Journal of KIIT. Vol. 16, No. 5, pp. 91-102, May 31, 2018. pISSN 1598-8619, eISSN 2093-7571 91
http://dx.doi.org/10.14801/jkiit.2018.16.5.91

클라우드 환경에서의 프라이버시 강화형 중복제거 기법 제안

이동혁*, 박남제**

A Proposal of Privacy-Enhanced Deduplication Technique in a Cloud Environment

Donghyeok Lee*, Namje Park**

이 논문은 2018학년도 제주대학교 교원성과지원사업에 의하여 연구되었음.

요 약

향후 4차산업 시대에는 대량의 비정형 데이터를 취급하게 될 것이며, 클라우드 스토리지의 비용절감을 위한 중복제거 기술이 필수적인 요소기술로 자리잡게 될 것이다. 그러나 현재의 중복제거 기술은 구조적으로 프라이버시 보호 문제에 취약점을 가지고 있다. 본 논문에서는 현재의 중복제거 기술에서 발생할 수 있는 프라이버시 침해 문제를 지적하고, 이러한 문제를 해결할 수 있는 새로운 중복제거 기법을 제안하였다. 제안한 기법은 서버상에 사용자와 파일간 매핑구조를 저장하지 않는 특성이 있어 메타분석 공격에 안전하며, 사용자의 PIN을 기반으로 파일에 대한 소유권에 대한 증명도 가능하다. 또한, 내부자 공격, 스니핑 공격에 안전하다는 장점도 가지고 있다.

Abstract

In the fourth industrial age of the future, large amounts of unstructured data will be handled, and data deduplication technology for cost reduction of cloud storage will become an essential element technology. However, current data deduplication technology is structurally vulnerable to privacy protection issues. In this paper, we point out the problem of privacy vulnerability that can occur in current data deduplication technology. So, we propose a new data deduplication technique to solve this problem. The proposed scheme is secure against meta analysis attacks because it does not store the mapping structure between users and files on the server. It is also possible to prove ownership of the file based on the user's PIN, and secure for insider attack and sniff attack.

Keywords
secure deduplication, cloud security, privacy protection, insider attack

* 제주대학교 초등교육연구소 특별연구원,
제주대학교 과학기술사회연구센터 박사
- ORCID: https://orcid.org/0000-0001-7516-469X
** 제주대학교 초등컴퓨터교육전공 교수(교신저자)
- ORCID: https://orcid.org/0000-0003-4434-8933

· Received: Feb. 10, 2018, Revised: Apr. 16, 2018, Accepted: Apr. 19, 2018
· Corresponding Author: Namje Park
Dept. of Computer Education, Teachers College, Jeju National University, Hwabuk 1-dong, Jeju-si, Jeju Special Self-Governing Province, Korea, Tel.: +82-64-754-4914, Email: namjepark@jejunu.ac.kr

Ⅰ. 서 론

최근 클라우드 서비스가 대중화되면서 다양한 클라우드 서비스 제공 업체가 등장하고 있다. 클라우드 플랫폼에서는 다양한 형태의 서비스 제공이 가능하며, 이 가운데 스토리지 기반의 클라우드 서비스가 가장 널리 사용되는 서비스 유형으로 향후에도 지속적으로 시장이 확대될 것으로 보인다[1][2].

특히, 향후 다가올 4차 산업시대에서는 데이터 사이즈가 기하급수적으로 늘어날 것이며, 이는 결국 스토리지 확대에 따른 비용 증가로 이어지게 되어 서비스 제공 업체에 큰 부담으로 다가올 것이다.

따라서 클라우드 환경에서는 중복제거 기술이 매우 중요한 요소기술 중 하나이다. 중복제거 기술이란 복수의 사용자에 의해 업로드된 동일한 데이터를 중복으로 저장하지 않는 기술을 의미하며, 이러한 중복제거 기술을 통하여 획기적인 스토리지 용량 절감이 가능하다는 장점이 있다[3][4].

따라서 현재 상당수의 클라우드 스토리지 환경에서는 중복제거 기술이 이미 적용되어 있다. 클라우드 환경의 특성상 대용량의 스토리지 서버가 필요하며, 저장 용량이 증가하면 추가 스토리지 증설이 필요하다. 그러나 중복제거 기술을 사용하면 이러한 비용상의 문제를 원천적으로 해결할 수 있다.

그러나 중복제거 기술은 구조적으로 프라이버시 문제를 안고 있다. 중복제거 기술이 적용되려면, 사용자와 파일의 매핑구조를 메타정보로 저장하게 되며, 이 과정에서 서버상의 메타 분석을 통하여 특정 파일을 업로드한 사용자의 리스트를 확보할 수 있기 때문이다. 특히, 정치성향, 사상, 특정 질병, 성생활 등 민감성을 가지고 있는 파일인 경우 문제는 매우 심각할 수 있다. 클라우드 서버에서 해당 파일의 업로드 사용자 리스트를 확보할 수 있게 된다면 향후 클라우드 환경이 일종의 감시 체제로 작용할 위험성도 배제할 수 없기 때문이다[5]-[7].

본 논문에서는 이러한 문제를 지적하고, 이를 해결하기 위한 새로운 중복제거 기법을 제안하였다. 제안한 기법은 사용자와 파일의 매핑구조를 서버상에 저장하지 않으므로 서버의 메타정보 분석을 통해서는 파일의 업로드 사용자 리스트를 확보할 수 없어 사용자의 프라이버시를 보호할 수 있다. 또한, PIN을 활용하여 파일의 소유권 문제를 해결할 수 있으며, 내부자 공격 및 스니핑 공격에 안전하다는 장점이 있다.

Ⅱ. 관련 연구

2.1 데이터 중복제거 기술

중복제거 기술이란, 동일한 데이터를 중복으로 저장하지 않고, 하나의 파일만 저장하는 기술을 의미한다. 일반적으로 중복제거는 파일 해쉬값 비교를 통하여 구현할 수 있으며, 중복된 특정 파일에 해당된 사용자를 매핑하는 형태로 구현하는 경우가 일반적이다. 예를 들어, 그림 1과 같은 형태로 중복제거 기법을 구현할 수 있다. 만약 사용자 A와 B가 동일한 파일을 가지고 있다고 가정할 때, 클라우드 서버에는 단일 파일만 저장하게 되고, 해당 파일은 사용자 A와 사용자 B가 동시에 사용할 수 있는 파일이라는 매핑구조를 정의하는 형태로 구현한다.

이 과정에서 사용자 B가 파일을 삭제하였을 경우에는 그림 2와 같이 매핑정보를 제거하는 것으로 삭제 처리가 완료된다. 이후 사용자 A 또한 파일을 삭제하였을 경우 사용자 A의 매핑정보와 파일을 모두 제거하게 되는 방식으로 중복제거가 구현된다.

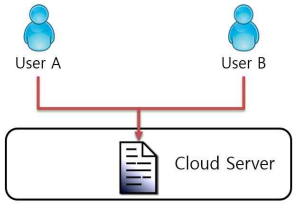

그림 1. 중복 제거 기술
Fig. 1. Deduplication technique

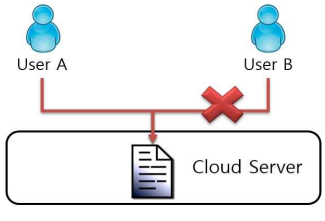

그림 2. 중복제거에서의 파일 삭제
Fig. 2. Delete files from deduplication

2.2 중복제거 보안기술

2.2.1 Convergent Encryption

중복제거 기술은 근본적으로 보안에 취약한 구조를 가지고 있다. 즉, 여러 사용자의 파일을 서버에서 단일하게 가지고 있게 된다면 구조적으로 다양한 보안 위협에 노출될 수 있다. 따라서, 중복제거 기술에 대한 적절한 보안 대책이 필요한 상황이다.

Douceur 등의 논문에서는 중복제거를 위한 수렴 암호화(Convergent Encryption) 기법이 제안되었다[1]. 수렴 암호화 기법은 원본 메시지의 해쉬를 이용하여 암호화 키를 유도하는 방식이다. 예를 들어 원본 메시지가 M이라면, 암호화 키는 M에 해쉬를 적용한 H(M)으로 생성할 수 있다. 이러한 기법은 기본적으로 파일단위로 각각 다른 키를 생성할 수 있으며, 원본 파일의 내용을 알지 못하는 경우 암호화 키를 파악하기 어렵기 때문에 안전하게 파일을 관리할 수 있고, 별도의 키 관리가 필요하지 않다는 장점이 있다. 그러나, 기본적으로 수렴 암호화 방식은 엔트로피가 낮거나 작은 데이터의 경우는 전수조사 공격에 취약하다는 문제점이 존재한다[4].

2.2.2 DupLESS

Bellare 등의 논문에서는 전수조사 공격을 막기 위한 기법이 제안되었다[5]. DupLESS는 수렴 암호화의 취약성을 보완하기 위하여, 별도의 키 서버를 두고, 해당 서버 측에서 클라이언트와 상호작용을 통하여 비밀키를 생성하는 방식의 서버측 중복제거 기술이다. 그러나 수렴 암호화 기법과 DupLESS 기법은 중복제거된 파일에 대한 암호화 키 생성에만 초점을 맞추고 있다는 점에서 한계가 있다.

실질적으로 중복제거 기술의 보안 문제는 단순히 파일을 적절한 방식의 키 관리로 암호화하는 것만으로 해결할 수 없다. 특히, 사용자 프라이버시 보호와 중복제거 기술은 서로 상충하는 성질이 있다. 3장에서는 이러한 중복제거 기술에 따른 사용자 프라이버시 침해 위협 및 기타 보안 문제점에 대하여 논의하고자 한다.

III. 중복제거기술의 프라이버시 이슈사항

3.1 사용자-파일간 매핑구조에 따른 문제

근본적으로 클라우드 스토리지에 중복제거 기법을 적용하려면, 2장에서 언급한 것과 같이 사용자-파일간 매핑구조가 확립되어야 한다. 그러나, 사용자-파일 매핑구조 문제는 심각한 프라이버시 문제가 존재한다. 단순 매핑구조만으로 보았을 때는 어떤 프라이버시 문제가 발생할 수 있는지 쉽사리 와닿지 않을 수 있으나, 사용자에게 민감성이 높은 파일이라면 사용자-파일간 매핑구조 자체만으로 사용자의 프라이버시의 침해 위협이 발생할 수 있다.

개인정보보호법 제23조(민감정보의 처리 제한)의 1항에 따르면, 개인정보처리자는 사상·신념, 노동조합·정당의 가입·탈퇴, 정치적 견해, 건강, 성생활 등에 관한 정보, 그 밖에 정보주체의 사생활을 현저히 침해할 우려가 있는 정보를 민감정보로 정의하고 있다. 또한, 동조 2항에 따르면 민감정보의 안전성 확보에 필요한 조치를 하여야 함을 명시하고 있다.

즉, 업로드 대상 파일이 민감정보가 담긴 파일이거나, 업로드 내역의 노출 자체가 민감한 상황인 경우에는 강력한 보안 대책이 필요하다. 특히, 특정 정당 지지 문건이나, 사상적 내용, 특정 질병 및 성생활 관련 파일을 업로드하는 경우, 중복제거 처리 시에는 사용자와 파일간의 매핑구조 자체가 프라이버시 침해에 해당될 수 있다. 특히, 중복제거 환경에서는 다수의 사용자와 단일 파일이 매핑된 구조로 처리되어 있으며, 이는 서버 측에서 특정 파일을 올린 사용자 리스트를 모두 확보할 수 있다는 것을 의미하며, 이러한 특성은 심각한 프라이버시 침해 요인이 될 수 있다. 심지어 클라우드 서버가 일종의 감시 역할을 수행할 가능성도 배제할 수 없다.

특정 사용자가 어떤 파일을 올렸는지 여부는 그 사람의 고유한 프라이버시 영역이다. 만약 클라우드 서버 관리자라 할지라도, 특정인이 올린 파일 리스트를 모두 확보하거나, 반대로 특정 파일을 올린 업로더 리스트를 확보하는 경우는 적극적으로 프라이버시를 침해하는 행위로 간주될 수 있다.

3.2 기밀성 확보와 중복제거의 비양립성

일반적으로 클라우드 시스템에서는 서버 측에서 암호화를 기반으로 데이터 기밀성 서비스를 제공한다. 즉, 평문 파일을 클라이언트 측에서 수신하면, 서버 측에서 암호화 처리 후 스토리지에 보관하는 것으로 해킹에 대한 대비를 하는 것이 일반적이다.

이러한 경우, 사용자가 다운로드를 요청하면 서버측에서 복호화된 평문 파일을 사용자에게 전송하게 된다. 즉, 서버 측에서는 암호화 키를 가지고 있으며 필요시 복호화도 가능하다는 것을 의미한다.

따라서 클라우드 서버측에서의 높은 수준의 권한이 있는 관리자라면 필요시 해당 키를 기반으로 파일을 복원할 가능성도 존재할 수 있다[6]-[9].

사용자의 프라이버시를 보호하기 위한 강력한 보안 대책으로, 클라이언트 측에서 파일 암호화를 사전에 적용한 이후 서버에 올리는 방법을 고려할 수 있다. 이러한 경우 서버 측에서도 파일은 안전하게 관리할 수 있게 되나, 데이터 중복제거 메커니즘이 작동하지 않는다. 그림 3에서와 같이, 사용자 A와 사용자 B가 동일한 파일을 올렸음에도 불구하고, 각각 다른 키로 암호화되어 실질적으로는 서버에서는 다른 파일로 인식하게 되어 중복제거가 불가능하다. 즉, 클라이언트 측에서의 사전 암호화를 통한 완전한 기밀성 확보와 중복제거를 통한 용량 절감을 동시에 달성하는 것은 실질적으로 매우 어렵다.

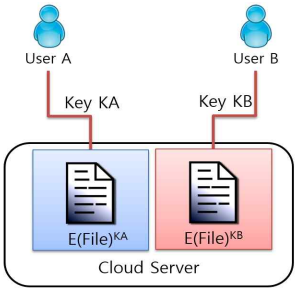

그림 3. 클라이언트에서의 사전 암호화
Fig. 3. Client-side encryption

3.3 파일의 소유권 문제

파일의 소유권 및 공유 문제는 서버 입장에서의 보안 정책과 관련되어 있다. 중복제거가 적용된 클라우드 환경에서 사용자 A와 사용자 B가 동일한 파일을 올렸다면, 실제로 저장된 파일은 하나이다. 이 경우, 해당 파일의 소유권이 누구한테 있는지를 고려해볼 필요가 있다. 반대로 생각하자면, 파일 다운로드를 요청하는 사용자는 자신이 해당 파일에 대한 소유권이 있다는 것을 증명할 필요가 있다.

이러한 문제는 특정 파일을 기존에 올린 사람들이 누구인지에 대한 리스트를 엄격히 관리하면 해결이 가능하다. 그러나 이러한 방식은 앞서 언급한 사용자-파일 매핑구조에 따른 프라이버시 문제를 그대로 가지게 된다. 특정 사용자가 특정 파일의 소유권을 행사하는 것은, 반대로 말해 해당 파일은 해당 사용자가 이전에 올린 파일이라는 것을 의미하며, 이것은 달리 말해 특정 파일을 올린 사람이 누구인지를 서버에서 명확히 알 수 있다는 것이다.

다시 예를 들면, 특정 정당 지지 홍보물, 특정 질병 및 성생활 관련 자료, 특정 사상 및 신념이 담긴 자료 등의 다양한 민감정보를 올린 사용자에 대한 리스트를 서버 측에서 메타 분석을 통하여 그대로 알 수 있다는 의미이다. 따라서 기존의 중복제거 방식이 적용된 클라우드 환경은 구조적으로 프라이버시 문제를 안고 있게 된다. 이러한 소유권 문제로 인하여 현재까지의 중복제거가 적용된 클라우드 서비스는 엄밀한 관점에서 완전한 프라이버시 보호 방안을 제공해 주지 않고 있다[10]-[13].

3.4 내부자 공격 취약성

클라우드 환경은 근본적으로 내부자에 의한 개인정보 노출 발생의 위험성이 존재한다. 과거에도 내부자에 의한 데이터 노출 사례가 다수 발생한 바 있어 내부자 공격은 실질적으로 클라우드 환경에서의 매우 큰 보안 위협으로 손꼽히고 있다.

일반적인 클라우드 환경은 데이터 암호화를 통한 최소한의 보안 장치를 마련하고 있다. 그러나 최고 수준의 권한을 가진 내부 관리자라면 파일 복호화 키를 수집할 수 있다는 가정을 배제할 수 없으며, 혹은 암호화 키를 모른다고 하더라도 메타데이터 분석을 통하여 파일과 사용자의 매핑 관계를 수집하게 될 수 있다. 이는 곧, 특정 파일을 올린 사용자 리스트를 모두 확보할 수 있거나, 반대로 특정

사용자가 올린 파일 리스트를 메타데이터 분석을 통하여 모두 확보할 수 있다는 것을 의미한다[14].

이렇게 중복제거와 프라이버시 문제를 동시에 해결하는 것은 실질적으로 매우 어렵다. 본 논문에서는 중복제거와 프라이버시 문제를 동시에 해결하는 방안을 제안한다.

IV. 새로운 중복제거 기법 제안

본 장에서는 기존 중복제거 방식에서의 프라이버시 문제를 해결하기 위한 새로운 기법을 제안한다.

4.1 제안 방식 개요

기존의 중복제거 환경에서의 프라이버시 이슈는 근본적으로 사용자와 파일이 매핑된 구조에서 발생한다. 실질적으로, 파일과 사용자의 매핑구조를 완전히 제거한다면 특정 파일에서 해당 파일을 업로드한 사용자 리스트나, 혹은 그 반대로 특정 사용자가 올린 파일 리스트를 확보할 수 없게 된다. 따라서 본 논문에서는 이러한 사용자-파일 리스트와 파일의 매핑관계를 원천적으로 차단하여 서버상에서의 메타데이터 및 파일구조 분석만을 통해서는 사용자를 유추할 수 없는 방식을 제안하였다[15][16].

제안한 방식에서는 PIN, RefID, FHV를 기반으로 사용자가 파일에 대한 업로드/다운로드 수행이 가능하다. 여기에서, PIN은 사용자만 알고 있는 값이며, RefID는 메타 서버에서 소유한 값으로 PIN을 기반으로 구성된다. 또한, FHV 값은 파일의 해쉬 결과값에 한번 더 해쉬를 취한 값으로, FHV를 기반으로 RefID를 추측할 수 없으며, 반대로 RefID를 기반으로 FHV 및 PIN도 추측할 수 없다.

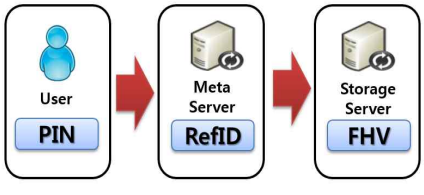

그림 4. 제안 방식 개요
Fig. 4. Overview of the proposed scheme

즉, 메타서버와 스토리지 서버가 보관하고 있는 RefID와 FHV만을 통해서는 어떤 사용자-파일 매핑 정보도 얻을 수 없으며, RefID를 기반으로 FHV를 추출하려면 반드시 사용자가 알고 있는 PIN값이 요구된다.

제안 방식의 설명을 위한 약어는 표 1과 같다.

표 1. 약어
Table 1. Notation

Abbreviation	Description
UserID	User's ID
SID	Session ID
PK	Pre-shared encryption key
File	File to upload / download
Filehash	The hash value of the original file
FHV	Hash value for Filehash
FilePath	The full path of the destination file
PIN	Value required to verify user ownership
RefID	Reference value stored in the meta server
$H(\cdot)^K$	Result of hashing a specific value
$E(\cdot)^K$	Result of encryption processing with key K
$D(\cdot)^K$	Result of decryption processing with key K

4.2 시스템 구성

그림 5는 클라이언트와 클라우드 서버의 구성을 나타낸다. 클라이언트에서는 모바일, 노트북, PC 등 다양한 장치에서 클라우드 서버에 접속할 수 있다.

한편, 클라우드 서버에는 메타서버와 스토리지 서버가 있다. 메타서버와 스토리지 서버는 엄격히 분할되어야 한다.

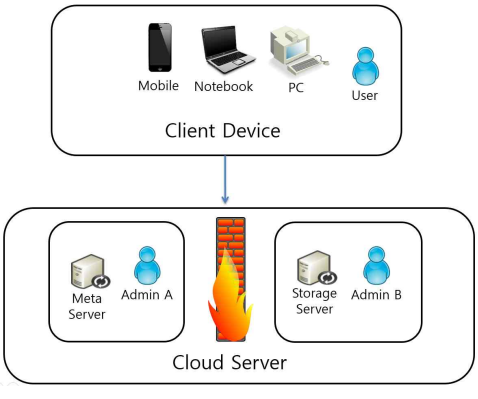

그림 5. 시스템 구성
Fig. 5. System configuration

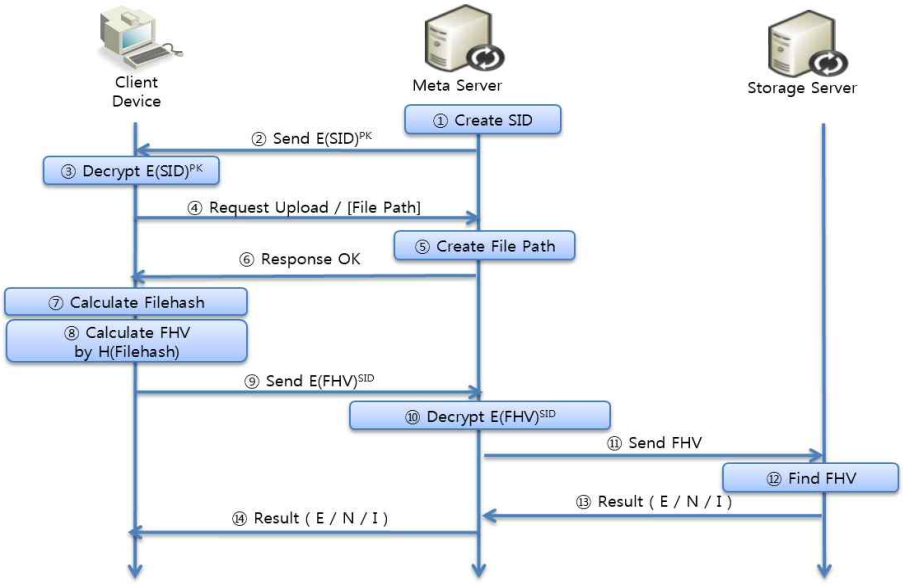

그림 6. 파일 중복 체크 프로토콜
Fig. 6. File duplicate check protocol

즉, 한대의 서버에 같이 구성될 수 없으며, 서로 다른 서버에 구성되어야 한다. 또한, 물리적으로도 메타서버와 스토리지 서버가 분할되는 것을 권장한다. 여기에서, 메타서버와 스토리지 서버의 담당 관리자는 별도로 구성하여야 한다.

4.3 파일 중복 체크 프로토콜

파일 중복 체크 프로토콜은 서버에 실제로 파일이 존재하는지를 확인하는 단계이다. 만약, 서버에 동일한 파일이 존재할 경우는 파일 업로드 과정이 필요하지 않으므로 효율적이다. 파일 중복 체크 프로콜을 설명하면 다음과 같다.

① 메타 서버는 세션아이디인 SID를 생성한다.
② 메타 서버는 SID를 사전 공유된 PK로 암호화하고 클라이언트 서버에 전송한다.
③ 클라이언트는 복호화를 통하여 SID를 얻는다.
④ 클라이언트는 특정 경로에 업로드를 요청한다.
⑤ 메타서버는 요청된 파일 경로를 생성한다.
⑥ 경로 생성이 완료되었음을 클라이언트에 알린다.
⑦ 클라이언트는 업로드 대상 파일을 해쉬처리하여 파일의 해쉬값을 추출한다.
⑧ 클라이언트는 추출된 파일의 해쉬값에 해쉬처리를 한번 더 취하여 FHV를 생성한다.
⑨ 클라이언트는 FHV를 암호화하여 전송한다.
⑩ 메타 서버는 복호화를 통하여 FHV를 얻는다.
⑪ 메타 서버는 스토리지 서버에 FHV를 전송하여, 해당 파일이 이미 존재하는지 여부를 질의한다.
⑫ 스토리지 서버는 파일의 등록 여부를 확인한다.
⑬ 해당 결과값을 E, N, I 중 하나로 구분하여 리턴한다. 여기에서 E는 Exist의 약자로써, 파일이 이미 존재하고 있다는 것을 의미하고, N은 Not Exist의 약자로써, 파일이 존재하지 않는다는 것을 의미한다. 또한, I는 Incomplete의 약자로써, 파일이 일부만 올라와 있다는 것을 의미한다.
⑭ 메타 서버는 E/N/I값을 클라이언트에 응답한다.

파일 중복 체크 결과를 정리하면 표 2와 같다. E를 수신할 경우는 서버상에 동일한 파일이 이미 업로드된 상태이므로, 별도로 추가 업로드는 필요하지 않다. 따라서 클라이언트는 업로드 과정 없이 매핑 과정만 수행하면 된다. 그러나 N 혹은 I를 수신하였을 경우, 서버상에 업로드가 필요하다는 의미이므로 클라이언트는 파일 업로드를 수행하여야 한다.

만약, 결과가 I인 경우는 부분적인 파일 업로드만 수행된 단계이므로, 서버상에 업로드된 파일을 제외한 나머지 부분을 추가적으로 업로드를 진행한다.

표 2. 파일 중복 체크 결과 설명
Table 2. Notation

Result	Description
E	File already exists on server
N	File does not exist on the server
I	File has partially uploaded to the server

4.4 파일 업로드/매핑 프로토콜

파일 업로드/매핑 프로토콜은 다음과 같다.

① 절차 ① 및 ②는 파일 중복 체크 결과값이 N과 I일 경우에만 해당한다. 즉, 파일 중복 파일 결과가 E일 경우는 별도의 업로드 과정이 필요 없다. 그러나, N, I의 값을 수신한 경우는 스토리지 서버로 실제 파일 업로드를 수행한다.
② 업로드가 완료되었음을 클라이언트 측에 알린다.
③ 사용자는 클라이언트를 통해 PIN을 입력한다.
④ 클라이언트는 PIN을 기반으로, RefID를 생성한다. RefID는 다음과 같은 식에 의하여 생성된다.

$$RefID = E(H(UserID) \oplus (H(PIN) \oplus FHV))^{H(PIN)} \quad (1)$$

⑤ 스토리지 서버는 파일의 해쉬값을 추출한다.
⑥ 스토리지 서버는 수신된 파일의 무결성을 파일의 해쉬값과 FHV값과의 비교를 통하여 파일의 무결성을 검증한다.
⑦ FHV와 업로드된 파일을 매핑하여 저장한다.
⑧ 클라이언트는 ④에서 생성한 RefID를 메타서버에 전송한다.
⑨ 메타서버는 수신된 RefID를 저장한다.
⑩ 메타서버는 RefID와 FilePath 값을 매핑한다.
⑪ 메타서버는 업로드가 완료되었음을 알린다.

4.5 파일 다운로드 프로토콜

파일 다운로드 프로토콜은 다음과 같다.
① 메타 서버는 세션아이디인 SID를 생성한다.
② 메타 서버는 SID를 PK로 암호화하고 클라이언트 서버에 전송한다.
③ 클라이언트는 복호화를 통하여 SID를 얻는다.
④ 클라이언트는 메타서버에 다운로드를 요청한다.
⑤ 메타서버는 해당 경로에 파일이 존재함을 확인한다. 존재하지 않을 경우에는 여기서 종료한다.
⑥ 메타서버는 클라이언트에 해당 경로에 파일이 존재함을 알린다.
⑦ 사용자는 클라이언트를 통하여 PIN을 입력한다.
⑧ 클라이언트는 PIN의 해쉬값을 SID 키로 암호화하여 메타 서버에 전송한다.

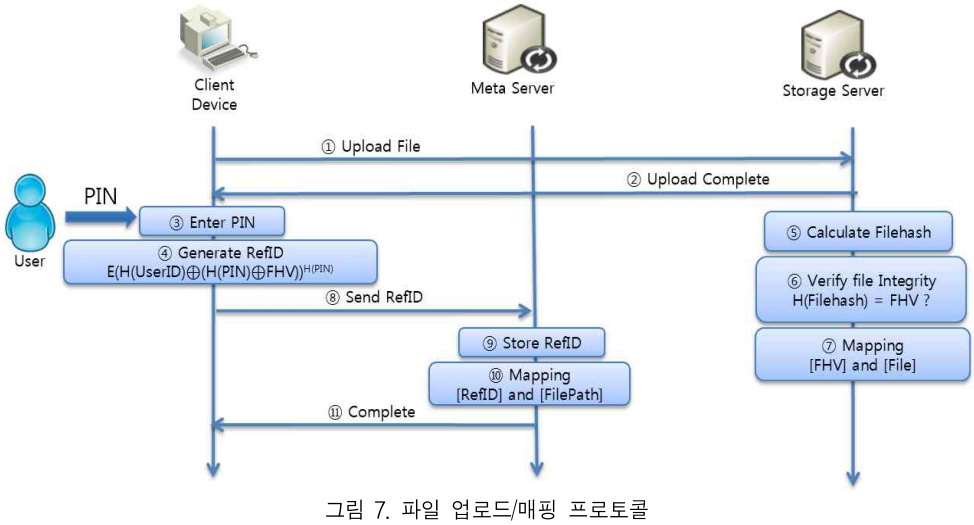

그림 7. 파일 업로드/매핑 프로토콜
Fig. 7. File upload/mapping protocol

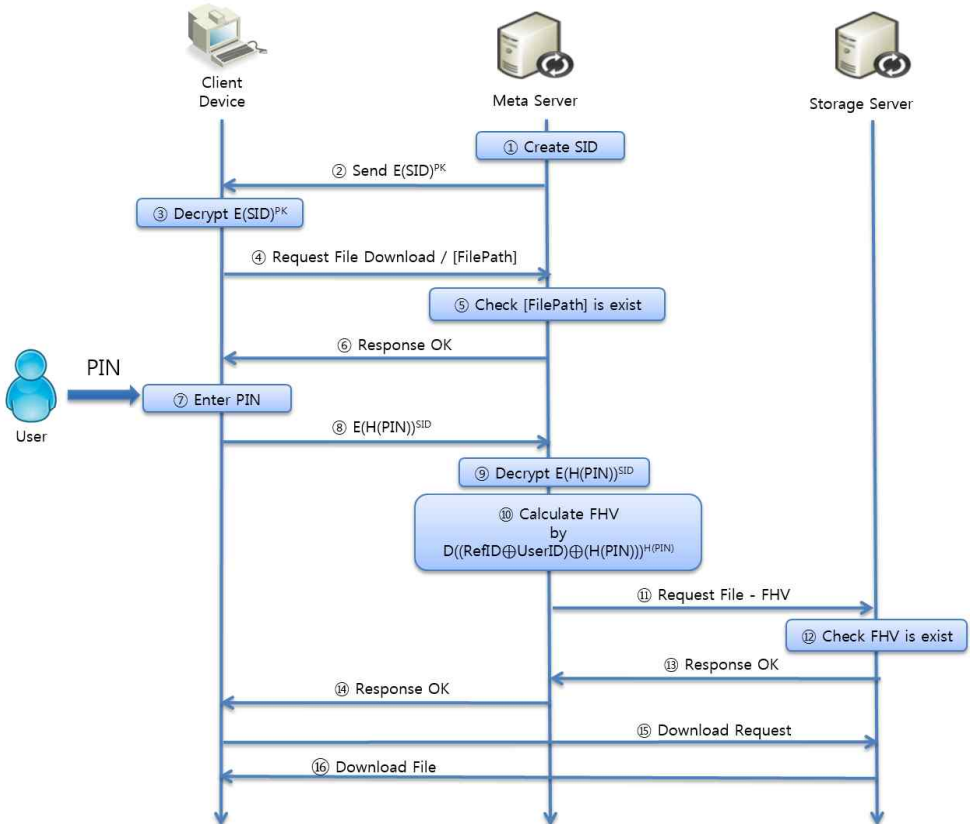

그림 8. 파일 다운로드 프로토콜
Fig. 8. File download protocol

⑨ 메타 서버는 H(PIN)을 복호화한다.
⑩ 메타 서버는 RefID와 H(PIN) 값을 기반으로 아래의 식을 통하여 FHV를 추출한다.

$$D((RefID \oplus UserID) \oplus (H(PIN)))^{H(PIN)} \quad (2)$$

⑪ 메타 서버는 스토리지 서버에 FHV를 전달한다.
⑫ 스토리지 서버는 FHV와 매핑된 특정 파일이 존재하는지 여부를 확인한다.
⑬ 스토리지 서버는 해당 FHV로 매핑된 파일이 있음을 메타 서버에 응답한다.
⑭ 메타서버는 클라이언트에 다운로드 준비 상태임을 알린다.
⑮ 클라이언트는 스토리지 서버에 파일의 다운로드를 요청한다.
⑯ 클라이언트는 스토리지 서버에서 파일을 다운로드 처리하여 완료한다.

Ⅴ. 안전성 및 효율성 분석

5.1 프라이버시 보호 측면

기존의 중복제거 방식은 파일 암호화 기법을 적용하여 파일 자체에 대한 보안성은 확보하고 있으나, 사용자-파일 매핑구조는 그대로 노출하고 있다. 이러한 문제에 따라 메타정보 분석이 가능하며, 심각한 사용자 프라이버시 침해로 이어질 수 있다.

본 논문에서 제안한 방식은 사용자와 파일의 정보가 완전히 분리된다. RefID는 메타서버에서 저장되고, FHV 값은 스토리지 서버에서 저장된다.

RefID는 PIN에 의하여 XOR 처리 및 암호화된 값이며, 해당 암호화된 값을 관리자도 복호화할 수 없다. 복호화를 위해서는 H(PIN)값이 필요하나, 이 정보는 서버에 저장되지 않는다[17]-[20].

따라서 RefID를 기반으로 FHV를 추정할 수 없으

며, 반대로 FHV를 기반으로 RefID를 연결지을 수 없다. 만약 해당 정보를 완전히 구성하여 파일을 다운로드하려면 반드시 사용자가 알고 있는 PIN 값이 필요하다. 특히, RefID는 H(PIN)값을 키로 암호화되어 있어 분석을 더욱 어렵게 한다. 즉, PIN 값을 알지 못하면 서버에 있는 정보를 이용해서는 사용자 정보와 파일의 매핑관계를 재구성할 수 없으므로 사용자의 프라이버시를 안전하게 보호한다.

5.2 소유권 문제 해결

기존의 중복제거 기술은 파일의 소유권 문제를 파일에 대한 사용자 매핑구조를 기반으로 처리하는 경우가 일반적이다. 그러나, 이러한 방법은 구조적으로 프라이버시 침해 가능성이 있어 문제가 된다.

본 논문에서 제안한 방식에서는 PIN값이 사용자의 특정 파일 소유권을 증명하는 단서가 된다. 만약, 권한이 없는 자가 파일에 대한 다운로드를 시도하는 경우에는 PIN를 알지 못하므로 RefID에서 정상적인 FHV를 추출할 수 없으며, 파일 다운로드 절차를 정상적으로 수행할 수 없다. 그러나 정상적인 사용자의 경우 업로드시에 적용한 PIN을 기반으로 RefID에서 FHV 값을 추출하여 다운로드를 수행할 수 있다. 이 경우, RefID로부터 해당 파일의 FHV를 추출하는 것은 PIN을 알고 있는 사용자만 가능하므로, 해당 사용자는 파일의 소유권 증명이 가능하다. 즉, 비밀정보는 PIN이며, 서버에는 비밀정보가 저장되어지지 않는다는 특성을 통하여 프라이버시를 보장한 상태에서 파일 소유권 문제를 해결할 수 있다.

5.3 내부자 공격

기존의 중복제거 기술에서는 파일과 사용자간의 매핑구조를 메타정보로서 가지고 있다. 이러한 경우 열람 권한을 가지고 있는 관리자는 메타정보에 대한 수집이 가능하다는 측면에서 내부자 공격에 매우 취약하다. 즉, 클라우드 서버 관리자는 특정 파일에 대한 업로더 리스트, 혹은 특정 사용자의 전체 파일 리스트 등을 용이하게 파악할 수 있다.

제안한 방법에서는 내부자 공격에 의해 메타서버와 스토리지 서버 전체가 노출된 경우를 가정하여도 공격자는 사용자의 파일 리스트, 혹은 특정 파일에 대한 업로드 사용자 리스트를 추출할 수 없다. 서버에 저장된 정보는 RefID와 FHV이며, 해당 값 자체는 상호간 연결성을 가지고 있지 않다. 특히, RefID 값은 H(PIN)을 키로 암호화되어 있어 이는 메타 분석을 더욱 어렵게 한다. 따라서 내부자에 의한 데이터 전수 노출이 발생하더라도 PIN 정보를 알지 못하면 파일과 사용자간 관계를 연관지을 수 없으므로, 내부자 공격 문제를 해결할 수 있다.

5.4 스니핑 공격

제안한 방식은 프로토콜 수행 과정에서 클라이언트와 서버 간의 파라미터가 암호화되어 전송된다. 이는 해커가 스니핑 공격을 수행하더라도 안전을 보장한다. 특히, 프로토콜 과정에서 전달되는 파라미터를 암호화하는 키로써, 메타서버가 생성하는 1회성 값인 SID를 사용한다. 따라서 해커가 스니핑에 따른 재연공격을 수행하더라도 서버와 클라이언트간 사전 공유된 PK를 알지 못하면 공격자는 파일에 대한 업로드 및 다운로드 프로토콜을 정상적으로 수행할 수 없게 된다[21]-[26].

5.5 무결성 측면

제안한 방식은 파일의 무결성을 보장하기 위해서 서버측에서의 검증 절차를 한번 더 거치게 된다. FHV값은 파일의 해쉬값에 대한 재해쉬값이며, 서버측에서는 업로드된 파일에 대한 두번의 해쉬 결과값과 FHV값이 일치하는지 여부를 확인하여 업로드된 파일의 무결성 여부를 확인할 수 있다. 즉, FHV값은 다운로드 프로토콜 수행 과정에서 파일 소유권 증명에 활용됨과 동시에 업로드된 파일에 대한 무결성 검증의 역할도 수행할 수 있다는 특징이 있다. 특히, 스토리지 서버에 저장된 파일에 대한 변조 여부를 감지할 수 있다는 장점도 존재한다.

5.6 효율성 측면

중복제거 방식은 일반적으로 Merkle-Tree 기반의 파일 해쉬값을 생성한다. 파일 해쉬 생성 부분은 실

질적으로 중복처리 과정에서 가장 시간이 많이 소요되는 부분이므로, 여기서는 파일 중복체크 시의 Merkle-Tree 생성속도와 제안한 알고리즘에서의 RefID 추출시간을 합산하여 비교한다. 아래 표 3은 성능 측정을 위한 환경을 나타낸다.

표 3. 성능 측정 환경
Table 3. Performance test environment

CPU	Intel i5-3470 @3.2GHz
RAM	4GB
O/S	Windows 7
Language	C++
Hash Algorithm	Merkle Tree (SHA-1)

표 4. 성능 측정 결과 비교
Table 4. Comparison of performance measurement results

Method	50K	100K	1M	5M	10M
Merkle Tree	150	142	317	749	1790
Merkle Tree + RefID	241	242	358	896	1901

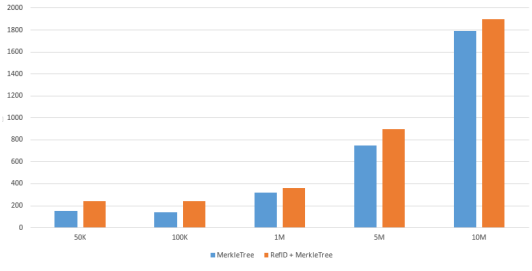

그림 9. 성능 측정 결과
Fig. 9. Perfomance measure result

제안한 논문은 기존의 중복제거 기술에 보안 기능을 더한 것으로, 실질적으로 처리시간은 기존의 중복제거 기술에 비하여 RefID 연산 등 보안처리에 필요한 시간이 추가된다. 그러나 실질적으로 기존의 Merkle Tree 기반의 중복제거 방식에 있어 현저한 성능저하를 보이지는 않으며, 프라이버시 보호 및 보안 기능을 가지고 있다는 장점을 가지고 있으므로, 기존 중복제거 방식에 대한 보안성 확보를 위하여 본 방식으로 대체할 수 있을 것으로 보인다.

Ⅵ. 결 론

클라우드 스토리지 시스템의 구축에 있어 중복제거 기술은 필수적인 요소기술로써 자리잡고 있다. 중복제거 기술은 스토리지의 용량을 획기적으로 절감할 수 있는 기술로써 직접적인 비용 절감의 장점이 있어 향후에도 클라우드의 핵심 기술로서 작용할 것이다. 그러나 중복제거 기술은 구조적으로 사용자의 프라이버시 침해 문제를 안고 있으며, 기존에 제안되었던 중복제거 보안 기술은 파일 암호화 자체에만 초점을 두었던 측면이 있다[27]-[29].

본 논문에서는 사용자-파일간 매핑구조에 따른 프라이버시 문제를 지적하고, 이를 보완하기 위한 새로운 기법을 제안하였다. 제안한 기법은 메타서버와 스토리지 서버에 각각 RefID와 FHV값만을 보관하며 두 값 자체만으로는 서로 연결성을 갖지 않는다는 특징이 있어 프라이버시를 안전하게 보호하고, 내부자 공격에 따른 메타정보 노출에도 안전을 보장할 수 있다. 또한, 사용자의 PIN값을 활용함으로써 파일의 소유권과 프라이버시 문제를 동시에 해결하였다.

제안한 방식의 설명을 위해, 2장에서는 기존의 중복제거 기술에 대한 관련연구를 살펴보았고, 3장에서는 중복제거 기술 적용에 따른 프라이버시 침해 및 기타 발생할 수 있는 보안 문제를 지적하였다. 4장에서는 새로운 중복제거 기법을 제안하였고, 5장에서는 제안한 기법의 안전성을 분석하였다.

중복제거 기술은 클라우드 스토리지 구축에 있어 반드시 필요한 기술이며, 현재 많은 클라우드 서비스에 중복제거 기술이 적용되어 있다. 향후 4차산업 시대에서의 안전한 클라우드 서비스 제공을 위해서는 중복제거 기술에서 발생할 수 있는 프라이버시 문제에 대한 연구가 지속적으로 필요할 것이다.

References

[1] J. R. Douceur, A Adya, W. J. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system", Distributed Computing Systems, Proceedings of 22nd International Conference on. IEEE, 14 pages, Jul. 2002.

[2] Namje Park and Namhi Kang, "Mutual Authentication Scheme in Secure Internet of

Things Technology for Comfortable Lifestyle", Sensors, Vol. 16, No. 1, pp. 1-16, Dec. 2015.

[3] Donghyeok Lee and Namje Park, "Geocasting-based synchronization of Almanac on the maritime cloud for distributed smart surveillance", The Journal of Supercomputing, Vol. 73, No. 3, pp. 1103-1118, Mar. 2017.

[4] Kyungsu Park, Ji Eun Eom, Jeongsu Park, and Dong Hoon Lee, "Secure and Efficient Client-side Deduplication for Cloud Storage", Journal of the Korea Institute of Information Security & Cryptology, Vol. 25, No. 1, pp. 83-94, Feb. 2015.

[5] Bellare, Mihir, Sriram Keelveedhi, and Thomas Ristenpart, "DupLESS: Server-Aided Encryption for Deduplicated Storage", IACR Cryptology ePrint Archive, 2013.

[6] Donghyeok Lee and Namje Park, "Teaching Book and Tools of Elementary Network Security Learning using Gamification Mechanism", Journal of the Korea Institute of Information Security & Cryptology, Vol. 26, No. 3, pp. 787-797, Jun. 2016.

[7] Namje Park, Jin Kwak, Seungjoo Kim, Dongho Won, and Howon Kim, "WIPI Mobile Platform with Secure Service for Mobile RFID Network Environment", Advanced Web and Network Technologies, and Applications, LNCS, Vol. 3842, pp. 741-748, Jan. 2006.

[8] Hyun-il Kim, Cheolhee Park, Dowon Hong, and Changho Seo, "Encrypted Data Deduplication Using Key Issuing Server", Korea Information Science Society, Vol. 43, No. 2, pp. 143-151, Feb. 2016.

[9] Cheolhee Park, Dowon Hong, Changho Seo, and Ku-Young Chang, "Privacy Preserving Source Based Deduplication In Cloud Storage", Korea Institute Of Information Security And Cryptology, Vol. 25, No. 1, pp. 123-132, Feb. 2015.

[10] Namje Park and Marie Kim, "Implementation of load management application system using smart grid privacy policy in energy management service environment", Cluster Computing, Vol. 17, No. 3, pp. 653-664, Sep. 2014.

[11] J. Li, X. Chen, M., Li, J. Li, P. P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management", IEEE transactions on parallel and distributed systems, Vol. 25, No. 6, pp. 1615-1625, Jun. 2014.

[12] Namje Park and Hyo-Chan Bang, "Mobile middleware platform for secure vessel traffic system in IoT service environment", Security and Communication Networks, John Wiley&Sons Ltd, Vol. 9, No. 6, pp. 500-512, Apr. 2016.

[13] Bellare, Mihir, Sriram Keelveedhi, and Thomas Ristenpart, "Message-locked encryption and secure deduplication", Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, 2013.

[14] Namje Park, Hongxin Hu, and Qun Jin, "Security and Privacy Mechanisms for Sensor Middleware and Application in Internet of Things (IoT)", International Journal of Distributed Sensor Networks, Vol. 2016, Article ID 2965438, 3 pages, 2016.

[15] Cheolhee Park, Dowon Hong, and Changho Seo, "A Secure and Practical Encrypted Data De-duplication with Proof of Ownership in Cloud Storage", Korea Information Science Society, Vol. 43, No. 10, pp. 1165-1172, Oct. 2016.

[16] Namje Park, "UHF/HF Dual-Band Integrated Mobile RFID/NFC Linkage Method for Mobile Device-based Business Application", The Journal of The Korean Institute of Communication Sciences, Vol. 38, No. 10, pp. 841-851, Oct. 2013.

[17] Namje Park, "Design and Implementation of Mobile VTS Middleware for Efficient IVEF Service", Journal of KICS, Vol. 39C, No. 6, pp. 466-475, Jun. 2014.

[18] 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, Mar. 2014.

[19] P. Puzio, R. Molva, M. Onen, and S. Loureiro, "ClouDedup: secure deduplication with encrypted data for cloud storage", Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on. Vol. 1. IEEE, pp. 363-370, Dec. 2013.

[20] Namje Park, Jungsoo Park, and Hyoungjun Kim, "Inter-Authentication and Session Key Sharing Procedure for Secure M2M/IoT Environment", International Information Institute(Tokyo) Information, Vol. 18, No. 1, pp. 261-266, Jan. 2015.

[21] Donghyeok Lee and Namje Park, "A Secure Almanac Synchronization Method for Open IoT Maritime Cloud Environment", Journal of Korean Institute of Information Technology Vol. 15, No. 2, pp. 79-90, Feb. 2017.

[22] Ma, Jingwei, Gang Wang, and Xiaoguang Liu., "DedupeSwift: Object-Oriented Storage System Based on Data Deduplication", Trustcom/Big Data SE/I SPA, 2016 IEEE, pp. 1069-1076, Aug. 2016.

[23] Tang, Yang, and Junfeng Yang, "Secure Deduplication of General Computations", USENIX Annual Technical Conference, pp. 319-331, Jul. 2015

[24] Young-Jun Yoo, Sun-Jeong Kim, and Young Woong Ko, "Cloud File Synchronization Scheme using Bidirectional Data Deduplication", Journal of KIIT, Vol. 12, No. 1, pp. 103-110, Jan. 2014.

[25] Byung Kwan Kim, Young Woong Ko, and Kwang Mo Lee, "Performance Enhancement for Data Deduplication Server Using Bloom Filter", Journal of KIIT. Vol. 12, No. 4, pp. 129-136, Apr. 2014.

[26] Donghyeok Lee and Namje Park, "Electronic identity information hiding methods using a secret sharing scheme in multimedia-centric internet of things environment", Personal and Ubiquitous Computing, Vol. 22, No. 1, pp. 3-10, Feb. 2017.

[27] Donghyeok Lee, Namje Park, Geonwoo Kim, and Seunghun Jin, "De-identification of metering data for smart grid personal security in intelligent CCTV-based P2P cloud computing environment", Peer-to-Peer Networking and Applications, DOI 10.1007/s12083-018-0637-1, pp. 1-10, Mar. 2018.

[28] Namje Park, "The Core Competencies of SEL-based Innovative Creativity Education", International Journal of Pure and Applied Mathematics, Vol. 118, No. 19, pp. 837-849, Jan. 2018.

[29] Namje Park, "STEAM Education Program : Training Program for Financial Engineering Career", International Journal of Pure and Applied Mathematics, Vol. 118, No. 19, pp. 819-835, Jan. 2018.

저자소개

이 동 혁 (Donghyeok Lee)

2007년 2월 : 동국대학교 전자상거래기술전공 공학석사
2018년 2월 : 제주대학교 컴퓨터교육전공 공학박사
2007년 6월 ~ 2008년 5월 : 한국전자통신연구원 연구원
2008년 11월 ~ 2015년 6월 : (주) KT 플랫폼개발단 과장
2018년 3월 ~ 현재 : 제주대학교 과학기술사회연구센터 학술연구교수
관심분야 : IoT 보안, 클라우드 보안, DB 보안 등

박 남 제 (Namje Park)

2008년 2월 : 성균관대학교 컴퓨터공학과 박사
2003년 4월 ~ 2008년 12월 : 한국전자통신연구원 선임연구원
2009년 1월 ~ 2009년 12월 : University of California at Los Angeles(UCLA) Post-doc.
2010년 1월 ~ 2010년 8월 : Arizona State University (ASU) Research Scientist
2010년 9월 ~ 현재 : 제주대학교 교육대학 초등컴퓨터교육전공 교수, 과학기술사회연구센터장
관심분야 : 컴퓨터교육, STEAM, 암호이론, 스마트그리드, IoT 보안 등