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.2017.27.4.981

원격 저장소 데이터 아웃소싱에서 발생하는 중복 식별 과정에서의 부채널 분석 및 제거*

구 동 영†‡
한성대학교

Analysis and Elimination of Side Channels during Duplicate Identification in Remote Data Outsourcing*

Dongyoung Koo†‡
Hansung University

요 약

클라우드 컴퓨팅의 대중화로 개인 및 기업의 로컬 저장소에서 관리되던 데이터가 클라우드 스토리지 등 제 3의 공간에 아웃소싱 되면서 유지, 관리 비용의 절감 효과를 얻을 수 있게 됨과 동시에, 다수의 원격저장 서비스 제공자는 공간 자원의 효율화를 위하여 아웃소싱된 데이터의 중복제거 기법을 도입하고 있다. 동일 데이터의 중복성 판단에 해시 트리가 사용되는 경우에는 검증 데이터의 크기 및 트리의 일부 정보에 대한 부채널이 존재하게 되는데, 이로부터 특정 데이터에 대한 정보 수집 및 검증의 우회 가능성이 증가하게 된다. 이러한 부채널로 인한 검증의 유효성 문제를 개선하기 위하여, 본 논문에서는 멀티 셋 해시함수를 이용한 동일성 검증 기법을 제시한다.

ABSTRACT

Proliferation of cloud computing services brings about reduction of the maintenance and management costs by allowing data to be outsourced to a dedicated third-party remote storage. At the same time, the majority of storage service providers have adopted a data deduplication technique for efficient utilization of storage resources. When a hash tree is employed for duplicate identification as part of deduplication process, size information of the attested data and partial information about the tree can be deduced from eavesdropping. To mitigate such side channels, in this paper, a new duplicate identification method is presented by exploiting a multi-set hash function.

Keywords: side channel, deduplication, hash tree, message size, information leakage

I. 서 론

클라우드 컴퓨팅 등 인터넷에 연결된 사용자가 사용량에 따른 비용지불을 통하여 필요한 자원을 신속하고 유연하게 이용할 수 있는 환경이 보편화되면서, 개인 및 기업은 물론 공공기관에서도 원격저장소를 주 저장 공간으로 활용하는 추세가 확산되고 있다. 이와 동시에 기하급수적으로 증가하는 데이터의 효율적 활용을 위하여 다수의 데이터 저장 서비스 제공자는 데이터 중복제거 기법을 도입하고 있다 [1]. 데이터 중복제거에서는 중복된 데이터의 판단을 위하여 동일성 검증 과정을 거치게 되는데, 가장 원시적인 방법은 사용자가 아웃소싱하고자 하는 데이터 전부를 원격저장소에 업로드 하고 서비스 제공자는 자신이 관리하는 저장소에 동일한 데이터가 저장되어 있는지를 판단한 후 중복된 데이터가 확인되는 경우 해당

Received(06. 30. 2017), Modified(1st: 07. 19. 2017, 2nd: 08. 07. 2017), Accepted(08. 07. 2017)
* 본 연구는 한성대학교 교내학술연구비 지원과제 임
† 주저자, dykoo@hansung.ac.kr
‡ 교신저자, dykoo@hansung.ac.kr(Corresponding author)

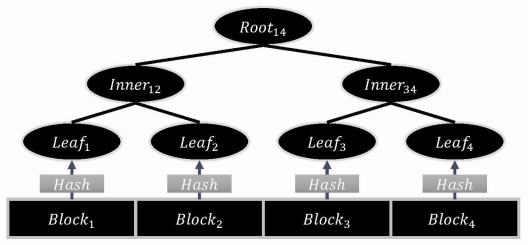

Fig. 1. Hash tree with 4 data blocks

데이터의 제거를 수행하는 것이다. 하지만 중복 데이터가 빈번히 업로드되는 경우에는 실제로 저장되지 않는 중복 데이터의 업로드로 인한 네트워크 대역폭의 낭비와 더불어 업로드된 데이터의 중복 (동일)성 여부 확인이 업로드 이후에 이루어짐에 따른 저장 공간 절감 효과의 지연 등을 보완하고자 해시 트리 [5] 등을 활용한 보다 효율적인 동일성 판단 기법이 다방면으로 연구되어 왔다.

1.1 해시 트리를 이용한 소유권 증명 (중복성 판단)

동일 저장소 내에서의 저장 공간 효율화를 위한 중복제거 기법은 1980년대 초반부터 지속적으로 연구되어 왔으나 [2,3], 클라우드 스토리지와 같은 원격저장소에서의 데이터 중복성 판단은 2000년대 들어 클라우드 컴퓨팅 도입과 함께 큰 관심을 받게 되었다. 원격 저장소에서의 중복제거에서는 저장 자원뿐 아니라 네트워크 자원의 효율적 활용이 중요시 되면서 해시함수 등을 이용하여 적은 통신량으로 중복성 판단을 수행하는 기법이 주로 활용되고 있다.

Halevi 등 [4]은 메시지의 고정된 단일 해시 값만으로 데이터의 중복성을 판단하게 되면, 데이터를 소유하지 않은 공격자라 할지라도 어렵지 않게 해당 해시 값 획득을 통하여 부정하게 소유권을 증명할 수 있는 문제점을 제시하면서, 해시 트리 [5]를 활용한 중복성 판단을 수행할 수 있는 방안을 제시하였다. Fig.1.과 같이 중복성 검증을 수행할 데이터는 동일 또는 가변 길이의 메시지 블록으로 분할되고 각 블록의 해시 값은 트리의 단말 노드에 할당된다. 이후 내부 노드는 단말 노드로부터 상향식으로 생성되는데, 좌측부터 이웃한 두 노드 값(의 결합)에 대한 해시 값을 부모 노드의 값으로 설정한다. 이와 같은 과정을 최상위 근 노드가 생성될 때까지 반복 수행함으로써 이진 (해시) 트리를 생성한다.

서비스 제공자가 사용자의 중복된 데이터에 대한

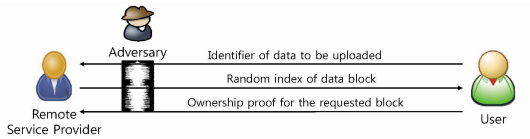

Fig. 2. Communication for ownership proof

소유권을 판단하기 위하여 해시 트리를 활용하는 방법은 Fig.2.와 같다. 먼저 사용자는 서비스 제공자에게 소유권을 증명하고자 하는 데이터의 식별자를 전송하고, 서비스 제공자는 해당 식별자에 대응되는 데이터를 저장하고 있는지 확인한다. 만약 동일 데이터가 원격저장소에 존재하는 경우에는 동일한 데이터를 반복하여 전달받을 필요가 없기 때문에 사용자의 소유권 증명을 요청하게 되는데, 이 때 데이터를 구성하는 임의 블록 번호를 사용자에게 전달하게 된다.

사용자는 요청받은 데이터 블록에 대응되는 소유권 증명 정보를 생성하게 되는데, 이는 해당 데이터 블록과 함께 해시 트리의 대응되는 단말 노드에서 근 노드에 이르는 경로 상에 위치한 노드들의 이웃 노드 값들로 이루어진다. 예를 들어, Fig.1.에서 $Block_1$에 대한 소유권 증명을 요청받은 사용자는 해시 트리를 생성한 후 $\{Block_1, Leaf_2, Inner_{34}\}$를 소유권 증명 정보로 서비스 제공자에게 전달한다.

서비스 제공자는 $Block_1$과 $Leaf_2$ 값으로부터 $Inner_{12}$에 해당하는 값을 계산하고, 이 계산 결과 값과 증명으로 전달받은 $Inner_{34}$의 값으로부터 $Root_{14}$에 대응되는 값을 계산한다. 서비스 제공자가 자신이 저장하고 있는 데이터에 대한 해시 트리의 근 노드 정보를 사전에 저장하고 있다고 가정했을 때, 앞서 사용자로부터 전달받은 정보로부터 계산된 결과 값이 저장된 값과 일치하는지를 판단함으로써 동일 데이터의 여부를 결정하게 된다.

II. 해시 트리 기반 소유권 증명에서의 부채널

해시 트리 기반 소유권 증명의 신뢰도 향상을 위해 서비스 제공자는 다수 데이터 블록 번호를 동시에 요청하고 그에 대응되는 소유권 증명 각각을 판단할 수 있는데, 소유권 증명이 평문 형태로 통신되는 환경에서는 Koo 등 [6]이 소개한 바와 같은 부채널이 존재하게 된다. 요약하면, 서비스 제공자와 사용자 간의 통신을 도청할 수 있는 공격자는 동일 데이터에 대한 소유권 판단을 위하여 해시 트리가 반복적으로

사용되는 경우, 트리의 부분 유출 정보를 누적하여 데이터를 소유하지 않고도 수집된 트리 정보만으로 해당 데이터에 대한 데이터의 소유권 획득 및 정보 열람이 가능해진다. 서비스 제공자가 전달한 데이터 블록에 대응되는 소유권 증명이 이미 수집된 해시 트리의 부분 정보인 경우, 공격자는 이를 재활용하여 소유권 증명을 회피할 수 있는 것이다.

2.1 해시 트리의 정보 유출 누적

해시 트리는 정적 해시함수를 이용하여 생성되기 때문에, 평문 형태로 통신되는 데이터를 공격자가 수집 및 누적하여 재활용할 수 있다.

서비스 제공자는 사용자의 소유권 증명을 위하여 임의의 데이터 블록 번호를 선택하게 되는데, 동일 데이터에서의 데이터 블록의 개수 (n라 하자)는 고정되어 있으므로 마지막 블록이 선택될 확률은 1/n이 된다. 앞선 예와 같이 4개의 블록으로 이루어진 데이터에서 최초로 블록 1에 대한 소유권 증명이 공개 채널을 통하여 전송된 경우 도청 가능한 공격자는 $\{Block1,\ Leaf_2,\ Inner_{34}\}$ 및 이로부터 계산 가능한 $\{Inner_{12},\ Root_{14}\}$의 값을 획득하게 되므로, 서비스 제공자가 다음 인증에서 $Block_1$을 선택하게 되면 공격자는 별도의 연산 없이도 소유권을 증명할 수 있게 된다. 대신, 서비스 제공자가 $Block_2$에 대한 소유권 증명을 요청하게 되면, 해당 블록 ($Block_2$)에 대응되는 값 (collision)만을 알게 되면 전체 데이터의 소유권을 얻을 수 있게 된다.[1] 따라서, 서비스 제공자는 이후 사용자의 업로드 요청에 대하여 $Block_3$ 또는 $Block_4$에 대한 소유권 증명을 요청하는 것이 바람직하며 이러한 과정이 반복될 때, 최대 블록의 개수에 대한 정보는 $O(n)$ 번의 중복 데이터 업로드 및 이에 대한 중복성 판단으로부터 획득 가능하게 된다.

데이터 크기에 대한 근사값은 소유권 증명의 크기와도 관련되는데, 4개의 블록으로 구성된 데이터의 예에서는 해시 트리의 높이가 2이기 때문에 2개의 해시값과 하나의 데이터 블록으로 소유권 증명이 구성되는 것을 알 수 있다. 엄밀히 말하면, 데이터 블록의 크기가 S_B이고 해시값의 크기가 S_H라 할 때, n개의 블록으로 구성된 데이터에 대한 소유권 증명의 크기는 $S_P = \lceil \log n \rceil S_H + S_B$가 된다. 역으로 도청가능한 공격자는 데이터 증명 크기 S_P로부터 증명하고자 하는 최소 데이터 크기 $2^{\lfloor S_P - S_B \rfloor} \cdot S_B$ 및 최대 데이터 크기 $2^{\lceil S_P - S_B \rceil}$를 알 수 있게 된다.

타깃 데이터의 크기 및 이진 해시 트리 정보를 습득한 공격자는 $O(n)$번의 도청으로부터 소유권을 획득할 수 있기 때문에, 해시 트리 기반의 소유권 증명 및 중복성 확인에서의 부채널은 중복제거 시스템의 신뢰성을 약화시키는 위험인자로 작용할 수 있다.

Table 1. Notations

notation	description
$\oplus$	bitwise exclusive-or (XOR)
h	collision-resistant hash function
K	secret key
M	multi-set data
B	set of unique elements in multi-set
B_i	i-th block of data (multi-set)
M_b	frequency of unique elements

III. 중복성 판단에서의 부채널 방지 기법

해시 트리를 이용한 중복성 판단 과정에서의 부채널 제거를 위하여 본 논문에서는 Clarke 등[7]이 소개한 멀티 셋 해시함수를 이용한 방법을 제시한다.

3.1 멀티 셋 해시함수 정리

멀티 셋은 동일한 원소가 한 번 이상 나타날 수 있는 순서가 정해지지 않은 유한 집합으로, [7]에서 제시된 네 가지 기법 중 가장 연산 효율적인 멀티 셋 해시함수 $H_K(\bullet)$는 다음과 같다 (사용된 용어는 Table 1에 정리한다):

$$H_K(M) = \left[\begin{matrix} h_K(0,r) \oplus \bigoplus_{b \in B} M_b \cdot h_K(1,b); \\ \sum_{b \in B} M_b \bmod 2^m; r \end{matrix} \right]_{r \leftarrow B}$$

위 멀티 셋 해시함수는 연산시마다 새로운 임의의

1) 물론 여기서 사용되는 해시함수는 충돌 회피 (collision resistance) 속성을 지니고 있지만, 전체 데이터에 대한 소유권 증명이 하나의 데이터 블록에 대한 소유권 증명과 동일시되는 점에서 바람직하지 않다.

값 r이 포함되는 확률적 (probabilistic) 알고리즘이기 때문에, 해시 트리에서의 결정론적 (deterministic) 알고리즘과 달리 생성되는 해시 값이 매번 달라진다. 따라서 해시 값의 중복성 판단을 위해서는 후처리 과정이 필요하게 된다. 동일 데이터 M에 대하여 생성된 서로 다른 두 해시 값을 각각 $H=(h_1,h_2,h_3)$와 $H'=(h'_1,h'_2,h'_3)$라 하면, $h_1\oplus h_K(0,h_3)$과 $h'_1\oplus h_K(0,h'_3)$이 같은 경우에만 두 해시 값은 동일하다고 간주된다.

3.2 멀티 셋 해시함수를 이용한 중복성 판단

앞선 멀티 셋 해시함수의 정의에서 해시 값의 두 번째 원소는 메시지 블록의 개수를 그대로 노출하게 되지만, [7]에서 언급된 바와 같이 데이터 크기를 숨기기 위한 목적에서 안전성 저하 없이 해당 원소를 생략할 수 있다. 또한 멀티 셋은 순서를 보장하지 않는다는 점에서 치환 공격 (replace attack) [8]에 대한 취약점을 가질 수 있으므로, 이에 대한 보완으로 두 번째 항의 배타적 논리합 연산을 수행하는 과정에서 고정된 값 1을 이용하는 대신 데이터 구성 순서에 따른 값을 할당할 수 있다. 즉, n개의 블록으로 구성된 데이터 $M=(B_0,B_1,...,B_{n-1})$의 중복성 판단 증명을 위한 해시함수는 다음과 같이 표현될 수 있다:

$$H_K=\left\{h_K(n,r)\oplus\bigoplus_{i=0}^{n-1}h_K(i,B_i);\ r\right\}.$$

위 연산에서는 멀티 셋에서 보장되지 않는 순서를 보존하기 위하여, 동일한 블록 B_i가 데이터 내에서 여러 번 발견되는 경우 반복되는 횟수를 곱하여 한번만 계산하는 대신 각각의 블록 번호와 결합하여 반복 횟수만큼 배타적 논리합 연산을 수행하도록 함으로써 데이터 블록의 순서 치환에 따른 공격을 방지한다.

또한, 비밀 키 (K)는 서비스 제공자와 올바른 데이터를 소유한 사용자가 공통으로 생성할 수 있되 공격자는 알 수 없는 값으로 설정하여야 하므로, 검증하고자 하는 데이터로부터 유도된 값이어야 함을 알 수 있다. 따라서 무열쇠 해시함수 (keyless hash function)를 이용한 $K=h(M)$으로 설정한다.

임의 값 r에 의하여 동일 데이터 M에 대한 서로 다른 해시 값 $H(M)=(h_1,h_2)$와 $H'(M)=(h'_1,h'_2)$의 중복성은 멀티 셋 해시함수의 속성을 그대로 활용하여 아래와 같이 검증한다:

$$h_1\oplus h_K(n,h_2)=\left(h_K(n,r)\oplus\bigoplus_{i=0}^{n-1}h_K(i,B_i)\right)\oplus h_K(n,r)$$
$$=h'_1\oplus h_K(n,h'_2)=\left(h_K(n,r')\oplus\bigoplus_{i=0}^{n-1}h_K(i,B_i)\right)\oplus h_K(n,r')$$
$$=\bigoplus_{i=0}^{n-1}h_K(i,B_i).$$

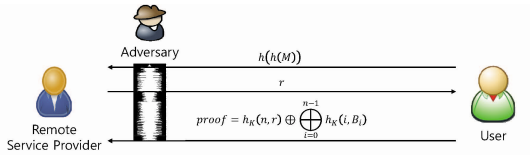

Fig. 3. Communication flow of proposed scheme

사용자의 데이터 식별자 선택에 있어서는 해시함수의 충돌 회피 속성 (collision resistance)을 활용하여, [9]에서와 같이 $h(h(M))$로 설정하여 서비스 제공자가 저장소 내 동일 데이터의 유무를 확인하도록 하며, 서비스 제공자는 동일 데이터 (식별자)가 저장소 내에서 발견될 경우 데이터 블록 번호 대신 해시 값에 사용될 임의 값 r을 선택하여 전송한다. 전송되는 데이터의 흐름은 Fig.3.과 같다.

IV. 성능 및 안전성 평가, 제안 기법의 특징

먼저 멀티 셋 해시함수를 변형한 중복성 식별 제안 기법의 특징은 아래와 같이 요약할 수 있다:

1. 해시 트리 기반의 소유권 증명에서 제공하는 기능을 동일하게 제공한다. 해시 트리는 하나의 데이터를 일련의 블록으로 분할하여 전체 데이터가 완전히 일치하는지를 무시할 수 없을 정도의 높은 확률로 판단하는데 활용되며, 멀티 셋 해시함수를 활용한 제안 기법에서도 이와 동일하게 전체 데이터의 중복성을 높은 확률로 판단할 수 있다 [7].

2. 해시 트리 기반의 중복성 판단 과정에서 발생하는 부채널을 제거한다. 데이터의 크기 및 해시 트리에 대한 부분 유출 정보의 누적으로부터 도청가능한 공격자가 정당한 권리 없이 소유권을 획득하는 것을 방지할 수 있다.

3. 해시 트리 기반 중복성 판단 기법에 비하여 통신 및 연산 효율성을 향상시킨다.

Table 2. Communication cost comparison

Primitive	Service provider	User
Hash tree [4]	S_N	$S_B + \lceil \log n \rceil \cdot S_H$
Multi-set hash	S_H	S_H

Table 3. Computation cost comparison

Primitive	Proof generation	Proof validation
Hash tree [4]	$O(n)(C_H + C_{Con})$	$O(\log n)(C_H + C_{Con})$
Multi-set hash	$O(n)(C_H + C_{XOR})$	$O(1)(C_H + C_{XOR})$

4.1 안전성 분석

멀티 셋 해시함수는 모든 구성 원소의 배타적 논리합으로 계산되며 임의 값 r을 통하여 그 연산 결과가 계속 달라지기 때문에 도청가능한 공격자라 할지라도 동일한 값을 그대로 사용하게 되면 서비스 제공자가 이를 용이하게 탐지할 수 있어, 재사용에 의한 소유권 증명의 우회 가능성을 제한할 수 있다. 제안 기법에서는 서비스 제공자가 이 임의 값 (r)을 선택하도록 함으로써 공격자의 수집 정보 재사용을 통한 부당한 소유권 획득을 차단한다 [10].

또한, 항상 동일하게 연산되는 부분 ($\oplus_{i=1}^{n-1} h_K(i, B_i)$)은 매번 상이한 임의 값 r을 이용하여 ($h_K(n,r)$) 난독화되기 때문에, 비밀 키 K를 소유하지 못한 도청가능한 공격자가 r로부터 $\oplus_{i=1}^{n-1} h_K(i, B_i)$에 대한 계산을 다항 시간 (polynomial time) 내에 수행하기는 불가능하다.[2)]

4.2 멀티 셋 해시함수의 통신 및 연산 효율성

먼저 네트워크 대역폭 사용에 따른 효율성을 살펴보도록 한다. 각각의 데이터 블록에 대한 해시 값을 계층적으로 형성하는 해시 트리와 달리, 멀티 셋 해시함수는 다수의 블록에 대한 해시 값을 하나로 표현할 수 있기 때문에, 데이터 크기에 영향을 받지 않고 오직 하나의 값만을 생성할 수 있다. 이는 사용자가 생성하는 소유권 증명의 크기가 메시지 블록 개수의 로그에 비례하는 해시 트리 기법에서 유추할 수 있는 데이터 크기에 대한 정보를 공격자로부터 차단할 수 있을 뿐 아니라 네트워크 대역폭의 절감 효과를 얻을 수 있음을 의미한다.

$S_N = \{0,1\}^{\lceil \log n \rceil}$ 을 0에서 n까지 표현할 수 있는 이진비트열 길이라 하고, $S_H = |h|$ 를 암호학적 해시함수의 비트 길이 (예를 들어, SHA-256의 경우 256), $S_B = |B_i|$ 를 데이터 블록 B_i의 비트 길이, 그리고 임의 값 r의 크기가 S_H라 할 때, 서비스 제공자와 사용자의 통신 (송신) 비용은 Table 2와 같다. 서비스 제공자는 해시 트리 기반 중복성 판단을 위하여 1에서 n 사이의 값을 전달하는데 반해 제안 기법에서는 임의 값 r을 전달하기 때문에 송신비용이 상대적으로 크다고 할 수 있으나, 두 경우 모두 한 번의 값 전송만 이루어지기 때문에 그 크기의 차이에 의한 성능 저하는 미미하다고 볼 수 있다. 반면에 사용자가 전달해야하는 증명 데이터의 크기를 살펴볼 때, 해시 트리 기반의 기법에서는 트리의 높이에 비례하여 전송해야하는 정보의 양이 증가하는 것에 비하여 제안 기법에서는 서비스 제공자와 마찬가지로 하나의 일정한 (해시) 값만을 전달하기 때문에 통신비용의 절감 효과가 탁월하다. 여기서, 해시 트리 기반의 접근법에서 전송되는 데이터 블록의 크기 S_B는 일반적으로 해시 값의 크기 S_H보다 매우 큰 값이기 때문에 사용자의 통신비용은 제안 기법이 월등히 우수함을 확인할 수 있다.

멀티 셋 해시함수를 응용한 제안 기법 및 해시 트리 기반의 기법에 대한 연산 소요 비용은 Table 3과 같이 정리할 수 있다. 여기서 C_H는 암호학적 해시함수 (SHA 등)의 연산에 소요되는 비용을 의미하고, C_{XOR}은 두 비트열의 비트단위 배타적 논리합 (bitwise exclusive or) 연산에 소요되는 비용을 의미하며 C_{Con}은 두 비트열의 결합 (concatenation)에 소요되는 비용을 가리킨다. Tobin과 Malone이 수행한 연산 소요 시간 비교 [11]에 따르면, 단순 비트열의 배타적 논리합 또는 비트열 결합에 소요되는 비용은 암호학적 해시함수를 연산하는데 소요되는 비용의 수십 분의 1에 불과하

2) 또한, 멀티 셋 해시함수를 이용하면 원격저장소의 데이터 일부가 빈번히 변경되는 동적 환경 (예를 들어, 다수가 참여하는 공동 프로젝트에서 생성된 데이터 일부에 대한 수시 변경/삽입/삭제에 대한 백업)에서 변경된 블록에 대한 해시 연산의 갱신만으로 전체 데이터에 대한 중복성 판단이 보다 유연하게 적용될 수 있을 것이다.

기 때문에 해시함수 여기서는 연산 횟수에 따른 효율성을 비교하도록 한다.

멀티 셋 해시함수를 활용한 제안 기법에서는 증명 데이터가 모든 구성 원소의 해시 값에 대한 배타적 논리합으로 계산되기 $(n+1)$번의 해시 연산과 n번의 배타적 논리합으로 이루어지며, 서비스 제공자가 수행하는 증명 데이터의 검증은 한 번의 해시 연산과 한 번의 배타적 논리합 연산을 통하여 임의 값 r로부터 생성된 마스킹 (masking) 값을 제거하는 것으로 볼 수 있다. 반면에, 해시 트리 기반 기법에서 증명 데이터를 생성하기 위해서는 트리 형성을 위하여 $O(n)$번의 해시 연산 및 비트열 결합을 필요로 하게 된다. 트리로부터 추출된 증명데이터의 검증 과정에서도 트리의 높이에 대응되는 일련의 해시 값들에 대한 해시 연산과 결합이 필요하게 되며 이는 데이터 블록의 개수에 영향을 받는 것을 의미한다. 따라서 증명 데이터의 생성에서는 해시 트리와 멀티 셋 해시 함수를 이용한 기법에서 연산 소요 비용의 차이가 미미하다고 볼 수 있지만, 검증 과정에서는 데이터 블록의 개수와 무관하게 한 번의 해시 연산 및 배타적 논리합 연산을 수행하는 제안 기법이 보다 효율적임을 알 수 있다.[3)]

네트워크 대역폭 및 검증 연산 소요량에서 살펴본 바와 같이, 제안 기법이 데이터 블록 개수에 무관하게 항상 일정한 통신 및 연산 비용을 필요로 하기 때문에 기존 해시 트리 기반의 중복성 판단에 비하여 효율성이 현저히 향상됨을 확인하였다. 또한, 증명 데이터의 생성 과정에서 데이터의 중복성을 증명하는 사용자는 모든 데이터를 활용하여 전체 데이터에 대한 정보를 요약한다는 측면에서 데이터 블록의 개수에 비례한 연산이 필요하지만 이는 기존 연구와 동일한 수준의 연산량을 요구하며 성능 하락을 유발하지는 않는다.

V. 결 론

원격저장 서비스와 데이터 중복제거 기술의 보편적 활용 과정에서 동일 데이터의 중복성 판단 시 이전에 고려되지 않았던 부채널이 발견되고 있고, 이는 전체 시스템의 안정성을 해치는 위험 요소가 되었다. 특히, 해시 트리를 이용한 중복성 판단 과정에서 노출되는 데이터 크기 및 트리의 부분 정보는 전체 데이터를 소유하지 않은 도청 공격자에게 데이터의 소유권을 부정한 방법으로 획득할 수 있는 기회를 제공하므로 이에 대한 대응책을 살펴보았다. 멀티 셋 해시 트리를 활용하여 데이터의 크기 정보를 숨길 수 있을 뿐 아니라, 난수를 활용함으로써 공격자의 재사용 공격을 차단하여 부채널을 통한 공격자의 정보 수집 능력을 제한할 수 있었다. 또한 효율성 분석을 통하여 제안 기법의 실용성을 검증하였다.

3) 관련 연구 [6]은 기존 해시 트리에 대한 변형으로, 데이터 크기 및 트리 노드의 값을 감추기 위하여 생성된 증명 데이터 각 원소에 대한 추가적인 배타적 논리합과 해시 연산을 요한다. 이러한 상수배의 추가 연산은 big-O 표기법 상에서 확인이 어려우며, 검증 데이터 길이는 서비스 제공자가 요구하는 바에 따라 가변적으로 변하므로 통신 및 연산 효율성에서의 비교에서는 제외하였다.

References

[1] D.T. Meyer and W.J. Bolosky, "A study of practical deduplication," ACM Transactions on Storage, vol. 7, no. 4, pp. 14:1-14:20, Jan. 2012

[2] Udi Manber, "A probabilistic lower bound for checking disjointness of sets," Information Processing Letters, vol. 19, no.1, pp. 51-53, Jul. 1984

[3] K.V.S. Ramarao, Robert Daley, and Rami Melhem, "Message complexity of the set intersection problem," Information Processing Letters, vol. 27, no. 4, pp. 169-174, Apr. 1988

[4] Shai Halevi, Denny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg, "Proofs of ownership in remote storage systems," Proceedings of the 18th ACM conference on Computer and Communications Security, pp. 491-500, Oct. 2011

[5] R.C. Merkle, "A digital signature based on a conventional encryption function," Advances in Cryptology, CRYPTO'87, LNCS 293, pp. 359-378, 1988

[6] Dongyoung Koo, Youngjoo Shin, Joobeom Yun, and Junbeom Hur, "An online data-oriented authentication based on Merkle tree with improved reliability," Proceedings of the 2017 IEEE International Confere

nce on Web Services, pp. 840-843, Jun. 2017
〔7〕 Dwaine Clarke, Srinivas Devadas, Marten van Dijk, Blaise Gassend, and G.E. Suh, "Incremental multiset hash functions and their applications to memory integrity checking," Advances in Cryptology, ASIACRYPT'03, LNCS 2894, pp. 188-207, Dec. 2003
〔8〕 Kan Yang and Xiaohua Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 9, pp. 1717-1726, Sep. 2013
〔9〕 Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart, "DupLESS: server-aided encryption for deduplicated storage," Proceedings of the 22nd USENIX conference on Security, pp. 179-194, Aug. 2013
〔10〕 S.R. Lohstroh and David Grawrock, "Method for providing a secure non-reusable one-time password," US5768373 A (US Patent), Symantec Corporation, Jun. 1998
〔11〕 R.J. Tobin and David Malone, "Hash pile ups: using collisions to identify unknown hash functions," Proceedings of the 2012 7th International Conference on Risk and Security of Internet and Systems, pp. 1-6. Oct. 2012

〈저자소개〉

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/엣지/포그 컴퓨팅 보안