는 기존 기법들의 보안 취약점 및 고려사항을 제시하고, 마지막 4장은 결론이다.

2. 관련 연구

2.1 중복 제거 기법에 대한 일반적인 공격 유형

중복 제거 기법에 관한 가장 일반적인 공격은 전수 조사 공격이다. 이 공격은 네트워크 트래픽의 관찰을 통해 수행 가능하다. 클라이언트 측 중복 제거에서 네트워크 트래픽 모니터링은 주어진 데이터의 조각(즉, 파일)에 중복 제거 적용 여부를 공격자가 판단할 수 있게 한다. 따라서 공격자는 다른 사용자가 클라우드 스토리지 시스템에 저장한 데이터 항목을 식별하거나 그 내용을 알 수 있다. 예를 들면, 파일 F가 스토리지 시스템에 저장되어 있는지를 판단하기 위해, 공격자는 단순히 파일 F의 업로드를 시도한 후, 자신의 장치에서 클라우드 스토리지 시스템으로의 네트워크 트래픽을 관찰한다. 만약 파일 F가 업로드된다면, 이것은 스토리지 시스템에 해당 파일이 존재하지 않는다는 것을 의미한다. 공격자는 업로드를 조기에 취소할 수 있기 때문에 완전한 파일 F를 업로드하지 않을 수 있다. 따라서 공격자는 나중에 이 공격을 반복하는 것이 가능하다. 반면에 파일 F가 업로드 되지 않는다면, 공격자는 스토리지 시스템에 해당 파일 F가 이미 존재하는 것으로 판단할 수 있다. 결국 이것은 스토리지 시스템을 파일 내용에 대한 전수 조사 공격(brute-force attack)에 취약하게 만든다. 전수 조사 공격은 온라인 공격과 오프라인 공격으로 분류할 수 있다.

온라인 전수 조사 공격은 클라이언트 측에서 중복 제거를 지원하는 클라우드 스토리지 시스템에 적용 가능한 공격이다. 이 스토리지 시스템에서는 몇몇 조건이 갖춰지면 “누군가가 이 파일을 업로드했는가?”라는 질문에 대답하는 오라클(oracle)로 이용될 수 있기 때문이다[9]. 공격자는 파일을 업로드하여 중복 제거가 발생하는지를 관찰하여 이 공격을 수행할 수 있다. 낮은 엔트로피를 가지는 예측 가능한(predictable) 파일에 대해, 공격자는 모든 가능한 파일들을 구성할 수 있고, 이 파일들을 업로드 시도하여 중복 제거가 발생하는지를 관찰할 수 있다.

Harnik 등[9]은 이러한 온라인 전수 조사 공격을 다루기 위해 랜덤화된 임계치(Randomized Threshold) 접근법을 제안했다. 이는 각 파일 F에 대해 서버는 랜덤한 임계치 $t_F(t_F \geq 2)$와 카운터 c_F를 유지한다. 여기서 카운터 c_F는 이전에 파일 F를 업로드한 클라이언트의 수를 나타낸다. 클라이언트 측 중복 제거는 $c_F \geq t_F$일 때만 발생한다. 그렇지 않으면, 클라이언트는 파일 F를 서버에 업로드하고 서버가 서버 측 중복 제거를 수행한다.

중복 제거를 지원하는 동시에 기밀성 제공을 위해 수렴 암호화(Convergent Encryption)기법이 중복 제거 기법에 폭넓게 활용되고 있다. 하지만, 수렴 암호화 기법은 오프라인 전수 조사 공격[10]에 취약하

다. 그 이유는 주어진 암호문 C의 평문 공간이 충분히 크지 않기 때문이다. 즉, 메시지는 보통의 경우에 예측 가능하다(predictable). 따라서 공격자는 오프라인 단계로 모든 가능한 평문들을 암호화하여 대응하는 평문 정보를 획득할 수 있다(암호화 기법 E가 결정적이고 수렴키 K는 데이터 파일 C에만 의존한다는 조건하에 공격이 가능하다).

이 문제를 해결하기 위해 Bellare 등[10]은 DupLESS라 불리는 기법을 제시했다. DupLESS에서 사용자는 키 서버 KS(Key Server)의 도움으로 수렴키를 생성한다. 여기서 수렴키는 사용자와 KS 간의 블라인드 서명 프로토콜을 수행함으로써 KS의 개인키가 적용된다.

중복 제거 기법이 적용된 스토리지 시스템에 대해 가능한 또 다른 공격으로는 CDN(Content Delivery Network) 공격이 있다[11]. 공격자는 클라우드 스토리지 시스템을 CDN으로 활용 가능하게 한다. 밥(Bob)이 파일 F를 앨리스(Alice)와 공유하기 원한다고 가정하자. 밥은 파일 F를 클라우드 스토리지 시스템에 업로드하고, 앨리스에게 파일 F의 식별자를 전달한다. 앨리스가 이를 수신하면, 파일 F의 파일 식별자를 동일한 클라우드 스토리지 시스템에 전송하여 파일 F의 업로드를 시도한다. 스토리지 시스템은 이 식별자가 이미 존재하는 것을 알고, 중복 제거를 통해 앨리스가 파일 F의 소유주임을 의미하는 참조를 스토리지 시스템에 저장할 것이다. 그 후, 앨리스가 파일 F를 다운로드하기 원하면, 클라우드 스토리지 시스템에 요청하여 다운로드받을 수 있다. 이러한 취약점을 방지하기 위해서는 클라이언트 측 중복 제거 기법에서는 항상 소유권 검증(Proofs of Ownership, PoW)[10] 절차를 수행해야 한다.

게다가 악의적인 클라이언트가 주장하는 식별자와 대응하지 않는 데이터(즉, 파일) 조각을 업로드하는 Targeted Collision 공격의 위협이 존재한다[12]. 예를 들면, 밥이 앨리스를 속이길 원한다고 가정한다. 클라우드 스토리지 시스템에 의한 제어가 없다면, 밥은 파일 F_1의 식별자를 가진 파일 F_2을 업로드할 수 있다. 그 후에, 앨리스가 파일 F_2를 파일 F_1의 식별자를 가지고 업로드 하려고 하면 스토리지 시스템은 파일 F_2의 식별자가 존재하는 것을 검출할 것이고 따라서 파일 F_2를 저장하지 않을 것이다. 다만, 앨리스 또한 파일 F_1(시스템에서 파일 F_2의 식별자에 대응하는 파일임)을 소유한다는 참조만 저장할 것이다. 나중에 앨리스가 파일 F_2의 식별자로 파일 F_2를 다운로드 요청하면, 스토리지 시스템은 앨리스에게 저장되어 있는 파일 F_1을 전송할 것이다. 이를 방지하기 위해서는 소유권 검증(PoW) 절차가 필요하며, 또한 클라이언트는 중복 제거 절차를 수행할 때, 클라우드 스토리지 시스템에 저장된 파일이 식별자에 대응되는 파일이 맞는지 확인하는 절차가 요구된다.

2.2 기존 기법들

2.2.1 Yan-Ding-Zhu의 기법[5]

Yan 등은 중복 제거를 지원하는 암호데이터 저장소를 관리하기 위한 PRE (Proxy-Re-Encryption) 기반 기법을 제안하였다. 제안 시스템은 사용자, CSP (Cloud Service Provider), AP(Authorized Party)로 구성되며, AP는 완전 신뢰 개체이며 CSP와 공모하지 않는다고 가정한다. 또한 CSP는 honest-but-curious 신뢰 모델로 가정한다. 사용자는 중복 확인을 위한 식별자 id로 데이터 M의 해시 값을 사용한다. 해시 값과 이에 대한 서명을 CSP에게 전달하면, CSP는 해시 값을 인덱스로 사용하여 중복 여부를 확인한 후, 중복이 없으면 랜덤 암호화 키 DEK_1로 암호화된 암호문 $CT_1 = Encrypt(DEK_1, M)$과 AP의 PRE 공개키 pk_{AP}를 사용하여 암호화된 $CK_1 = E(pk_{AP}, DEK_1)$, $H(M)$의 서명 값 등을 포함한 $DP_1(Data\,Package)$를 CSP에 업로드한다. 중복이 있는 경우, CSP는 AP에게 재 암호화된 키 값 $CK_1 = R(rk_{AP\to u_2}; E(pk_{AP}, DEK_1)) = E(pk_2; DEK_1)$을 요청한다. 여기서 $rk_{AP\to u2}$는 PRE 재암호화 키 생성 알고리즘에 의해 생성된 재암호화 키이다. AP가 $rk_{AP\to u2}$을 CSP에게 전송하고, CSP는 사용자에게 CK_1'을 전달한다. 사용자는 CK_1'을 이용

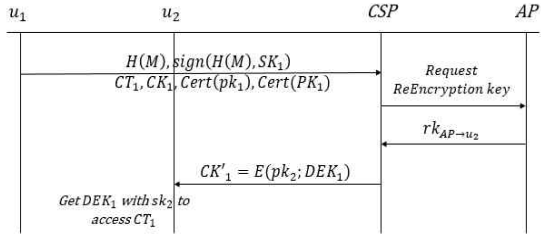

Fig. 1. Yan-Ding-Zhu's scheme.

하여 CT_1에 접근 가능하게 된다.

2.2.2 Hur-Koo-Shin-Kang의 기법[6]

Hur 등은 암호화된 데이터에 대한 중복 제거 기법을 제시하였다. 제안 기법은 동적인 데이터 소유권 관리를 지원한다. 제안 기법은 사용자와 CSP로 구성되며, CSP는 honest-but-curious 신뢰 모델로 가정한다. 제안 기법에서 사용자는 랜덤한 데이터 암호화키 L을 이용하여 메시지 M_i를 암호화한다. $C_i^1 \leftarrow E_L(M_i)$. 또한 키 $K_i \leftarrow H(M_i)$를 계산하여 키 L을 $C_i^2 \leftarrow L \oplus K_i$ 형태로 암호화한다. 암호문은 $C_i = C_i^1 \| C_i^2$로 구성된다. 메시지 태그로 $T_i \leftarrow H(K_i)$를 계산하여 인덱스 정보로 사용한다. 태그 T_i를 통해 클라우드는 중복 여부를 확인한다.

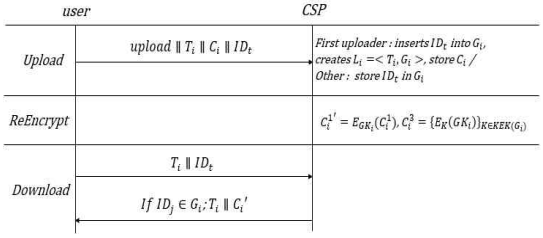

Fig. 2. Hur-Koo-Shin-Kang's scheme.

2.2.3 Yan-Ding-Yu-Zhu-Deng의 기법[7]

2016년 Yan 등은 중복 제거를 지원하는 암호화된 데이터 스토리지 관리를 위해 데이터 소유권 challenge와 PRE에 기반한 기법을 제안하였다. 이 기법은 클라이언트의 데이터 소유권을 검증하였다. 이를 통해, 중복 제거와 데이터 접근 제어 기능을 결합하였다. 제안 기법은 CSP와 데이터 소유자, 그리고 AP로 구성된다. CSP는 honest-but-curious 신뢰 모델로 가정하며, AP는 완전 신뢰 개체이며 CSP와 공모하지 않는다고 가정한다.

데이터 중복 제거를 위한 토큰으로 사용자 u_1은 데이터 M에 대한 토큰 $x_1 = H(H(M)*P)$를 생성하여, $\{x_1, pk_1, Cert(pk_1)\}$을 CSP에게 전송한다. 여기서 P는 ECC (Elliptic Curve Cryptography)에서 base point로 공개 파라미터 값이다. CSP는 x_1을 통해 중복이 없는 경우, 데이터 업로드 요청을 하고, 사용자 u_1은 데이터 M을 DEK_1으로 암호화하여 CT_1을 얻으며,

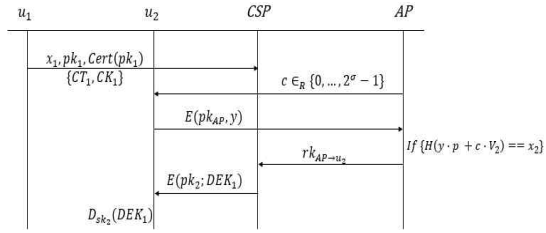

Fig. 3. Yan-Ding-Yu-Zhu-Deng's scheme.

pk_{AP}로 DEK_1을 암호화하여 CK_1을 생성한다. $\{CT_1, CK_1\}$을 CSP에게 전송한다. 중복이 있는 경우에는 다음과 같이 소유권 검사 과정을 수행한다. AP는 랜덤하게 선택한 $c \in_R 0, \ldots, 2^\sigma - 1$를 사용자 u_2에게 challenge로 전송하며, 사용자 u_2는 $y = H(M) + (s_2 * c)$를 계산하여 $E(pk_{AP}, y)$를 AP에게 전달한다. 여기서 s_2는 사용자 u_2의 개인키이다. AP는 sk_{AP}로 복호화하여 y를 복구하고, $H(y \bullet P + c \bullet V_2)$를 계산하여 x_2와 비교한다. 여기서 V_2는 사용자 u_2의 공개키이다. 두 값이 일치하면, 소유권 검증이 완료되고 AP는 재암호화 키 생성 알고리즘 $RG(pk_{AP}; sk_{AP}; pk_2)$을 호출하여 재암호화 키 $rk_{AP \to u_2}$를 생성하고, 이를 CSP에게 전달한다. CSP는 재암호화 알고리즘 $R(rk_{AP \to u_2}; E(pk_{AP}; DEK_1)) = E(pk_2; DEK_1)$을 호출하여 $E(pk_{AP}; DEK_1)$을 재 암호화한 후, 재암호화된 키 $E(pk_2; DEK_1)$를 사용자 u_2에게 전송한다. 사용자 u_2는 자신의 개인키 sk_2로 DEK_1을 복호화 할 수 있게 된다.

2.2.4 XDeup 기법[8]

2016년 Yu는 기존 PAKEDedup[13] 기법의 문제점을 해결하기 위해 업로더와 클라우드 서버만을 수반하는 전수 조사 공격 저항 대칭 암호화 기반 중복 제거 기법 제안하였다. 제안 기법인 XDeup은 은클라우드 서버 S가 확장된 룩업 테이블 $\mathcal{L}^+$을 유지 관리한다. 이 테이블 $\mathcal{L}^+$은 파일 청크 f의 짧은 해시 값인 $sh(f)$ 에 의해 인덱스된다. $\mathcal{L}^+(sh(f)) = 0$인 경우 새로운 파일의 업로드 절차가 진행되며, $\{h_{h(f)}(r), \varepsilon_{h(f)}(r), \varepsilon_{h(f)}(k_f), \varepsilon_{k_f}(f), \varepsilon_{k_i}(k_f)\}$를 서버 S에 업로드한다. 여기서 $\varepsilon_k(x)$는 대칭 비밀키 k를 이용한 입력 메시지 x 의 암호화를 나타낸다. k_f는 파일 청크 f의 청크 키이고 k_i는 사용자 c_i의 개인 비밀 키이다. $\mathcal{L}^+(sh(f)) \neq 0$인 경우 중복 제거 과정을 진행한다.

3. 기존 기법들의 보안 취약점

3.1 Yan-Ding-Zhu 기법의 보안 취약점

Yan-Ding-Zhu의 제안 기법에서는 데이터의 해시 값이 보호되고, 공격자가 이를 획득할 수 없다고 가정하고 있다. 하지만, 현실 상황에서 이는 매우 강한 가정이다. 일반적으로 해시 값은 비밀이 아니며, 여러 가지 상황에서 해시 값은 노출될 수 있다[5]. 그러므로 제안 기법은 해시 값이 노출되는 경우에는 기존의 중복 제거 기법들이 가지고 있는 취약점을 그대로 가진다. 특히, 예측 가능한 메시지들에 대해서 오프라인 전수 조사 공격에 대한 가능성이 존재한다. 공격자는 예측 가능한 메시지 공간의 모든 메시지들에 대해서 중복 데이터 식별자 id를 계산하여 중복 여부를 확인 할 수 있다. 이는 사용자의 중요 데이터에 대한 정보를 노출하는 프라이버시 문제를 초래한다. 또한 제안 기법은 기본적인 서버 측 중복 제거 기법으로 제안되고 있지만, 클라이언트 측 중복 제거 기법으로 적용하는 것도 가능하다고 기술하고 있다. 하지만, 클라이언트 측 중복 제거 기법을 적용하기 위해서는 소유권 검증 기법의 도입이 필요하며, 그렇지 않은 경우, 공격자는 CSP를 CDN으로 활용하는 공격이 가능하다.

그리고, Targeted Collision 공격의 가능성 역시 존재한다. 최초 업로더가 $DP_1 = CT_1, CK_1, H(M), Sign(H(M), SK_1), Cert(pk_1), Cert(PK_1)$를 CSP에 업로드하게 되는데, 이 경우 서버는 해시 값에 대한 서명 검증을 통해 수신된 데이터를 인증한다. 하지만 공격자가 $H(M)$과 다른 암호문 CT_1' 을 업로드하게 되면, CSP는 이를 확인할 수 없다. 따라서 CSP에서 수신된 데이터에 대한 정확성을 검증하거나 중복된 데이터를 업로드하는 사용자가 CSP에 저장된 데이터가 자신의 데이터와 동일한지를 검증할 수 있는 방안이 필요하다.

마지막으로 AP가 평문 데이터에 접근 가능하다는 취약점이 존재한다. 제안 논문에서는 AP가 Raw Data(즉, 암호데이터)에 접근할 수 없다고 기술되어 있지만, 이것은 매우 강한 가정이다. AP가 평문 데이터가 아닌 암호 데이터에 접근할 수 없다는 것은 일반적이지 않은 가정이다. 일반적으로 AP 역시 완전히 신뢰되기 보다는 semi-trust 개체로 가정하는 것이 좀 더 타당하며, 많은 관련 논문에서도 AP를 honest-but-curious 신뢰 모델로 가정하고 있다. 따라서 AP를 honest-but-curious 신뢰 모델로 가정하게 되면, AP는 DEK를 획득 할 수 있게 되며, 사용자의 평문 데이터에 접근 가능하게 된다. 실제로 첫 번째 업로드 과정에서 $CK = E(pk_{AP}, DEK)$를 암호문 CT와 함께 업로드하기 때문에 AP가 CK를 획득하게 되면 이를 sk_{AP}로 복호화 하여 DEK를 획득 가능하고 따라서 AP는 평문 데이터에 접근 가능하다.

3.2 Hur-Koo-Shin-Kang 기법의 보안 취약점

Hur등의 제안 기법은 데이터 프라이버시 위협이 존재한다. 이는 낮은 엔트로피를 가지는 예측 가능한(predictable) 파일에 대해, 공격자는 모든 가능한 파일들을 구성할 수 있고, 이 파일들로부터 태그를 계산하여 업로드를 요청함으로써 중복제거가 발생하는지를 관찰할 수 있다. 또는 트래픽 도청을 통해 사용자에 의해 업로드 되는 (T_i, C_i)쌍을 획득할 수 있다. 따라서 이 기법 역시 기본적인 수렴 암호화 기법의 문제점을 그대로 가지고 있다. 따라서, 오프라인 전수 조사 공격에 취약하다.

제안 기법에서 키는 $K_i \leftarrow H(M_i)$이고, 메시지 태그는 $T_i \leftarrow H(K_i)$이므로, 전수 조사 공격을 통해 인덱스 정보로 사용되는 태그 $T_i \leftarrow H(H(M_i))$에 대한 추측 공격이 가능하다. 태그 추측이 성공하면, K_i 역시 획득 가능하며, 이를 통해 C_i^2로부터 랜덤 암호화 키 L을 유도할 수 있다. $L \leftarrow C_i^2 \oplus K_i$. 공격자가 랜덤 암호화 키 L을 획득하면, $M_i \leftarrow D_L(C_i^1)$을 통해 평문 M_i를 획득할 수 있다.

3.3 Yan-Ding-Yu-Zhu-Deng 기법의 보안 취약점

첫 번째로 Yan 등의 기법은 클라우드 스토리지에 대해 CDN 공격이 수행될 수 있다. 사용자는 데이터 M의 배포에 클라우드 스토리지를 활용하기 위해 다른 사용자들에게 $H(M)$ 값만을 배포한다. $H(M)$을 획득한 사용자들은 CSP에게 업로드 요청을 하게 되며, CSP는 중복을 발견하여 소유권 검증 과정을 수행하게 된다. $H(M)$만을 가진 사용자 u_i는 AP의 challenge c에 대해 $y = H(M) + (s_i * c)$를 계산하여 $E(pk_{AP}, y)$를 AP에게 전달함으로써 소유권 검증을 통과할 수 있으며, 이를 통해 데이터 M에 대한 접근 권한을 획득하게 된다. 그리고 최초 업로더에 의해 전송된

$\{CT_1, CK_1\}$에 대해 CSP는 태그 x_1과 암호문 CT_1의 적절성을 검증하지 않기 때문에, Targeted Collision 공격의 가능성 역시 존재한다.

더 심각한 문제는 사용자의 개인키가 유출될 수 있다는 취약점이 존재한다는 것이다. 데이터 M의 해시 값 $H(M)$은 일반적으로 예측 가능(predictable)하며 (즉, 메시지의 공간이 크지 않음), 따라서 오프라인 전수 조사 공격에 의해 $H(M)$을 알아낼 수 있다. 토큰 $x_i = H(H(M)*P)$를 공격자가 획득할 수 있으며, 이를 통해 메시지 공간상의 메시지들에 대해 $H(M)$을 계산하여 x_i와 일치하는지 여부를 판단할 수 있다. $H(M)$을 알아내게 되면, 소유권 검사 과정에서 사용되는 정보 $y=(H(M)+(s_i*c))$을 얻어낼 수 있으며, 이를 통해 사용자의 개인키 s_i를 알아낼 수 있게 된다. c는 AP에 의해 전달되는 랜덤한 challenge 값으로 비밀 값이 아니다. y값에서 $H(M)$을 빼주면 $y-H(M)=(s_i*c)$를 얻게 되고, 이 값에 c^{-1}를 곱해주면 사용자의 개인키 $(s_i*c)*c^{-1}=s_i$를 얻을 수 있게 된다. 이는 제안 논문의 'Proposition 2'의 증명이 틀렸다는 것을 말하며, 소유권 증명 과정에서 사용자의 개인키가 유출될 수 있는 취약점이 존재한다.

3.4 XDeup 기법의 보안 취약점

XDedup은 낮은 최소 엔트로피(min-entropy) 청크의 가정하에서 오프라인 및 온라인 전수 조사 공격에 대해 안전하다고 주장하고 있다. 하지만, 제안 기법 역시 저자의 주장과 달리 오프라인 전수 조사 공격에 대해 안전하지 않다.

사용자 c_i는 $\mathcal{L}^+(sh(f))=0$인 경우에 $\{h_{h(f)}(r), \varepsilon_{h(f)}(r), \varepsilon_{h(f)}(k_f), \varepsilon_{k_f}(f), \varepsilon_{k_i}(k_f)\}$를 서버 S에 업로드한다. 공격자가 이 값을 획득하게 되면, $h_{h(f)}(r)$와 $\varepsilon_{h(f)}(r)$ 값을 이용하여 오프라인 전수 조사 공격을 수행한다. 가능한 청크 메시지 공간의 모든 경우에 대해 $x=D_{h(f)}(r)$을 계산한 후, 이 값 x를 해시한 결과가 $h_{h(f)}(r)$와 일치하는 지를 검사한다. 일치하는 값을 발견하면, 공격자는 $h(f)$를 찾은 것이 되고, 이를 통해 청크 암호화 키 k_f $(=D_{h(f)}(k_f))$도 획득할 수 있다. 따라서, 낮은 최소 엔트로피 청크인 경우, 제안 기법인 Xdedup은 오프라인 전수 조사 공격에 대해 안전하지 않다.

3.5 분석 및 고려사항

앞의 분석을 통해 기존의 몇 가지 중복 제거 기법들이 일반적인 공격 방법인 전수 조사 공격, CDN 공격, Targeted Collision 공격에 대해서 취약성이 존재한다는 것을 알 수 있었다. 이러한 취약성의 주된 이유는 로컬 스토리지에 데이터를 저장하는 것과는 달리 원격지의 스토리지에 데이터를 저장하는 과정에서 부적절한 가정이나 잘못된 상호 작용에 의한 것이다. 프로토콜의 안전성을 중복 여부의 확인을 위한 식별자로 사용되는 파일의 해시 값의 비밀성에 의존하는 것은 너무 강한 가정이며 현실적으로 부적절한 가정이다. 또한 AP와 같은 개체를 완전 신뢰로 가정하는 것 역시 다소 강한 가정이며, 현실적으로 semi-trust 개체로 가정하여 프로토콜의 안전성을 논하는 것이 적절하다.

그리고, 프로토콜의 입력 데이터(또는 파일)은 낮은 엔트로피를 가지는 예측 가능하다고 가정하는 것이 현실적이다. 따라서 이러한 경우에는 항상 오프라인 전수 조사의 공격이 가능하므로 이에 대해 안전하도록 프로토콜을 설계해야 한다.

오프라인 전수 조사 공격에 안전성을 제공하기 위해서는 DupLESS와 같이 수렴 키를 생성하는 데 도움을 주는 키 서버와 같은 개체를 도입할 필요가 있고, 이 경우 역시 이 개체는 semi-trust로 가정하는 것이 적절하다. 또한 키 서버를 도입하는 경우에도 온라인 전수 조사 공격이 가능성이 존재하므로, 이에 대처할 수 있는 방안을 고려해서 프로토콜을 설계해야 한다.

기존 기법이 CDN 공격에 취약한 원인은 클라이언트 측 중복 제거 기법에서 중복이 발생한 경우 클라이언트가 해당 파일 전체를 실제 소유하고 있는지를 클라우드 서버가 확인하지 않기 때문에 발생하므로 클라우드 서버는 파일의 실제 보유를 검증하기 위한 PoW 과정을 반드시 수행해야 한다. Targeted Collision 공격에 대한 취약점은 중복이 발생한 경우 클라이언트가 클라우드 서버에 해당 파일이 실제 저장되어 있는지를 확인하지 않기 때문에 발생한다. 따라서 중복 제거 과정에서 클라이언트는 중복 파일이 실제 클라우드 서버에 이미 저장되어 있는 파일과 일치하는지 확인하도록 프로토콜을 설계해야 한다.

4. 결 론

본 논문에서는 클라우드 스토리지의 효율적인 사

용을 위해 제안된 몇 가지 중복 제거 기법에 대해서 살펴보고, 보안 취약점에 대해서 분석하였다. 기존의 로컬 스토리지에서의 중복 제거 기법과는 달리 중복 제거 기법이 적용된 클라우드 스토리지의 데이터 아웃소싱은 많은 장점을 제공하지만, 그에 따르는 보안 취약점이 존재한다. 이러한 보안 취약점은 원격지의 스토리지에 데이터를 저장하는 과정에서 부적절한 가정이나 잘못된 상호 작용에 의한 것이다. 클라이언트 측 안전한 중복 제거 기법을 설계할 때 파일의 해시 값의 비밀성에 의존하지 않도록 하고 AP와 키 서버와 같은 개체는 semi-trust로 가정하고 또한 입력 데이터 역시 낮은 엔트로피를 가지는 것으로 가정하는 것이 좀더 현실적이다. 또한 CDN 공격에 대처하기 위해 PoW 프로토콜을 수행하는 것이 필요하며, 이 과정에서 클라이언트는 중복 파일이 실제 클라우드 서버에 이미 저장되어 있는 파일과 일치하는지 확인하도록 프로토콜을 설계하는 것이 필요하다.

REFERENCE

[1] K. He, J. Chen, R. Du, Q. Wu, G. Xue, and X. Zhang, "Deypos: Deduplicatable Dynamic Proof of Storage for Multi-user Environments," *IEEE Transactions on Computers,* Vol. 65, No. 12, pp. 3631-3645, 2016.

[2] T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, "Secure and Efficient Cloud Data Deduplication With Randomized Tag," *IEEE Transactions on Information Forensics and Security,* Vol. 12, No. 3, pp. 532-543, 2017.

[3] Cisco Global Cloud Index: Forecast and Methodology, https:// www.cisco.com/c/dam/m/en_us/service-provider/ ciscoknowledgenetwork/files/622_11_15-16-Cisco_GCI_CKN_2015-2020_AMER_EMEAR_NOV2016.pdf (accessed Jun., 18, 2018).

[4] C. Kim, D. Kim, H. Kim, Y. Kim, and D. Seo, "Torus Network Based Distributed Storage System for Massive Multimedia Contents," *Journal of Korea Multimedia Society,* Vol. 19, No. 8, pp. 1487-1497, 2016.

[5] Z. Yan, W. Ding, and H. Zhu, "A Scheme to Manage Encrypted Data Storage with Deduplication in Cloud," *Proceeding of International Conference on Algorithms and Architectures for Parallel Processing,* Vol. 9530, pp. 547-561, 2015.

[6] J. Hur, D. Koo, Y. Shin, and K. Kang, "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage," *IEEE Transactions on Knowledge and Data Engineering,* Vol. 28, No. 11, pp. 3113-3125, 2016.

[7] Z. Yan, W. Ding, X. Yu, H. Zhu, and R.H. Deng, "Deduplication on Encrypted Big Data in Cloud," *IEEE Transactions on Big Data,* Vol. 2, No. 2, pp. 138-150, 2016.

[8] C.M. Yu, *XDedup: Efficient Provably-secure Cross-user Chunk-level Client-side Deduplicated Cloud Storage of Encrypted Data*, International Association for Cryptologic Research Cryptology ePrint Archive: Report 2016/1041, 2016.

[9] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side Channels in Cloud Services: Deduplication in Cloud Storage," *IEEE Security and Privacy,* Vol. 8, No. 6, pp. 40-47, 2010.

[10] S. Keelveedhi, M. Bellare, and T. Ristenpart, "DupLESS: Server-aided Encryption for Deduplicated Storage," *Proceeding of the 22nd Unix Users Group Security Symposium,* pp. 179-194, 2013.

[11] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of Ownership in Remote Storage Systems," *Proceeding of the 18th Association for Computing Machinery Conference on Computer and Communications Security,* pp. 491-500, 2011.

[12] M.W. Storer, K. Greenan, D.D. Long, and E.L. Miller, "Secure Data Deduplication," *Proceeding of the 4th Association for Computing Machinery International Workshop on Storage Security and Survivability*, pp. 1-10, 2008.

[13] J. Liu, N. Asokan, and B. Pinkas, "Secure Deduplication of Encrypted Data without Additional Independent Servers," *Proceeding*

of the 22nd Association for Computing Machinery Special Interest Group on Security, Audit and Control Conference on Computer and Communications Security, pp. 874-885, 2015.

박 지 선

2017년 2월 부경대학교 IT융합응용공학과(학사)
2017년 3월～현재 부경대학교 정보보호학협동과정 재학
관심분야: 블록체인, 암호 프로토콜

신 상 욱

1995년 2월 부경대학교 전자계산학과(학사)
1997년 2월 부경대학교 전자계산학과(석사)
2000년 2월 부경대학교 전자계산학과(박사)

2000년 4월～2003년 8월 한국전자통신연구원 선임연구원
2003년 9월～현재 : 부경대학교 IT융합응용공학과 교수
관심분야 : 암호 프로토콜, 블록체인, 디지털 포렌식, IT융합보안