JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 27 No. 7, pp. 17-25, July 2022
https://doi.org/10.9708/jksci.2022.27.07.017

Hash Based Equality Analysis of Video Files with Steganography of Identifier Information

Wan Yeon Lee*, Yun-Seok Choi*

*Professor, Dept. of Computer Science, Dongduk Women's University, Seoul, Korea
*Associate Professor, Dept. of Computer Science, Dongduk Women's University, Seoul, Korea

[Abstract]

Hash functions are widely used for fast equality analysis of video files because of their fixed small output sizes regardless of their input sizes. However, the hash function has the possibility of a hash collision in which different inputs derive the same output value, so there is a problem that different video files may be mistaken for the same file. In this paper, we propose an equality analysis scheme in which different video files always derive different output values using identifier information and double hash. The scheme first extracts the identifier information of an original video file, and attaches it into the end of the original file with a steganography method. Next the scheme calculates two hash output values of the original file and the extended file with attached identifier information. Finally the scheme utilizes the identifier information, the hash output value of the original file, and the hash output value of the extended file for the equality analysis of video files. For evaluation, we implement the proposed scheme into a practical software tool and show that the proposed scheme performs well the equality analysis of video files without hash collision problem and increases the resistance against the malicious hash collision attack.

▸ **Key words**: Steganography, Equality Analysis, Video File, Hash Collision, Double Hash

[요 약]

해시 함수는 입력 크기와 상관없는 고정된 작은 출력 크기로 인해서, 최근 들어 동영상 파일들의 해시 출력 값들을 빠르게 비교하는 동일성 검사에 널리 사용되고 있다. 그러나 해시 함수는 상이한 입력들이 동일한 출력 값을 도출하는 해시 충돌 가능성이 존재하여, 상이한 동영상 파일들을 동일한 파일로 오인할 수 있는 문제가 존재한다. 본 논문에서는 구별자 정보와 이중 해시를 이용하여 상이한 동영상 파일들은 항상 다르게 판단하는 동일성 검사 기법을 제안하였다. 제안된 기법은 먼저 원본 동영상 파일에서 구별자 정보를 추출하고, 이 구별자 정보를 원본 동영상에 은닉하여 추가하는 확장 파일을 생성한다. 그리고 구별자 정보, 원본 파일의 해시 출력 값, 확장 파일의 해시 출력 값을 사용하여 파일의 동일성 검사를 수행한다. 제안된 기법의 성능 평가를 위해서 실제 프로그램으로 구현하였고, 해시 충돌 없이 동영상 파일의 동일성 검사를 잘 수행하고 또한 인위적 해시 충돌 공격에도 뛰어난 내성을 가짐을 확인하였다.

▸ **주제어**: 정보 은닉, 동일성 검사, 동영상 파일, 해시 충돌, 이중 해시

• First Author: Wan Yeon Lee, Corresponding Author: Wan Yeon Lee
*Wan Yeon Lee (wanlee@dongduk.ac.kr), Dept. of Computer Science, Dongduk Women's University
*Yun-Seok Choi (cooling@dongduk.ac.kr), Dept. of Computer Science, Dongduk Women's University
• Received: 2022. 05. 11, Revised: 2022. 06. 13, Accepted: 2022. 06. 13.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

해시 함수(hash function)는 단방향 함수로 역방향 함수를 찾을 수 없다는 것이 주요 특징이다. 즉, 해시 함수의 출력 값을 가지고 해시 함수의 입력 값을 수치적(computationally)으로 유추할 수 없도록 설계되었다[1]. 이러한 해시 함수의 역방향 함수 무존재 특성으로 인해서 해시 함수는 보안 시스템에서 널리 사용되고 있다[2,3]. 또한 해시 함수의 입력 값 크기는 임의로 변경할 수 있어서, 대용량의 입력 크기에 대해서도 출력 값의 크기는 상대적으로 작은 125 ~ 512 바이트 사이의 고정된 크기를 가진다는 장점이 있다.

최근 들어 감시카메라 녹화 동영상들을 사건/사고의 증거로 활용하기 위해서 대용량의 동영상 파일들을 클라우드 시스템에 저장하는 경우가 증가하고 있으며, 중복 저장을 방지하기 위해서 해시 함수를 기반으로 중복 검사를 빠르게 수행하는 시스템이 보편화되고 있다[4]. 만약 해시 함수를 사용하지 않고 다수의 대용량 동영상 파일들을 직접 비교하면 엄청난 비교 연산을 요구한다. 그러므로 동일한 내용의 동영상이 존재하는지 여부를 빠르게 확인하기 위해서, 각각의 동영상 파일에 해시 함수를 적용하여 고정된 작은 크기를 가지는 해시 출력 값으로 변환하고 이 해시 출력 값들만을 빠르게 비교한다. 그러나 해시 함수가 대용량 크기의 입력 값들을 소량 크기의 출력 값으로 변환하게 되면 불가피하게 다대일 매핑(many-to-one mapping)이 발생하게 되고, 다른 입력 값들이 동일한 출력 값으로 도출되는 해시 충돌(hash collision) 문제가 발생한다[5].

구글과 네덜란드 CWI 연구소에서 SHA-1 해시 함수의 충돌쌍을 생성하는 서비스를 공개하여 해시 충돌의 실질적 위험성을 알리고 있다[6]. 산업체에서는 해시 충돌의 위험성을 줄이기 위해서 기존의 해시 함수들을 최신 SHA-256 해시 함수[1]로 교체하여 해시 충돌의 가능성을 현격하게 줄였으나, 여전히 해시 충돌의 가능성을 완전히 제거하지는 못하여 해시 충돌이 발생할 위험성을 항상 내포하고 있다. 해시 충돌로 인해서 대용량의 동영상 파일들을 비교할 때, 다른 내용의 동영상 파일들이 동일한 해시 출력 값을 가지게 되고 중복 동영상으로 오판되어 저장되지 않고 버려지는 문제가 발생한다.

최근 들어 CCTV, 블랙박스와 같은 감시카메라 녹화 동영상들을 사건/사고의 증거로 활용하기 위해서 클라우드 저장소에 보관하는 추세가 증가하고 있다[7]. 또한 COVID-19 팬데믹 이후의 비대면 업무의 증가로 인해서, 다수의 응시생들의 비대면 시험 과정 동영상을 녹화하여 클라우드 저장소에 보관하는 경우도 증가하고 있다. 어떤 동영상이 증거로 활용될지 사전에 알수 없는 경우에는, 감시카메라로 녹화된 전체 동영상들을 모두 보관해야 하고 또한 편집으로 인한 분쟁을 방지하기 위해서는 원본 동영상 파일을 전체를 편집이 없는 무결성 유지 상태로 보관해야 한다. 이 과정에서 중요한 증거가 포함된 일부 감시 카메라 동영상이 해시 충돌로 인해서 저장 과정에서 누락될 수 있는 문제점은 반드시 개선되어야 한다.

본 논문에서는 동영상 파일들의 동일성 검사에 해시 함수를 사용할 때, 상이한 동영상 파일들은 항상 다르게 판단하는 동영상 비교 기법을 제안한다. 제안된 기법에서는 우선 원본 동영상에서 구별자(identifier) 정보를 추출하고, 이후에 추출된 구별자 정보를 은닉 기법(steganography)을 사용하여 입력 동영상 파일에 추가 내장하는 확장 변경 파일을 생성한다. 그리고 제안된 기법이 동영상 파일들의 동일성 검사를 수행할 때에는, 원본 동영상 파일 해시 출력 값, 확장 동영상 파일 해시 출력 값, 그리고 구별자 정보를 모두 비교하여 동영상 파일들의 동일성 여부를 판단한다. 동일성 검사를 통과하여 다른 동영상 파일로 판정되면, 원본 동영상 파일의 해시 출력 값과 확장 변경 파일의 해시 출력 값, 그리고 구별자 정보를 비교 검사를 위한 데이터베이스에 저장한다. 본 논문에서는 구별자 정보가 유일성(uniqueness)을 가지도록, 파일의 소유자 ID와 파일의 마지막 수정 시간의 묶음(pair)을 사용하였다.

제안된 기법은 유일성을 가지는 구별자 정보를 사용하여, 인위적 조작이 없는 자연적 동영상 파일 비교 검사에서 다른 내용의 동영상 파일들이 동일한 내용의 파일로 오판되는 현상을 완전히 제거하였다. 그리고 해시 충돌을 유도하기 위해서 인위적으로 입력 동영상 파일 내용을 지속적으로 변경하는 해시 충돌 공격(hash collision attack)[5]에 대해서도, 구별자 정보를 조작하기 어렵게 구별자 정보를 은닉하도록 설계하여 매우 견고한 공격 저항력을 가진다. 본 논문에서는 제안된 방법을 실제 프로그램 툴로 개발하여 실행한 필요한 연산 소요 부담을 측정하였고, 분석을 통해서 자연적 해시 충돌 가능성과 인위적 해시 충돌 공격에 대한 저항력을 분석하였다.

본 논문의 나머지 부분은 다음과 같이 구성된다. II 장에서는 해시 함수의 성능 개선에 관련된 기존 연구들에 대해서 설명한다. III 장에서는 제안된 기법의 동작 원리에 대해서 상세히 설명하고, IV 장에서는 제안된 방법의 동작 성능을 측정하며 장단점을 기존 연구들과 비교하여 분석한다. 마지막으로 V 장에서는 본 논문의 내용을 요약하고 정리한다.

II. Previous Work

해시 함수는 메시지 전송의 무결성 검사에서 매우 중요한 역할을 수행하였다. 송신자가 보낸 메시지와 수신자가 받은 메시지가 동일한지 아니면 중간에 악의적으로 변경되어 수신자에게 전달되었는지 여부를 확인하는 무결성 검사에서 해시 함수가 주로 사용되었다[1]. 송신 메시지의 해시 출력 값과 수신 메시지의 해시 출력 값을 비교하여 일치하는 경우에만 메시지가 변경되지 않고 무결성이 유지되었다고 판단한다. 따라서 중간의 공격자는 메시지를 변경하면서, 변경 메시지와 원본 메시지가 동일한 해시 출력 값을 가지도록 해시 충돌이 발생하는 변경 메시지를 찾는 공격 과정을 수행하는데, 이 해시 충돌 공격 문제의 난이도를 높이는 방향으로 해시 함수 설계 연구가 진행되어 왔다. 기존에는 MD5와 SHA-1 알고리즘이 해시 함수로 많이 사용되어 왔으나, 최근에는 해시 충돌 공격에 대한 저항력을 개선하기 SHA-2와 SHA-3 알고리즘이 해시 함수로 점진적으로 채용이 증가하는 추세이다[2,3].

기존의 해시 함수에 관련된 대부분의 연구들을, 주어진 단일 해시 출력 값에 대해서 상이한 다수의 입력 값들을 수치적으로 찾기 어렵도록 해시 함수의 동작 알고리즘을 설계하는 문제를 중점적으로 다루었다[1,2]. 그러나 Maetouq et al.[3]의 연구에서 현재 널리 사용되고 있는 대부분의 해시 함수들은 해시 충돌 공격에 대해서 영향을 받을 수 있음을 보여주었다.

Table 1. Summary of Previous Work

Previous Work	Main Characteristics	Probability of Hash Collision
Ref. [4]	simple hash for fast comparison	not zero
Ref. [5]	fast search operation with double hash functions	not zero
Ref. [7]	reduced hash collision with steganography	not zero
Ref. [8]	defence with encryption and steganography against hash collision attack	not zero

표 1은 기존 관련 연구들의 주요 특성과 해시 충돌 가능성을 요약하여 보여주고 있다. Hwang과 Kim[4]의 연구에서는 해시 함수를 사용하여 대용량 파일의 중복 업로드를 방지하는 시스템 구조를 제안하였으나, 빠른 연산을 위해서 간단한 해시 알고리즘을 사용하여 해시 충돌이 발생할 가능성을 내포하는 단점이 존재한다. Rahim et al.[5]의 연구에서는 해시 함수를 사용하여 동일한 문장을 찾는 탐색(search) 연산을 효율적으로 구현하기 위해서 이중 해시 함수를 사용하는 방법을 제안하였다. 이 연구에서는 첫 번째 해시 함수의 해시 충돌로 인해서 발생하는 부작용을 줄이기 위해서 두 번째 해시 함수를 첫 번째 해시 함수와 별개로 독립적으로 적용하여 해시 발생 가능성을 현저하게 줄였으나, 여전히 해시 충돌 발생 가능성을 완전히 제거하지는 못하였다. Kellinis와 Papapanagiotous[7]의 연구에서는 정보 은닉(steganography) 과정을 해시 함수 전단계에 추가하여 해시 충돌의 가능성을 줄였으나, 마찬가지로 해시 충돌 발생 가능성을 완전히 제거하지는 못하였다. Saraireh et al.[8]의 연구에서는 암호화(encryption) 연산, 정보 은닉 연산, 해시 연산을 모두 혼합하여 메시지 전달 과정에서 메시지 변조 공격을 차단하는 방법을 제안하였다. 이 연구에서는 해시 충돌 공격을 시도하기 어렵게 해시 함수 입력 값에 암호화 연산을 적용하여 외부에 입력 값이 노출되지 않도록 하였다.

위의 기존 연구들에서는 해시 충돌 발생 가능성을 줄이도록 노력을 하였으나 해시 충돌 발생 가능성을 완전히 제거하지는 못하였다. 따라서 본 연구에서는 해시 충돌 발생 가능성을 완전히 차단하면서 동영상 파일의 동일성 검사를 빠르게 수행하는 기법을 다루었다.

III. The Proposed Scheme

1. Double Hashing with Identifier Information

해시 충돌 현상을 제거하기 위해서 제안된 기법은 유일성(uniqueness)을 가지는 구별자 정보를 동영상 파일의 동일성 비교 검사에 이용한다. 구별자 정보가 각각의 동영상 파일에 대해서 유일성을 가진다면, 상이한 구별자 정보를 가진 동영상 파일들은 동일성 검사에서 항상 불일치 결과가 도출된다. 본 논문에서는 구별자 정보로 파일 소유자 ID와 파일의 마지막 수정 시간 정보의 묶음(pair)을 사용하였다.

해시 충돌 공격 과정에서 구별자 정보가 쉽게 노출되어 구별자 정보가 조작되는 것을 방지하기 위하여, 제안된 기법에서는 구별자 정보를 파일 내에 은닉하여 보관한다. 제안된 기법은 두 번의 해시 함수 연산을 적용하는데, 첫 번째 해시 함수 연산은 원본 동영상 파일을 입력 값으로 사용하고, 두 번째 해시 함수 연산은 구별자 정보를 원본 동영상 파일에 은닉한 변경 확장 동영상 파일을 입력 값으로 사용한다. 그리고 동영상 파일들의 내용 일치 비교를 위해서 원본 파일 해시 출력 값, 변경 확장 파일 해시 출력 값, 그리고 구별자 정보를 모두 비교하도록 설계되었다. 제안

된 방법에서는 특정 해시 함수를 고려하여 설계되지 않았으므로, 기존에 알려진 MD5, SHA-1, SHA-2, SHA-3 등에서 임의로 선택하여 사용해도 된다.

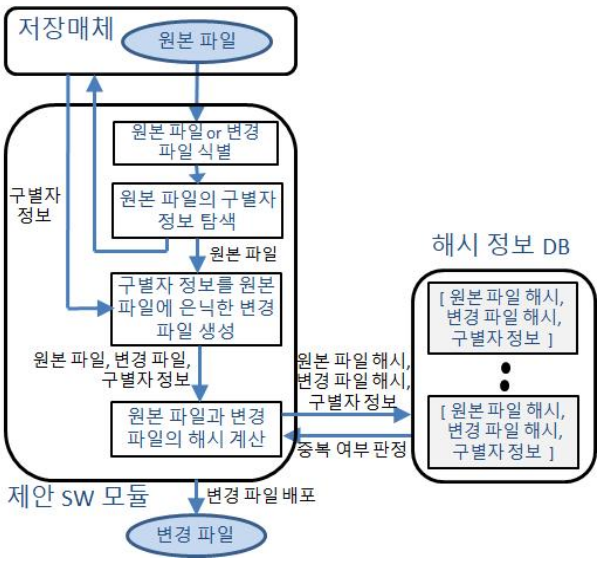

Fig. 1. System Structure of Proposed Scheme

제안된 기법은 그림 1과 같은 시스템 구조도를 가진다. 제안 SW 모듈에서 원본 동영상 파일을 입력 받으면, 입력 동영상 파일이 원본 동영상인지 변경 확장 동영상인지 구분한다. 구분 방법은 입력 동영상 파일에서 은닉 구별자 정보 추출을 시도하고, 추출이 실패하면 원본 동영상으로 판단한다. 원본 동영상 파일이 확인되면 제안 SW 모듈이 저장 매체의 파일 시스템으로부터 파일 소유자 ID와 마지막 수정 시간 정보를 추출하고, 추출된 이 정보들을 문자 은닉 방법을 사용하여 원본 동영상에 내장(embedding)한 변경 확장 파일을 생성한다. 보다 상세한 구별자 정보 은닉 과정과 은닉 구별자 정보 추출 과정은 III.2 절에서 별도로 다룬다.

제안 SW 모듈은 입력 동영상의 원본 파일 해시 출력 값, 변경 확장 파일 해시 출력 값, 그리고 추출 구별자 정보를 묶음으로 관리하고 이 정보들을 모두 사용하여 동일성 검사를 수행한다. 동일성 검사를 통과하여 데이터베이스에 등록된 기존 동영상 파일들의 묶음 정보들과 일치 여부를 비교한다. 원본 파일 해시 출력 값과 변경 파일 해시 출력 값, 그리고 구별자 정보의 묶음과 전체 내용이 일치하는 묶음이 데이터베이스에 존재하면 '완전 일치' 판정을 출력한다. '완전 일치' 판정은 입력된 동영상 파일이 기존에 등록된 동영상 파일들 중에 한 개와 파일 내용이 일치한다는 의미이다. 입력 파일의 묶음 정보와 일치하는 묶음이 데이터베이스에 존재하지 않으면 '완전 불일치' 판정을 출력하고, 이 파일의 묶음 정보를 데이터베이스에 추가한다.

데이터베이스에 '완전 일치' 판정을 가지는 묶음은 없고, 대신 입력 파일의 묶음 정보 중에서 구별자 정보는 상이하면서 원본 파일 해시 출력 값이 동일거나 또는 변경 파일 해시 출력 값이 동일한 묶음 항목이 존재하면 '부분 불일치' 판정을 출력한다. 그리고 이 파일의 묶음 정보도 데이터베이스에 추가한다. '부분 불일치' 판정은 입력 파일이 기존에 등록된 동영상 파일들과 상이하고, 해시 함수의 충돌로 문제로 인해서 묶음의 일부분에서 동일한 값을 도출된 것을 의미한다. '부분 불일치' 판정의 주요 기준은 구별자 정보가 상이하면서 다른 정보가 동일한 상황이다.

'완전 일치' 판정을 가지는 묶음은 존재하지 않고, 대신 입력 파일의 묶음 정보 중에서 구별자 정보는 동일하면서 원본 파일 해시 출력 값 또는 변경 파일 해시 출력 값이 상이한 묶음 항목이 존재하면 '이상 입력' 판정을 출력한다. '이상 입력' 판정은 동일한 구별자 정보를 가진 파일은 하나만 존재하기 때문에 정상적인 상황에서는 발생하지 않는 경우이고, 조작된 입력 파일을 사용한 해시 충돌 공격을 시도한 경우이다. '이상 입력' 판정의 주요 기준은 구별자 정보가 동일하면서 다른 정보가 상이한 상황이다.

그림 2는 제안된 기법의 동작 흐름도를 보여주고 있으며, 그림 1의 시스템 구조도를 시간 흐름에 따라 동작 과정으로 재배치한 내용을 보여주고 있다. 제안 SW 모듈에서 입력 파일이 원본 파일인지 수정 변경 파일인지 먼저 판단한다. 입력 파일에서 구별자 정보 추출이 실패하면 원본 파일로 인정하고 그림 2의 왼쪽 흐름도를 따르고, 구별자 정보 추출이 성공하면 변경 확장 파일로 인정하고 그림 2의 오른쪽 흐름도에 따라 동작한다.

그림 2의 왼쪽 흐름도에서 원본 파일로 판정되면 저장 매체의 파일 시스템에서 구별자 정보를 추출하고, 이후에 구별자 정보를 원본 파일에 은닉한 변경 확장 파일을 생성한다. 그리고 원본 파일의 해시 출력 값과 변경 확장 파일의 해시 출력 값을 계산한다. 제안 SW 모듈은 계산된 원본 파일 해시 출력 값, 변경 파일 해시 출력 값, 구별자 정보의 묶음을 해시 정보 DB에 기존에 등록된 묶음 항목들과 비교하여 '완전 일치', '완전 불일치', '부분 불일치', '이상 입력'의 판정들 중에 한 개를 결정하고 사용자에게 출력한다. 그리고 '완전 불일치' 또는 '부분 불일치' 판정을 받은 입력 동영상의 묶음 정보는 해시 정보 DB에 추가로 등록한다.

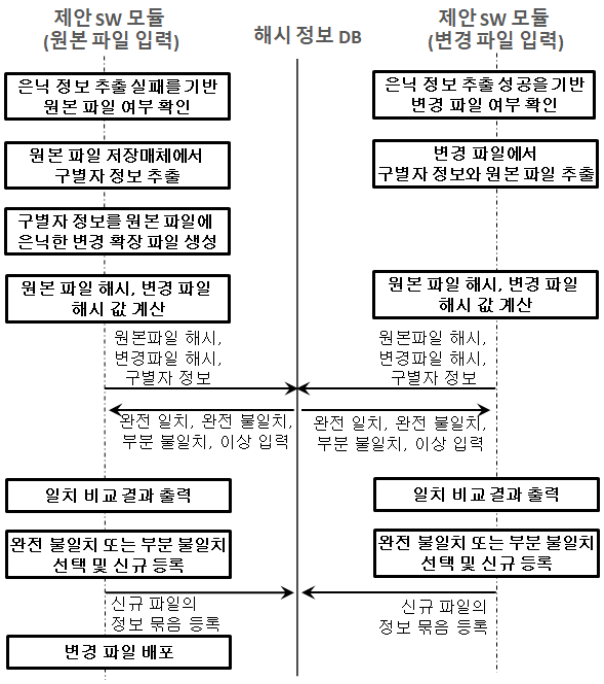

Fig. 2. Flow Chart of Proposed Scheme

제안 SW 모듈은 '완전 불일치' 또는 '부분 불일치' 판정을 받은 동영상 파일에 대해서만, 변경 확장 파일을 다른 사용자들에게 배포하여 다른 사용자들이 원본 동영상 파일 대신에 변경 확장 동영상 파일을 사용할 수 있도록 제공한다. 원본 파일 대신에 변경 파일을 사용하면 구별자 정보를 저장 매체의 파일시스템으로부터 추출하는 과정을 생략할 수 있는 장점을 가진다.

그림 2의 오른쪽 흐름도는 원본 파일 대신에 변경 확장 파일을 입력 값으로 수행하는 흐름도를 보여주고 있다. 그림 2의 왼쪽 흐름도에서는 저장 매체의 파일 시스템에서 구별자 정보를 추출하고 변경 확장 파일을 생성하였다면, 오른쪽 흐름도에서는 변경 확장 파일에서 은닉된 구별자 정보와 원본 파일을 추출한다. 이후에는 왼쪽 흐름도와 유사하게 원본 파일의 해시 출력 값과 변경 확장 파일의 해시 출력 값을 계산하고, 구별자 정보를 포함한 묶음을 해시 정보 DB에 보내서 기존에 등록된 묶음 항목들과 비교한다. 왼쪽 흐름도와 다른 점이라면 변경 파일을 생성하지 않으므로 변경 파일을 배포하는 과정이 생략된다.

2. Information Steganography Method

구별자 정보를 동영상 파일에 은닉하는 방법은 기존에 제시된 방법[9]을 활용하였다. 그림 3은 현재 가장 많이 사용되고 있는 동영상 파일 형식 AVI, MP4, MOV의 내부 구조를 보여주고 있다. AVI 동영상 파일 형식은 'RIFF'라는 4바이트 식별자와 함께 동영상 파일의 크기 정보로 시작되면, MP4와 MOV 동영상 파일 형식은 'ftyp'라는 4바이트 식별자와 함께 동영상 파일의 크기 정보로 시작된다[10]. 따라서 그림 3과 같이 파일 크기 정보를 이용하여 동영상 파일의 끝 이후에 임의의 영역을 추가해서 확장 연결해도 동영상 파일의 재생에는 아무런 영향을 미치지 않는다. 이러한 특성을 이용하여 본 연구에서는 원본 동영상 파일의 마지막 이후에 확장 영역을 추가하고 이 확장 영역에 구별자 정보를 은닉한다.

Fig. 3. Internal Structures of AVI, MP4 and MOV Video File Formats

그림 4는 추가 확장 영역에 파일 소유자 ID와 마지막 수정 시간을 첨부하는 은닉 방법의 수행 예시를 보여주고 있다. 본 예시에서 파일 소유자 ID는 'gildong'이며 마지막 수정 시간은 2021년 12월 25일 14시 50분 59초인 경우를 가정하였다. 소유자 ID는 아스키(ASCII) 디지털 코드로 변환하여 한 글자를 1 바이트에 순차적으로 저장하고, 마지막 수정 시간은 2자리 정수로 분리하여 1 바이트에 순차적으로 저장하였다. 소유자 ID와 마지막 수정 시간 사이에 분리자 표시로 제어문자 "NULL"의 아스키 디지털 코드 '00' 값을 가진 1 바이트를 추가하고, 마지막 종료를 표시하기 위해 16진수 'FF' 값을 가진 1 바이트를 추가하였다.

추가된 정보를 은닉하기 위해서 소유자 ID와 마지막 수정 시간 정보에 '더하기 2' 연산을 수행한다. 반면 은닉된 정보를 추출하기 위해서는 '빼기 2' 연산을 수행하고, 이후에 분리자 표시 '00' 값과 종료 표시 'FF' 값을 기준으로 소유자 ID와 마지막 수정 시간 정보를 찾는다.

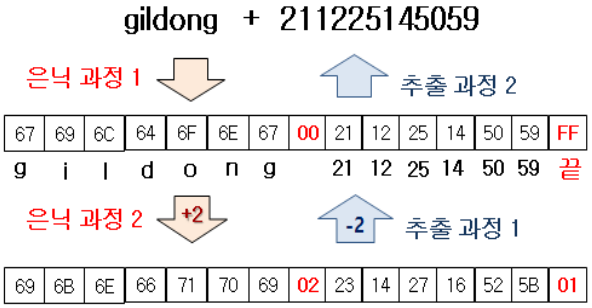

Fig. 4. Hiding Process and Extracting Process of Identifier Information

그림 4에서 보여주는 정보 은닉 방법과 추출 방법은 제안 SW 모듈 내에서만 수행되고, 연산 과정은 외부에는 공개되지 않도록 동작한다. 본 논문의 예에서는 비교적 간단한 은닉 방법을 사용하였지만 고난도의 은닉 연산을 사용하면 외부의 은닉 정보 추출 공격에 대한 견고성을 더욱 증가시킬 수 있다.

IV. Evaluation

성능 평가를 위해서 제안된 방법을 실제 프로그램 툴로 구현하였다. 구현된 프로그램은 JAVA 언어를 사용하여 Eclipse IDE Jee Neon 환경에서 개발되었으며, Sqlite 데이터베이스 라이브러리를 사용하여 구현하였다. 적용된 해시 함수는 128 비트의 출력 값을 가지는 SHA-1 해시 함수이다.

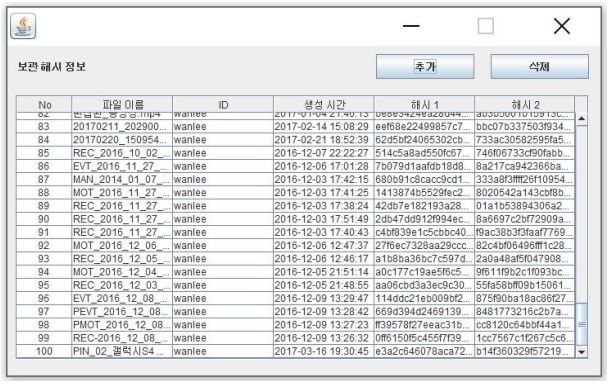

Fig. 5. Implementation Result

그림 5는 구현된 프로그램의 동작 결과를 보여주고 있으며, 100개 다른 동영상들에 대해서 내용 일치 비교를 수행하였다. 100개의 동영상 파일들을 두 번씩 입력 값으로 주어 기존에 등록된 동영상 파일들과 동일성 여부를 검사하는 평가를 수행하였다. 100개의 동영상 파일들 각각에 대해서 첫 번째 입력 때에는 신규 등록의 결과를 받았고, 두 번째 입력 때에는 기존에 등록된 해시 정보 DB 내에서 '완전 일치'의 비교 결과를 가지는 항목이 발견되어 중복 판정이 내려지는 정상적인 수행 결과를 확인하였다.

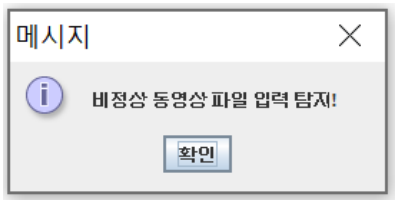

Fig. 6. Detection of Abnormal File Input

인위적 해시 충돌 공격에 대항력 실험을 위해서, 그림 5의 실험에서 이미 등록된 100개 동영상들 중에 한 개와 원본 파일 해시 값은 동일하면서 입력 원본 파일 내용은 상이한 변조 파일을 사전에 미리 찾아서 준비하였다. 그리고 해시 충돌이 발생한 등록된 동영상 파일의 구별자 정보를 준비된 변조 파일에 은닉하여 변경 확장 파일을 생성하고, 이 생성된 변경 확장 파일을 구현된 프로그램 툴의 입력으로 지정하여 구동 결과를 확인하였다. 구현된 툴에서는 이 변조 확장 파일의 입력에 대해서, 그림 6과 같이 '이상 입력' 판정의 출력인 "비정상 동영상 파일 입력 탐지"라는 경고 메시지 창이 사용자에게 출력되는 것을 확인하였다. 이 출력 기능은 인위적 해시 충돌 공격을 탐지하여 시스템 관리자에게 알리는 위해서 설계되었다.

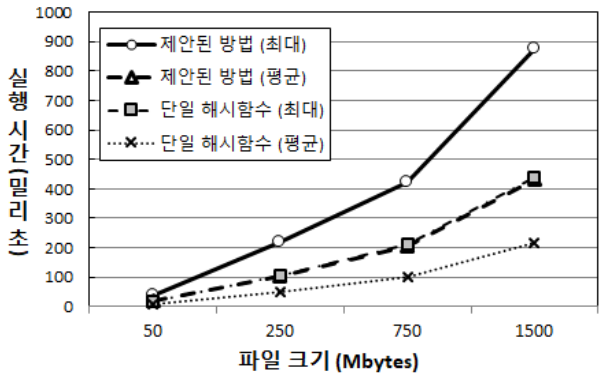

Fig. 7. Execution Time

그림 7은 구현된 프로그램 툴이 소요하는 시간을 측정한 실험 결과들 보여주고 있다. 기존 방법은 해시 함수를 한번만 수행할 때의 소요되는 시간을 측정하였고, 제안 방법은 구현된 프로그램 툴이 2번의 해시 함수와 은닉된 구별자 정보를 추출하는데 소요한 시간을 측정하였다. 연산 시간 측정이 수행된 하드웨어 환경은 인텔 i5-6500 CPU, 8

Gbyte 메모리, 64비트 윈도우 10 운영체제를 탑재한 PC이다. 그림 7에서 기존의 단일 해시 함수만을 적용할 때의 소요 시간과 비교하여, 제안된 방법은 약 2배의 시간을 소요하였고 은닉된 구별자 정보를 추출하기 위해서 평균적으로 약 1 밀리 초(milli second)를 추가로 소요하였다. 실험에 사용된 100개동영상 파일들 중에서 최대 크기는 1.98 Gbytes이고, 이때 소요되는 시간은 단일 해시 방법이 438 밀리초이고 제안된 방법은 877 밀리초로 측정되었다. 최대 1.98 Gbytes의 크기를 가지는 동영상 파일에 대해서 단일 해시 함수 방법과 제안된 방법은 모두 1초 이내에 실행을 완료하여 연산 부담이 낮음을 확인하였다. 또한 두 가지 방법의 소요 시간 차이가 0.5 초 이내로, 일반 사용자가 속도 차이를 현실적으로 체감되지 않는 수준이었다.

Table 2. Conditions of Hash Collision

Hash Function	Single Hash Function		Proposed Scheme	
	necessary condition	sufficient condition	necessary condition	sufficient condition
SHA-1	none	2^{128} times inputs	stegano. hacking	$2^{128} X 2^{128}$ times inputs + stegano. hacking
SHA-3	none	2^{256} times inputs	stegano. hacking	$2^{256} X 2^{256}$ times inputs + stegano. hacking

표 2는 주어진 해시 값이 한 개인 경우에, 기존의 단일 해시 방법과 제안된 방법에서 해시 충돌이 발생하기 위한 필요조건과 충분조건들을 정리한 도표이다. 표 2에서 SHA-1 해시 함수는 128 비트 길이의 출력 값을 가지며, SHA-3 해시 함수는 256 비트 길이의 출력 값을 가진다. 기존의 단일 해시 방법은 해시 충돌이 발생하기 위한 필요조건이 없고, 따라서 매우 불운하다면 단 한번의 다른 입력 시도에 대해서도 해시 충돌이 발생할 수 있다. 또한 해시 출력 값의 길이가 n 비트인 경우에, 2^n 번의 다른 입력 변경 시도를 적절하게 수행하면 주어진 해시 값과 동일한 출력 값이 항상 도출될 수 있다는 충분조건이 성립된다. 반면 제안된 방법은 자연적인 상황에서는 구별자 정보로 인해서 인위적 조작이 없는 경우에는 해시 충돌이 절대 발생하지 않고, 인위적으로 구별자 정보를 조작해야만 해시 충돌이 발생하므로 필요조건으로 구별자 정보의 정보 은닉을 파해하는 기술이 반드시 요구된다. 또한 제안된 방법은 이중 해시 함수를 사용하므로, 첫 번째 해시 함수의 출력 값 동일 조건과 두 번째 해시 함수의 출력 값 동일 조건을 동시에 만족해야 한다. 따라서 해시 출력 값의 길이가 n 비트인 경우에, $2^n X 2^n$ 번의 입력 변경 시도를 적절하게 수행하면 해시 충돌의 결과가 항상 발생한다는 충분조건이 성립된다.

Table 3. Number of Hash Collisions

No. of Input Tries	Single Hash Function	Proposed Scheme
100,000번	0	0
1,000,000번	1	0
10,000,000번	3	0

표 3은 SHA-1 해시 함수의 입력 값을 지속적으로 변동하여, 해시 충돌이 발생한 회수를 측정한 결과를 보여주고 있다. 표 2에서는 비교 대상인 주어진 해시 출력 값이 한 개로 고정되었다면, 표 3에서는 해시 출력 값이 상이하면 이 출력 값을 추가로 등록하여 비교 대상 개수를 증가하도록 하였다. 인위적으로 입력 값을 점진적으로 변경하는 실험에서, 10만 번의 시도에서는 해시 충돌이 발생하지 않았지만, 100만 번에서는 1번의 해시 충돌이 발생하였고, 1000만 번의 시도에서는 3번의 해시 충돌이 발생하였다. 반면 제안된 방법은 해시 입력 파일의 다른 생성 시간 비교를 통해서 해시 충돌이 전혀 발생하지 않았다. 단일 값을 비교하는 경우와는 다르게, 다수의 값들을 비교하는 경우에는 비교 대상 개수에 비례하여 해시 충돌 발생 가능성이 증가하게 된다. 즉, 등록된 비교 대상 개수가 X 이고 해시 출력 값의 길이가 n 비트이면, 해시 충돌의 충분조건은 $2^n/X$ 번의 다른 입력 시도이다. 이 실험 결과를 통해서 등록된 해시 값들이 추가되어 비교 대상 개수가 증가하게 환경에서는, 기존 단일 해시 방법의 해시 충돌 가능성은 무시할 수 없는 수준임을 확인하였다.

표 4는 제안된 방법과 기존 관련 연구들과의 장단점 및 추가 비용을 분석하여 요약 비교한 도표이다. 해시 충돌 가능성의 완전 차단 기능에 대해서, 기존 방법들은 모두 제공하지 못하지만 제안된 방법만이 구별자 정보를 기반으로 제공한다. 인위적인 해시 충돌 공격 방어 기능에 대해서는, 참고 문헌 [7,8] 방법과 제안된 방법에서 정보 은닉 또는 암호 기법을 기반으로 입력 값 변동 이외의 추가 파해 기술을 요구하는 1차 방어 기능을 제공한다. 인위적 해시 충돌 시도 탐지 기능에 대해서는, 기존 방법들은 모두 제공하지 못하지만 제안된 방법은 그림 6의 출력 메시지와 같은 해시 충돌 시도 탐지 기능을 제공한다. 저장 공간 부담의 관점에서, 기존 방법들과 제안된 방법은 각각의 동영상 파일에 대해서 고정된 크기의 저장 공간을 요구하므로 N개의 동영상 파일들에 대한 저장 공간 복잡도 분석(complexity analysis)은 O(N)으로 동일하다. 그림 5에서

보여주듯이 제안된 방법은 단일 해시 함수 기반의 참고 문헌 [4,7,8] 방법과 비교하여 1번째 해시 함수 출력 값 이외에 2번째 해시 함수 출력 값과 구별자 정보의 저장을 위해서 약 3배의 저장 공간을 요구한다.

Table 4. Comparison between Previous and Proposed Schemes

Comparison Factors	Ref. [4,5]	Ref. [7]	Ref. [8]	Proposed Scheme
Prevention of Natural Hash Collision	No	No	No	Yes
Defence of Artificial Hash Collision Attack	No	Yes	Yes	Yes
Detection of Artificial Hash Collision Attack	No	No	No	Yes
Complexity Analysis of Memory Overhead	O(N)	O(N)	O(N)	O(N)

마지막으로 추가 연산 부담의 관점에서, 제안된 방법은 단일 해시 함수만을 사용하는 참고 문헌 [4,7] 방법들과 비교하여 추가 연산 부담의 차이가 미미함을 표 1의 실험을 통해서 확인하였다. 이중 해시를 사용하는 참고 문헌 [5] 방법과 비교하여 제안된 방법의 계산 연산 부담은 비슷하며, 암호화 연산을 사용하는 참고 문헌 [8] 방법보다는 제안된 방법의 해시 연산 부담이 상대적으로 낮다. 본 논문에서는 연산 부담을 줄이기 위해서 정보 은닉 기법을 기반으로 구별자 정보를 보호하였지만, 정보 은닉 기법 대신에 암호화 기법을 사용하여 구별자 정보를 보호하는 방법도 적용 가능하다.

V. Conclusions

본 논문에서는 빠른 동영상 파일 비교를 위해서 해시 연산 출력 값을 사용할 때, 해시 충돌 현상을 방지하기 위해서 구별자 정보를 기반의 동영상 파일들의 동일성을 분석하는 방법을 제안하였다. 제안된 방법에서는 동영상 원본 파일의 구별자 정보를 저장 매체에서 추출하고, 추출된 구별자 정보를 원본 파일에 은닉 내장하여 확장 변경 파일을 생성하였다. 그리고 구별자 정보, 원본 파일 해시 출력 값, 변경 확장 파일 해시 출력 값의 묶음을 사용하여 동영상 파일의 일치 비교를 수행하였다.

제안된 방법은 유일성을 가지는 구별자 정보를 기반으로 다른 내용의 동영상 파일들이 동일한 내용의 파일로 오판되는 문제를 완전히 제거하였다. 또한 구별자 정보를 공격자에 쉽게 노출되지 않도록 은닉하여 해시 충돌 공격에 대한 저항력을 향상시켰다. 제안된 방법을 실제 프로그램 툴로 개발하여 정상적 동작 수행 능력을 확인하였고, 수행한 필요한 연산 소요 시간을 측정하여 수행 부담이 미미함을 확인하였다.

제안된 방법은 소유자 ID가 중복되지 않는 환경에서만 해시 충돌 가능성을 완전히 차단할 수 있고, 동일한 소유자 ID가 다수가 존재하는 환경에서는 구별자 정보의 중복 가능성으로 인해서 해시 충돌 가능성을 완전히 차단시키지 못하는 한계성을 가진다. 또한 제안된 방법은 동영상 파일의 메타 정보를 포함한 전체 내용이 동일한지 여부를 비교할 수 있고, 동영상 파일은 상이하지만 동영상의 멀티미디어 콘텐츠 내용이 동일한 경우에는 적용할 수 없다는 한계점이 존재한다.

향후 후속 연구에서는 해시 정보 DB에 대한 쿼리 증가를 대응하기 위한 병렬화 기법에 관한 연구를 진행하고자 한다. 또한 해시 정보 DB에 정상 동영상과 연관성이 없는 랜덤 해시 값을 지속적으로 저장하여 해시 정보 DB를 훼손시키는 외부 해킹 공격에 대응하기 위하여, 동영상 업로드 SW 모듈과 해시 정보 DB 간의 상호 인증 프로토콜 절차에 관한 연구도 진행할 예정이다.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2021R1F1A1045676).

REFERENCES

[1] Richa Purohit, Upendra Mishra and Abhay Bansl, “A Survey on Recent Cryptographic Hash Function Design,” International Journal of Emerging Trends and Technology in Computer Science, vol. 2, no. 1, pp. 117-122, Jan.-Feb. 2013.

[2] Zulfany E. Rasjid, Benfano Soewito, Gunawan Witjaksono, and Edi Abdurachman, “A Review of Collisions in Cryptographic Hash Function Used in Digital Forensic Tools,” International Conference on Computer Science and Computational Intelligence, vol. 2, pp. 381-392, Oct. 2017.

[3] Ali Maetouq et al., “Comparison of Hash Function Algorithms

Against Attacks: A Review" International Journal of Advanced Computer Science and Applications, vol. 9, no. 8, 2018.

[4] Sung-Min Hwang and Seog-Gyu Kim, "Design of System for Avoiding Upload of Identical-file Using SA Hash Algorithm," Journal of the Korea Society of Computer and Information, vol. 19, no. 10, pp. 81-89, Oct. 2014.

[5] Robbie Rahim, Iskandar Zulkarnain, and Hendra Jaya, "Double Hashing Technique in Closed Hashing Search Process," IOP Conference Series: Materials Science and Engineering, vol. 237, 2017.

[6] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, Yarik Markov, Alex Petit Bianco, and Clement Baisse, Announcing the first SHA1 collision, https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html, Feb. 2007.

[7] Emmanuouel Kellinis and Konstantinos Papapanagiotou, "Using Steganography to Improve Hash Fuctions' Collision Resistance," International Conference on Security and Cryptiography, vol. 2, pp. 337-340, 2007.

[8] Saleh Saraireh, Jaafer Al-Saraiereh, and Mohammad Saraireh, "Integration of Hash-Crypto -Steganography for Efficient Security Technique," International Journal of Circuits, Systems and Signal Processing, vol. 12, 2018.

[9] Wan Yeon Lee and Yun-Seok Choi, "Ultra-light Mutual Authentication Scheme based on Text Steganography Communication," Journal of the Korea Society of Computer and Information, vol. 24, no. 4, pp. 11-18, April. 2019.

[10] Sangwook Lee, Ji Eun Song, Wan Yeon Lee, Young Woong Ko, and Heejo Lee, "Integrity Verification Scheme of Video Contents in Surveillance Cameras for Digital Forensic Investigations" IEICE Trans. Information and Systems, vol. E98-D, no. 1, pp. 95-97, Jan. 2015.

Authors

Wan Yeon Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from POSTECH, Korea, in 1994, 1996 and 2000, respectively. From 2000 to 2003, he was a research engineer in LG electronics. From 2003 to 2011, he was an associate professor in Hallym University. He is currently a Professor in the Department of Computer Science, Dongduk Women's University. He is interested in digital video forensic, blockchain system, embedded system, system security.

Yun-Seok Choi received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Soong-sil University, Korea, in 1997, 1999 and 2009, respectively. He is currently a Professor in the Department of Computer Science, Dongduk Women's University. He is interested in digital video forensic, blockchain system, software architecture, and mobile software.