Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.4, Aug. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.4.971

암호화된 클라우드 데이터의 중복제거 기법에 대한 부채널 공격*

신 형 준,[1†] 구 동 영,[2] 허 준 범[1‡]
[1]고려대학교, [2]한성대학교

Side-Channel Attack against Secure Data Deduplication over Encrypted Data in Cloud Storage*

Hyungjune Shin,[1†] Dongyoung Koo,[2] Junbeom Hur[1‡]
[1]Korea University, [2]Hansung University

요 약

클라우드 환경에서 대량으로 발생하는 데이터들에 대해 효율적인 저장 공간을 제공하는 기법으로 단일의 데이터만을 저장하여 중복을 제거하는 중복제거 기법을 활용할 수 있다. 위탁 데이터에 대한 기밀성에 민감한 사용자들은 안전한 암호 알고리즘을 이용 가능하지만 중복제거 기법의 효율성을 떨어뜨린다는 단점을 가지고 있다. 사용자의 데이터 프라이버시를 보장하면서 저장 공간의 효율성을 올리기 위해 2015년에 PAKE(Password Authenticated Key Exchange) 프로토콜을 활용한 서버 측면의 사용자간 중복제거 기법이 제안되었다. 본 논문에서는 부채널을 통하여 제안된 기법이 CoF(Confirmation-of-File) 또는 중복 확인 공격(ducplicate identification attack)에 대해 안전하지 않음을 증명한다.

ABSTRACT

Data deduplication can be utilized to reduce storage space in cloud storage services by storing only a single copy of data rather than all duplicated copies. Users who are concerned the confidentiality of their outsourced data can use secure encryption algorithms, but it makes data deduplication ineffective. In order to reconcile data deduplication with encryption, Liu et al. proposed a new server-side cross-user deduplication scheme by exploiting password authenticated key exchange (PAKE) protocol in 2015. In this paper, we demonstrate that this scheme has side channel which causes insecurity against the confirmation-of-file (CoF), or duplicate identification attack.

Keywords: Secure Data Outsourcing, Server-side Deduplication, Client-side Deduplication

I. 서 론

대량으로 발생하는 데이터의 저장을 위해 사용되는 클라우드 저장 서비스에서 위탁 데이터의 저장 공간 효율성을 향상시키는 방법으로 중복제거 기법을 활용할 수 있다. 중복제거 기법은 중복된 데이터에

Received(06. 26. 2017), Modified(08. 07. 2017), Accepted(08. 07. 2017)
* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No.2017-0-00380,차세대 인증 기술 개발)과 한국연구재단의 지원을 받아 수행된 연구임(No.2016R1A2A2A05005402). 본 연구는 한성대학교 교내학술연구비 지원과제 임(구동영).
† 주저자, hjshin@isslab.korea.ac.kr
‡ 교신저자, jbhur@korea.ac.kr(Corresponding author)

대해서 단일의 데이터만을 저장하고 타당한 데이터 소유자들에게 이미 저장된 데이터에 대한 링크(소유권)만을 제공하기 때문에 백업 환경과 같이 중복된 데이터가 많이 발생하는 환경에서 보다 효율적으로 저장 공간을 관리할 수 있다[1]. 특히, 클라이언트 측 중복 제거 기법(client-side deduplication)은 해쉬 값과 같은 데이터로부터 생성된 비교적 작은 크기의 값을 통하여 사전에 데이터의 중복을 확인하고 이를 통해 불필요한 중복된 데이터 전체 업로드를 방지할 수 있다. 중복된 데이터에 대해서 비교를 위해 작은 크기의 값만을 전송하는 클라이언트 측 중복제거 기법은 저장 공간 및 대역폭 효율성에 장점을 가지고 있지만 CoF 공격 (Confirmation-of-File attack)에 취약하다는 단점을 가지고 있다[14, 19]. 클라이언트 측 중복 제거 기법에서의 CoF 공격자는 적은 수의 업로드 요청을 보낸 후 지정된 데이터 전체가 업로드 되는지 혹은 그렇지 않은지를 관찰함으로써 해당 데이터가 서버에 존재(저장)하는 지를 확인할 수 있다. 반면에 서버 측 중복 제거 기법(server-side deduplication)은 모든 클라이언트들이 데이터 전체를 업로드 한 후 서버에 의해 중복 제거가 이루어지기 때문에 CoF 공격에 대한 데이터 프라이버시 문제를 해결한다.

자신의 위탁 데이터에 대한 기밀성을 중요하게 생각하는 사용자들은 위탁 전에 데이터를 암호화하여 클라우드에 저장할 수 있다. 데이터 암호화에서는 사용자들 마다 서로 다른 임의의 키를 사용하기 때문에 같은 데이터에 대한 암호화는 서로 다른 암호문을 생성하게 된다. 즉, 데이터 중복 제거에서는 중복된 데이터들을 확인 가능해야하는 반면에 안전한 암호화 알고리즘은 중복된 데이터에 대해 구별 불가능한 암호문들을 생성하므로 암호화된 데이터에 대한 중복제거 기술의 적용은 클라우드 저장 공간의 높은 효율성을 보장하지 못 한다[2].

데이터 기밀성의 보장과 저장 공간의 효율성을 높이기 위해 암호화된 데이터에 중복제거 기법을 적용시키기는 방안으로 결정론적 암호 알고리즘(deterministic encrytpion)과 데이터로부터 파생되어진 키를 이용하여 같은 데이터에 대해서 항상 같은 암호문을 생성하는 MLE(Message-Locked Encryption)의 하나의 예로 CE(Coverenct Encryption)를 볼 수 있다[15, 3]. CE의 결정론적인 특성은 암호화에 사용된 키에 대한 정보 없이 암호문의 간단한 비교를 통하여 중복 제거를 가능하게 하지만 오프라인 전수조사 공격(offline brute-force attack)에 취약하다는 단점을 가지고 있다[14, 18, 20].

CE에서 발생하는 취약점을 해결하기 위해 추가적인 키 서버를 활용하여 CE에서의 오프라인 전수조사 공격을 방어 가능한 기법들이 제안되었다[4,5,6,7]. 결정론적 블라인드 서명(deterministic blind signatures)[8]을 이용하여 안전하게 키 서버의 랜덤 키를 통한 암호화 키를 생성함으로써 제안된 기법들은 키 서버가 오프라인 전수조사 공격을 하지 않는다는 가정 하에 CE보다 높은 안전성을 보장한다. 하지만, 안전하고 효율적인 키 서버의 관리는 실제 상업 클라우드 시나리오에서는 실현되기 힘들다는 한계점을 가지고 있다.

2015년에 Liu 등[9]은 추가적인 제 3의 서버가 필요 없는 안전한 서버 측 중복제거 기법을 제안했다. 그들은 중복제거 과정에서 추가적인 키 서버의 필요성과 CE 기반의 중복제거 기법들에서 발생하는 정보 노출을 제거했다. 제안된 기법은 PAKE(Password Authenticated Key Exchange)[10]를 이용하여 같은 데이터를 가진 사용자들끼리 공통의 세션 키(임시 키)를 생성하게 하고 이를 통하여 데이터의 중복을 확인한다. 데이터의 중복 여부에 따라서 제안된 기법은 lifted ElGamal 암호[11]와 Paillier 암호[12]와 같이 공개키 기반의 동형암호(public key homomorphic encryption)를 사용하여 안전하게 같은 데이터를 소유한 사용자들끼리 암호화키를 공유 가능하게 한다. 공유된 암호화 키는 데이터로부터 파생된 값이 아니라 임의로 선택된 값이기 때문에 CE 기반의 중복제거에서 발생하는 공격에 안전하다. 또한, 결정론적 암호 알고리즘을 통하여 생성된 암호문은 중복된 데이터에 대해서 같은 암호화키로 암호화되기 때문에 암호문의 비교를 통해서 중복 제거가 가능하게 된다.

본 논문에서는 Liu 등[9]이 제안한 기법이 동형암호를 통한 암호화키를 공유하는 과정에서 부채널이 발생하며 이를 통하여 CoF 공격이 가능함을 보인다. 부채널을 통한 CoF 공격은 두 개의 순차적인 데이터 업로드 클라이언트를 사용한다. 자신이 원하는 데이터가 서버에 저장되어 있는지에 대한 정보를 알고 싶은 공격자는 두 번의 순차적인 업로드를 통하여 해당 정보를 알아낼 수 있다. 본 논문에서 제안하는 공격 방법은 Liu 등이 제안한 기법이 그들이 정

의한 안전성 요구사항에 부합하지 않음을 보이며 기존의 서버 측 중복제거 기술에서는 발생하지 않았던 정보의 노출이 발생가능하다는 점을 확률적인 방법을 통하여 증명한다.

II. 분석 대상

본 장에서는 부채널 공격의 대상이 되는 Liu 등[9]이 제안한 기법과 해당 기법이 만족해야 하는 안전성 요구사항에 대해 설명한다.

2.1 안전성 요구사항

Liu 등[9]에 의해 제안된 서버 측 중복제거 기법은 다음과 같은 안전성을 요구한다.

- 클라우드 서버를 제외한 어떠한 개체도 업로드 요청을 한 데이터에 대해 같은 내용을 가진 중복된 데이터가 이미 서버에 존재하는지에 대한 정보를 알 수 없다.

문헌에서 다뤄지는 다른 공격 위협들 중에 본 논문에서는 다음과 같은 [9](3.2절)에서 정의한 온라인 공격자에 초점을 맞춘다.

- 공격자는 프로토콜 참여를 거부할 수 있고 그들의 입력 값을 다른 값으로 변경 가능하며 프로토콜이 완료되기 전에 중단시킬 수 있다.

2.2 분석 대상 기법 개요

2.2.1 해쉬 함수(hash function)

해쉬 함수 $H: \Phi \rightarrow \{0,1\}^n$는 Φ에 있는 임의의 길이의 이진 문자열을 n의 길이를 갖는 이진 문자열 h로 대응시키는 결정론적 함수이다. 암호학적 해쉬 함수(cryptographic hash function)는 임의의 값을 생성하며 단 방향성(one-way)을 만족하고 결과 값에 대한 충돌을 찾아내는 게 현실적으로 실행 불가능한 해쉬 함수를 가리킨다. 분석 대상 기법에서의 해쉬 함수 H는 암호학적 해쉬 함수를 의미한다.

충분히 긴 길이의 결과 값을 가진 암호학적 해쉬 함수는 충돌저항성(collision-resistance)을 만족하지만, 짧은 길이의 결과 값을 가진 해쉬 함수는 많은 충돌을 야기한다(서로 다른 입력 값에 대해서 같은 결과 값을 생성하기 쉽다). 분석 대상 기법에서는 효율성과 기밀성의 보장을 위해 짧은 길이의 해쉬 함수 SH를 사용한다.

2.2.2 PAKE 프로토콜

PAKE 프로토콜은 사용자 인증에서 흔히 사용되는 패스워드 기반 프로토콜에서 오프라인 전수조사 공격에 취약하다는 문제를 해결하기 위해 제안된 방법이다[10, 16, 17]. 분석 대상 기법[9]에서는 Abdalla와 Pointcheval이 제안한 SPAKE2 프로토콜[13]을 통하여 PAKE 기능을 구현하였다. Fig.1.은 SPAKE2 프로토콜에서 공통의 키를 생성하는 과정을 보여준다. Fig.1.에서 볼 수 있듯이 Alice와 Bob이 같은 패스워드를 가지고 있을 경우 ($pw_A = pw_B$), 같은 세션 키($SK_A = SK_B$)를 생성함을 알 수 있다. [9]에서 제안된 기법은 SPAKE2 프로토콜에서 패스워드 대신에 위탁할 데이터의 해쉬 값을 사용한다.

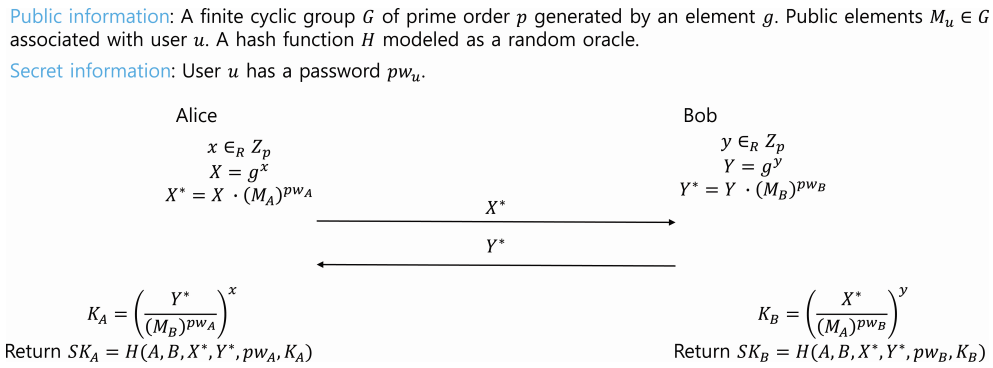

Fig. 1. SPAKE2 protocol used in [9]

2.2.3 합 연산 동형 암호(additively homomorphic encryption)

다음의 조건을 만족하는 공개키 암호화 알고리즘은 합 연산 동형 성질을 가지고 있다고 말할 수 있다.

- 주어진 두 개의 암호문 $c_1 = Enc(pk, m_1; r_1)$와 $c_2 = Enc(pk, m_2; r_2)$에 대해서 pk에 대응되는 개인키에 대한 정보 없이 효율적으로 새로운 임의의 값 r을 통한 $Enc(pk, m_1 + m_2; r)$을 계산 가능하다.

합 연산 동형 암호의 대표적인 예로 평문을 지수승하여 사용할 경우 Paillier 암호[12]와 lifted ElGamal 암호[11]를 들 수 있다. [9]에서는 제안된 기법을 구현하기 위해 lifted ElGamal 암호를 이용한다.

2.2.4 시스템 구성요소

Liu 등[9]이 제안한 기법은 다음의 개체들을 가지고 있다.

- 저장 서버(storage server, S)는 온라인 저장 서비스를 제공하고 위탁된 데이터에 대해 중복제거를 실행한다.
- 온라인 확인자 집합(online checkers, $C_i s$)은 위탁된 데이터에 대해서 그들의 소유권이 검증된 타당한 데이터 소유주들의 집합이다. (이전에 어떤 파일 F_i를 저장 서버에 업로드한 사용자)
- 데이터 소유주(data owner, C)는 S에 자신의 데이터를 위탁하려는 클라이언트이다. 데이터 소유주는 대응되는 $C_i s$와 공통의 세션 키(암호화키)를 생성하기 위해 다수의 PAKE 프로토콜에 참여한다.

[9]에서 서버 S는 위탁된 데이터에 대한 높은 충돌 확률을 가진 짧은 길이의 해쉬 값(sh_i), 암호화된 위탁 데이터($SE(k_{F_i}, F_i)$), 그리고 대응되는 타당한 데이터 소유주들의 집합을 포함하는 메타 데이터(C_i)를 가지고 있다. Fig.2.는 S가 저장하고 있는 메타 데이터의 구조를 보여준다. Fig.3.는 Liu 등이 제안한 기법의 전체적인 흐름을 보여주며 Table 1.은 이때 사용되는 표기법들을 나타낸다. Liu 등이 제안한 기법의 전반적인 절차는 다음과 같다.

Table 1. Notation

Notation	Description
SH	Short hash function with high collision
H_1, H_2	Cryptographic hash functions
PRF	Public pseudorandom function
Enc/Dec	Additively homomorphic encryption/decryption
$\oplus/\ominus$	Additively homomorphic addition/subtraction
$\|\|$	Concatenation
$a \in_R A$	Uniform random selection of a from set A
P	Plaintext space
k_i, k'_i	Session keys established through PAKE
k_{F_i}, k_F	Symmetric encryption key of data F

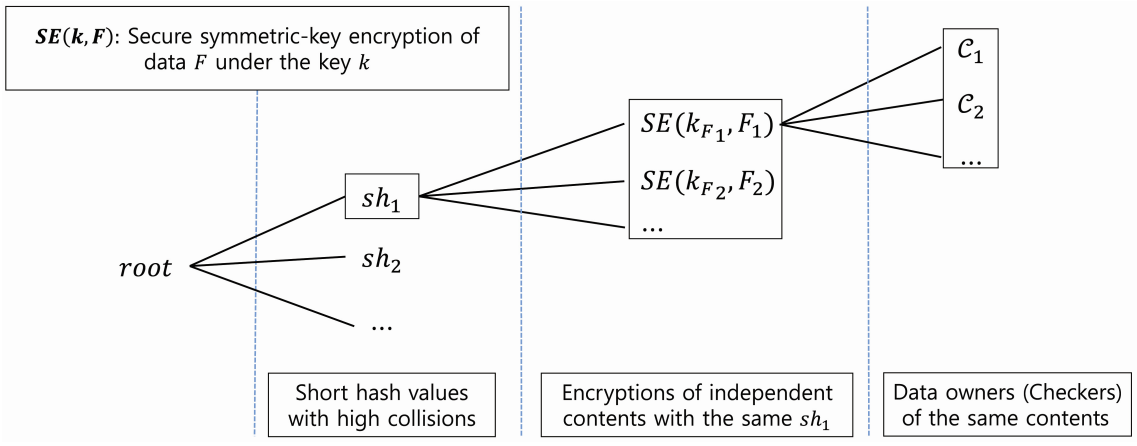

Fig. 2. Metadata structure in the server S

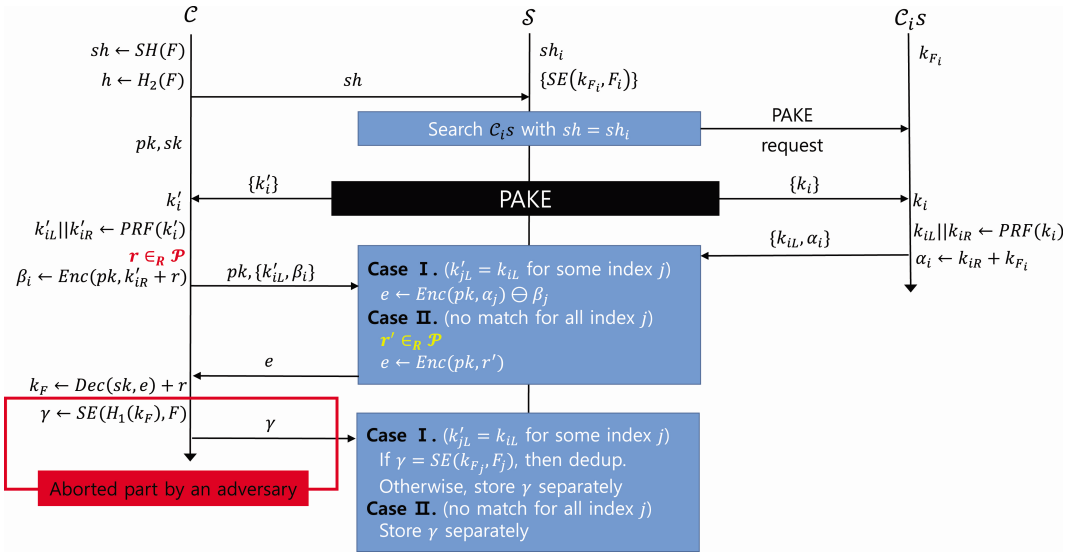

Fig. 3. Liu et al.'s server-side deduplication protocol

1) 데이터 소유주는 파일 F를 저장 서버에 저장하기 위해 우선 F를 이용하여 짧은 길이의 해쉬 값 sh와 암호학적 해쉬 함수를 통한 해쉬 값 h를 생성하고, sh를 저장 서버 S에 전송한다.

2) S는 Fig.2.에서 볼 수 있는 저장된 메타 데이터를 통해 전송 받은 sh가 이전에 저장되어 있는 값인지를 확인한다. 만약 이미 저장되어 있는 값이라면 S는 해당 값에 대응되는 암호문 각 각에 대해서 현재 온라인인 소유주들($C_i s$)을 확인하고, 각각의 암호문에 대해서 한명의 확인자를 선택한다.

3) 파일을 업로드하려고 하는 데이터 소유주 C와 선택 받은 온라인 확인자 C_i들은 S를 통하여 PAKE 프로토콜을 진행하게 된다. 이때, C는 PAKE 프로토콜의 패스워드 값으로 h를 사용하며 C_i는 $h_i(\leftarrow H_2(F_i))$를 사용한다($sh = sh_i$).

4) PAKE 프로토콜을 진행한 후에, C는 온라인 확인자 수만큼의 세션 키 k_i'를 얻게 된다. 각각의 k_i'에 대해서 C는 PRF를 이용하여 키의 길이를 확장하고 세션 키의 왼쪽 부분 k_{iL}'과 오른쪽 부분 k_{iR}'을 나누고, 임의의 값 r을 선택하며 자신의 공개키 pk와 $\{k_{iL}'\}$, 그리고 동형 암호를 이용한 $Enc(pk, k_{iR}' + r)$을 S에게 전송한다. 온라인 확인자 C_i 각자는 PAKE 프로토콜을 진행한 후에 단일의 세션 키 k_i를 얻게 된다. C_i는 k_i에 대해서 PRF를 이용하여 키의 길이를 확장한 후 왼쪽 부분 k_{iL}과 오른쪽 부분 k_{iR}로 나누며, 이전에 파일을 암호화하는데 사용한 암호화키 $k_{F_i}(sh = sh_i)$를 이용하여 $k_{iR} + k_{F_i}$와 k_{iL}을 S에게 전송한다.

5) C와 C_i로부터 메시지($[pk, k'_{iL}, \beta_i]$와 $[k_{iL}, \alpha_i]$)를 받은 S는 세션 키의 왼쪽 부분을 통하여 F의 중복을 확인한다. 만약에 중복이 확인되었을 경우에는 동형 암호의 성질을 통하여 e를 생성하고 (Case Ⅰ), 그렇지 않을 경우에는 자신이 선택한 임의의 값을 이용하여 생성 한다 (Case Ⅱ). S는 생성된 e를 C에게 전송한다.

6) S로부터 e를 전송받은 C는 자신의 개인키로 동형 암호를 복호화 하고 이전에 사용된 r 값을 사용하여 암호화키 k_F를 생성 한다 (만약 어떤 온라인 확인자 C_j가 이전에 올린 파일 F_j와 F가 같다면 $k_F = k_{F_j}$이다[9]). C는 생성된 암호화키를 이용하여 일반적인 결정론적(deterministic) 대칭키 암호 알고리즘을 이용하여 파일 F에 대한 암호문을 생성하고 이를 S에 전송한다.

7) 만약에 Case Ⅰ에 의해 e가 생성되었을 경우 S는 전송받은 암호문이 C_j가 이전에 올린 암호문과 같은지 확인하고 같다면 중복 제거를 진행한다. Case Ⅱ에 의해서 e가 생성되었을 경우와 C_j가 이전에 올린 암호문과 다를 경우, S는 전송 받은 암호문을 따로 저장한다.

C와 C_i에 대한 온라인 전수조사 공격(online brute-force attack)을 막기 위해 Liu 등이 제안한 기법은 PAKE 요청의 수에 대한 비율 제한 전술(rate limiting strategy)을 이용한다. 또한, 네트워크 대역폭의 보다 좋은 효율성을 위해 임의로 추출된 임계값 접근 방법(randomized threshold approach)[14]을 이용한 클라이언트 측 중복 제거 기법에 대한 변형이 간단하게 [9]에 언급되어 있다.

III. 부채널을 통한 CoF 공격

본 장에서는 Liu 등이 제안한 기법이 부채널을 통한 CoF 공격에 취약하며, 그들이 제안한 안전성 요구조건에 위배됨을 보인다.

공격자 A는 지정된 데이터 F가 S에 이미 존재하는지에 대한 정보를 알고 싶으며 이를 위해 순차적으로 A_1, A_2 두 개의 인위적인 업로더(클라이언트)를 조작한다고 가정하자. 첫 번째로, 공격자는 $r_1 \neq r_2$을 만족하는 두 개의 임의의 값 $r_1, r_2 \in P$를 선택한다. A_1은 원래 프로토콜로부터 γ의 전송을 제외한 암호문의 마지막 전송까지의 Fig.2.에서 보이는 프로토콜 과정을 진행한다. 그 후에 A_1은 서버로부터 전송된 암호문 e를 통하여 다음 두 개의 값을 계산하고 저장한다.

$$e_1' = Dec(sk_{A_1}, e)$$

$$k_F^{A_1} = Dec(sk_{A_1}, e) + r_1$$

다음으로 A_2는 r_2를 이용하여 A_1과 같은 절차를 진행하며, 새로 서버로부터 전송된 암호문 e를 통하여 e_2'과 $k_F^{A_2}$를 계산한 후 이를 저장하고 프로토콜을 중단한다. Table 2.는 공격자 A에 의해 획득 가능한 정보를 나타낸다.

$k_F^{A_1} = k_F^{A_2}$ 일 때, 공격자는 압도적인 확률로 저장 서버 S에 공격자가 업로드 요청한 데이터 F와 중복된 데이터가 존재한다는 사실을 알 수 있다[1]. Table 3.는 발생 가능한 경우에 대응되는 확률들을 보여준다.

1) 오직 $r'' = r' + r_1 - r_2$ 일 때, 공격자 A는 실제로는 데이터 F와 중복된 파일이 S에 있음에도 불구하고, 중복된 파일이 없다고 잘못 판단할 수 있다. 그러나 Table 2.에서 볼 수 있듯이, 서버가 평문 공간(plaintext space) P로부터 r'과 r''을 선택하기 때문에 공격자가 잘못 판단할 확률은 무시할 만큼 작다.

Table 2. Obtained information by adversary A

Adversary	k_F	$e' = Dec(sk, e)$
A_1	$k_F^{A_1} = r' + r_1$	$e_1' = r'$
A_2	$k_F^{A_2} = r'' + r_2$	$e_2' = r''$

서로 다른 두 개의 임의의 값 r_1, r_2를 선택하는 공격자에 대해 $k_F^{A_1} = k_F^{A_2}$ 가 가능하기 위해서는 $e_1' + r_1$와 $e_2' + r_2$ 가 서로 같아야하기 때문에 $e_1' = e_2'$인 경우는 발생하지 않는다. $e_1' \neq e_2'$인 경우에는 공격자가 지정한 데이터 F가 이미 S에 저장되어 있는 경우(Case Ⅰ)와 그렇지 않은 경우(Case Ⅱ)가 발생한다. $k_F^{A_1} = k_F^{A_2}$ 이면서 $e_1' \neq e_2'$인 경우에 Case Ⅰ이 발생할 확률은 전체 확률 1에서 Case Ⅱ가 발생할 확률을 감하여 구할 수 있다. Case Ⅱ에서는 S가 r'과 r''을 임의로 선택하기 때문에 r'과 r''이 같거나 다를 경우가 발생할 수 있다. 이때, $e_1' = r'$이고 $e_2' = r''$ 이기 때문에 $e_1' \neq e_2'$인 경우에 $r' = r''$인 경우는 발생하지 않는다. 반면에 $r'' = r' + r_1 - r_2$의 조건을 만족해야하기 때문에 $r' \neq r''$인 경우는 평문 공간의 크기에 의해서 결정되어 $1/|P|$가 되고 그러므로 Case Ⅰ일 경우는 $1 - 1/|P|$가 된다.

$k_F^{A_1} \neq k_F^{A_2}$인 경우에는 $e_1' + r_1$와 $e_2' + r_2$가 서로 달라야만 한다. Case Ⅰ인 경우에 $e_1' = e_2'$을 만족하기 위해서는 $Enc(pk, k_{F_j} - r_1)$와 $Enc(pk, k_{F_j} - r_2)$가 서로 같아야 한다. 이때 공격자는 서로 다른 두 개의 임의의 값 r_1, r_2를 선택하기 때문에 Case Ⅰ인 경우에 e_1'과 e_2'은 서로 같아질 수 없다. Case Ⅱ인 경우에는 $e_1' = r'$이고 $e_2' = r''$이기 때문에 항상 $r' = r''$인 경우만 발생한다.

$k_F^{A_1} \neq k_F^{A_2}$을 만족하면서 $e_1' \neq e_2'$일 때 Case Ⅰ인 경우에는 $Enc(pk, k_{F_j} - r_1)$와 $Enc(pk, k_{F_j} - r_2)$가 서로 다른 경우로 볼 수 있다. 하지만, 두 값이 서로 다름에도 불구하고 r_1, r_2가 e_1'과 e_2'에 각 각 더해지기 때문에 $k_F^{A_1} \neq k_F^{A_2}$을 만족시키지 못하게 된다. 그러므로 Case Ⅰ인 상황은 발생하지 않는다. Case Ⅱ인 경우에는 $e_1' = r'$이고 $e_2' = r''$이기 때문에 항상 $r' \neq r''$인 경우만 발생한다.

Table 3. Obtained information by adversary A

Cases determined by A		Possible cases by S		Probability
$k_F^{A_1} = k_F^{A_2}$	$e_1' = e_2'$	Case Ⅰ(Dedup.)		0
		Case Ⅱ (No Dedup.)	$r' = r''$	0
			$r' \neq r''$	0
	$e_1' \neq e_2'$	Case Ⅰ(Dedup.)		$1-1/\|P\|$
		Case Ⅱ (No Dedup.)	$r' = r''$	0
			$r' \neq r''$	$1/\|P\|$
$k_F^{A_1} \neq k_F^{A_2}$	$e_1' = e_2'$	Case Ⅰ(Dedup.)		0
		Case Ⅱ (No Dedup.)	$r' = r''$	1
			$r' \neq r''$	0
	$e_1' \neq e_2'$	Case Ⅰ(Dedup.)		0
		Case Ⅱ (No Dedup.)	$r' = r''$	0
			$r' \neq r''$	1

정의 1. 〔IND-COF(Indistinguishability under CoF attack)〕 Liu 등〔9〕이 제안한 기법에 대한 CoF 공격 게임에서의 공격자 A가 가지는 이점은 다음과 같다.

$$Adv_{\hat{\pi},\pi,A}^{IND-COF}(\lambda) = |2 \times \Pr[\mathrm{Exp}_{\hat{\pi},\pi,A}^{IND-COF}(\lambda) = 1] - 1| \quad (1)$$

이때, π는 Liu 등〔9〕이 제안한 기법을 가리키며, $\hat{\pi}$은 원래 프로토콜에서 γ의 전송을 제외한 암호문의 마지막 전송까지의 Fig.2.에서 보이는 프로토콜을 가리키고, λ는 보안 파라미터를 나타낸다. $\mathrm{Exp}_{\hat{\pi},\pi,A}^{IND-COF}(\lambda)$은 Fig.4.와 같다.

정의 2. Liu 등〔9〕이 제안한 기법이 CoF 공격에 안전하기 위한 조건은 다음과 같다. 만약 모든 확률적 다항 시간 공격자 A 및 모든 λ에 대해서 다음과 같은 사소 함수(negligible function) $negl(\lambda)$가 존재한다.

$$Adv_{\hat{\pi},\pi,A}^{IND-COF}(\lambda) \leq negl(\lambda) \quad (2)$$

정리 1. 모든 다항시간 공격자 A 및 모든 λ에 대해서 (2)를 만족하는 $negl(\lambda)$은 존재하지 않는다.

증명. $\Pr[\mathrm{Exp}_{\hat{\pi},\pi,A}^{IND-COF}(\lambda) = 1]$을 구해보면 다음과 같다.

$$\begin{aligned}
&\Pr[\mathrm{Exp}_{\hat{\pi},\pi,A}^{IND-COF}(\lambda) = 1] \\
&= \frac{1}{2} Pr[A^{Store}(\lambda, m_0) = 0] + \frac{1}{2} Pr[A^{Store}(\lambda, m_1) = 1] \\
&= \frac{1}{2} \times (1 - \frac{1}{|P|}) + \frac{1}{2} \\
&= 1 - \frac{1}{2|P|}
\end{aligned}$$

(1)의 수식에 대입하면

$$\begin{aligned}
&Adv_{\hat{\pi},\pi,A}^{IND-COF}(\lambda) \\
&= 2 \times (1 - \frac{1}{2|P|}) - 1 \\
&= 1 - \frac{1}{|P|}
\end{aligned}$$

즉, 모든 확률적 다항 시간 공격자 A 및 모든 λ에 대해서 항상 $1-\frac{1}{|P|}$의 값을 가지고 $\frac{1}{|P|}$는 매우 작은 값이기 때문에 (2)의 수식을 만족하는 사소 함수는 존재하지 않는다. 그렇기 때문에 Liu 등〔9〕이 제안한 기법은 제안된 CoF 공격에 취약하다. □

Fig.4.에서 공격자의 시간 복잡도는 Liu 등〔9〕이 제안한 기법에서 2번의 파일 업로드 요청을 보냈을 때 걸리는 시간과 같다. 단일의 파일 업로드 요청에 대해서 k_F 값을 생성하는데 걸리는 시간은 k_F의 길

$Exp_{\hat{\pi},\pi,A}^{IND-COF}(\lambda)$

1. Let D_0 be a set of messages stored in S and D_1 be a set of messages not to be stored in S
2. Choose uniform $m_0 \in D_0$ and $m_1 \in D_1$
3. Choose uniform $b \in \{0,1\}$
3. Plaintext m_b and λ is given to A
4. A output a bit b' by using $Store$
5. The output of the experiment is defined to be 1 if $b' = b$, otherwise 0.

$Store$:

1. Get k_0 from $\hat{\pi}(\lambda, m_b)$
2. Get k_1 from $\pi(\lambda, m_b)$
3. Output $b' = 0$ if $k_0 = k_1$, otherwise $b' = 1$

Fig. 4. The IND-COF experiment for Liu et al's scheme

이에 비례하여 증가하게 된다. 즉, 키의 길이를 λ라고 했을 때, 공개 키 암호로 ElGamal 암호[11]를 사용할 경우 λ에 대한 순환 그룹 내에서의 지수 연산 및 곱 연산에 비례한 시간이 걸리게 된다. 지수 연산으로 제곱-곱 연산을 사용하고 곱 연산에 걸리는 시간을 $M(\lambda)$라고 했을 때 공격자의 시간 복잡도는 $O(M(\lambda)\log\lambda)$로 나타낼 수 있다. 만약 간단한 곱 연산을 사용할 경우($O(\lambda^2)$) 공격자의 시간 복잡도는 $O(\lambda^2\log\lambda)$로 나타난다.

예측 가능한 파일과 같은 낮은 엔트로피(무질서도, low entropy)를 지닌 데이터에 대해서 공격자는 적은 수의 시도를 통하여 지정된 메시지가 클라우드 저장 서버에 이미 저장되어 있는지 혹은 아닌지에 대한 정보를 알아낼 수 있다. 부채널을 통한 CoF 공격은 키 합의(key agreement) 동안 (암호화된 파일을 저장 서버에 전송하기 이전)에 진행되기 때문에 서버 측 혹은 클라이언트 측과 같은 중복 제거가 발생하는 측면과는 상관없이 Liu 등[9]이 제안한 기법에서 항상 발생 가능하다. 그러므로 [9]에서 사용한 임계값 접근 방법 혹은 비율 제한 전술은 제안된 공격을 방어하지 못한다.

IV. 결 론

서버 측 중복 제거 기법의 접근 방법은 클라이언트 측 중복 제거 기법에 비해 CoF와 같은 공격에 대한 정보 노출이 없기 때문에 안전성 측면에서 더 선호된다. 본 논문에서는 Liu 등[9]이 제안한 PAKE 프로토콜 기반의 서버 측 중복 제거 기법을 대상으로 확률적인 분석을 통해 부채널이 있음을 보였고, 이를 통하여 CoF 공격이 가능하다는 것을 증명했다. 이와 같은 공격의 가능성은 불행하게도 Liu 등[9]이 제안한 기법이 안전성 요구조건을 만족하지 못함을 보여주며 서버 측 중복 제거 기법임에도 불구하고 CoF 공격이 가능하기에 기존 기법들에 비해 높은 안전성을 보장하지 못한다는 사실을 보여준다.

References

[1] Dutch T. Meyer and William J. Bolosky, "A study of practical deduplication," Proceedings of the 9th USENIX Conference on File and Storage Technologies, pp. 1-1, Feb. 2011.

[2] Mark W. Storer, Kevin Greenan, Darrell D.E. Long, and Ethan L. Miller, "Secure data deduplication," Proceedings of the 4th ACM international workshop on Storage security and survivability, pp.1-10, Oct. 2008.

[3] J.R. Douceur, A. Adya, W.J. Bolosky, P. Simon and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," Proceedings of the 22nd International Conference on Distributed Computing Systems, pp. 617-624, July. 2002.

[4] M. Bellare, S. Keelveedhi and T.

Ristenpart, "Dupless: server-aided encryption for deduplicated storage," Presented as part of the 22nd USENIX Security Symposium, pp. 179-194, Aug. 2013.

〔5〕 P. Puzio, R. Molva, M. nen and S. Loureiro, "ClouDedup: secure deduplication with encrypted data for cloud storage," in 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 363-370, Mar. 2013.

〔6〕 J. Stanek, A. Sorniotti, E. Androulaki and L. Kencl, "A secure data deduplication scheme for cloud storage," in 18th International Conference on Financial Cryptography and Data Security, pp. 99-118, Nov. 2014.

〔7〕 Yitao Duan, "Distributed key generation for encrypted deduplication: achieving the strongest privacy," Proceedings of the 6th edition of the ACM Workshop on Cloud Computing Security (CCSW), pp. 57-68, Nov. 2014.

〔8〕 J. Camenisch, G. Neven and A. Shelat, "Simulatable adaptive oblivious transfer," Proceedings of the 26th Annual International Conference on Advances in Cryptology (EUROCRYPT), pp. 573-590, May. 2007.

〔9〕 J. Liu, N. Asokan, and B. Pinkas, "Secure deduplication of encrypted data without additional independent servers," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 874-885, Oct. 2015.

〔10〕 S.M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in IEEE Computer Society Symposium on Research in Security and Privacy, pp. 72-84, May. 1992.

〔11〕 T. ElGamal, "A public key cryptosystem and a signature scheme based on discret logarithms," Proceedings of Advances in Cryptology (CRYPTO), pp. 10-18, Aug. 1985.

〔12〕 P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," Proceedings of the 17th international conference on Theory and Application of Cryptographic Techniques (EUROCRYPT), pp. 223-238, May. 1999.

〔13〕 M. Abdalla and D. Pointcheval, "Simple password-based encrypted key exchange protocols," Proceeding of the Cryptographers' Track at the RSA Conference (CT-RSA), pp. 191-208, Feb. 2005.

〔14〕 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, Dec. 2010.

〔15〕 M. Bellare, S. Keelveedhi and T. Ristenpart, "Message-locked encryption and secure deduplication," in 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 296-312, May. 2013.

〔16〕 A. Jain and S. Pankanti, "Biometrics: A tool for information security," in IEEE Transactions on Information Forensics and Security, vol. 1, no. 2, pp. 125-143, Jun. 2006.

〔17〕 Daniel V. Klein, "Foiling the cracker: A survey of, and improvements to, password security," Proceedings of the 2nd USENIX Security Workshop, pp. 5-14, Jul. 1990.

〔18〕 Zheng Yan, Mingjun Wang and Yuxiang Li, "Encrypted data management with deduplication in cloud computing," IEEE Cloud Computing, vol. 3, no. 2, pp. 28-35, May. 2016.

〔19〕 P. Puzio, R. Molva, M. Önen, and S.

Loureiro, "PerfectDedup: secure data deduplication," In International Workshop on Data Privacy Management, pp. 150-166, Sep. 2015.
[20] Youngjoo Shin, Dongyoung Koo, and Junbeom Hur. "A survey of secure data deduplication schemes for cloud storage systems," ACM Computing Surveys (CSUR), vol. 49, no. 74, Feb. 2017.

〈저자소개〉

신 형 준 (Hyungjune Shin) 정회원
2015년 2월: 중앙대학교 컴퓨터공학부 졸업
2017년 2월: 고려대학교 컴퓨터학과 석사
2017년 3월~현재: 고려대학교 컴퓨터학과 박사과정
〈관심분야〉 정보보호, 응용 암호, 클라우드 보안

구 동 영 (Dongyoung Koo) 정회원
2009년 2월: 연세대학교 컴퓨터.산업공학과 졸업
2012년 2월: 한국과학기술원 전산학과 석사
2016년 2월: 한국과학기술원 전산학부 박사
2016년 3월~2017년 3월: 고려대학교 정보대학 컴퓨터학과 연구교수
2017년 4월~현재: 한성대학교 기계전자공학부 조교수
〈관심분야〉 정보보호, 응용 암호, 네트워크 보안, 클라우드 보안

허 준 범 (Junbeom Hur) 종신회원
2001년 2월: 고려대학교 컴퓨터공학 졸업
2005년 8월: 한국과학기술원 전산학 석사
2009년 8월: 한국과학기술원 전산학 박사
2009년 9월~2011년 8월: University of Illinois at Urbana-Champaign 박사후 연구원
2011년 9월~2015년 2월: 중앙대학교 컴퓨터공학부 조교수
2015년 2월~2016년 8월: 고려대학교 컴퓨터학과 조교수
2016년 9월~현재: 고려대학교 컴퓨터학과 부교수
〈관심분야〉 정보보호, 응용 암호, 네트워크 보안, 클라우드 보안