Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.1, Feb. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.1.123

클라우드 스토리지 상에서의 프라이버시 보존형 소스기반 중복데이터 제거기술*

박 철 희,[1†] 홍 도 원,[1‡] 서 창 호,[1] 장 구 영[2]
[1]공주대학교, [2]ETRI

Privacy Preserving Source Based Deduplication In Cloud Storage*

Cheolhee Park,[1†] Dowon Hong,[1‡] Changho Seo,[1] Ku-Young Chang[2]
[1]Kongju National University, [2]ETRI

요 약

최근 클라우드 스토리지 사용이 급증함에 따라 스토리지의 효율적인 사용을 위한 데이터 중복제거 기술이 활용되고 있다. 그러나 외부 스토리지에 민감한 데이터를 저장할 경우 평문상태의 데이터는 기밀성 문제가 발생하기 때문에 중복처리를 통한 스토리지 효율성 제공뿐만 아니라 데이터 암호화를 통한 기밀성 보장이 필요하다. 최근, 스토리지의 절약뿐만 아니라 네트워크 대역폭의 효율적인 사용을 위해 클라이언트측 중복제거 기술이 주목을 받으면서 다양한 클라이언트측 중복제거 기술들이 제안되었지만 아직까지 안전성에 대한 문제가 남아있다. 본 논문에서는 암호화를 통해 데이터의 기밀성을 보장하고 소유권 증명을 이용해 데이터 접근제어를 제공하여 신뢰할 수 없는 서버와 악의적인 사용자로부터 프라이버시를 보존할 수 있는 안전한 클라이언트측 소스기반 중복제거 기술을 제안한다.

ABSTRACT

In cloud storage, processing the duplicated data, namely deduplication, is necessary technology to save storage space. Users who store sensitive data in remote storage want data be encrypted. However Cloud storage server do not detect duplication of conventionally encrypted data. To solve this problem, Convergent Encryption has been proposed. But it inherently have weakness due to brute-force attack. On the other hand, to save storage space as well as save bandwidths, client-side deduplication have been applied. Recently, various client-side deduplication technology has been proposed. However, this propositions still cannot solve the security problem. In this paper, we suggest a secure source-based deduplication technology, which encrypt data to ensure the confidentiality of sensitive data and apply proofs of ownership protocol to control access to the data, from curious cloud server and malicious user

Keywords: Cloud Storage, Deduplication, Privacy

I. 서 론

클라우드 스토리지 서비스 제공자들은 중복된 데이터의 처리를 통하여 저장 공간을 효율적으로 사용할 수 있다. 즉, 서로 다른 사용자가 같은 데이터를 아웃소싱 할 경우 스토리지 서버는 중복된 데이터를 단 한 번만 저장함으로써 스토리지 효율성을 증대시킨다. 스토리지의 효율적인 사용을 위한 중복제거 기술은 수행

접수일(2014년 11월 13일), 수정일(2015년 1월 5일), 게재확정일(2015년 1월 5일)
* 본 논문은 ETRI 주요사업(15ZS1500) 및 교육부와 한국연구재단의 지역혁신 창의인력 양성사업(NO.2013H1138A2032077)의 지원을 받아 수행된 연구임.
† 주저자, newpch89@kongju.ac.kr
‡ 교신저자, dwhong@kongju.ac.kr(Corresponding author)

주체에 따라 서버측 타겟기반 중복제거 기술과 클라이언트측 소스기반 중복제거 기술로 구분된다.

◉ 서버측 타겟기반 중복제거: 서버는 클라이언트로부터 전체 데이터를 수신한 후 서버측에서 중복을 제거하는 방식이다. 프라이버시 침해가 적지만 데이터의 송·수신과 중복검사를 함께 수행하므로 네트워크 혼잡이 발생할 수 있다.

◉ 클라이언트측 소스기반 중복제거: 클라이언트가 직접 중복을 제거하는 방식으로써 네트워크 혼잡이 적고 대역폭의 효율적인 사용이 가능하다. 하지만 상대적으로 크기가 작은 *Tag* 값만으로 전체 데이터를 대체하는 것은 데이터 소유권에 대한 문제가 발생할 수 있다.

클라우드 스토리지(DropBox[1], Mozy[2], Google-Drive[3]등)에 민감한 데이터를 아웃소싱할 경우 평문상태의 데이터가 저장되는 것은 프라이버시에 위협이 되므로 데이터를 암호화하여 저장해야 한다. 그러나 일반적인 암호화 방식의 경우 클라이언트들이 서로 다른 비밀키를 사용한다면 같은 평문일지라도 암호문은 서로 다른 값을 갖기 때문에 동일한 데이터에 대한 중복처리가 불가능하다. 이러한 문제점을 보완하기 위하여 Douceur 등에 의해 Convergent Encryption 기법이 제안되었다[4]. Convergent Encryption은 데이터의 해시 값을 키로 사용하기 때문에, 동일한 데이터에 대해 동일한 암호문이 생성되어 데이터의 중복 처리가 가능하다. 그러나 Convergent Encryption은 전수조사 공격에 매우 취약하기 때문에 인가되지 않은 사용자일지라도 암호문만으로 평문 데이터를 예측할 수 있는 문제점을 가지고 있다. 이를 보완하기 위하여 Bellare 등은 키 서버의 도움을 받아 데이터를 암호화하는 방식인 DupLESS를 제안했다[5]. DupLESS는 암호문만으로 평문데이터를 예측할 수 없으므로 전수조사 공격에 저항성을 가진다. 그러나 DupLESS는 서버측 타겟기반 중복제거 기술로써 외부 스토리지의 저장 공간을 효율적으로 사용할 수 있지만 네트워크 대역폭의 낭비가 발생한다. 따라서 클라우드 서비스 제공자들은 저장 공간뿐만 아니라 네트워크 대역폭의 효율적인 사용을 위하여 클라이언트 측 소스기반 중복제거 기술을 필요로 한다.

클라이언트 측 소스기반 중복제거 기술은 상대적으로 크기가 큰 전체 데이터 대신 데이터에 대한 해시 값만으로 중복을 감지하기 때문에 불필요하게 낭비되는 대역폭을 절약할 수 있다. 하지만 클라이언트 측 소스기반 중복제거 기술은 상대적으로 크기가 작은 해시 값이 전체 데이터를 대신하기 때문에 해당 해시 값이 위협을 받는다면 전체 데이터에 대한 위협이 발생할 수 있다. 이에 따라 클라이언트 측 중복제거 기술은 데이터 소유권에 대한 증명이 필요하며 Halevi 등은 Merkle-Tree 기반의 소유권 증명방법인 PoW(proofs of ownership)를 제안했다[6]. PoW는 데이터를 encoding(또는 universal hashing, streaming)한 후, 그 값을 이용해 Merkle-Tree 기반의 소유권 증명을 수행한다. 그러나 Halevi 등의 방법은 평문데이터에 대한 소유권 증명만을 제공하고 있어, 데이터의 기밀성을 보장하지 않는다. 따라서 암호화된 데이터에 대한 클라이언트측 소스기반 중복제거 기술이 필요하다.

최근 Kaaniche 등은 Convergent Encryption을 통한 데이터 암호화 및 Merkle-Tree 기반의 소유권 증명방법인 PoW가 결합된 클라이언트측 중복제거 기법을 제안했다[7]. 이 기법은 암호화된 데이터를 기반으로 소유권 증명을 수행하기 때문에 데이터의 기밀성을 유지할 수 있다. 하지만 Kaaniche 등의 방법은 기존의 Convergent Encryption을 통하여 데이터를 암호화하기 때문에 여전히 전수조사공격인 사전공격에 대해 취약한 문제점을 가지고 있다.

본 논문에서는 이와 같은 약점을 보완하기 위하여 클라우드 스토리지로부터 독립적인 키 서버의 도움을 받아 데이터 암호화를 위한 키를 발급하며, 이러한 키를 이용해 암호화를 실행하고 Merkle-Tree를 구성한 후 PoW를 수행하는 클라이언트측 소스기반 중복제거 기술을 제안한다. 이는 현재까지 알려진 모든 취약점을 보완할 수 있는 프라이버시 보존형 소스기반 데이터 중복제거 기술이다.

II. 관련 기술

◉ Convergent Encryption: Douceur 등[4]에 의해 제안된 Convergent Encryption은 데이터 f를 해시한 값 $H(f)$를 키로 사용하며 대칭키 암호 알고리즘 E를 이용해 데이터 f를 암호화 한다.

$$C \leftarrow E(H(f), f)$$

따라서 같은 평문은 동일한 암호문이 되므로 중복데이터의 처리가 가능하다. 하지만 Convergent Encryption은 $H(f)$를 키로 사용하므로 전수조사 공격인 사전공격(dictionary attack)에 매우 취약하다.

◉ DupLESS: Convergent Encryption의 취약점인 사전공격을 보완하기 위하여 Ballere 등은 키 서버를 도입한 DupLESS를 제안했다[5]. DupLESS는 키 서버를 통하여 클라이언트에게 키를 분배하고 클라이언트는 분배받은 키를 이용해 데이터를 암호화한다. 이때 키 서버는 클라이언트와의 통신에서 RSA-OPRF[8][9]프로토콜을 이용해 클라이언트는 키 서버의 비밀정보를, 키 서버는 클라이언트의 비밀정보를 서로 알 수 없게 메시지로부터 유도되는 키를 분배하여 데이터의 중복 처리를 수행한다. 그러나 DupLESS는 키 서버와 공모하여 키 서버의 비밀 정보를 알면서 암호화된 데이터에 접근할 수 있는 공격자에 대해 엔트로피가 매우 낮은 데이터는 안전하지 않을 수 있다. 이 경우 DupLESS의 안전성은 Convergent Encryption의 안전성과 같아진다. 따라서 키 서버는 상당히 신뢰할 수 있는 제 3자(Semi-Trusted Third Party)로 가정하고 클라이언트로부터 요구되는 키 요청을 비율적으로 제한함으로써 안전한 데이터 아웃소싱을 가능하게 한다. 하지만 DupLESS는 서버측 타겟기반 중복제거 기술이므로 중복된 데이터 저장을 요청할 경우 대역폭의 낭비가 발생한다.

◉ Merkle-Tree: 1989년 R. C. Merkle 등은 해시트리를 이용한 인증 방식을 제안했다[10]. 전체 데이터를 다운로드 하지 않고 데이터의 일부만을 이용해 무결성을 검증할 수 있도록 고안되었다. Merkle-Tree는 전체 데이터를 고정길이를 갖는 블록단위로 분할 한 뒤 인접한 두 블록에 대한 해시 값을 계산한다. 연속적으로, 인접해 있는 해시 값들은 다시 해시의 입력으로 사용되며 새로운 해시 값을 출력한다. 이 과정을 반복하여 최종적으로 하나의 해시 값이 출력될 때까지 반복한다.(이때 고정길이를 갖는 최하위

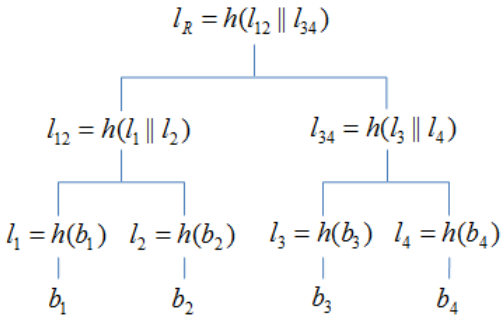

Fig. 1. Merkle-Tree

블록의 개수는 2^m으로 가정한다. 만일 최하위 블록의 개수가 2^m이 아닌 경우 패딩 등의 방식으로 블록의 개수를 2^m으로 맞춘다.) 결과적으로 최하위 노드인 leaf node는 파일의 블록이며 중간노드는 해시를 사용해 계산된 값으로 높이 m의 이진 Tree를 구성한다. Fig 1.은 4개의 블록 b_1, b_2, b_3, b_4로 이루어진 파일 f의 Merkle-Tree 구성이다. leaf node l_1,...,l_4는 b_1,...,b_4를 각각 해시한 값 $l_1 = h(b_1)$,..., $l_4 = h(b_4)$이고 중간 node는 인접한 두 leaf node의 해시 값 $l_{12} = h(l_1 \| l_2)$, $l_{34} = h(l_3 \| l_4)$이다. 동일한 방법으로 root 값인 l_R은 인접한 l_{12}와 l_{34}의 해시를 통해 출력된다. $MT_{h,b}(X)$는 데이터 X를 길이 b-bit를 갖는 블록으로 분할하고 해시함수 h를 이용해 구성된 Merkle-Tree를 의미한다. Fig 1.의 예에서 $MT_{h,b}(X) = \{l_1, l_2, l_3, l_4, l_{12}, l_{34}, l_R\}$이다. 또한 leaf node l에 대하여 l로부터 root까지의 경로 중 형제 node들을 sibling path $P(\cdot)$라고 한다. 예를 들어, Fig 1.에서 l_1의 $P(1) = \{l_2, l_{34}\}$이다.

◉ 소유권 증명(proofs of ownership): Halevi 등[6]에 의해 제안된 PoW(proofs of ownership)는 Merkle-Tree 기반의 소유권 증명 프로토콜이며 클라이언트는 서버로부터 요구되는 무작위 색인에 대하여 올바른 sibling path를 제시해야 한다[6][10]. Halevi 등은 최소 엔트로피가 낮은 데이터에 대한 안전성 문제를 보완하기 위하여 서로 다른 3가지 버전의 PoW를 제안했다. 클라이언트가 데이터의 소유

를 서버에게 증명할 수 있는 이 방식은, Merkle-Tree를 구성한 후 Tree의 최하위 node인 leaf node의 개수와 최상위 node인 root 값을 서버에게 전송한다. 서버는 무작위 leaf node의 색인을 클라이언트에게 전송하고 클라이언트는 해당 색인의 leaf node와 올바른 sibling path를 응답한다. 이때 서버는 클라이언트로부터 전송받은 값을 이용해 자체적인 root′ 값을 계산하고 만일 root′=root이면 해당 클라이언트에게 데이터 소유를 인정한다. 하지만 Halevi 등의 방식은 데이터를 암호화하지 않기 때문에 평문 데이터에 대한 소유권 증명을 수행한다.

◉ 클라이언트측 소스기반 중복제거 기술: 클라이언트측에서 직접 중복을 제거하는 방식으로써 데이터의 *Tag* 값을 계산하고 이를 외부 스토리지 서버로 전송한다. 이때 *Tag* 값이 이전에 스토리지에 저장된 데이터의 *Tag* 값이라면 클라이언트는 데이터 전체를 전송하지 않고 이미 저장되어있는 데이터에 대하여 해당 클라이언트의 정보인 메타데이터만을 전송하는 방식이다. 따라서 클라이언트측 소스기반 중복제거 기술은 외부 스토리지의 저장 공간뿐만 아니라 네트워크의 대역폭을 절약할 수 있다. 그러나 실제 데이터를 소유하지 않은 악의적인 사용자는 *Tag* 값만을 이용해 데이터의 소유를 주장하고 외부 스토리지에 저장되어있는 데이터를 다운로드 할 수 있다. 또한 클라이언트측에서 직접 중복제거를 수행하므로 클라이언트는 데이터 중복제거의 발생여부를 판단할 수 있다. 이는 외부 스토리지에 특정 데이터의 존재유무를 확인하여 온라인상으로 전수조사 공격을 가능하게 한다[11].

민감한 데이터에 대하여 위와 같은 문제를 보완하기 위해 Kaaniche 등[7]은 최근 Convergent Encryption을 이용해 데이터를 암호화하고 암호화된 데이터를 이용해 Merkle-Tree를 구성한 후 소유권 증명을 수행하는 클라이언트측 소스기반 중복제거 기술을 제안하였다. 이 기법은 Convergent Encryption을 통해 비인가 사용자와 신뢰할 수 없는 스토리지로부터 데이터를 보호한다. 그러나 Kaaniche 등의 제안 기법은 기존의 Convergent Encryption을 통한 암호화 방식을 사용하므로

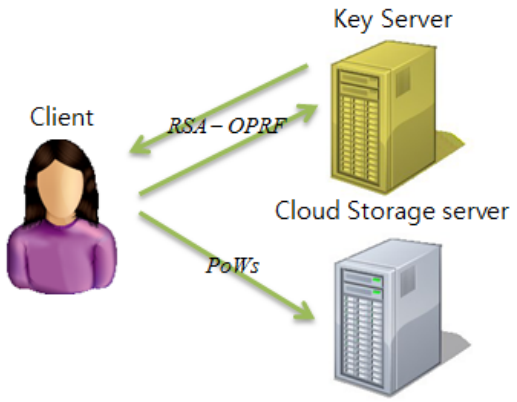

Fig. 2. System construction

여전히 사전공격에 매우 취약할 수 있다. 따라서 본 논문에서는 이러한 취약점을 보완하기 위하여 DupLESS의 키 서버를 이용한 암호화 방식과 Merkle-Tree 기반의 소유권 증명을 결합한 안전한 클라이언트측 소스기반 중복제거 기술을 제안한다.

III. 프라이버시 보존형 소스기반 중복제거 기술

3.1 시스템 구성

Fig 2.와 같이 본 논문에서 제안하는 시스템 구성 요소는 3가지로 이루어진다.

◉ Cloud Storage Server(CSS) : 외부 스토리지인 CSS는 클라이언트와 상호작용하는 주체로써 사용자로부터 받은 파일 및 해당 파일에 관련된 정보를 데이터베이스에 저장하며 CSS는 신뢰할 수 없다고 가정한다.

◉ Key-sever(KS) : KS는 클라이언트와 상호작용하는 주체로써 CSS로부터 독립적이며 클라이언트에게 데이터를 암호화 할 키를 분배한다. 또한 KS는 상당히 신뢰할 수 있는 제 3자(Semi-Trusted Third Party)라고 가정한다.

◉ Client: 클라이언트는 CSS에 데이터를 저장하는 주체로써, 새로운 데이터에 대한 저장을 요청하는 주체인 $Client_1$과 중복된 데이터에 대한

저장을 요청하는 주체인 $Client_2$로 구분한다.

3.2 제안 기법

본 논문에서 제안하는 암호화된 데이터 중복제거 기술은 KS-클라이언트, CSS-클라이언트의 상호작용으로 이루어진다. 이때 해시함수 H는 충돌 저항성을 만족하고 대칭키 암호화 알고리즘 E는 안전한 의사난수 함수이다.

◉ 설계 목표: 소스기반 중복제거 기술은 클라이언트 측에서 데이터에 대한 Tag 값을 생성하며 전체 데이터 보다 비교적 크기가 매우 작은 Tag 값을 이용해 중복을 감지하므로 중복제거 기술에서의 저장 공간 절약뿐만 아니라 불필요하게 낭비되는 네트워크 대역폭의 사용을 줄일 수 있다. 하지만 서버측에서 수행하는 타겟기반 중복제거 기술과 다르게 소스기반 중복제거 기술은 클라이언트측에서 Tag 값을 생성하기 때문에 악의적인 사용자의 경우 상대적으로 크기가 작은 Tag 값을 이용해 불법적인 파일의 전송을 가능하게 한다. 또한 Tag 값이 전체 데이터를 대신하기 때문에 Tag 값만을 가지고 데이터에 대한 소유를 주장할 수 있다. 이러한 문제점을 해결하기 위하여 Halevi 등[6]은 데이터의 소유권 증명 기술인 PoW를 제안하였다. 하지만 PoW는 평문상태로 소유권 증명을 수행하기 때문에 데이터의 기밀성 및 무결성을 보장하는 것은 아니다. 이를 보완하기 위하여 최근 Kaaniche 등[7]은 Convergent Encryption을 이용해 암호화를 하고 암호화된 데이터를 기반으로 소유권 증명을 수행하는 클라이언트측 소스기반 중복제거 기술을 제안하였다. 하지만 여전히 Convergent Encryption은 예측 가능한 평문에 대한 사전공격에 매우 취약한 문제를 가지고 있다. 따라서 본 논문의 목표는 위에서 언급된 문제점들을 보완하는 안전하고 효율적인 클라이언트측 소스기반 중복제거 기술의 설계이다.

3.2.1 키 분배 프로토콜

KS는 Fig 3.과 같이 KS가 클라이언트에게 키를 분배해 주는 과정은 Bellare 등[5]에 의해 제안된 RSA-OPRF프로토콜을 이용한다. 이 과정은 RSA-

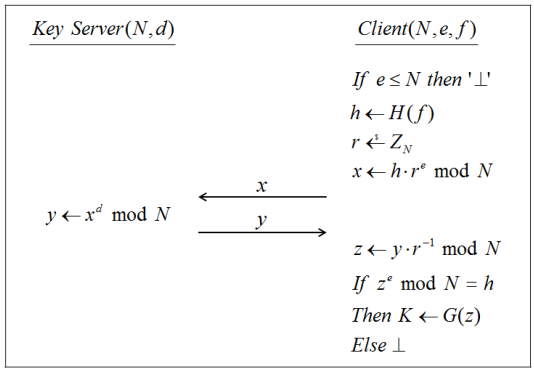

Fig 3. Key distribution protocol

blind-signature[8,9]의 성질을 이용해 클라이언트에게 데이터를 암호화할 키 K를 발급한다.

◉ 키 분배: KS는 RSA 기반 공개키와 개인키 쌍을 생성한다. 즉, 큰 두 소수의 곱 $N = p \cdot q$과 $e \cdot d \equiv 1 \pmod{\varphi(N)}$을 계산하고, N과 e를 공개키로 사용하며, p, q 그리고 d를 개인키로 사용한다.

클라이언트는 Z_N에서 일정한 확률로 선택한 무작위 값 r에 KS의 공개키 e를 이용하여 r^e를 계산하고 자신의 데이터 f을 해시한 값 $h = H(f)$와 r^e을 곱한 값 $x \leftarrow h \cdot r^e \bmod N$을 KS에게 전송한다. 이때 만일 $e \leq N$이면 '$\perp$'한다. KS는 전송받은 x에 자신의 개인키 d를 이용하여 $y \leftarrow x^d \bmod N$을 계산한 후 클라이언트에게 전송한다. 클라이언트는 전송받은 y에 r^{-1}을 곱한 값 $z \leftarrow y \cdot r^{-1} \bmod N$을 계산한 후, 만일 $z^e \bmod N = h$이면 $K \leftarrow G(z)$를 데이터 f를 암호화할 키로 사용하고, $z^e \bmod N \neq h$이면 '$\perp$'한다. 여기서 함수 G는 충돌 저항성을 갖는 해시함수이다.

3.2.2 업로드 프로토콜

◉ $Client_1$-CSS : 클라이언트가 CSS에 존재 하지 않는 데이터 즉, 새로운 데이터의 저장을 요청할 경우, 클라이언트는 먼저 KS에게 데이터 f를 암호화 할 키 K를 발급받은 후 Fig 4.와 같이 데이터를 암호화 한 값 $C_1 \leftarrow E(K, f)$과 자신의 개인

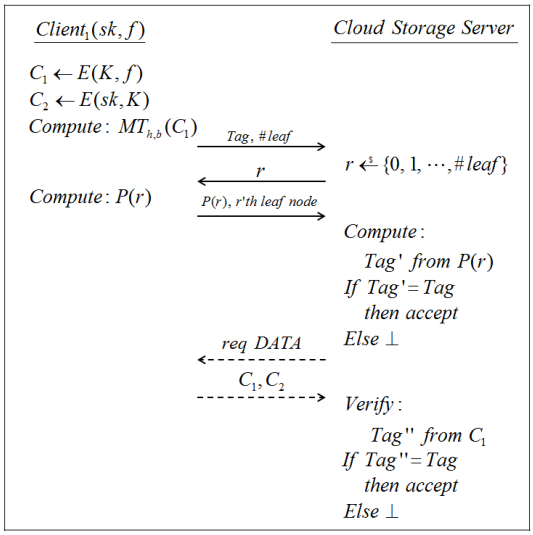

Fig 4. $Client_1$ upload protocol

키 sk로 키 K를 암호화 한 값 $C_2 \leftarrow E(sk, K)$를 계산한다. 이때 C_1을 이용하여 Merkle-Tree를 구성한 후 CSS에게 leaf node의 개수와 Merkle-Tree의 root 값인 Tag 값($Tag = l_R$)를 전송한다. CSS는 클라이언트에게 leaf node의 무작위 색인을 보내고 클라이언트는 해당 색인의 leaf node와 sibling path $P(r)$로 응답한다. 이때 서버는 클라이언트에게 전송받은 leaf node와 $P(r)$을 이용해 자체적인 Tag' 값을 계산한 후 만일 $Tag' = Tag$이면($Tag' \neq Tag$이면 '$\perp$'한다.) 클라이언트에게 데이터에 대한 전송을 요청하고 클라이언트는 계산된 C_1과 C_2를 CSS에 전송한다. 이때 서버는 C_1에 대한 데이터 무결성을 검증하기 위하여, 클라이언트로부터 전송받은 C_1을 이용해 자체적으로 Merkle-Tree의 root 값인 Tag''을 계산한다. 만일 $Tag'' = Tag$이면($Tag'' \neq Tag$이면 '$\perp$'한다.) CSS는 최종적으로 클라이언트에게 C_1에 대한 소유권을 인정하고 C_1을 저장 공간에 저장하며 C_2와 Tag 값을 메타데이터로 저장한다.

◉ $Client_2$-CSS : 앞서 언급한 클라이언트와 달리 CSS에 이미 존재하는 데이터 즉, 중복된 데이터의 저장을 요청하는 클라이언트의 경우, $Client_2$는 우선 키 서버로부터 키 K를 발급 받은 후 Fig 5.와 같이 데이터를 암호화한 값 $C_1 \leftarrow E(K, f)$과 자신의 개인키 sk'을 이용해 키 K를 암호화한 값 $C_2' \leftarrow E(sk', K)$을 계산한다. 이때 C_1을 이용해 Merkle-Tree를 구성하고 CSS에게 leaf node의 개수와 Tag 값을 전송한다. 이때 CSS에 Tag 값이 이미 존재하는 데이터이므로 CSS는 중복을 감지하고 소유권 증명을 수행한 후 메타데이터인 C_2'의 전송만을 요구한다. 따라서 중복된 데이터인 C_1은 전송하지 않고 C_2'만을 전송하기 때문에 네트워크 혼잡이 줄고 대역폭의 효율적인 사용이 가능하다.

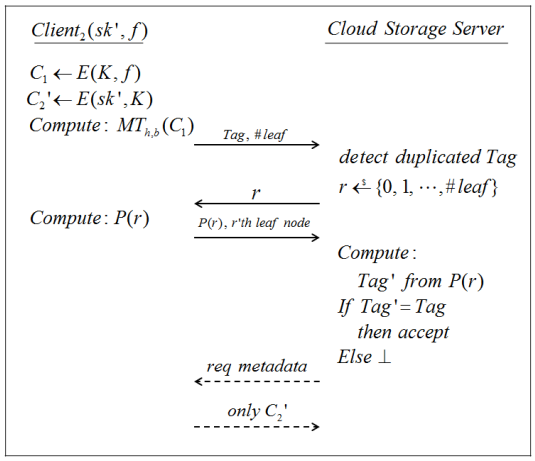

Fig 5. $Client_2$ upload protocol

위와 같이 CSS-클라이언트의 상호작용에서 최종적으로 클라이언트는 데이터를 아웃소싱한 후 $MT_{h,b}(C_1)$과 자신의 개인키만을 유지하고 CSS는 데이터 C_1, C_1에 대한 Tag, 그리고 메타데이터인 C_2, C_2'만을 유지한다.

3.2.3 다운로드 프로토콜

데이터의 소유자가 클라우드 스토리지에 저장해 놓은 데이터에 대한 복구를 요청할 경우 Fig 6.과 같이 클라이언트는 유지하고 있는 $MT_{h,b}(C_1)$에서 Tag 값인 l_R과 leaf node의 개수를 전송한다. CSS는 클라이언트에게 leaf node의 무작위 색인 r을 전송하고 클라이언트는 해당 색인의 leaf node와 sibling

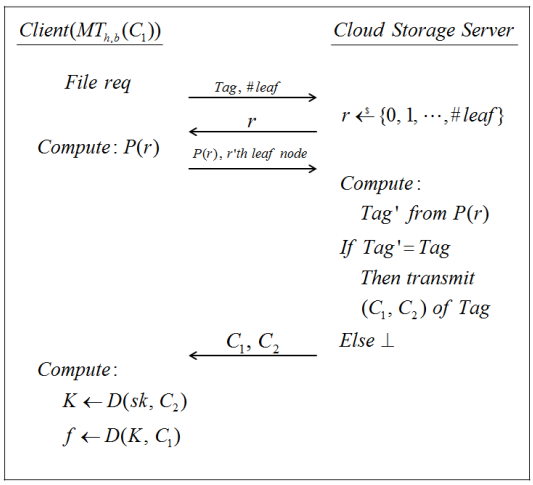

Fig 6. Download protocol

path $P(r)$로 응답한다. 이후 해당 클라이언트의 소유권이 인정되면 CSS는 클라이언트에게 해당 Tag에 대한 C_1과 해당 클라이언트에 대한 C_2(또는 C_2')를 전송한다. 이때 클라이언트는 자신의 개인키인 sk(또는 sk')로 C_2(또는 C_2')를 복호화하여 데이터를 암호화한 키 K를 복구하고 C_1을 복호화하여 데이터 f를 복원한다.

IV. 안전성 분석

Table 1.은 본 논문에서 제안하는 기술과 타 소스 기반 중복제거 제안 기법과의 비교표이다. 본 논문에서 제안하는 기법은 키 서버로부터 발급받은 키를 이용해 데이터를 암호화하기 때문에 키 서버가 안전성에 영향을 미치는 중요한 주체가 된다. 공격자와 키 서버가 공모할 경우 공격자가 키 서버의 개인키인 d를 안다면 클라이언트의 암호화된 데이터는 전수조사 공격에 대하여 안전하지 않을 수 있다. 또한 키 서버는 정당한 클라이언트와 악의적인 목적을 가진 클라이언트를 구분 할 수 없기 때문에 키 서버는 악의적인 사용자의 키 발급 요청일지라도 정당하게 키를 발급한다. 이 경우, 엔트로피가 매우 낮은 데이터는 온라인 전수조사 공격에 위협을 받을 수 있다. 따라서 본 논문에서 제안하는 키 서버의 구성은 Bellare 등[5]에 의해 제안된 DupLESS의 키 서버와 동일한 조건을 가정한다.

◉ 키 서버는 상당히 신뢰할 수 있는 제 3자(Semi-Trusted Third Party)로 가정한다.

◉ Rate limiting: 온라인 전수 조사 공격에 대한 저항성을 갖기 위해 클라이언트마다 키 발급 요청을 비율적으로 제한한다. 일정한 기간 동안 키 발급 요청의 횟수를 제한하며, 키 발급 요청이 누적될 때마다 고정된 지연 시간을 추가한다. 따라서 엔트로피가 낮은 데이터의 경우 비율적으로 제한을 한다면 온라인 전수조사 공격에 대한 저항성을 가질 수 있다.

본 논문에서 제안하는 기법은 3가지 구성인 클라우드 스토리지 서버, 키 서버, 클라이언트로 이루어지며, 공격자가 클라우드 스토리지 서버와 공모한 경우, 키 서버와 공모한 경우와 악의적인 클라이언트에 대한 경우로 구분하여 안전성을 분석한다.

4.1 클라우드 스토리지 서버로부터의 안전성

클라이언트들의 데이터를 저장하고 있는 CSS가 공격자와 공모한 경우, 공격자는 암호화된 클라이언트의 데이터에 쉽게 접근할 수 있다. 따라서 클라이언트들의 암호화된 데이터 C_1과 C_1에 대한 메타데이터인 C_2를 위협한다. 이 경우 C_1은 키 서버로부터 발급 받은 키 K를 이용해 암호화된 데이터이다. 따라서 키 서버의 개인키 d를 알지 못한다면 엔트로피가 낮은 데이터 일지라도 평문 데이터를 예측할 수 없다. 또한 C_2는 클라이언트의 개인키 sk로 키 K를 암호화한 값이므로 sk를 알지 못한다면 C_2를 복호화 할 수 없다. 따라서 클라우스 스토리지 서버만이 공모된 경우 데이터의 안전성을 보장한다.

4.2 키 서버로부터의 안전성

키 서버는 클라이언트에게 데이터를 암호화할 키를 분배한다. 키 서버가 악의적인 의도를 가지고 클라이언트의 정보를 위협할 경우라도 키 서버와 클라이언트는 RSA-OPRF 프로토콜을 이용하여 통신하기 때문에 클라이언트의 민감한 정보인 $H(f)$ 값은 무작위 값 r^e에 가려진 상태로 전송되어 키 서버는 클라이언트의 데이터 정보를 도출해 낼 수 없다. 따라서 키 서버는 자신이 클라이언트에게 분배한 키를 알 수 없다.

Table 1. Comparison of client-side source-based deduplication

	Convergent Encryption	Halevi etc.[6]	Kaaniche etc.[7]	Proposed method
Data encryption	○	×	○	○
Proof of ownership	×	○	○	○
Brute-force attack resilience	×	×	×	○

또한 서로 다른 클라이언트가 같은 파일에 대한 키 발급을 요청할 경우 즉, $H(f)$ 값이 같을 경우 무작위 값 r에 의해 KS는 키 발급이 요청되는 데이터의 동일함을 확인할 수 없다.

4.3 악의적인 사용자로부터의 안전성

클라이언트는 키 서버 및 클라우드 스토리지 서버와 통신을 한다. 이때 정당한 클라이언트가 악의적인 행동을 하는 경우 키 서버의 비밀 정보를 위협하고 CSS에 저장 되어있는 다른 사용자들의 데이터를 위협한다. 또한 악의적인 사용자는 파일 업로드 시 정상적인 파일로부터 생성된 Merkle-Tree를 이용해 소유권 증명 프로토콜을 수행한 후, 데이터 전송 시 악의적인 파일을 업로드 하여 정상적인 데이터 저장을 위협할 수 있다.

첫 번째, 키 서버의 비밀 정보인 (p, q, d)를 위협할 경우 RSA-OPRF 성질에 의해 공격자는 인수분해 문제를 해결해야만 키 서버의 비밀 정보를 위협할 수 있다.

두 번째, CSS에 저장되어 있는 암호화된 데이터를 위협할 경우 악의적인 사용자는 데이터에 대한 소유권 증명을 수행해야 한다. 이때 CSS는 검증자, 클라이언트는 증명자이다. 공격자는 엔트로피 t인 데이터 f에서 최대 b-bit의 정보를 알고 있을 때 $t-b$가 충분히 크다면 공격자는 소유권 증명 프로토콜을 통과할 수 없다. 본 논문에서 제안하는 기법은 암호화된 데이터에 대한 Merkle-Tree를 생성하므로 공격자는 소유권 증명 프로토콜에서 무작위 색인에 대한 sibling path를 제시할 수 없다[6][10]. 또한 공격자는 평문 데이터를 예측 하여 온라인상으로 전수조사 공격을 시도할 수 있다. 이 경우 키 서버의 Rate Limiting에 의해 온라인 전수조사 공격에 대한 저항성을 가질 수 있다[5].

마지막으로, 악의적인 사용자는 데이터 C_1에 대한 소유권 증명을 정당하게 수행한 후, 데이터 업로드 과정에서 C_1'을 전송한다. 따라서 서버는 데이터 C_1'에 대한 메타데이터를 C_1에 대한 Tag 값으로 저장하게 된다. 이후 C_1을 업로드 하는 정당한 사용자는 데이터 다운로드 시 C_1'을 받게 된다. 이러한 공격을 보완하기 위해 서버는 전송받은 C_1에 대해 자체적인 Tag''을 계산하여 소유권 증명 시 전송된 Tag와 비교하여 전송된 데이터의 무결성을 검증한다. 따라서 악의적인 사용자는 소유권 증명을 통과하는 악의적인 데이터를 업로드 할 수 없다.

위와 같이 단일 주체가 공격자에게 공모된 경우 본 논문에서 제안하는 기법은 전수조사공격에 대한 저항성을 갖는다. 그러나 키 서버가 클라우드 스토리지 및 악의적인 사용자와 함께 공모된 경우, 즉, 키 서버의 비밀 정보인 d를 알고 암호화된 데이터에 접근할 수 있는 공격자는 전수조사 공격을 시도할 수 있다. 따라서 엔트로피가 낮은 데이터는 사전 공격에 취약할 수 있으며, 이 경우 Convergent Encryption을 통한 암호화 방식과 같은 안전성을 갖는다.

V. 결 론

본 논문에서는 최근 클라우드 스토리지 서비스에 적용되고 있는 중복제거 기술에 대하여 데이터 프라이버시를 보존하고 외부 스토리지의 저장 공간과 네트워크 대역폭을 절약할 수 있는 방법을 제안하였다. 신뢰할 수 없는 외부 스토리지 서버로부터 안전하게 데이터의 프라이버시를 보존하고 비인가 사용자로부터 데이터의 접근을 차단할 수 있는 프라이버시 보존형 소스기반 중복제거의 구현을 가능하게 한다.

향후 키 서버를 사용하지 않고 예측 가능하지 않은 암호화를 할 수 있는 방식, 소유권 증명 프로토콜에서 오버헤드를 줄일 수 있는 효율적인 소유권 증명방식 등이 남아있는 과제라고 할 수 있다.

References

[1] DropBox, http://www.dropbox.com

[2] Mozy, http://www.mozy.com

[3] google-Drive, http://www.drive.google.com

[4] Douceur, John R., et al. "Reclaiming space from duplicate files in a serverless distributed file system." Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. IEEE, pp. 617-624, 2002.

[5] Bellare, Mihir, Sriram Keelveedhi, and Thomas Ristenpart. "DupLESS: server-aided encryption for deduplicated storage." Proceedings of the 22nd USENIX conference on Security. USENIX Association, pp. 179-194, August. 2013.

[6] Halevi, Shai, et al. "Proofs of ownership in remote storage systems." Proceedings of the 18th ACM conference on Computer and communications security. ACM, pp. 491-500, October. 2011.

[7] Kaaniche, Nesrine, and Maryline Laurent. "A Secure Client Side Deduplication Scheme in Cloud Storage Environments." New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on. IEEE, pp. 1-7, March. 2014.

[8] Camenisch, Jan, and Gregory Neven. "Simulatable adaptive oblivious transfer." Advances in Cryptology-EUROCRYPT 2007. Springer Berlin Heidelberg, pp. 573-590, 2007.

[9] Naor, Moni, and Omer Reingold. "Number-theoretic constructions of efficient pseudo-random functions." Journal of the ACM (JACM) 51.2, pp. 231-262, 2004.

[10] Merkle, Ralph C. "A certified digital signature." Advances in Cryptology—CRYPTO'89 Proceedings. Springer New York, pp. 218-238, January. 1990.

[11] Harnik, Danny, Benny Pinkas, and Alexandra Shulman-Peleg. "Side channels in cloud services: Deduplication in cloud storage." Security & Privacy, IEEE 8.6, pp. 40-47, 2010.

〈저자소개〉

박 철 희 (Cheolhee Park) 정회원
2014년 2월: 공주대학교 응용수학과 학사 졸업
2014년 9월~현재: 공주대학교 수학과 석사 재학
〈관심분야〉 암호모듈 구현, 데이터 보호 기술

홍 도 원 (Dowon Hong) 종신회원
1994년 2월: 고려대학교 수학과 학사
2000년 2월: 고려대학교 수학과 박사
2000년 4월~2012년 2월: 한국전자통신연구원 팀장, 책임연구원
2012년 3월~현재: 공주대학교 응용수학과 교수
〈관심분야〉 암호기술, 프라이버시 보호기술

서 창 호 (Changho Seo) 종신회원
1990년: 고려대학교 수학과 학사
1992년: 고려대학교 수학과 이학석사
1996년: 고려대학교 수학과 이학박사
1996년~1996년: 국방과학연구소 선임연구원
1996년~2000년: 한국전자통신연구원 선임연구원, 팀장
2000년~현재: 공주대학교 응용수학과 교수
〈관심분야〉 암호알고리즘, PKI, 무선인터넷 보안 등

장 구 영 (Ku-Young Chang) 정회원
1995년 2월: 고려대학교 수학과 졸업
1997년 2월: 고려대학교 수학과 석사
2000년 8월: 고려대학교 수학과 박사
2000년 12월~현재: 한국전자통신연구원/암호기술연구실 책임연구원
〈관심분야〉 암호 알고리즘, 암호 프로토콜, 데이터 프라이버시