Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.5, Oct. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.5.1107

손상된 ZIP 파일 복구 기법

정 병 준,† 한 재 혁, 이 상 진‡
고려대학교 정보보호대학원

A Method of Recovery for Damaged ZIP Files

Byungjoon Jung,† Jaehyeok Han, Sang-jin Lee‡
Center for Information Security Technologies, Korea University

요 약

압축파일 형식으로 가장 많이 쓰이는 PKZIP형식은 ZIP 파일뿐만 아니라 MS Office 파일과 안드로이드 스마트폰의 어플리케이션 파일 등에서 사용되는 파일형식이다. 이처럼 다양한 영역에서 널리 쓰이는 PKZIP 형식의 파일은 디지털 포렌식 관점에서 구조분석이 필수적이고 파일이 손상된 경우 복구를 할 수 있어야 한다. 하지만 기존 연구들의 경우 ZIP 파일에서 사용하는 Deflate 압축 알고리즘이 적용된 데이터를 복구하거나 의미 있는 데이터를 추출해 내는 것에만 초점이 맞춰져 있다. 비록 대부분의 데이터가 ZIP 파일의 압축된 데이터에 존재하지만 그 외의 영역에서도 포렌식적으로 의미 있는 데이터가 존재하기 때문에 정상적인 ZIP 파일 형태로 복구를 해야 한다. 따라서 본 논문에서는 손상된 ZIP 파일이 주어졌을 때 이를 정상적인 형태의 ZIP 파일로 복구하는 기법을 제시한다.

ABSTRACT

The most commonly used PKZIP format is a ZIP file, as well as a file format used in MS Office files and application files for Android smartphones. PKZIP format files, which are widely used in various areas, require structural analysis from the viewpoint of digital forensics and should be able to recover when files are damaged. However, previous studies have focused only on recovering data or extracting meaningful data using the Deflate compression algorithm used in ZIP files. Although most of the data resides in compressed data in the ZIP file, there is also forensically meaningful data in the rest of the ZIP file, so you need to restore it to a normal ZIP file format. Therefore, this paper presents a technique to recover a damaged ZIP file to a normal ZIP file when given.

Keywords: PKZIP, ZIP, Deflate, Recovery

I. 서 론

디지털 기기의 사용이 보편화되고 데이터의 양이 방대해지면서 저장장치의 공간을 절약하기 위해 파일을 압축하여 저장하고 있다. 이 과정에서 주로 사용되는 압축파일 형식으로 PKWare에서 만든 PKZIP 형식[1]이 있으며, 이 파일 형식은 ZIP 파일에서 사용될 뿐만 아니라 안드로이드 스마트폰의 어플리케이션 파일(apk), 자바 패키지 파일(jar)등 다양한 파일을 생성하는데 사용된다. 또한 Microsoft Office 2007 버전부터 사용되는 OOXML 형식의 파일(DOCX, PPTX, XLSX)에서도 사용되고 있다[2]. 이처럼 PKZIP 형식은 다양한 영역에서 널리 쓰이고 있는 파일 형식이다.

디지털 포렌식 관점에서 보았을 때 신뢰할 수 있는 디지털 증거를 수집하기 위해서는 PKZIP 형식과 같이 자주 사용되는 파일 형식에 대한 분석이 필수적이고 이를 바탕으로 손상된 파일을 복구할 수 있

Received(09. 19. 2017), Modified(10. 13. 2017), Accepted(10. 13. 2017)
† 주저자, skruwk@gmail.com
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

는 도구가 필요하다.

본 논문에서는 PKZIP형식의 ZIP 파일이 손상되었을 경우 이를 복구하기 위한 기법을 제시한다.

II. 관련 연구

일반적인 파일 손상의 정의는 손상되기 전 파일과 손상된 후 파일이 1비트라도 다를 경우를 의미한다[3]. 또한 손상된 파일이 주어진 경우 손상되기 전과 동일하게 복구하는 것은 일반적으로 불가능하다. Karl Wust[3]는 손상된 파일에 대한 정의가 너무 광범위하다는 것과 손상된 파일을 손상되기 전과 완전히 동일하게 복구하는 것은 어렵다는 것에 착안하여 파일 손상과 복구에 대한 정의를 새롭게 하였다. Karl Wust의 논문에서는 파일 손상에 대해 다음과 같이 정의하였다.

- 정의 : 파일 명세조건이 주어졌을 때, 만약 파일이 적어도 하나의 파일 명세조건을 만족하지 않을 경우 이를 손상된 파일이라 한다.

또한 파일 복구에 대해 다음 3가지 조건을 만족할 경우 성공적인 복구로 정의하였다.

1. 유효한 프로그램이 충돌이나 오류 없이 파일을 열어야 한다.
2. 파일이 원본 파일에 포함된 대부분의 정보를 포함해야 한다.
3. 파일이 원본 파일에 포함되지 않은 정보를 거의 포함하지 않아야 한다.

ZIP 파일의 필드 중에는 데이터가 변조되어도 파일의 동작과 디지털 포렌식 관점에서 의미 있는 데이터에 어떠한 영향도 주지 않는 필드가 존재한다. 예를 들어 **Fig.1**, 의 센트럴 디렉터리 파일 헤더에 속하는 필드 중 압축 버전(Comp Version)이 변조되었을 때, ZIP 파일은 어떠한 오류도 없이 압축 해제가 가능하고, 압축된 파일의 파일명과 같은 포렌식적으로 의미 있는 데이터를 얻을 수 있다. 손상된 데이터는 원본 데이터를 가지고 있지 않다면 원본과 동일하게 복구하는 것은 당연히 불가능하다. 하지만 손상된 ZIP 파일을 명세조건에 맞게 필드 값을 재구성하여 압축 전 데이터와 압축된 파일의 파일명과 같은 메타데이터의 일부라도 얻을 수 있다면, 이는 포렌식적으로 의미 있는 복구라 할 수 있다. 따라서 본 논문에서는 Karl Wust가 제시한 손상된 파일의 정의와 파일 복구의 정의를 사용하여 ZIP 파일 복구 방안을 제시한다.

ZIP 파일은 저장된 데이터의 종류에 따라 메타데이터와 압축된 데이터로 구분할 수 있다[1]. 메타데이터는 로컬 파일의 헤더, 센트럴 디렉터리 파일 헤더, 엔드 오브 센트럴 디렉터리에 저장하며, 로컬 파일의 데이터 영역에 ZIP 파일을 만들 때 사용된 파일들의 압축된 데이터를 저장한다[1].

압축된 데이터는 Deflate 압축 알고리즘을 이용하여 압축이 된다. Deflate를 구현한 것으로는 Jean-loup Gailly와 Mark Adler[4]가 공동으로 개발한 zlib 라이브러리가 있는데 ZIP 파일에서 이를 이용하여 압축을 진행한다. Ralf D. Brown[5-6]은 Deflate로 압축된 데이터의 일부가 유실되거나 변조되었을 경우, 남아있는 정상 데이터를 이용하여 압축되기 전 데이터 일부를 복구해내는 방법에 대해 연구를 진행하였다. Bora Park[7]은 미할당 영역에서 Deflate로 압축된 데이터를 식별하고 이를 압축 해제하여 압축되기 전 데이터를 추출해내는 방법에 대해 연구를 진행하였다.

이처럼 기존 연구들은 ZIP 파일의 메타데이터와 압축된 데이터 중 압축된 데이터의 일부가 변조되거나 유실되었을 때 이를 복구하거나 일부 유의미한 데이터를 추출하는데 중점을 두고 연구를 진행하였다. 기존 연구의 한계점은 일부가 손상된 압축데이터를 복구하거나 추출하기 위해선 특정 조건을 만족해야만 한다는 것이다. 만약 특정 조건을 만족하지 않을 경우 압축된 데이터가 손상된 ZIP 파일에서 어떠한 정보도 얻을 수 없다.

ZIP 파일의 메타데이터에는 ZIP 파일을 만들 때 사용된 파일의 파일명과 확장자, 압축된 시간·날짜 정보가 저장되어 있고 폴더를 압축했을 경우엔 압축된 폴더의 내부구조도 함께 저장된다. 만약 기존 연구와 달리 ZIP 파일 형태로 복구를 할 경우 ZIP 파일의 메타데이터를 통해 포렌식적으로 의미 있는 정보를 얻을 수 있다.

예를 들어 기밀자료가 포함된 폴더를 ZIP 파일 형태로 압축하여 유출한 뒤 압축 파일의 압축데이터 영역에 데이터 유실이나 변조가 일어났을 경우 기존 연구방법으로는 유출에 대한 어떠한 흔적도 얻을 수 없다. 하지만 ZIP 파일 형태로 복구를 하여 메타데이터에 저장되어 있는 파일명과 폴더 구조를 확인할 경

우 파일이 유출되었다는 사실을 추론해 낼 수 있다.

시중에서 쉽게 다운받을 수 있는 zpi2fix와 zip_repair와 같은 ZIP 파일 복구 도구의 경우 손상된 ZIP 파일을 정상적인 ZIP 파일 형태로 복구를 해주지만 복구가 가능한 파일임에도 불구하고 복구가 되지 않는 경우가 존재한다는 문제점이 있다. 따라서 본 논문에서는 손상된 ZIP 파일이 주어졌을 때 압축된 데이터와 메타데이터 복구에 중점을 두고, 복구성능을 향상시킬 수 있는 복구 기법에 대해 설명한다.

III. ZIP 파일 구조

손상된 ZIP 파일을 정상적인 ZIP 파일 형태로 복구하기 위해선 ZIP 파일의 구조와 각 필드가 의미하는 값이 무엇인지 알고 있어야 한다. 따라서 본 절에서는 ZIP 파일의 구조를 살펴보도록 한다. ZIP 파일은 크게 두 개의 영역으로 구분할 수 있다. 첫 번째 영역은 로컬 파일(LF) 영역이고, 두 번째 영역은 센트럴 디렉터리(CD) 영역이다. **Fig.1,** 은 ZIP 파일의 전체 구조와 각 필드가 의미하는 값을 나타낸 그림이다.

3.1 로컬 파일 영역의 구조

로컬 파일 영역은 1개 이상의 로컬 파일로 구성되며 로컬 파일의 개수는 ZIP 파일을 만들 때 사용된 파일의 개수에 의해 결정된다. 로컬 파일은 압축된 파일의 압축 정보와 같은 메타데이터를 저장하고 있는 헤더 영역과 Deflate 압축 알고리즘으로 압축된 데이터를 저장하고 있는 데이터 영역으로 구분되는데, 로컬 파일의 헤더 영역의 경우 시그니처(Signature)부터 추가설명 길이(Extra Length)까지는 0x1E의 고정된 크기를 가지고 이름(Name)과 추가설명(Extra) 부분만 가변적인 길이를 갖는다. 로컬 파일의 데이터 영역의 경우 헤더 바로 뒤에 이어서 나오는 영역으로 압축된 데이터가 들어있다. 압축된 데이터는 4GB 미만의 가변크기로 저장된다.

로컬 파일을 복구할 때 중요한 필드는 7개가 있다. 첫 번째 필드인 시그니처는 각 Local File들의 시작점을 나타내고 항상 특정 값(0x504B0304)으로 고정되어있다. 두 번째 필드인 제너럴 비트(General Bit)는 ZIP 파일이 암호화가 되었는지 여부를 알려주는 영역이다. 만약 이 값이 0일 경우 암호화가 되지 않은 ZIP 파일이고, 1일 경우 암호화가 된 ZIP 파일이다. 세 번째 필드는 CRC32이다. 로컬 파일마다 압축된 데이터를 저장하고 있는데, 이 필드에는 데이터가 압축되기 전 데이터의 CRC32값을 저장한다. 네 번째 필드인 압축 크기(Compression Size)는 로컬 파일에 저장된 압축된 데이터의 크기를 나타낸다. 다섯 번째 필드는 압

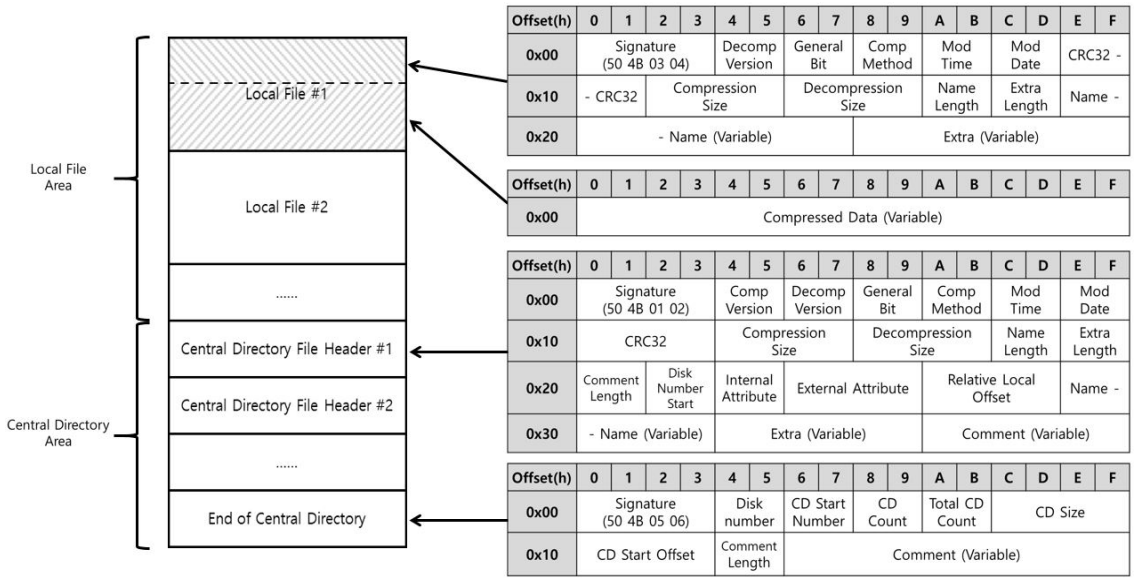


Fig. 1. ZIP File Format

축해제 크기(Decompression Size)이다. 이 필드는 로컬 파일에 저장된 압축된 데이터의 압축되기 전 데이터 크기를 나타낸다. 여섯 번째 필드인 이름에는 ZIP 파일 내에서 해당 파일의 이름을 나타내고 폴더의 경우 내부 파일구조를 나타낸다. 예를 들어 A폴더 하위에 B.txt 파일이 존재할 경우 ZIP 파일의 이름에는 A/B.txt로 저장된다. 마지막 일곱 번째 필드는 데이터(Data)이다. 이 영역에는 압축된 파일 데이터가 저장되는 곳으로 ZIP 파일 복구에 있어 가장 핵심적인 데이터를 저장하고 있는 영역이다.

3.2 센트럴 디렉터리 영역의 구조

센트럴 디렉터리 영역은 로컬 파일 영역의 뒤에 위치하며 로컬 파일의 개수와 동일한 개수의 센트럴 디렉터리 파일 헤더와 1개의 엔드 오브 센트럴 디렉터리(EOCD)로 구성되어 있다. 센트럴 디렉터리 파일 헤더들은 로컬 파일들과 일대일로 쌍을 이루고 있고, 각 내부에는 쌍을 이루는 로컬 파일의 위치와 로컬 파일이 저장하고 있는 메타데이터를 포함한 정보를 저장하고 있다.

센트럴 디렉터리 파일 헤더는 시그니처부터 상대 로컬 파일 주소(Relative Local Offset)까지는 0x2E의 고정된 크기를 가지고 그 뒤에 가변적인 길이의 이름과 추가설명, 주석(Comment)을 갖는다. ZIP 파일의 가장 마지막에 위치한 엔드 오브 센트럴 디렉터리에는 압축된 파일의 수, 센트럴 디렉터리영역의 시작 위치 등의 ZIP 파일의 메타데이터를 저장하고 있다. 해당 영역은 시그니처부터 주석 길이(Comment Length)까지는 0x16의 고정된 크기를 갖고, 가변크기의 주석으로 구성되어 있다.

센트럴 디렉터리 파일 헤더를 복구하는 과정에서 중요한 필드는 7개이다. 첫 번째 필드인 시그니처의 경우 각 센트럴 디렉터리 파일 헤더의 시작점을 나타내고 항상 특정 값(0x504B0102)으로 고정되어 있다. 7개의 필드 중 5개의 필드(제너럴 비트, CRC32, 압축 크기, 압축해제 크기, 이름)는 앞서 설명한 로컬 파일의 필드와 동일한 값을 나타낸다. 일곱 번째 필드는 상대 로컬 파일 주소이다. 해당 필드에는 자신과 쌍을 이루는 로컬 파일이 ZIP 파일 내에서 어느 위치에 있는지를 나타낸다.

엔드 오브 센트럴 디렉터리를 복구하는 과정에서는 3개의 필드가 중요하다. 첫 번째는 시그니처인데 이 경우 엔드 오브 센트럴 디렉터리의 시작점을 나타내는 특정 값(0x504B0506)이 저장되어있다. 두 번째는 센트럴 디렉터리 크기(CD Size)이다. 이 필드에는 센트럴 디렉터리 파일 헤더들 크기의 합이 저장된다. 마지막 필드인 센트럴 디렉터리 시작 위치(CD Start Offset)는 ZIP 파일 내에서 센트럴 디렉터리 영역의 위치를 저장하고 있다.

IV. 손상된 ZIP 파일의 복구 방안

대부분의 손상된 ZIP 파일은 미할당 영역에서 카빙을 진행하면서 발견된다. ZIP 파일을 카빙해 주는 상용도구의 경우 파일의 시작 위치를 찾아 카빙을 진행하기 때문에 ZIP 파일의 시작부분부터 복원이 된다. 만약 ZIP 파일이 단편화 되거나 유실된 상태로 미할당 영역에 존재할 경우 파일의 일부가 존재하지 않은 형태로 복원이 되거나 쓰레기 값으로 채워져 복원이 된다. 이 경우 ZIP 파일은 파일의 뒷부분이 손상된 형태로 복원이 된다.

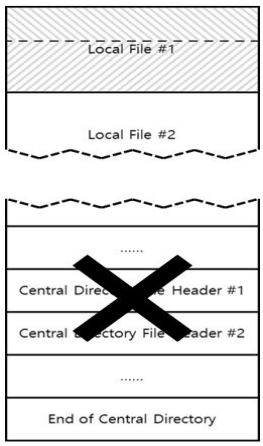
ZIP 파일 손상이 발생할 수 있는 다른 경우는 삭제된 ZIP 파일을 복원할 때 발생할 수 있는데, 메타데이터는 남아있지만 실제 데이터 영역의 일부가 다른 데이터로 덮어써졌을 경우 ZIP 파일의 앞부분 일부나 뒷부분 일부가 손상된 형태로 복원이 된다. 본 장에서는 손상된 ZIP 파일 유형을 4가지로 분류하고 유형별 복구하는 기법에 대해 설명한다. **Fig.2,** 부터 **Fig.5,** 는 ZIP 파일이 손상되었을 때 발생할 수 있는 4가지 유형을 그림으로 나타낸 것이다.

4.1 로컬 파일 영역 일부부터 손상된 ZIP 파일 복구

Case 1은 파일의 시작부터 로컬 파일 영역 일부까지는 남아있고 나머지 부분이 유실되거나 변조된 것을 의미한다. 이러한 형태의 손상된 파일을 복구하는 방법은 다음과 같다.

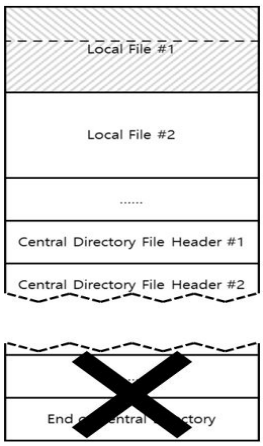
1. 파일의 시작위치를 찾는다.
2. 파일의 시작위치부터 로컬 파일 단위로 건너뛰면서 손상되지 않은 로컬 파일들을 찾는다.
3. 손상되지 않은 로컬 파일들을 이용하여 센트럴 디렉터리 일부를 재구성한다.
4. 3번째 단계의 결과물을 이용하여 엔드 오브 센트럴 디렉터리를 재구성한다.

Case 1의 경우 로컬 파일 영역의 일부부터 파일



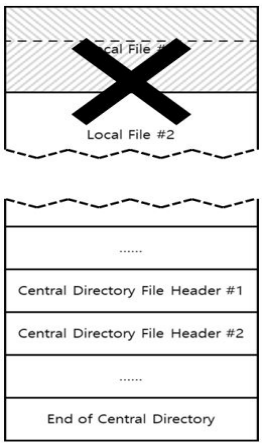
[Case 1]

Fig. 2. Corrupted from Local File area



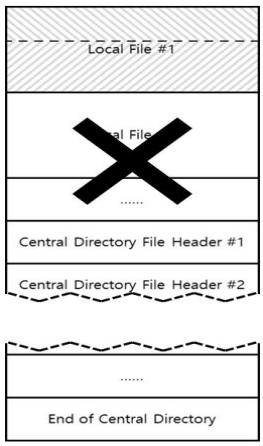
[Case 2]

Fig. 3. Corrupted from Central Directory area



[Case 3]

Fig. 4, Corrupted to Local File area



[Case 4]

Fig. 5, Corrupted to Central Directory area

끝까지 손상된 파일을 복구하는 것이기 때문에 손상되기 전 ZIP파일과 동일한 상태로 압축데이터를 복구하는 것은 불가능하다. 하지만 정상적인 ZIP 파일 형식으로 복구하여 압축된 데이터와 메타데이터의 일부를 얻는 것은 가능하다. Case 1은 손상이 시작된 부분이 로컬 파일의 메타데이터인지 압축된 데이터인지에 따라 3번째 복구단계의 세부과정이 나누어진다. 만약 메타데이터에서 손상이 시작되었을 경우 해당 로컬 파일은 복구가 어렵기 때문에 ZIP 파일 복구에서 제외한다. 반면에 압축된 데이터에서 손상이 시작되었을 경우 해당 로컬 파일의 메타데이터를 손상된 압축데이터를 참고하여 재구성함으로써 파일 복구에 포함시킬 수 있다.

남아있는 정상 로컬 파일을 이용하여 센트럴 디렉터리 파일 헤더를 재구성할 때 주의해야하는 사항이 2가지가 있다. 첫 번째는 로컬 파일과 센트럴 디렉터리 파일 헤더에는 동일한 필드가 존재하는데 동일한 필드에는 동일한 값이 들어가야 한다는 것이다. 예를 들면 제너럴 비트의 경우 로컬 파일에서 0x0000이 저장되어 있는 경우 센트럴 디렉터리 파일 헤더의 제너럴 비트에도 0x0000가 들어가야 한다. 두 번째는 센트럴 디렉터리 파일 헤더에는 센트럴 디렉터리 시작 위치라는 필드가 존재하는데 이곳의 값은 재구성된 ZIP 파일에 맞게 구성해야 된다는 것이다. 센트럴 디렉터리 시작 위치라는 필드는 쌍을 이루는 로컬 파일의 시작위치를 저장하는 필드인데 만약 잘못된 값이 들어있을 경우 정상적인 형태의 ZIP 파일로 복구가 되지 않는다.

3번째 복구단계에서 복구된 로컬 파일들과 센트럴 디렉터리 파일 헤더들을 이용하여 엔드 오브 센트럴 디렉터리를 복구할 때에는 3가지 주요필드를 주의하여 복구해야 한다. 첫 번째로 주의해야할 필드인 시그니처는 항상 고정된 값(0x504B0506)을 갖고 있기 때문에 이 값을 넣어 엔드 오브 센트럴 디렉터리를 재구성해야 한다. 두 번째는 센트럴 디렉터리 크기이다. 이 필드에는 3번째 복구단계에서 복구된 센트럴 디렉터리 파일들 크기의 합이 저장되는 필드인데 만약 값이 잘못되어있을 경우 정상적인 형태의 ZIP 파일로 복구가 되지 않는다. 세 번째는 센트럴 디렉터리 시작 위치이다. 이 필드는 3번째 복구단계에서 복구된 센트럴 디렉터리 파일들의 시작위치를 넣어 구성해야 된다. 이 필드 역시 잘못된 값이 들어있을 경우 정상적인 형태의 ZIP 파일로 복구가 되지 않는다.

4.2 센트럴 디렉터리 영역 일부부터 손상된 ZIP 파일 복구

Case 2는 파일의 시작부터 센트럴 디렉터리 영역 일부까지는 남아있고 나머지 부분이 유실되거나 변조된 것을 의미한다. 이러한 형태의 손상된 파일을 복구하는 방법은 다음과 같다.

1. 파일의 시작위치를 찾는다.
2. 파일의 시작위치부터 로컬 파일 단위로 건너뛰면

서 센트럴 디렉터리 영역의 시작위치를 찾는다.
3. 센트럴 디렉터리 영역의 시작위치부터 센트럴 디렉터리 파일 헤더 단위로 건너뛰면서 손상되지 않은 센트럴 디렉터리 파일 헤더를 찾는다.
4. 로컬 파일을 이용하여 손상된 센트럴 디렉터리 파일 헤더를 재구성한다.
5. 4번째 단계의 결과물을 이용하여 엔드 오브 센트럴 디렉터리를 재구성한다.

Case 2의 경우 로컬 파일 영역이 모두 정상적으로 존재하기 때문에 손상되기 전 ZIP 파일과 동일한 상태로 압축데이터를 복구하는 것이 가능하다. Case 2를 복구하는 과정의 4번째 복구단계와 5번째 복구단계는 센트럴 디렉터리 영역을 재구성하는 것인데 이는 Case 1 복구과정의 3번째, 4번째 복구단계와 동일하기 때문에 Case 1에서 설명한 방법으로 복구가 가능하다.

4.3 로컬 파일 영역 일부까지만 손상된 ZIP 파일 복구

Case 3은 파일의 앞부분이 손상되고 로컬 파일 영역의 일부부터 남아있는 것을 의미한다. 이러한 형태의 손상된 파일을 복구하는 방법은 다음과 같다.

1. 엔드 오브 센트럴 디렉터리의 시작위치를 찾는다.
2. 엔드 오브 센트럴 디렉터리의 메타데이터를 이용하여 센트럴 디렉터리 영역의 시작위치를 찾는다.
3. 센트럴 디렉터리 파일 헤더마다 쌍을 이루는 로컬 파일이 남아있는지 확인한다.
4. 센트럴 디렉터리 파일 헤더를 이용하여 손상된 로컬 파일을 재구성한다.

Case 3의 경우 파일 시작부터 로컬 파일의 일부가 손상된 것을 복구하는 것이기 때문에 손상되기 전 ZIP파일과 동일한 상태로 압축데이터를 복구하는 것은 불가능하다. 하지만 정상적인 ZIP 파일 형식으로 복구하는 것은 가능하다. Case 3은 앞선 두 유형과 다르게 ZIP 파일의 마지막에 위치한 엔드 오브 센트럴 디렉터리를 먼저 찾는다. 그 이유는 센트럴 디렉터리 영역에는 ZIP 파일의 메타데이터가 저장되어 있는데 엔드 오브 센트럴 디렉터리에는 센트럴 디렉터리의 시작 위치가 저장되어 있기 때문이다. 엔드 오브 센트럴 디렉터리를 이용하여 센트럴 디렉터리 영역의 시작위치를 찾으면 각 센트럴 디렉터리 파일 헤더별로 쌍을 이루는 로컬 파일이 존재하는지 확인한다. 센트럴 디렉터리 파일 헤더에는 쌍을 이루는 로컬 파일의 위치가 저장되어 있기 때문에 쉽게 손상유무를 판단할 수 있다. 손상여부 판단이 끝난 후 손상되지 않은 데이터를 이용하여 손상된 로컬 파일을 재구성한다.

센트럴 디렉터리 파일 헤더를 이용하여 손상된 로컬 파일을 재구성할 때에는 주의해야할 점이 있다. 센트럴 디렉터리 영역에는 압축된 파일이 저장되어있지 않기 때문에 로컬 파일의 압축 데이터가 손상될 경우 이를 복구할 수 없다. 따라서 로컬 파일을 재구성할 때 압축데이터가 없는 상태로 재구성해야 한다. 재구성을 위해 로컬 파일과 센트럴 디렉터리 파일의 주요 필드에 Table 1,에 명시된 값을 넣어줘야 한다.

시그니처의 경우 넣어야 하는 값이 고정되어 있기 때문에 로컬 파일의 시그니처일 경우 0x504B0304를 넣고 센트럴 디렉터리 파일 헤더의 시그니처일 경우 0x504B0102를 넣어준다. 센트럴 디렉터리 파일 헤더를 이용해 로컬 파일을 복구할 경우에는 압축된 데이터가 없기 때문에 제너럴 비트부터 압축해제 크기까지 모두 0으로 넣어줘야 한다. 로컬 파일의 이름에는 쌍을 이루는 센트럴 디렉터리 파일 헤더의 이름에 있는 값을 넣어줘야 한다. 위와 같은 방법으로 복구된 ZIP 파일의 센트럴 디렉터리 파일 헤더의 상대 로컬 파일 주소 필드에는 쌍이 되는 로컬파일의 주소를 넣어준다.

Table 1. Value

Field	Value
Signature	0x504B0304 (Local File) 0x504B0102 (CD File Header)
General Bit	0x00 00
CRC32	0x00 00 00 00
Compression Size	0x00 00 00 00
Decompression Size	0x00 00 00 00
Name	Name in CD File Header
Data	No data
Relative Local Offset	Start offset of LF pair

4.4 센트럴 디렉터리 영역 일부까지만 손상된 ZIP 파일 복구

Case 4는 파일의 앞부분이 손상되고 센트럴 디렉터리 영역의 일부부터 남아있는 것을 의미한다. 이러

한 형태의 손상된 파일은 압축된 데이터가 전혀 남아 있지 않기 때문에 정상적인 ZIP 파일 형태로 복구하는 것은 비효율적이다. 따라서 Case 4에서는 메타데이터만 뽑아내고 방법은 다음과 같다.

1. 엔드 오브 센트럴 디렉터리의 시작 위치를 찾는다.
2. 엔드 오브 센트럴 디렉터리의 메타데이터를 이용하여 손상되기 전 센트럴 디렉터리의 시작위치를 찾는다.
3. 손상되기 전 센트럴 디렉터리의 시작위치부터 엔드 오브 센트럴 디렉터리 사이에 존재하는 센트럴 디렉터리 파일 헤더의 개수를 파악한다.
4. 손상되지 않은 센트럴 디렉터리를 이용하여 메타데이터를 추출한다.

Case 4의 경우 로컬 파일 영역이 존재하지 않기 때문에 압축된 데이터를 전혀 복구할 수 없다. 이 경우 남아있는 데이터를 이용하여 정상적인 ZIP 파일로 복구하는 것보다 단순히 남아있는 데이터 중 메타데이터만 추출하는 것이 효율적이다. 추출된 메타데이터는 원본 파일의 ZIP 파일의 일부 메타데이터만 포함한다. Case 4는 Case 3과 동일하게 ZIP 파일의 마지막에 위치한 엔드 오브 센트럴 디렉터리를 먼저 찾는다. 이후 엔드 오브 센트럴 디렉터리의 메타데이터를 이용하여 센트럴 디렉터리 영역의 시작위치를 찾는다. Case 4는 센트럴 디렉터리 영역의 앞부분도 손상이 되어 있는 유형이기 때문에 센트럴 디렉터리 영역의 시작위치에 센트럴 디렉터리 파일 헤더의 시그니처가 존재하지 않는다. 따라서 센트럴 디렉터리 영역의 시작위치부터 엔드 오브 센트럴 디렉터리까지 바이트스캔을 통해 남아있는 센트럴 디렉터리 파일의 개수를 파악한다. 남아있는 센트럴 디렉터리를 이용해 메타데이터를 추출한다.

4.5 손상된 ZIP 파일 복구 알고리즘

4개의 케이스에 맞게 손상된 ZIP 파일을 복구하기 위해선 가장 먼저 손상 파일이 어떤 케이스에 속해있는지를 판단해야 한다. 이후 각 케이스에 맞는 복구 방법을 적용하여 정상적인 ZIP 파일 형태로 복구를 하거나 메타데이터만 추출해야 한다. 따라서 손상된 ZIP 파일을 입력받았을 때 복구하는 전체 알고리즘은 Fig.6,과 같다.

첫 번째 단계로 입력된 파일을 대상으로 바이트 스캔을 통해 로컬 파일과 센트럴 디렉터리의 개수를 파악한다. 두 번째 단계로 로컬 파일과 센트럴 디렉터리의 개수를 비교하여 Case1/Case2중 하나에 해당하는지 Case3/Case4중 하나에 해당하는지 파악한다. 세 번째 단계로는 조건문을 통해 손상된 ZIP파일이 몇 번째 Case에 속하는지를 파악한 후 앞서 설명한 방법을 통해 정상적인 ZIP파일로 복구를 진행한다.

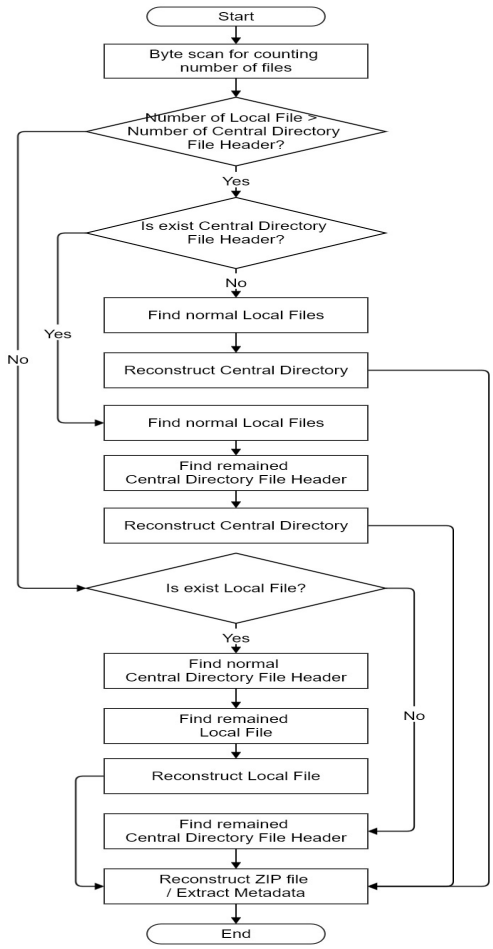


Fig. 6. An algorithm of recovery for damaged ZIP files

V. 복구 기법이 적용된 도구와 기존도구 성능 비교

본 장에서는 4장에서 기술한 손상된 ZIP 파일 복구 기법을 적용하여 개발한 도구의 성능이 기존 ZIP 파일 복구도구들의 성능보다 좋은지를 확인하기 위해 시중에서 사용되고 있는 기존도구와 2가지 실험을 통해 성능비교를 하였다.

5.1 2016 디지털 포렌식 챌린지 문제를 이용한 성능 비교

첫 번째 실험에서 사용된 손상된 ZIP 파일 샘플로는 KDFS 2016 디지털 포렌식 챌린지의[1] 1번 문제에서 제공된 것을 사용하였다. 해당 샘플은 한국 디지털 포렌식 학회에서 만든 다양한 유형의 손실된 ZIP 파일을 대상으로 실험을 진행할 수 있기 때문이다. 손상된 ZIP 파일 샘플에는 확장자가 ZIP인 200개의 파일이었고, 각 파일은 손상된 ZIP 파일이거나 ZIP 파일이 아닌 파일이 섞여 있었다.

비교대상으로 사용한 도구는 챌린지에서 비교대상 도구로 사용한 *Zip2fix*와 *Zip_repair*를 사용하였다. 또한 챌린지에서 개발된 도구 중 복구 성공률이 높은 상위 3개 도구와도 비교하였다. **Table 2.** 는 200개의 샘플 중 정상적인 ZIP 파일로 복구된 파일의 개수를 복구 도구별로 정리한 것이다.

위 표에서 알 수 있듯이 복구 기법이 적용된 도구는 Case3과 Case4에 해당하는 ZIP 파일도 복구하기 때문에 나머지 기존도구들에 비해 성능이 뛰어났다. 도구가 복구하지 못한 59개의 파일들 중 대부분은 원래 ZIP 파일이 아닌 파일이었고, 나머지 파일은 인위적으로 파일 헤더의 일부를 손상시켜 만든 파일이었다.

Table 2. Recovery Result

	Recovered ZIP file	Description
Recovery tool	141	Author's tool
Zip2fix	113	Freeware
Zip_repair	125	Freeware
Tool #1	135	Developed in Challenge
Tool #2	137	Developed in Challenge
Tool #3	133	Developed in Challenge

5.2 ZIP 파일 카빙 결과물을 이용한 성능비교

두 번째 실험에서 사용된 손상된 ZIP 파일 샘플로는 기밀유출 사건의 하드디스크 이미지에서 미할당 영역을 추출하여 카빙한 결과물을 사용하였다. 카빙 결과물로 156개의 ZIP 파일이 생성되었다. 그 중 정상적인 ZIP 파일 44개를 제외한 112개의 손상된 ZIP 파일을 이용하여 성능 비교를 진행하였다. **Table 3.** 은 112개의 샘플 중 정상적인 ZIP 파일로 복구된 파일의 개수를 복구 도구별로 정리한 것이다.

Table 3. Recovery Result

	Recovered ZIP file	Description
Recovery tool	72	Author's tool
Zip2fix	0	Freeware
Zip_repair	0	Freeware
Tool #1	67	Developed in Challenge
Tool #2	69	Developed in Challenge
Tool #3	67	Developed in Challenge

위 표에서 알 수 있듯이 두 개의 프리웨어에서는 손상된 ZIP 파일을 전혀 복구해 내지 못하였다. 복구 기법이 적용된 도구는 챌린지에서 개발된 도구보다 4%~7%정도 복구 성능이 우수하였다.

VI. 결론 및 향후 계획

PKZIP 형식의 파일은 ZIP 파일을 포함하여 다양한 확장자에서 쓰이고 있다. 디지털 포렌식 관점에서 보았을 때 PKZIP 형식과 같이 널리 사용되는 파일 형식은 파일 구조분석이 필수적이고 이를 바탕으로 손상된 파일을 복구하는 기법이 필요하다.

기존에 진행된 연구에서는 ZIP 파일의 압축된 데이터가 손상되었을 경우 손상된 압축데이터에서 의미있는 데이터를 추출하거나 복원하는 것에 초점이 맞춰져 있었다. 하지만 ZIP 파일은 메타데이터에도 포렌식적으로 의미 있는 데이터가 존재하기 때문에 ZIP 파일 형태로 복구를 해야 한다.

따라서 본 논문에서는 손상된 ZIP 파일을 정상적인 ZIP 파일로 복구할 수 있는 기법에 대해 기술하였다. 또한 논문에서 제시한 복구 기법을 적용하여 개발한 도구와 시중에서 사용되는 상용도구의 성능을 비교하여 복구 기법에 적용된 도구의 성능이 좋다는 것을 확인하였다. 하지만 도구의 구현상의 문제로 일부 복구 가능한 파일을 복구하지 못한 경우도 존재하였다. 따라서 향후 복구도구의 문제점을 파악하여 개선한 후 오픈소스로 공개할 예정이다.

1) (사)한국 디지털포렌식 학회에서 주최한 대회

References

〔1〕 PKWARE Inc, ZIP File Format Specification〔Internet〕, https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT Oct. 2014.

〔2〕 ECMA, Standard ECMA-376 Office Open XML File Formats〔Internet〕, https://www.ecma-international.org/publications/standards/Ecma-376.htm, Dec. 2016.

〔3〕 Karl Wust, "Force Open: Lightweight black box file repair," Proceedings of the Fourth Annual DFRWS Europe, pp. 75-82, January. 2017.

〔4〕 Jean-loup Gailly and Mark Adler, Zlib〔Internet〕, https://zlib.net/, January 2017.

〔5〕 Ralf D. Brown, "Improved recovery and reconstruction of DEFLATEd files," Digital Investigation, pp. 21-29, June. 2013.

〔6〕 Ralf D. Brown, "Reconstructing corrupt DEFLATEd files," Digital Investigation, pp. 125-131, May. 2011.

〔7〕 Bora Park, "Determinant Whether the Data Fragment in Unallocated Space is Compressed or Not and Decompressing of Compressed Data Fragment", Journal of the Korea Institute of Information Security & Cryptology, pp. 175-185, Aug. 2008.

〈저자소개〉

정 병 준 (Byungjoon Jung) 학생회원
2016년 2월: 광운대학교 수학과 졸업
2016년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 디지털 포렌식, 파일시스템, 역공학

한 재 혁 (Jaehyeok Han) 학생회원
2011년 2월: 서울시립대학교 수학과 졸업
2016년 2월: 고려대학교 정보보호대학원 공학석사
2016년 3월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 디지털 포렌식, 파일시스템, 데이터 마이닝

이 상 진 (Sang-jin Lee) 종신회원
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~현재: 고려대학교 정보보호대학원 교수
2008년 3월~현재: 고려대학교 디지털포렌식연구센터 센터장
2017년 3월~현재: 고려대학교 정보보호대학원 원장
〈관심분야〉 디지털 포렌식, 심층 암호, 해쉬 함수