정 보 보 호 학 회 지
제 31 권 제2호, 2021. 4

안전한 데이터 중복 처리 기술 연구 동향

윤 택 영*

요 약

중복 제거 기술(Deduplication)은 동일한 데이터에 대하여 중복 저장을 방지함으로써 다수의 클라이언트가 공용으로 사용하는 서버의 저장 성능을 향상하기 위한 기술이다. IT 서비스에서 사용되는 데이터의 크기 및 양이 증대함에 따라 중복 제거 기술의 중요성은 더욱 높아지고 있다. 모든 사용자를 신뢰할 수 있는 환경에서는 다수의 사용자가 동일 데이터를 저장하고자 하는 경우, 외부 서버에 하나의 데이터만 저장하고 반복된 데이터 저장 요청에 대해서는 해당 데이터에 대한 소유권만 인정하는 방식으로 중복 데이터 관리의 효율성을 높일 수 있다. 그러나 다양한 목적으로 악의적인 행위를 수행하는 공격자에 의해 특정 시스템의 작은 취약점도 해당 시스템 기반으로 제공되는 서비스들의 안전성을 훼손하기 위해 악용될 수 있고, 중복 제거 기술도 이러한 위협에 노출되어 있다. 본 논문에서는 중복 제거 기술을 대상으로 알려져있는 공격 방법에 대해 살펴보고, 안전하게 데이터 중복 처리를 제공하기 위한 요소 기술 및 관련 기술 동향에 대해 소개하고자 한다.

Ⅰ. 서 론

클라우드 스토리지 서비스 제공자들은 스토리지 서버에 저장된 중복된 데이터를 모두 저장하지 않고 대표로 한 개만 저장하는 중복 처리 기술을 적용하여 저장 공간을 효율적으로 사용할 수 있다. 즉, 다수의 사용자가 동일한 데이터를 같은 스토리지 서버에 저장하는 경우 스토리지 서버는 중복된 데이터가 존재하는지 확인하고 동일 데이터의 경우 단 한 개만 저장하는 방법으로 스토리지 효율성을 증대시킨다. 물론, 다수의 사용자에게 동일 데이터에 대한 접근을 허용하기 위해 타당한 데이터 소유자들에게 이미 저장된 데이터에 대한 접근을 허용하기 위한 부가적인 기술이 요구된다. 각 사용자에게 소유권을 부여하고 적절한 효율적 검증 기술을 활용하여 각 사용자의 서비스 요청에 따라 저장된 데이터 기반의 서비스를 제공해야 한다. 이와 같은 부가적인 비용에도 불구하고 중복된 데이터가 많이 발생하는 환경에서는 효율적으로 스토리지 서비스를 제공할 수 있다. 이는 데이터의 크기가 커지고 있는 요즘 IT 환경에서 매우 중요한 기술로 여겨지고 있다. 이와 같은 요구에 따라, 국내에서도 안전하게 중복된 데이터를 처리하기 위한 기술에 대한 연구가 진행되었다 [1-4].

데이터 중복 처리 기술의 경우 스토리지를 효율적으로 관리하기 위한 기술로만 여겨져 왔다. 그러나 스토리지 서비스에서 다루어지는 데이터로 인한 다양한 보안 위협이 알려짐에 따라 중복 처리 기술에 대한 보안 기술에 대해서도 관심이 크게 증가하였다. 데이터 중복 처리 기술에 대한 보안 요구사항은 보안 위협에 대한 분석에서 시작된다. 각 사용자들은 다음과 같은 위협에 대해 대응할 수 있는 기술을 필요로 한다.

클라이언트 입장에서는 자신이 스토리지 서버에 저장한 데이터에 대한 기밀성이 가장 중요한 보안 요구사항이 된다. 즉, 자신의 데이터가 외부에 드러나지 않는 것이 가장 중요한 보안 요구사항이 된다. 기밀성을 위해 클라이언트는 데이터를 스토리지 서버에 저장하기 전에 암호화하여 암호문의 형태로 클라우드에 저장할 수 있다. 이 경우, 일반적인 암호화 기술의 상용법의 경우 사용자들 마다 서로 다른 임의의 키를 사용하기 때문에 각 사용자들은 동일한 데이터에 대해 서로 다른 암호문을 생성하게 된다. 즉, 데이터 중복 제거에서는 중복된 데이터들을 확인해야 하는 것과는 달리 안전한 암호화 알고리즘은 중복된 데이터에 대해 구별 불가능한 암호문들을 생성하므로 기밀성을 위한 암호화 기술을 사용하면 클라우드 저장 공간의 효율성을 높이기 위해 중복 처리 기술을 적용하기 어렵다. 데이터 기밀성의 보장과 저장 공간의 효율성을 높이기 위해 암호화된 데이터에

* 단국대학교 산업보안학과 (조교수, taekyoung@dankook.ac.kr)

중복제거 기법을 적용시키기는 방안으로 데이터에서 유도된 키를 기반으로 암호화하는 기술인 MLE (Message-Locked Encryption)이 개발되었다 [9]. MLE는 특정 기법의 이름이기 보다는 데이터를 기반으로 키를 생성하고 이를 기반으로 암호화를 수행하는 기술들에 대한 일반적인 이름이다. MLE가 학계의 관심을 받기 이전에 CE(Coverenct Encryption)라는 이름으로 암호 데이터에 대한 중복 처리 기술이 처음 소개되었으나 MLE가 발표되면서 암호 데이터에 대한 중복 처리 기술에 대한 연구가 체계적으로 진행되었다. 이후, 데이터에서 고정된 키를 생성하는 방식이 가지는 구조가 전수조사 공격에 취약하다는 분석에 따라 해당 공격에 안전성을 향상시키기 위한 연구가 진행되었다 [5].

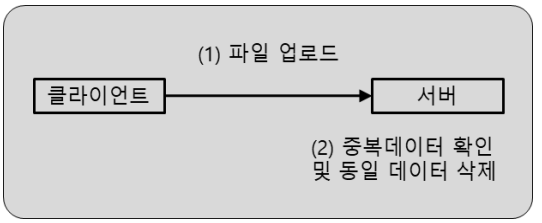
서버의 경우, 동일한 데이터들이 다수 존재하는 경우 하나의 파일만 저장하여 스토리지 효율성은 높일 수 있으나 해당 데이터에 대해 권한이 있는 사용자들을 정확하게 식별하는 것이 중요하게 되었다. 서버 중심으로 수행되는 중복 처리의 경우 적법한 사용자에 대한 리스트를 안전하게 관리하면 되지만 클라이언트와 서버가 동적으로 프로토콜을 수행하여 중복 데이터를 제거하는 기술의 경우 사용자가 실제로 데이터를 업로드 하지 않는 방식이기에 실제 데이터 보유 여부를 정확하게 검증할 수 있는 기술이 필수적으로 요구된다. 이러한 목적으로, 소유권 증명 기술인 PoW이 제안되었다 [6]. PoW와 유사한 기술로 PoR이라는 기술이 제안되었다 [7]. 두 기술은 특정 데이터를 보유하고 있는지 확인하기 위한 기술이라는 측면에서 여러모로 유사한 특징을 가진다. 그러나 각각 클라이언트와 서버가 특정 데이터를 소유하고 있음을 증명해야 하고 서버와 클라이언트가 가지는 환경적인 차이점에 의해 PoW는 PoR과 별도의 기술로 연구되고 있다.

본 고에서는 상기에서 기술한 스토리지 서비스에서의 안전한 중복 처리를 위한 기술들을 소개한다. 2장에서는 중복 처리 방식에 대해 간단히 설명하고, 3장에서는 중복 처리 기술에 대한 공격 방법을 소개한다. 4장에서는 안전한 중복 처리를 위한 기술 동향을 소개하고 5장에서 본 고를 마친다.

Ⅱ. 중복 처리 기술의 종류

스토리지의 효율적인 사용을 위한 중복 데이터 제거 기술은 중복 데이터 제거를 수행하는 주체에 따라 서버측 중복 제거 기술과 클라이언트측 중복 제거 기술로 구분된다.

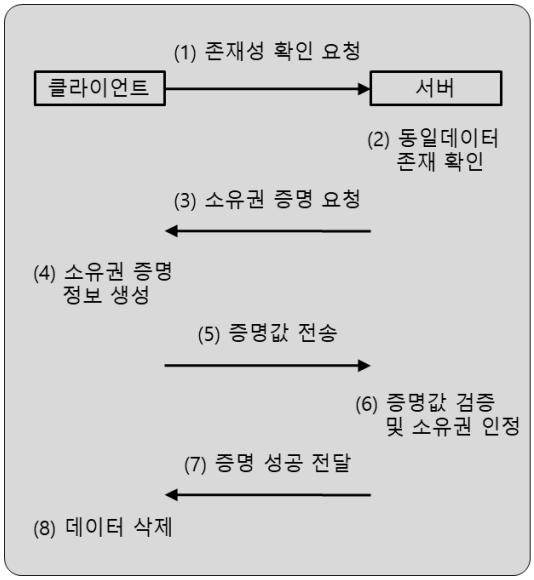
- 서버측 중복 제거 기술
  - 스토리지 서비스 제공자는 다수의 클라이언트로부터 데이터를 수신한다. 주기적으로 동일한 데이터의 경우 중복된 데이터를 제거하는 방식으로 중복 처리를 수행한다. 동일 데이터는 제거하기에 스토리지 저장 성능을 높일 수 있으나 모든 클라이언트가 서버에 중복된 데이터도 업로드하므로 통신 성능을 높일 수는 없다.



〔그림 1〕 서버측 중복 처리 기술 동작도

- 클라이언트측 중복 제거 기술
  - 중복 데이터를 제거하는 과정에 클라이언트도 참여하는 방식이다. 클라이언트와 서버는 프로토콜의 수행을 통해 업로드하려는 데이터가 서버의 스토리지에 있는 경우에는 실제로 데이터 업로드를 수행하지 않고 해당 클라이언트의 보유 여부만 확인한 뒤 해당 데이터에 대한 소유권을 부여하는 방식이다. 동일 데이터는 중복되어 저장되지 않으므로 스토리지 성능을 향상시키 수 있으며 동일 데이터가 업로드 되기 위해 전송되지 않으므로 통신 비용 또한 절감할 수 있다. 전체 데이터를 업로드 하지 않고 작은 값으로 소유권을 확인해야하기에 클라이언트의 데이터 보유 여부를 정확하게 확인해야 하는 것이 중요한 보안 요구사항으로 부각된다.

중복 처리를 수행하는 주체에 따른 구분 외에도 블록 처리 단위에 따른 구분도 존재한다. 파일 자체의 동일 여부를 확인하는 파일 단위 중복처리 기술 외에도 파일을 구성하는 블록 단위로 중복 데이터를 확인하고 제거



〔그림 2〕 클라이언트측 중복 처리 기술 동작도

하는 블록 단위 중복 제거 방식이 존재한다. 그러나 블록 단위의 중복 데이터를 관리하기 위해서는 부가 비용이 발생한다. 일반적으로는 블록 단위의 중복 데이터가 발생하지 않으나, 지속적인 작업 과정에서 동일 파일에 대해 생성되는 다수의 버전 파일의 경우 상당 부분의 블록 내지는 구간이 중복되기에 블록단위 중복 처리를 적용하는 경우 상당 부분의 저장 공간을 절약할 수 있다.

Ⅲ. 중복 처리 기술에 대한 공격 방법

본 단원에서는 중복 처리 기술에 대한 공격 방법에 대해 살펴보고자 한다. 동작 방식이 상이한 서버측 중복 처리 기술과 클라이언트측 중복 처리 기술의 안전성에 대해 3.1과 3.2장에서 각각 분석하고, 중복 처리 기술에 적용 가능한 공격 모델을 3.3에서 정리하고자 한다.

3.1. 서버측 중복 처리 기술의 안전성

서버측 중복 처리 기술의 경우, 동일한 데이터를 제거하는 방식이 간단하기에 클라이언트측 중복 처리 기술에 비해 공격에 안전하다는 장점을 가진다. 그러나, 기본적으로 중복 처리를 위해 동일한 데이터를 식별하기 위해서는 동일 데이터에 대해 동일한 암호문이 생성되는 특성이 제공되어야 한다. 이러한 기술적인 한계 내지는 근본적인 특성에 의해 메시지에서 유도되는 키를 암호화를 위해 사용하는 것은 피할 수 없는 구성상의 특성이다. 메시지에서 유도된 키의 경우, 키의 복잡도가 키를 생성하기 위해 사용한 평문 데이터의 복잡도에 의해 결정된다. 즉, 낮은 복잡도를 가지는 평문의 경우 낮은 복잡도를 가지는 키로 암호화 된다는 취약점을 가지게 된다.

3.2. 클라이언트측 중복 처리 기술의 안전성

서버측 중복 처리 기술과 클라이언트측 중복 처리 기술의 가장 큰 차이점은 중복 데이터의 존재 여부를 확인하기 위한 소유권 확인 절차가 존재하는지 여부이다. 서버측 중복 처리 기술에서는 동일 파일의 존재 여부에 상관 없이 사용자가 업로드 한 뒤에 서버가 동일 데이터를 제거하는 방식이기에 동일 데이터 존재 여부를 확인하지 않더라도 중복 처리 기능을 제공할 수 있다. 그러나 클라이언트측 중복 처리 기술은 데이터 업로드 이전에 동일 데이터 존재 여부를 확인한다. 그 뒤에 해당 데이터가 존재하는 경우만 업로드 동작의 수행 없이 해당 클라이언트의 동일 데이터 소유여부 검증 후 소유권을 인정하는 방식이므로 사용자의 소유권 증명이 매우 중요한 절차가 된다. 이러한 구조적 차이에서 발생하는 추가 비용은 존재 여부를 확인하기 위한 검증 절차, 그리고 해당 데이터를 실제로 보유하고 있는지 검증하기 위한 소유권 증명 비용이다. 이러한 구조적인 차이에 의해 새로운 취약점이 발생한다. 동일 데이터를 확인하기 위한 프로토콜의 특성에 의해 동일 데이터에 대한 소유자 식별이 가능한 파일 신원 확인 공격(identification attacks)이나 실제 데이터를 보유하지 않으면서 소유권을 획득하고자 하는 형태의 공격이 가능하다.

3.3. 공격 모델

3.1장과 3.2장에서는 서버측 중복 처리 기술과 클라이언트측 중복 처리 기술에 대해 공격자가 시도할 수 있는 공격 가능성에 대해 기술하였다. 이러한 공격 가능성에 대한 고찰을 기반으로 기존에 연구된 공격 모델은 다음과 같이 정리할 수 있다.

• 데이터 프라이버시 훼손 공격 - 중복 데이터 처리 과

정에서 동일 데이터 확인을 위해 암호화를 적용하지 않고 평문 상태의 데이터를 외부 클라우드 스토리지에 저장하는 경우, 특정 사용자가 보유한 데이터에 대한 프라이버시 침해가 발생할 수 있다. 이러한 형태의 공격은 안전한 중복 처리를 위해 아무런 보안 대책을 적용하지 않은 경우에만 발생한다.

- 데이터 오염 공격(Poison attacks) - 중복 처리 과정에서 발생 가능한 가장 큰 위협이다. 데이터 중복 처리 과정에 동일한 파일이 다수 존재하는 경우 하나만 제외하고 나머지는 제거하기 때문에 기존에 저장되어 있던 데이터가 훼손된 데이터임에도 불구하고 서버가 동일 데이터로 판단하고 새로운 데이터 업로드를 받지 않는 경우 많은 사용자들이 본인의 자산에 해당하는 데이터를 잃을 수 있다. 즉, 다른 데이터와 동일한 데이터로 오인할 수 있는 가치 없는 데이터를 서버에 저장하고 이럴 실제 사용자들의 데이터 대신 저장되도록 함으로써 가치 있는 데이터가 저장되지 않는 문제를 야기할 수 있다.

- 사전 공격(Dictionary attacks) - 이는, 암호데이터에 대한 중복 처리를 위해 데이터에서 유도된 키를 기반으로 암호화를 수행하는 경우 암호화를 사용한 목적이 무력화 되도록 할 수 있는 공격이다. 데이터 중복 처리 시 데이터에서 유도된 키를 사용하는 가장 근본적인 이유는 동일 데이터에 대해 동일한 암호문을 생성하기 위해서이다. 이 과정에서 사용되는 데이터를 기반으로 계산되기 때문에 근본적으로 암호화를 위한 키가 가지는 랜덤성이 평문이 가지는 랜덤성보다 클 수 없다는 정보량 측면의 한계를 가진다. 즉, 낮은 엔트로피를 가지는 평문의 경우 평문 추측을 통한 키 예측이 가능하고, 평문의 엔트로피에 따라 이러한 예측 공격의 난이도가 매우 낮을 수 있다. 이러한 경우 암호데이터 중복 처리 기술이 적용된 데이터들은 사전 공격에 매우 취약하며, 암호화 키의 유추가 쉽다는 약점은 다른 공격을 수행하기 위한 정보로 활용될 수 있다.

- 데이터 소유자 신원 확인 공격(Identification attacks) - 중복 처리 과정에서 발생 가능한 가장 큰 위협이다. 데이터 중복 처리 과정에 동일한 파일이 다수 존재하는 경우 하나만 제외하고 나머지는 제거하기 때문에 기존에 저장되어 있던 데이터가 훼손된 데이터임에도 불구하고 서버가 동일 데이터로 판단하고 새로운 데이터 업로드를 받지 않는 경우 많은 사용자들이 본인의 자산에 해당하는 데이터를 잃을 수 있다. 즉, 다른 데이터와 동일한 데이터로 오인할 수 있는 가치 없는 데이터를 서버에 저장하고 이럴 실제 사용자들의 데이터 대신 저장되도록 함으로써 가치 있는 데이터가 저장되지 않는 문제를 야기할 수 있다.

Ⅳ. 안전한 중복 처리를 위한 기술 동향

본 장에서는 3장에서 기술된 위협에 대응하기 위해 제안된 기술을 정리하고 관련된 연구 동향에 대해 소개한다.

4.1. 기밀성 제공 기술

기밀성을 제공하기 위해 일반적으로 사용되는 기본적인 도구는 암호화 기법이다. 그런데 데이터 중복 처리가 필요한 응용 환경에서는 일반적인 암호화 기법의 사용은 적합하지 않다. 동일 평문에 다른 암호문이 생성되는 것이 기본적인 특성인데 동일 평문에서 동일 암호문이 생성되어야 중복 처리가 가능하기 때문이다. 동일 평문에 동일 암호문을 생성하기 위한 기반 기술로 MLE가 제안되었다 [9]. MLE는 평문에서 유도된 키를 사용하여 암호화를 수행하는 방식으로 동일 평문에서 동일 암호문을 생성하는 기술이다. 데이터를 해쉬한 값과 같이 동일 데이터에서 동일한 값이 계산되는 일방향 함수를 사용하여 비밀키를 생성하므로 동일 평문에서 동일 암호문이 생성되는 것을 보장할 수 있다. 그런데, 평문이 가지는 복잡도가 낮은 경우에는 평문을 예측하고 이를 기반으로 암호화키를 생성하여 암호문을 만들 수 있기에, 공격자는 평문을 추측하여 맞추는 일종의 사전공격을 시도할 수 있다. 사전공격에 대한 내용은 아래에서 별도로 설명한다.

4.2. 사전공격 대응 기술

상기 기술한 바와 같이 MLE는 동일 평문에서 동일

암호문이 생성된다는 특성을 제공하기 위해 낮은 복잡도를 가지는 평문의 경우 안전성을 보장하기 어렵다는 약점을 동시에 가진다. 이러한 약점을 극복하기 위한 연구가 다수 진행되었다. 대표적인 연구 결과로는 Dupless를 들 수 있다 [5]. Dupless는 키를 데이터를 가진 모든 사용자가 직접 계산하는 점에서 사전 공격이 수행된다는 점에 착안하여 키 생성 함수를 키 생성 서버로 대체하는 구성으로 변경하였다. 특정 데이터에 대한 키를 생성하기 위해 클라이언트와 서버는 일종의 블라인드 서명을 생성하기 위한 프로토콜을 수행한다. 결과적으로, 서버는 클라이언트가 서명을 만들고자 하는 데이터를 알지 못하면서도 해당 데이터에 대한 서명을 생성한다. 서명 생성을 위해 알고리즘으로 결정적 함수(deterministic function)을 사용하기에 동일 평문에 대해 동일 키가 생성되는 것은 보장한다. 그러나, 키를 생성하기 위해 키 생성 서버와 통신을 수행해야 하기에 임의의 데이터에 대한 키를 생성할 수 있는 능력을 제한하는 효과를 제공한다.

사전 공격에 대한 대응 기술로 제안된 Dupless는 낮은 복잡도를 가지는 파일에 대해 충분한 안전성을 제공할 수 있다. 그러나, 키 생성 서버가 공격자이거나 키 생성 서버와 공모하는 공격자의 경우 쉽게 사전 공격을 시도할 수 있다. 이러한 취약점에 대응하기 위해 비밀분산 등의 기반 기술을 활용하여 위험은 경감시킬 수 있다. 그러나 이와 같이 다수의 신뢰할 수 있는 제 3자(trusted third party)를 고려하는 것은 상대적으로 큰 비용을 야기할 수 있다.

4.3. 중복 처리 기술에 대한 부채널 공격 대응 기술

데이터 중복 처리 기술 중에서 서버에 저장하고자 하는 파일과 동일한 파일이 존재하는지 확인하고 동일 파일이 있는 경우만 소유권 증명을 수행하는 클라이언트측 중복 처리 기술의 경우, 동일 파일의 존재 여부를 확인하기 위한 과정에 필수 불가결하게 동일 파일의 존재성에 대한 정보를 노출해야한다. 이 과정에서 해당 파일을 가장 먼저 업로드한 사용자에 대한 파일 소유자 프라이버시가 훼손될 수 있다. 이와 같은 공격을 파일 소유자 신원 공격(identification attack)이라 한다. 파일 소유자 신원 공격은 기본적으로 중복 처리 기술에서 발생하는 부채널 정보를 활용하는 것인데, 이는 클라이언트측 중복 제거 기법이 가지는 기본적인 구성상 특징을 기반으로 하고 있다. 클라이언트측 중복 처리 기법의 경우 클라이언트가 업로드 하려는 파일과 동일 파일이 있는 경우와 그렇지 않은 경우로 나뉘어 동작하는데 이와 같은 조건부 동작으로 인해 특정 데이터 존재 여부에 대한 부수적인 정보가 노출된다. [10]에서는 이와 같은 부채널 공격 방법에 대한 체계적인 분석과 함께 효과적인 대응 기술을 제시하고 있다. 지금까지 알려진 대응 기술로는 서버가 각 파일마다 임의의 횟수만큼 중복처리를 수행하지 않는 방법이다. 이는 서버 중심의 대응 기술로 중복 처리를 수행하지 않는 횟수가 큰 경우 안전성은 높아지지만 중복 처리로 인한 성능 향상은 감소하는 단점이 있다. [10]에서는 이러한 단점을 극복하기 위해 각 클라이언트가 안전성을 위한 설정을 하고 설정된 시간 이후에만 중복 처리가 가능한 형태로 저장할 수 있는 기술을 제안하였다. 즉, 데이터의 소유주가 안전성 관련된 설정을 결정할 수 있는 클라이언트 중심의 대응 기술이 [10]에서 제안되었다.

4.4. 증명 가능한 데이터 소유권 확인 기술

중복 처리 기술은 클라이언트의 참여 여부에 따라 서버측 중복 처리 기술과 클라이어언트측 중복 처리 기술로 나누어진다. 이중, 클라이언트 중복 처리 기술의 경우 데이터를 업로드하려는 클라이언트가 실제 데이터를 소유하고 있는지만 확인한 뒤에 서버에 저장된 동일 데이터에 대한 권한을 인정하는 대신 실제 데이터 업로드를 수행하지 않는 방식으로 동작하고 있다. 이로 인해 실제로 데이터를 보유하고 있는지 확인하는 소유권 증명의 중요성이 매우 크다. 실제 데이터의 소유 여부에 대한 정확한 증명이 이루어지지 않으면 데이터를 보유하고 있지 않으면서 데이터에 대한 권한을 획득할 수 있기 때문이다. 이에 따라, 특정 데이터를 소유하고 있는지 정확하게 증명가능한 방식으로 확인하기 위한 기술로 Halevi 등에 의해 PoW(proofs of ownership)가 Merkle-Tree 라는 데이터 구조를 기반으로 제안되었다 [7]. Merkle-Tree 기반의 PoW에서는 서버가 데이터의 임의의 위치를 문제로 제시하고 클라이언트는 해당 데이터에 대한 sibling path를 제시해야 한다. Merkle-Tree에 대한 자세한 설명은 관련 논문을 참고하길 바란다. PoW가 처음 소개된 이후 다양한 기술들이

안전성과 효율성을 개선하기 위해 제안되었으나 기본적인 구조는 거의 유사하게 challenge-response 구조를 취하고 있다.

4.5. 기타 연구

상기 기술한 공격 대응 기술들의 경우, 안전한 중복 처리를 위해 요구되는 대표적인 보안 요구사항을 위한 기술들이다. 이 외에도 안전성을 해치지 않으면서 성능을 향상시키거나 부가 기능을 제공하기 위한 기술들에 대한 연구도 수행되고 있다. [11]에서는 중복 데이터의 존재성을 확인하기 위해 계산하는 테그를 생성하기 위한 비용을 줄일 수 있는 방안을 제시하고 있다. 중복 데이터가 없는 경우는 존재성을 확인하기 위한 연산이 불필요한 비용이 되므로 이를 최소화하여 꼭 필요한 연산만 수행하도록 최적화된 테그 생성 방법을 제시하고 있다. [12]에서는 안전한 중복 처리 기술과 저장 데이터에 대한 무결성 검증을 위한 PoR을 동시에 제공하는 기술을 제안하였다. 특히, 무결성 증명을 수행함에 사용할 수 있으나 비밀 정보가 드러나지 않는 값을 생성함으로써 무결성 검증 기능의 위탁이라는 부가 기능을 제공하는 기술을 제안하였다.

Ⅴ. 결 론

본 고에서는 안전하게 데이터 중복 처리를 수행하기 위한 연구 동향을 살펴보았다. 중복 처리를 제공하면서도 기밀성을 보장하기 위한 기술과 저장된 데이터의 무결성을 검증하기 위한 기술 등 효율적인 스토리지 서비스를 제공하면서도 안전성을 보장하기 위한 기술은 데이터 기반의 서비스가 더욱 증대되고 있는 현재 IT 서비스 환경에서 중요하게 사용될 것으로 기대된다.

참 고 문 헌

[1] 박경수, 엄지은, 박정수, 이동훈, *안전하고 효율적인 클라이언트 사이드 중복 제거 기술,* Vol. 25, No. 1, pp. 83-94, 2월. 2015

[2] 박철희, 홍도원, 서창호, 장구영, *클라우드 스토리지 상에서의 프라이버시 보존형 소스기반 중복데이터 제거기술,* Vol. 25, No. 1, pp. 123-132, 2월. 2015

[3] 신영주, *분산 스토리지 시스템에서 데이터 중복제거를 위한 정보분산 알고리즘 및 소유권 증명 기법,* Vol. 25, No. 1, pp. 155-164, 2월. 2015

[4] 신형준, 구동영, 허준범, *암호화된 클라우드 데이터의 중복제거 기법에 대한 부채널 공격,* Vol. 27, No. 4, pp. 971-980, 8월. 2017

[5] M. Bellare, S. Keelveedhi, and T. Ristenpart, *DupLESS: server-aided encryption for deduplicated storage*, in Proc. of USENIX 2013, 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] A. Juels and B. Kaliski, *PORs: Proofs of retrievability for large files*, In Proc. of CCS 2007, pp. 584-597, 2007

[8] Kevin D. Bowers, Ari Juels, and Alina Oprea, *HAIL: A High-Availability and Integrity Layer for Cloud Storage*, in Proc. of CCS 2009, pp. 1-12, ACM, 2009.

[9] M. Bellare, S. Keelveedhi, and T. Ristenpart, Message-Locked Encryption and Secure Deduplication, In Proc. of EUROCRYPT 2013, LNCS 7881, pp. 296-312 (2013)

[10] T.-Y. Youn, N.-S. Jho, K. Kim, K.-Y. Chang, and K. Park, *Locked Deduplication of Encrypted Data to Counter Identification Attacks in Cloud Storage Platforms*, Energies, 13(2742):1-20, May 2020

[11] K. Kim, T.-Y. Youn, N.-S. Jho, and K.-Y. Chang, *Client-Side Deduplication to Enhance Security and Reduce Communication Costs*, ETRI Journal, 39(1):116-123, June 2017

[12] T.-Y. Youn, K.-Y. Chang, K.-H. Rhee, and S.-U. Shin, *Efficient Client-Side Deduplication of Encrypted Data with Public Auditing in Cloud Storage*, IEEE Access, 06:26578-26587, May 2018.

〈저 자 소 개〉

윤 택 영 (Youn Taek Young)
정회원
2009년 8월 : 고려대학교 정보보호대학원 박사
2009년 9월~2010년 7월 : 고려대학교 연구교수
2010년 7월~2020년 8월 : 한국전자통신연구원(ETRI) 선임연구원
2020년 9월~현재 : 단국대학교 산업보안학과 조교수
<관심분야> 암호, 정보보호, 프라이버시, 데이터보안