http://dx.doi.org/10.3745/KTCCS.2013.2.11.483

Outsourced Storage Auditing Scheme using Coefficient Matrix

Hasoo Eun† · Heekuck Oh†† · Sangjin Kim†††

ABSTRACT

Users can access their data anywhere, at any time by using outsourced storage. But they cannot know how service provider manage the data. Even user cannot know when data damaged. To solve these problems, the outsourced storage auditing schemes has been proposed. Most proposed schemes are based on Homomorphic Verifiable Tags. But it has computational efficiency limitation because data used to exponent. In this paper, we propose a novel approach to outsourced storage auditing scheme using coefficient matrix. In the proposed scheme, data used to auditing by coefficient matrix form. Auditing procedures are proceed as solving the linear simultaneous equation. The auditor can audit easily by solving the equation using solution vector. The auditor can audit the n size data using sqrt(n) size data through out proposed scheme.

Keywords : Outsourced Storage Auditing, Homomorphic Verifiable Tags, Coefficient Matrix

계수행렬을 이용한 외부 스토리지 무결성 검증 기법

은 하 수† · 오 희 국†† · 김 상 진†††

요 약

사용자들은 외부 스토리지를 사용함으로써 언제, 어디서나 자신의 데이터에 접근할 수 있다. 하지만 자신의 데이터가 어떻게 관리되고 있는지는 알 수 없다. 심지어 자신의 데이터에 손상이 발생하여도 인지할 수 없다. 이와 같은 불편을 해결하기 위해 외부 스토리지 검증기법들이 제안되었다. 대부분의 기법들이 준동형 검증 태그(homomorphic verifiable tags)를 사용하고 있지만, 이는 데이터를 지수로 하여 계산하기 때문에 효율성에 한계가 있다. 본 논문에서는 외부 스토리지 무결성 검증의 새로운 접근 방법으로써 계수행렬을 이용한 기법을 제안한다. 제안하는 기법은 데이터를 계수행렬의 형태로 변환하여 검증에 사용한다. 검증과정은 선형연립방정식의 해를 구하는 형태로 진행되며, 검증자는 수식에 해벡터를 대입함으로써 쉽게 검증을 수행할 수 있다. 제안하는 기법을 사용하면 검증자는 sqrt(n) 크기의 데이터로 크기가 n인 데이터를 검증할 수 있다.

키워드 : 외부 스토리지 검증, 준동형 검증 태그, 계수행렬

1. 서 론

최근 클라우드 컴퓨팅의 사용이 일반화되었다. 사용자들은 클라우드 컴퓨팅을 이용하여 언제, 어디서든 자신의 데이터에 접근하고 업무를 수행할 수 있다. 대부분의 사용자들은 클라우드 컴퓨팅을 사용하기 위해 자신의 데이터를 업로드한다. 단순히 저장을 목적으로 할 수도 있고, 이를 이용하여 데이터 마이닝 등 특별한 연산을 수행할 수도 있다. 데이터의 소유 측면에서 보았을 때 이와 같은 행동은 사용자가 클라우드 시스템에 데이터를 위탁하는 것으로 간주할 수 있다. 사용자가 언제, 어디서든 데이터에 접근할 수 있다는 말은, 데이터를 가지고 있지 않더라도 시스템을 사용할 수 있음을 의미한다.

클라우드 서비스 제공자는 사용자의 데이터를 관리하기 위해 많은 노력을 기울일 것이다. 하지만 클라우드 서비스 제공자의 노력에도 불구하고, 천재지변이나 해킹 등으로 인해 의도치 않게 데이터에 손상이 발생할 수 있다. 이렇게 발생한 데이터 손상은 클라우드 서비스 제공자가 공지하거나, 사용자가 직접 사용해보기 전까지 알 수 없다. 이와 같은 상황에 대처하기 위해 데이터 손상여부를 능동적으로 확인할 수 있는 방법이 필요하다.

※ 본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(NIPA-2013-H0301-13-1002).
※ 이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2012-R1A2A2A01046986).
※ 이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임(No.2012-R1A1A2009152).
† 준 회 원 : 한양대학교 컴퓨터공학과 박사과정
†† 종신회원 : 한양대학교 컴퓨터공학과 교수
††† 종신회원 : 한국기술교육대학교 컴퓨터공학부 부교수
논문접수 : 2013년 10월 7일
심사완료 : 2013년 11월 2일
* Corresponding Author : Sangjin Kim(sangjin@koreatech.ac.kr)

외부 스토리지의 데이터 무결성 검증문제가 제기된 이래, 이를 해결하기 위한 많은 기법들이 제안되었다. 가장 기본적인 무결성 검증 방법으로써 해쉬(hash)를 예로 들 수 있다. 해쉬는 입력 값이 1bit만 바뀌어도 출력 값에 큰 변화가 생기는 특성이 있다. 데이터의 무결성을 검증하기 위해 해쉬를 이용하게 되면, 약간의 데이터 손상에도 해쉬 값이 크게 바뀌므로 데이터 손상 여부를 쉽게 판단할 수 있다. 하지만 해쉬는 외부에 저장된 데이터의 무결성 검증에는 적합하지 않다. 서비스 제공자가 미리 계산된 해쉬를 검증자에게 전달함으로써 쉽게 우회할 수 있기 때문이다. 검증자가 데이터를 받아 해쉬를 계산해볼 수도 있지만, 클라우드 스토리지와 같은 대용량 스토리지에는 적합하지 않다. 이에, 준동형 특성에 기반을 둔 외부 스토리지 무결성 검증 기법이 제안되었다.

최근의 외부 스토리지 무결성 검증은 지수, 겹선형 쌍곡선 등의 준동형 특성을 이용한 기법이 주류를 이루고 있다. 준동형 특성을 이용하면 쉽게 영지식 증명 프로토콜(zero-knowledge proof protocol)을 구성할 수 있다. 검증과정에 영지식 증명 프로토콜을 사용하면 검증자가 유지해야 할 정보를 최소한으로 줄일 수 있다. 하지만 준동형 특성을 이용하는 많은 기법들이 지수 연산에 기반을 두고 있으며, 데이터를 지수부에 두고 계산하기 때문에 연산의 효율성 측면에서 일정 수준이상 개선할 수 없는 한계가 있다.

본 논문은 외부 스토리지 검증 기법의 연산 효율 한계를 극복하기 위한 연구의 일환으로써 계수행렬을 이용한 외부 스토리지 무결성 검증 기법을 제안한다. 제안하는 기법은 데이터를 계수행렬의 형태로 다루어 검증을 수행한다. 서비스 제공자는 선형연립방정식의 해를 구하는 형태로 검증식을 생성하며, 검증자는 서비스 제공자로부터 받은 검증식에 해를 대입함으로써 검증을 수행한다. 제안하는 기법은 연립방정식의 풀이와 해의 대입만으로 검증을 수행하기 때문에 검증에 필요한 연산량이 적다. 이와 더불어 기존 기법들에서 제공되었던 배치검사, 샘플링을 통한 확률적 검사, 블록 단위 동적 업데이트 등을 제공할 수 있다.

이후 논문 구성은 2장에서 연구배경에 대해 설명하고, 3장에서 기존의 외부 스토리지 무결성 검증 기법들을 소개한다. 4장에서 계수행렬을 이용한 외부 스토리지 무결성 검증 기법을 제안하며, 5장에서는 앞서 제안한 기법을 안전성과 효율성 측면에서 분석한다. 끝으로 6장에서 결론을 맺는다.

2. 배경지식

이 장에서는 본 연구와 관련된 배경지식에 대해 알아본다. 우선 참여자들이 무엇을 원하는지 알아보고, 해쉬 기반 무결성 검증 기법과 준동형 검증 태그 기반 무결성 검증 기법에 대해 설명한다.

2.1 참여자

외부 스토리지 무결성 검증의 참여자는 역할에 따라 사용자, 서비스 제공자, 검증자로 나눌 수 있다. 시스템에 따라 사용자와 검증자는 동일한 개체일 수 있다. 사용자는 외부 스토리지에 자신의 데이터를 저장하는 개체이다. 사용자는 자신의 데이터가 외부 스토리지에 온전히 저장되어 있음을 알고 싶어 한다. 서비스 제공자는 사용자의 데이터를 관리하는 개체이다. 여러 사용자의 데이터를 유지하고 있으며, 가능한 효율적으로 저장 및 관리하고 싶어 한다. 검증자는 서버에 저장된 데이터의 무결성을 검증하는 개체이다. 무결성 검증의 결과를 사용자에게 제공한다.

본 논문에서는 외부 스토리지에 저장된 데이터와 별도로 사용자가 데이터를 유지하지 않는다고 가정한다. 또한 검증을 수행하는 개체는 데이터를 검증하기 위한 별도의 값을 유지할 수 있다고 가정한다. 이와 같은 가정은 사용자가 데이터의 저장 및 관리를 외부 스토리지에 완전히 위탁함을 의미한다. 또한, 검증자가 전체 데이터를 유지하지 않는 것은 역할의 구분을 명확히 하기 위함이다. 만일 검증자가 전체 데이터를 유지하며 무결성을 검증한다면 외부 스토리지 서비스 제공자를 별도로 둘 필요가 없다.

2.2 해쉬 기반 기법과 준동형 검증 태그 기반 기법

대표적인 무결성 검증 방법은 해쉬를 이용하는 것이다. 하지만 외부에 저장된 데이터의 무결성 검증은 데이터를 관리하는 개체와 검증하는 개체가 서로 다르다. 따라서 기존의 로컬 데이터 검증처럼 해쉬를 사용하는 경우 서비스 제공자가 미리 계산해둔 해쉬 값을 재전송하여 무결성 검증을 우회할 수 있다. 즉, 기존의 해쉬를 이용하게 되면, 검증 결과의 최신성(freshness)를 제공할 수 없다. 최신성을 제공하기위한 방법으로써 식 (1)과 같이 챌린지 C를 데이터 m과 함께 연산하도록 요구할 수 있다. 여기서 $H(.)$는 해쉬 함수, R은 해쉬의 결과로 얻은 해쉬 값을 의미한다.

$$R = H(m, C) \tag{1}$$

이 경우 해쉬 값 R을 미리 계산해둘 수 없으므로 최신성을 제공할 수 있다. 하지만 2.1에서 설명했듯이 검증자는 데이터 m을 유지하지 않으므로 해쉬 값 R을 검증할 수 없다. 이와 같은 문제를 해결하기 위해 준동형 검증 태그 기법이 제안되었다[1]. 이는 해쉬에 챌린지를 포함할 수 있도록 개선한 기법이다. Deswarte 등이 처음 제안한 이 기법은 지수연산의 준동형 특성에 기반을 두고 있다. 하지만 Deswarte 등의 기법은 데이터를 지수부에 두고 계산을 하기 때문에 연산량 측면에서 효율적이지 않다. 이후 Deswarte 등의 기법에서 연산량을 개선한 다양한 기법들이 제안되었다[2-6].

3. 관련 연구

이 장에서는 기존에 제안되었던 준동형 검증 태그 기반 기법들에 대해 소개한다. 외부 스토리지 검증 기법으로서

초기에 제안된 Deswarte 등의 기법에서부터 준동형 검증 태그에 기반을 둔 주요 논문들에 대해 서술한다.

3.1 준동형 검증 태그 기반 기법들

Deswarte 등은 2004년에 처음으로 외부 스토리지 검증 기법을 제안했다[1]. 제안된 기법은 디피-헬만 키 교환 프로토콜(Diffie-Hellman key exchange protocol)에 기반을 둔 것으로서, 지수연산의 준동형 특성을 이용하고 있다. 사용자가 데이터 m을 저장한다면, 검증자에게 $\alpha \equiv g^m \bmod N$을 전달하고, 서비스 제공자에게 m을 전달한다. 검증자는 랜덤 값 r을 선택하여 $req \equiv g^r \bmod N$을 서비스 제공자에게 전달한다. 서비스 제공자는 m과 req를 이용하여 $res \equiv req^m \bmod N$을 만들어 검증자에게 전달한다. 검증자는 $res \equiv ?\alpha^r \bmod N$을 계산함으로써 서비스 제공자의 데이터가 손상되지 않았음을 알 수 있다. Deswarte 등의 기법은 지수 연산을 통하여 검증자의 챌린지를 검증 값에 반영할 수 있다. 하지만 데이터를 통째로 지수부에 두어야 하기 때문에 무결성 검증에 많은 연산량이 필요하다.

Filho 등은 오일러 함수의 특성을 이용하여 검증자가 유지해야 하는 데이터의 크기를 줄였다[2]. 하지만 제안된 기법 역시 Deswarte 등의 기법과 연산량의 차이는 크지 않다. 이후 Yamamoto 등은 지수연산의 특성을 이용하여 데이터 블록을 나누고, 이들을 합하여 검증 값을 생성하는 기법을 제안했다[3]. 즉, $g^{\Sigma m_i} = \Pi g^{m_i}$인 지수연산의 특성을 이용하여 블록단위로 계산하고 이들을 곱하여 데이터 전체의 무결성을 검증한다. Yamamoto등의 기법은 데이터를 분할하여 계산하고, 이들을 모아 일괄검증 할 수 있는 최초의 기법으로써 의미가 있다. Ateniese 등은 공개키에 기반을 둔 준동형 검증 태그 기법을 제안했다[4]. 공개키와 개인키를 곱하여 1이 되는 특성을 이용하여, 각 블록에 사용자가 서명하고, 검증자가 서명된 데이터를 확인하는 형태로 진행된다. Shacham 등은 기본 블록 단위 기법을 더욱 잘게 나누고, 이들마다 임의의 수로 챌린지 함으로써 무결성을 검증하는 기법을 제안했다[5]. 챌린지하는 과정에서 임의로 블록을 선택하여 검증할 수 있으며, 이와 더불어 처음으로 확률적 검사를 제안했다. 제안된 내용에 따르면 1000개의 블록 중 360개만 검사하여도 99% 확률로 데이터 손상여부를 판단할 수 있다. 즉, 검증에 필요한 연산량을 1/3로 줄일 수 있다. 마지막으로, 최근 Wang 등은 기존의 기법을 포괄한 무결성 검증기법을 제안했다[6]. 이 기법은 기존 기법들의 장점을 취하고, 검증자에게 데이터를 노출하지 않도록 랜덤 값을 곱했다. 하지만 페어링 연산을 사용함으로써 기본적인 연산량이 증가하는 문제가 있으며, 블록단위로 메타데이터를 관리한 결과 검증 값을 관리하기 위해 많은 저장공간이 필요하다. 이 외에도 최근에 제안된 많은 기법들이 준동형 검증 태그에 기반을 두고 있다[7-14]. 그만큼 준동형 검증 태그 기법이 위탁된 데이터의 무결성 검증에 적합함을 의미한다. 하지만 현재까지 제안된 기법들은 준동형 특성을 이용하기 위해 기본적으로 지수연산을 사용하고 있으며, 특히 데이터를 지수부에 두어야하기 때문에 연산량을 효율적으로 개선하는데 한계가 있다.

4. 제안하는 기법

이 장에서는 계수 행렬에 기반을 둔 무결성 검증 기법을 제안한다. 기법을 제안하기에 앞서 외부 스토리지 무결성 검증 기법이 신뢰할 수 있는 무결성 검증 결과를 제공하기 위해 갖추어야할 보안 요구사항 및 시스템 요구사항을 서술한다.

4.1 보안 요구사항

외부 스토리지 무결성 검증 기법의 목표는 검증자가 사용자의 데이터 손상여부를 확인하는 것이다. 외부 스토리지에 저장된 데이터에 대해 신뢰할 수 있는 무결성 검증결과를 제공하기 위해 다음의 보안 요구사항을 만족해야 한다.

요구사항 1. 서비스 제공자가 검증 값 생성 함수 $f(.)$의 연산 과정을 임의로 생략할 수 없어야 한다.

요구사항 2. 검증 값 생성 함수 $f(.)$는 데이터 m 전체의 특성을 반영할 수 있어야 한다.

요구사항 3. 임의의 데이터 $\widetilde{m}(\neq m)$를 이용하여 검증 과정을 통과할 수 없어야 한다.

4.2 시스템 요구사항

본 논문에서 제안하는 기법은 디자인 면에서 기존의 기법들과 다른 접근을 시도하고 있다. 이와 더불어 검증 연산의 효율을 높이기 위해 배치(batch)검사, 확률적 검사 등을 제공할 수 있어야 하며, 동적 업데이트 또한 제공하고자 한다. 본 논문에서 달성하고자 하는 시스템 요구사항은 다음과 같다.

요구사항 1. 검증자는 적은 비용으로 외부 스토리지의 무결성을 검증할 수 있어야 한다.

요구사항 2. 무결성 검증을 위해 검증자가 유지해야하는 정보의 양을 최소화해야 한다.

요구사항 3. 여러 블록의 무결성을 한 번에 검증할 수 있어야 한다.

요구사항 4. 검증자는 데이터 블록들에 대해 선택적으로 무결성을 검증할 수 있어야 한다.

요구사항 5. 기존 블록과 함께 업데이트 된 블록의 무결성을 검증할 수 있어야 한다.

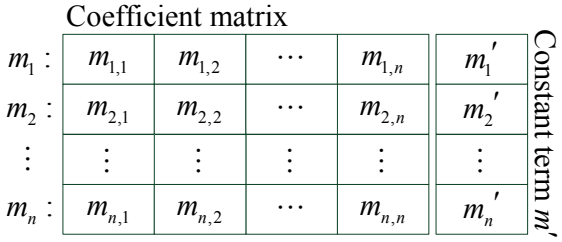


Fig. 1. Structure of coefficient matrix

4.3 계수 행렬에 기반을 둔 무결성 검증 기법

최근 사용되는 하둡(hadoop)과 같은 클라우드 파일 시스템은 사용자의 데이터를 블록 단위로 나누어 관리한다. 본 논문에서는 이에 착안하여 각 데이터 블록을 n개의 조각으로 나누고, 각 조각을 정수군에 매핑한다. 마지막으로 정수군에 매핑된 작은 블록들을 모아 행렬을 만든다. 본 논문에서 사용하는 계수 행렬은 Fig. 1과 같다. 여기에서 데이터 블록 m_i와 상수항 m'_i는 선형 방정식의 계수와 상수 관계로 생각할 수 있다.

프로토콜을 수행하기에 앞서 사용자는 데이터를 정수군에 매핑하여 계수를 구하고, 상수항을 정한다. 선형 연립방정식을 풀기 위해서는 각 선형 방정식들이 선형 독립(linearly independent)이어야 한다. 본 논문에서는 선형 독립을 만들기 위해 상수항을 조절한다. 다른 방정식과 의존 관계가 발견되지 않을 때까지 0이 아닌 임의의 값을 대입한다. 연립방정식의 구성이 완료되면 소거법을 통해 해벡터 (solution vector)를 구한다. 계수 행렬 $\boldsymbol{F}$, 해벡터 $\boldsymbol{x}$, 상수벡터 $\boldsymbol{m'}$사이에는 식 (2)와 같은 관계가 성립한다.

$$\boldsymbol{Fx} = \boldsymbol{m'} \tag{2}$$

사용자는 데이터 m을 서버에 전달하고, 해벡터 $\boldsymbol{x}$와 상수벡터 $\boldsymbol{m'}$을 검증자에게 전달한다.

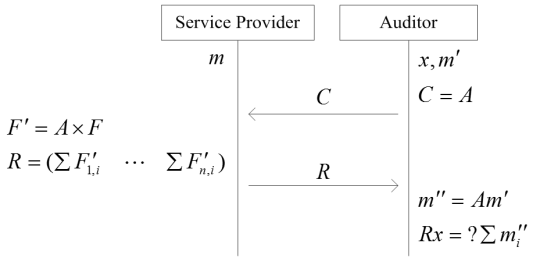


Fig. 2. Proposed auditing scheme

데이터의 분배가 끝난 후 검증자는 서비스 제공자에게 Fig. 2와 같이 검증을 요청한다. 검증자가 데이터의 무결성을 검증하는 과정은 다음과 같이 진행된다.

- Step 1. 검증자는 챌린지 행렬 $\boldsymbol{A}$를 생성하여 서비스 제공자에게 전달한다.
- Step 2. 서비스 제공자는 챌린지 행렬 $\boldsymbol{A}$와 계수 행렬 $\boldsymbol{F}$를 곱하여 행렬 $\boldsymbol{F'}$을 계산한다.
- Step 3. 행렬 $\boldsymbol{F'}$을 1행부터 n행까지 더하여 행벡터 $\boldsymbol{R}$을 생성한다.
- Step 4. 서비스 제공자는 행벡터 $\boldsymbol{R}$을 검증자에게 전달한다.
- Step 5. 검증자는 챌린지 행렬 $\boldsymbol{A}$와 상수벡터 $\boldsymbol{m'}$을 곱하여 열벡터 $\boldsymbol{m''}$을 계산한다.
- Step 6. 행벡터 $\boldsymbol{R}$과 해벡터 $\boldsymbol{x}$의 곱이 열벡터 $\boldsymbol{m''}$의 각 인자들의 합과 같은지 확인한다.

4.4 제안하는 기법의 타당성

수식 (2)의 양변에 챌린지 행렬 $\boldsymbol{A}$를 곱하면 식 (3)을 얻을 수 있다.

$$\boldsymbol{AFx} = \boldsymbol{Am'} \tag{3}$$

여기에서 $\boldsymbol{AF}$를 $\boldsymbol{F'}$으로, $\boldsymbol{Am'}$을 $\boldsymbol{m''}$으로 대치하면 식 (4)를 얻을 수 있다.

$$\boldsymbol{F'x} = \boldsymbol{m''} \tag{4}$$

$\boldsymbol{F'}$의 i번째 행벡터 $\boldsymbol{F'_{i*}}$와 해벡터 $\boldsymbol{x}$의 곱을 임의의 벡터 $\boldsymbol{y_i}$라 하면, 식 (5)와 같이 표현할 수 있다.

$$\boldsymbol{F_{i*}'x} = \boldsymbol{y_i} \tag{5}$$

$\boldsymbol{F'}$의 i번째 행벡터 $\boldsymbol{F'_{i*}}$의 해가 벡터 $\boldsymbol{x}$라면 $\boldsymbol{y_i} = \boldsymbol{m_i''}$가 된다. 양 변에 $\boldsymbol{m_{i+1}''}$을 더하면 식 (6)을 얻을 수 있다.

$$\boldsymbol{y_i} + \boldsymbol{m_{i+1}''} = \boldsymbol{m_i''} + \boldsymbol{m_{i+1}''} \tag{6}$$

만일, $\boldsymbol{F_{(i+1)*}'}$의 해도 벡터 $\boldsymbol{x}$라면 $\boldsymbol{y_{i+1}} = \boldsymbol{m_{i+1}''}$이 된다.

$$\boldsymbol{y_i} + \boldsymbol{y_{i+1}} = \boldsymbol{m_i''} + \boldsymbol{m_{i+1}''} \tag{7}$$

$\boldsymbol{F'}$의 모든 행벡터에 대한 해가 $\boldsymbol{x}$라면, 식 (7)은 다음과 같이 확장될 수 있다.

$$\sum_{i=1}^{n} \boldsymbol{y_i} = \sum_{i=1}^{n} \boldsymbol{F_{i*}'x} = \boldsymbol{Rx} = \sum_{i=1}^{n} \boldsymbol{m_i''} \tag{8}$$

식 (8)은 $\boldsymbol{Rx} \neq \sum_{i=1}^{n} \boldsymbol{m_i''}$일 경우, $\boldsymbol{F'}$의 행벡터 중 해벡터 $\boldsymbol{x}$를 해로 갖지 않는 것이 포함되어 있음을 의미한다. 이것은 $\boldsymbol{F}$가 해벡터 $\boldsymbol{x}$를 생성할 때와 달라졌음을 의미하며, 이를 통해 데이터가 손상되었음을 알 수 있다.

4.5 동적 업데이트 지원

제안하는 기법은 블록 단위의 업데이트를 지원한다. 블록 단위로 업데이트를 해야 조각으로 분할하고 방정식을 세울 수 있기 때문이다. 본 논문에서는 동적 업데이트를 지원하기 위해, 이미 계산된 해에 맞추어 해벡터를 조절하는 방법을 사용한다.

- Update: 새로운 블록을 조각으로 분할하고 방정식을 세운다. 업데이트할 파일의 해벡터를 방정식에 대입하여 상수항을 구한다. 계산이 완료된 블록은 서버에, 상수항은 검증자에게 전달한다.

Delete: 블록의 삭제를 위해 서버와 검증자에게 삭제할 블록의 인덱스를 알린다. 서버는 해당 블록을 삭제하고, 검증자는 삭제된 블록의 인덱스를 관리함으로써 검증에 사용되지 않도록 한다.

해를 계산하는 과정에서는 선형독립이 해의 존재 여부에 영향을 주었지만, 업데이트를 수행할 때는 이미 해를 계산한 후이기 때문에, 동일한 데이터에 대해 상수항이 겹쳐도 시스템에는 영향을 주지 않는다. 삭제과정 또한 각각의 블록이 독립적이므로 하나의 블록을 지워도 전체 파일과 해에는 영향을 미치지 않는다.

5. 분 석

이 장에서는 제안하는 기법이 안전성과 효율성에 대해 분석한다. 안전성은 4.1의 보안 요구사항을 만족하는지에 대해 논의하고, 효율성은 제안된 기법이 소량의 데이터 만으로 검증과정을 효율적으로 수행할 수 있는지에 대해 논의한다.

5.1 안전성 분석

서비스 제공자는 검증자와 공모하지 않는 한, 검증 값 생성 함수 $f(.)$의 연산 과정을 임의로 생략할 수 없다. 서비스 제공자가 이를 임의로 생략하기 위해서는 $\boldsymbol{AF}=\boldsymbol{F}$를 만족해야 한다. 즉, $\boldsymbol{A}$가 항등 행렬(identity matrix)여야 한다. 따라서 서비스 제공자가 챌린지로써 항등행렬이 아닌 매번 새로운 행렬을 보낸다면, 서비스 제공자는 검증 값 생성 함수 $f(.)$의 연산 과정을 임의로 생략할 수 없다.

본 논문에서 제안한 기법은 m 전체의 특성을 반영한다. 다시 말해 데이터의 일부가 누락된 경우 검증자가 발견할 수 있다. 만일, 임의의 블록 m_j가 누락된 경우 다음을 만족하면 검증을 통과할 수 있다.

$$\sum_{i=1}^{n} \boldsymbol{F}_{i*}'\boldsymbol{x} - \sum_{i=1}^{n} \boldsymbol{F}_{ij}'\boldsymbol{x} = \sum_{i=1}^{n} \boldsymbol{m}_i''$$

$$\therefore \sum_{i=1}^{n} \boldsymbol{F}_{ij}'\boldsymbol{x} = 0 \qquad (9)$$

즉, m_j의 상수항이 0이어야 한다. 하지만 상수항은 0이 아닌 임의의 수를 사용하므로 식 (9)는 성립되지 않는다. 즉, 임의의 블록 m_j가 누락된 경우, 검증자는 이를 발견할 수 있다.

검증자는 m에 오류가 발생한 경우, 즉, 서비스 제공자가 $\widetilde{m}(\neq m)$으로 검증을 수행하는 경우 이를 발견할 수 있다. 오류로 인해 $\boldsymbol{F}_{i,j}$가 $\boldsymbol{F}_{i,j}+\Delta t$로 바뀌었다고 가정하자. 여기서 Δt는 오류로 인해 생긴 데이터의 편차를 의미한다. 이를 통해 검증을 수행한다면 열벡터 $\boldsymbol{F}_{*i}'$의 모든 값에 Δt가 포함된다. 다시 말해, 데이터에 발생한 오류는 검증과정에서 오류가 확산되어 모든 수식이 해를 만족하지 않는다.

5.2 효율성 분석

제안하는 기법은 블록의 개수에 무관하게 검증을 수행할 수 있다. 즉, i개의 블록을 선택하여 확률적 검증을 수행할 수 있고, n개의 블록을 선택하여 전체 데이터를 검증할 수도 있다. 검증하려는 블록의 수에 따라 챌린지 행렬의 크기를 달리하고, 선택한 블록과 동일한 상수항을 선택하여 연산함으로써, 검증자는 자신이 원하는 블록들을 선택적으로 검증할 수 있다.

제안하는 기법은 상수벡터를 숨김으로써 동일한 해를 갖는 블록의 수가 늘어나도 서비스 제공자가 해를 얻지 못하도록 하고 있다. 사용자가 새로운 블록을 추가하는 경우, 해에 맞추어 상수항을 계산함으로써 기존 블록들과 함께 배치검사를 수행할 수 있다.

제안하는 기법을 검증하는데 필요한 연산량은 블록조각의 수에 비례한다. 즉, 제안하는 기법은 차후 업데이트된 블록의 수와 무관하게 연산량이 유지된다. 연산 또한 곱셈과 덧셈만으로 구성되어, 지수연산 및 페어링 연산을 사용하는 준동형 기법에 비해 빠른 검증이 가능하다.

최근에 제안된 Wang 등의 기법은 영지식 증명을 이용함으로서 검증자가 파일에 대한 정보를 유지하는 것만으로 검증이 가능하다. 하지만 제안하는 기법은 파일 정보 대신 해벡터와 상수벡터를 유지한다. 만일 서버에 저장된 데이터의 크기를 n이라 했을 때, 검증자가 유지하는 데이터의 크기는 $2\sqrt{n}$이다. 이는 서버에 저장된 1TB의 데이터를 724KB로 검증할 수 있음을 의미한다.

6. 결론 및 향후과제

본 논문은 클라우드 스토리지에 저장된 데이터의 무결성을 검증할 수 있는 새로운 접근 방법을 제시하고 있다. 제안된 기법은 블록의 집합을 이용해 계수 행렬을 구성하고 이에 대한 해를 구해 검증을 수행한다. 해의 크기는 블록조각의 수에 비례하기 때문에, 업데이트를 통해 블록의 수가 증가해도 $O(\sqrt{n})$의 검증비용을 보장한다. 여기서 n은 서버에 저장된 데이터의 크기이다. 제안하는 기법은 기존 연구들에서 제안된 배치검증, 선택적 검증, 동적 업데이트를 지원하면서, 적은 데이터로 모든 블록을 검증할 수 있다는 장점이 있다.

제안하는 기법은 적은 비용으로 연산을 수행할 수 있으나, 초기 설정 과정에 계수 행렬의 해벡터, 상수벡터를 계산하기 위해 추가적인 연산이 필요하다. 또한, 행렬을 챌린지 함으로써 결과 값을 계산하는데 $O(n^3)$만큼의 곱셈 연산이 필요하다. 향후에는 초기 설정과정 및 챌린지의 연산량을 효율적으로 개선하기 위한 연구가 진행되어야 할 것이다. 또한 클라우드 파일 시스템에 소규모 업데이트 시 발생하는 오버헤드 등 최근 지적되고 있는 문제점들을 고루 반영하여 연구가 진행되어야 할 것이다.

참 고 문 헌

[1] Y. Deswarte, J.-J. Quisquater, and A. Saïdane, "Remote Integrity Checking," in Integrity and Internal Control in Information Systems VI. Vol.140, S. Jajodia and L. Strous, Eds., ed: Springer US, 2004, pp.1-11.

[2] D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," IACR Cryptology ePrint Archive, Vol.2006, pp.150, 2006.

[3] G. Yamamoto, S. Oda, and K. Aoki, "Fast integrity for large data,"" in Workshop on Software Performance Enhancement for Encryption and Decryption (SPEED 2007), 2007, pp.21-32.

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, et al., "Provable data possession at untrusted stores," presented at the Proceedings of the 14th ACM conference on Computer and communications security, Alexandria, Virginia, USA, 2007.

[5] H. Shacham and B. Waters, "Compact Proofs of Retrievability," presented at the Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, Melbourne, Australia, 2008.

[6] W. Cong, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Transactions on Computers, Vol.62, pp. 362-375, 02/01 2013.

[7] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Probable Data Possession," Proceedings of the 16th ACM conference on Computer and Communications Security (CCS '09), pp.213-222, Nov., 2009.

[8] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proceedings of the 14th European conference on Research in computer security (ESORICS '09), pp.355-370, 2009.

[9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proceedings of IEEE INFOCOM 2010, pp.1-9, 2010.

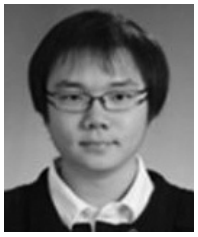
[10] Q. Wang, C. Wang, K. Ren, and W. Lou, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Transactions on Parallel and Distributed Systems, pp.847-859, Mar., 2011.

[11] K. Yang and X Jia, "Data storage auditing service in cloud computing challenges, methods and opportunities," World Wide Web, Vol.15, pp.409-428, 2012.

[12] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud," Applied Cryptography and Network Security (ACNS), LNCS 7341, pp.507-525, 2012.

[13] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," 2012 IEEE Fifth International Conference on Cloud Computing, pp.295-302, 2012.

[14] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Trans. on Parallel and Distributed Systems, Vol.24, No.9, pp.1717-1726, Sept., 2013.



은 하 수

e-mail : hseun@hanyang.ac.kr
2010년 한양대학교 컴퓨터공학과(공학사)
2012년 한양대학교 컴퓨터공학과(공학석사)
현 재 한양대학교 컴퓨터공학과 박사과정
관심분야 : 모바일 보안, NFC, 암호학

오 희 국

e-mail : hkoh@hanyang.ac.kr
1983년 한양대학교 전자공학과(학사)
1989년 아이오와주립대학교 전자계산학과(석사)
1992년 아이오와주립대학교 전자계산학과(박사)
현 재 한양대학교 컴퓨터공학과 교수
관심분야 : 네트워크 보안, 암호프로토콜

김 상 진

e-mail : sangjin@koreatech.ac.kr
1995년 한양대학교 컴퓨터공학과(공학사)
1997년 한양대학교 컴퓨터공학과(공학석사)
2002년 한양대학교 컴퓨터공학과(공학박사)
현 재 한국기술교육대학교 컴퓨터공학부 부교수
관심분야 : 프라이버시 보호, 애드혹 네트워크 보안, 클라우드 컴퓨팅 보안