Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.1, Feb. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.1.139

손상된 전자기록물 구분과 복원 방법에 관한 연구

김 지 훈,† 최 원 혁‡
누 리 랩

A Study on Classification and Recovery Method of Damaged Electronic Records

Jihun Kim,† Wonhyok Choi‡
NURILAB

요 약

본 연구에서는 손상된 전자기록물을 구분할 수 있는 방법과 손상 형태별 전자 기록물 복원 방법을 제안한다. 제안되는 구분 엔진과 복원 엔진은 전자기록물의 형태 및 구조를 기반으로 손상 전자 기록물을 분류하고, 손상된 형태에 따라 복원 확률을 높인다. 이러한 과정을 통해 멸실되는 전자기록물을 최소화하는 방법을 제안하고, 이를 실험을 통해 검증한다.

ABSTRACT

In this paper we propose a method to distinguish damaged electronic records and an electronic record recovery method according to damage type. The proposed classification engine and recovery engine classify damaged electronic records based on the form and structure of electronic records and increase the probability of recovery according to the damaged form. Through this process, we propose a method to minimize the electronic records that are destroyed and verify them through experiments.

Keywords: Classfication method of damaged electronic records, Recovery method of damaged electronic records, OLE File

I. 서 론

(전자기록물 이관 및 대상물) 현재 전자기록 생산 시스템에서 생산된 보존기간 30년 이상(30년 · 준영구 · 영구)의 전자기록물은 각급 기관의 기록 관리시스템(RMS)에서 국가기록원 중앙 영구 기록 관리 시스템(CAMS)로 이관하게 된다. 오늘날 많은 자료들이 전자 기록물로 보존되며 이러한 전자 기록물은 하드웨어, 소프트웨어 결함에 의해 손상 되어질 수 있다. 손상된 전자 기록물은 영구적으로 복원이 힘들거나 일부 내용을 상실할 가능성이 크다.

따라서 전자기록물이 이관되는 과정에서 손상된 파일을 구분하고, 이관된 전자 기록물 이라면 해당 기록물을 복원 할 수 있는 기술이 필요하다. 현재 (품질 검사 수행 과정) 국가기록원의 'RMS 주요 기능' 문서에서 확인할 수 있듯이 RMS에서 CAMS로 이관된 파일에 대해 가인수 확인 후 진본확인 및 품질검사 및 검수 과정을 거치게 된다. 사람이 수동으로 전자 기록물이 이관되기 전 전자 기록물이 손상되었는지 여부를 판단하는 것은 기록물 원본 유지에 큰 기여를 할 수 있다. 하지만 이러한 과정에서 손상된 문서를 놓치거나 많은 파일을 한꺼번에 검수 하는 것은 인력적인 한계가 있다. 이렇게 이관되기 전 기록물의 훼손 여부를 판단하여 훼손된 경우 재요청 하는 등

Received(11. 27. 2018), Modified(12. 13. 2018), Accepted(12. 14. 2018)
† 주저자, digitalforensic@nate.com
‡ 교신저자, whchoi@nurilab.com(Corresponding author)

원본 기록물을 유기하기 위해 꼭 필요한 부분이라 할 수 있으며 이를 검수 및 복원 할 수 있는 시스템을 구성하는 것은 중요한 역할을 하게 된다.

2015년에서 2017년까지 이관된 전자기록물의 수는 총 633만 건이며, Fig.1. 의 프로세스에서 표시한 빨간색 박스 단계에서 품질 검사를 수행한 결과, 손상된 전자기록물의 수는 59,398개로 확인된다.

손상된 전자기록물의 세부적인 통계는 Table 1. 에서 볼 수 있듯이 2015년 15,955개, 2016년 11,996개, 2017년 31,447개로 대부분의 업무 형태를 고려할 때 전자기록물의 수가 급증하는 만큼 손상된 상태로 가인수되는 전자기록물의 수는 계속적으로 증가할 것으로 예상된다.

전자기록물의 훼손 여부 판단과 복원의 기술을 구현하기 위해서는 전자기록물의 형태와 구성을 파악할 필요가 있다.

먼저 최근 1년간 손상된 전자기록물 샘플 4724개를 확보하여 전자기록물의 형태를 분석 한 결과는 다음 Table 2. 와 같다.

위와 같이 최근 1년간 손상된 전자기록물 샘플을 다시 파일의 형태 별로 분류해 보면 OLE 파일 구조

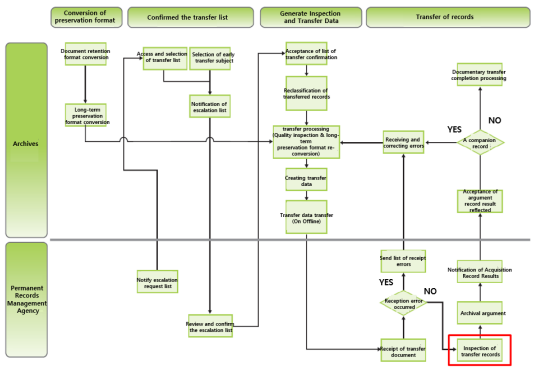

Fig. 1. Electronic records transfer process from RMS to CAMS[1]

Table 1. Number and growth rate of damaged files in electronic records

	2015	2016	2017	Remark
Corrupted files electronic records	15,955	11,996	31,447	Total 59,398
Growth rate	-	-24.81%	162.15%	

Table 2. Count by electronic record file type

	OLE[1)]	HML[2)]	PDF	Etc
Count	1115	1746	145	1718

(HWP, DOC, XLS, PPT 등)와 HML, PDF 파일인 경우가 대부분인 것을 알 수 있다. 기타로 구분된 1718개의 파일은 대부분 대체된 파일 즉 악성코드에 감염되어 원본 문서 파일이 텍스트 파일로 대체 된 경우이기 때문에 복원이 절대 불가능한 케이스이다.

본 논문에서는 이와 같이 손상된 전자기록물의 형태별 분류 개수에 기반 하여 OLE 파일의 구조 진단 및 XML 기반의 문서 파일인 HML, PDF 파일의 구조 진단을 통한 손상파일 분류와 분류된 손상 파일 중 복원이 가능한 파일의 복원을 시도하고자 한다.

따라서 본 논문의 구성은 다음과 같다. 2장에서 전자기록물을 구성하는 문서 파일의 포맷 기반 기술을 조사 분석하고 각 포맷별 손상 판단 방법을 살펴보고, 3장에서는 제안하고자 하는 손상 문서 분류기(separator), 손상 문서 복원기(recovery)의 구현 방법과 시스템 구성에 대해 설명한다. 4장에서 실험 결과 및 성능을 살펴본다. 마지막으로 5장에서 결론을 내리고 앞으로의 연구방향을 제시한다.

II. 전자기록물의 포맷

본 장에서는 전자기록물 파일의 종류와 구성 포맷에 대해 연구하여 복원에 필요한 정보를 확인한다.

2.1 OLE 파일의 구조

2.1.1 OLE 헤더 및 섹터의 구성

OLE 파일 포맷은 MS 오피스 워드, 파워포인트, 엑셀의 문서 포맷으로 사용되어 왔으며 내부는 하나의 작은 파일시스템 구조를 가진다[2]. OLE 파일 내부에는 폴더와 같은 개념인 Storage, 파일과 같은 개념인 Stream을 가지며 FAT 파일 시스템과 유사한 형태로 구성되어 있다. 현재는 마이크로소프트 오피스 문서뿐만 아니라 HWP, GUL 등 국내

1) OLE 파일은 HWP, Doc, Ppt, Xls 등의 문서에서 널리 사용되어 지는 문서 파일의 구조이다.
2) HML 은 한글과 컴퓨터 사에서 만든 XML 기반의 워드 문서 형식이다.

Fig. 2. Structure of OLE File

워드 문서 또한 OLE 파일 구조로 구성되어 있다.

OLE 파일은 아래 Fig.2. 와 같이 Header Block과 Data Block으로 크게 구분된다.

Header Block은 OLE 파일 전체의 주요 정보들을 가지고 있으며, 데이터 블록은 아래와 같은 정보들을 가지게 된다.

- 프로퍼티 (스토리지 및 스트림 정보를 보관)
- 스트림 데이터
- Big Block Allocation Table (BBAT)
- Small Block Allocation Table (SBAT)

OLE 헤더 블록은 OLE 파일의 주요 정보를 담고 있는 블록이며, 여기서 말하는 블록은 512 바이트씩 나누어 블록 번호를 부여하게 된다. 이때 가장 앞서 있는 첫 번째 블록(OLE Header Block)은 -1 블록임을 기억해야 한다.

OLE 헤더는 해당 OLE 구조의 파일에 대한 주요 정보를 저장 한다. (파일 시작~512바이트)

따라서, OLE 헤더는 Magic ID, CLSID, Minor Version, Major Version, Byte Order, Sector Shift, Small Sector Shift, Reserved Data, Number of big block Allocation table depot, Root Storage Sector ID, Max Small Stream Size, SBAT Depot Start Sector ID, SBAT Sector Count, Start Block of extra big block allocation table, Number of extra big block allocation table, BBAT Depot 으로 구성된다.

BBAT가 109개 이하인 경우 다음과 같이 BBAT를 구성할 수 있다.

- "Number Of Big Block Allocation Table"

Fig. 3. Block of OLE File sector

값을 읽어 BBAT 개수를 알아낸다.

- 1번에서 읽은 BBAT 개수가 109개 이하인 경우 0x4C 위치부터 BBAT Depot을 읽어 들이면 된다.
- BBAT Depot 값은 Sector ID 이므로 해당 값에 Sector Size(보통 512)를 곱하고, 다시 한 번 Sector Size(OLE 헤더 크기)를 더해서 정확한 Depot의 Offset을 구할 수 있게 된다.
- BBAT Depot를 읽게 되면 4바이트 단위로 각 섹터의 정보를 기록하는데, 그 값이 의미하는 바는 Table 3. 과 같다.
- 만약 BBAT 개수가 1 이상이면 다음 Depot 위치를 읽어서 위 과정을 반복하여 Table을 구성한다.

Table 3. Meaning of each sector information value

Value	Meaning
0xFFFFFFFD	Special Block
0xFFFFFFFE	End of Chain
0xFFFFFFFF	Not Used Block
0 ~	Next Chain Value(Next Sector ID)

2.1.2 Root Storage 분석

앞서 BBAT Depot, SBAT Depot 및 Root Storage의 구성 방법에 대해서 알아보았으며, 이렇게 읽어 들인 Root Storage에서 각각의 OLE 파일을 구성하고 있는 Storage, Stream을 읽어야만 OLE 파일 구조 분석이 완료 된다.

Root Storage나 다른 Storage는 각 0x80 바이트 단위로 하위에 있는 Stream 혹은 Storage 정보를 가진다. 0x80 바이트 내부에는 이름과 위치 값 등의 정보가 있다.

2.1.3 OLE 파일의 손상 판단

많은 종류의 문서 파일이 OLE 구조로 되어 있다. 따라서 OLE 파일의 구조상의 문제점을 파악하여 문서의 손상 여부를 판단할 수 있다. 예를 들어 특정 Stream, Storage를 읽기 위해 시작 섹터를 얻어 와서 BBAT, SBAT를 따라가서 읽는 도중 파일 사이즈를 넘어서는 위치를 가리키게 되면 파일 뒷부분이 잘려서 손상된 케이스 이며, 특정 필드의 값이 허용하는 범위를 넘어서거나 예측할 수 있는 값이 아니라면 파일이 변조되었다고 판단 할 수 있다. 예

를 들어 한 섹터의 사이즈가 512이고 특정 Stream의 크기가 4096(8개 섹터)인데, BBAT를 보면 12개의 섹터가 연결되어 있다면, Stream 크기 부분이 변조되었거나 BBAT가 손상된 케이스로 구분할 수 있다. 이렇듯 파일이 외부 자극(파일 이동 실패, 악성코드 감염 등)에 의해 손상 된 경우라면 충분히 OLE 구조상 문제점을 확인하여 손상 여부를 판단할 수 있다. 나머지 문서 특징 별 오류는 해당 문서 포맷의 스펙을 확인하여 문서 손상을 판단해야 할 것이다. 또한 OLE 구조가 아닌(PDF 등)의 문서는 따로 해당 파일의 구조를 확인하여야 한다.

2.2 XML 파일의 구조

2.2.1 XML 파일의 정의

XML(Extensible Markup Language)는 확장성 있는 마크업 언어이다. HTML은 이미 약속된 태그들만 사용이 가능하다면, XML은 사용자가 임의로 만들어 태그를 사용 할 수 있다.

대표적으로 HWPML(Hangul Word Processor Markup Language)이 있다. 이는 한컴에서 지정한 태그를 기반으로 문서의 정보를 표현하고 사용하며, XML 표준을 따르는 문서이다.

2.2.2 Well Form XML

XML 문서에는 정형식 문서 (well-formed document)와 유효한 문서(valid document)가 있다. 일반적으로 정형식 문서 well form 문서는 XML 문서 규칙에 맞게 만들어진 오류가 없는 문서를 의미한다. 규칙이 잘 지켜진 문서를 가지고 우리는 Well-Formed XML 이라고 한다.

2.2.3 XML 문서의 손상 판단

XML 문서는 2.2.2 에 언급한 바와 같이 특별한 규칙을 가지고 해당 규칙에 벗어나지 않으면 Well-Formed XML이라 부른다. XML 구조로 만들어진 문서 파일 또한 이러한 규칙을 잘 지켜야 하며 이를 어기면 문제가 발생 한다[3].

예를 들어 HWPML 문서는 문서 최상위 태그가 〈HWPML〉이다. 하지만 문서 마지막에 〈/HWPML〉과 같은 닫힘 태그가 없거나 다른 태그가 온다면 이 파일은 뒷부분이 손상된 파일이라 판단 할 수 있다[4]. 실제로 이러한 파일은 HWP 워드 프로세스에서 정상적으로 열지 못한다.

이외에도 태그의 대소문자가 다른 경우도 있으며 여러 가지 상황에서 Well-Formed 검사만으로 해당 파일의 손상 여부를 비교적 쉽게 판단해 낼 수 있다.

2.3 PDF 파일의 구조

2.3.1 PDF 파일의 구조

PDF 파일은 Fig.4. 와 같이 크게 PDF Header, Body, Cross Reference Table, Trailer 부분으로 이루어진다. PDF Header는 일반적으로 "%PDF" 문자열로 시작하는 PDF Signature가 오고 다음으로 PDF 버전 정보가 ASCII 문자로 기록된다[5].

PDF의 Body 영역은 PDF 문서가 가지는 내용에 대한 Object들이 기록되어 있다. 해당 오브젝트는 다양한 압축, 암호화 형태를 가질 수 있으며, 일반적으로 obj ~ endobj 문자열 사이에 데이터가 존재한다.

PDF의 Cross Reference Table(이하 xref table)는 Body에 존재하는 Object들에 대한 위치 정보, 오브젝트가 유효한지 등의 정보를 가진다. 해당 테이블은 일반적으로 문자열로 존재하는 Table 형태를 가지지만, 최신 문서의 경우 Table이 아닌 Binary Stream 형태를 가질 수도 있으며, 두 가지가 결합된 경우도 있을 수 있다. 또한 1개의 PDF에서 여러 개의 xref table이 존재 할 수 있다.

PDF의 Trailer는 PDF 분석 시 Header의 유

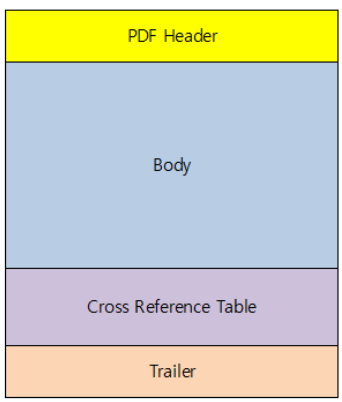

Fig. 4. Structure of PDF File

효성 체크 후 가장먼저 분석 되어야 할 위치이다. Trailer은 xref table의 위치, 특수 오브젝트(암호화 오브젝트, Root 오브젝트, ID 값 등)의 위치 정보를 가진다. 이 Trailer는 1개의 PDF에서 여러 번 존재 할 수 있다.

2.3.2 PDF 파일의 손상 판단

PDF 파일은 OLE파일과 달리 고유의 구조를 가지고 있다. 특히 오브젝트 단위로 문서 내 특정 정보들을 담고 있으며, 이 구조가 맞지 않는 경우 손상되었다고 판단 할 수 있다. 예를 들어 xref table에서 오브젝트 1번 위치가 100 인 경우 100 오프셋으로 이동했을 때 '1 0 obj'라는 문자열이 없다면 이는 손상된 PDF라고 볼 수 있다. 또한 PDF 파일의 끝에는 항상 startxref라는 문자열이 오고 이를 통해 xref table를 분석할 수 있어야 하는데, 이것이 없는 경우 파일 뒷부분이 잘려서 손상되었다고 판단 할 수 있다.

만약 startxref가 없어서 오브젝트 위치를 파악할 수 없더라도, PDF를 전체 탐색하여 카빙 한 후 오브젝트를 모두 추출 한다면, 문서의 내용(문자)을 추출할 가능성은 있다. 장지원[6]은 Page Object를 먼저 탐색하고, 페이지 단위의 복구를 진행하지만 이 경우 Page Object를 찾을 수 없다면 복구하지 못하게 된다. 따라서 본 논문의 결과물은 PDF의 구조적 결함의 보정 뿐 아니라 복원이 불가능한 PDF 파일에서 본문 및 이미지 오브젝트만 카빙하여 추출하는 기법을 사용한다.

III. 손상 파일 분류기와 복원기

3.1 손상 파일 분류기

손상 파일 분류기는 다음과 같은 특징으로 파일을 대분류 하였다.

위와 같이 코드를 부여 후 파일들을 분류 엔진으로 구분 한 결과 총 4724개의 손상 샘플 파일 중 정상파일 505개를 제외한 4219개의 샘플 파일을 손상파일로 구분 할 수 있었다.

손상 파일로 구분 된 4219 개의 파일은 OLE 파일 1115개, HML 파일 1746 개, PDF 파일 145 개를 대부분 탐지 하고 있다는 것을 알 수 있으며 기타 파일 중 악성코드에 감염되어 텍스트 파일로 대체

Table 4. Code Assignment of corrupted file separator

Code Prefix	Meaning
S	Substitution
L	Link
B	Broken
E	Encoding
U	Unknown

되는 등 복원이 불가능한 형태의 파일도 대부분 손상파일로 진단하도록 하였다.

3가지 전자문서 형태의 손상 탐지 방법에 대해서는 2장에서 설명 하였다.

3.2 손상 파일 복원기

손상 파일의 복원기는 단일 코드 R로 코드를 부여하였다. 손상 파일 복원기는 손상된 파일을 최대한 원문으로 복원하도록 구성하였지만, 불가능 할 경우 내부에서 강제로 문서 내용을 추출하는 기능을 추가하였다.

예를 들어 HML(Hangul Word Processor Markup Language) 문서가 Well-Form이 아닌 경우 정확하게 Well-Form 구조를 맞춰 줌으로써 손상된 HML 파일 1746개중 100% 인 1746개를 복원 할 수 있었다.

Fig.5. 를 보면 마지막 태그인 HWPML이 닫혀야 하지만, 원본에서는 없는 것을 알 수 있다. 따라서 복원기를 통해 복원하여 태그를 강제로 삽입해줌으로써 원본 그대로 복원이 가능하다. 또한 여러 가지 형태로 Well-Form이 아닌 xml 문서를 Well-Form이 되도록 해주는 것만으로 문서를 완벽히 복원해 낼 수 있었다.

이렇게 Well-Form이 되기 위해 가장 중요한 것은 태그의 쌍을 맞춰주는 것인데, 이는 자료구조의

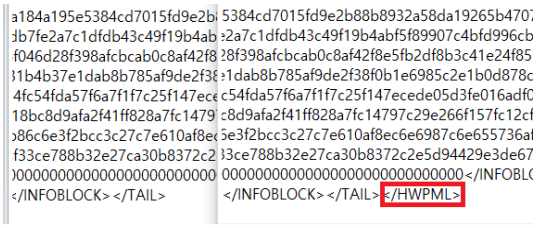

Fig. 5. Recovered HML files

Stack과 같은 원리이다.

예를 들어 〈DOC〉〈TEXT〉Text〈/TEXT〉〈/doc〉와 같은 XML 태그가 있을 때 이는 현재 DOC 태그의 대소문자가 맞지 않아 Well-Form이 아닌 경우이다. 따라서 본 논문의 HML 복원기는 Stack 자료구조에 문자 'DOC', 'TEXT'를 순서대로 Push하고 / 로 시작하는 닫힘 태그를 만났을 때 Stack 자료구조에서 1개씩 Pop 하여 일치하면 통과 시키고 불일치한다면 일치 하지 않는 원인에 따라 조치를 취한다.

복원기가 수행하는 조치는 Pop된 열림 태그와 현재 만난 닫힘 태그가 완전히 다르다면 Pop된 열린 태그에 해당하는 닫힘 태그를 삽입하고, 다시 한 번 Pop하여 현재 닫힘 태그와 비교 후 3 Depth까지 실패하면 복원 실패로 간주 한다. 다만, 닫힘 태그가 잘못 삽입 되었을 수 있으므로 다음 닫힘 태그를 추출하여 맞는지 비교하고 맞으면 현재 닫힘 태그를 삭제하는 조치를 취한다.

만약 Pop된 열림 태그와 대소문자만 다르다면 대소문자가 맞도록 수정하여 삽입한다.

현재 HML 형태의 전자기록물 대부분은 위와 같은 형태로 손상되어져 있었으므로 샘플 군 1746개의 손상 HML을 100% 복원해 낼 수 있었다.

Fig.6. 은 파일이 완전 손상되어(뒷부분이 잘림) 정상 파일로 복원 할 수 없지만, 내부에 문자열을 카빙 기술을 통해 강제로 추출하여 내용을 알아 볼 수 있도록 복원 하였다.

위의 파일은 PPT파일이며, 해당 파일은 OLE 파일 구조의 손상 또는 내부의 Stream 손상으로 판단되어 진다. 일반적으로 정상 PPT 문서에서 Plain Text를 추출하기 위해 "PowerPoint Document" Stream을 읽어 내부 레코드를 탐색하여 문자열을 추출

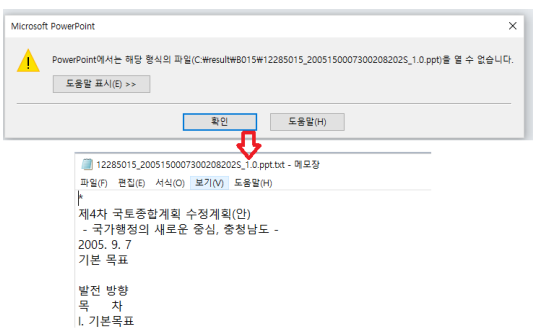

Fig. 6. Extract text from corrupted files

하도록 되어 있으나 현재 PPT파일은 손상이 되어 정상적인 OLE 구조에서 해당 Stream을 읽을 수 없다. 따라서 복원기는 이와 같은 형태의 파일에서 Plain Text를 최대한 추출해 내기위해 카빙방식을 통해 문자열이 존재하는 레코드를 탐색하고 추출한다. PPT파일의 구조 문서[7]를 보면 PPT의 레코드 형태는 크게 '레코드헤더', '레코드' 로 구분되는데 레코드 헤더의 4번째 2바이트를 보면 해당 레코드가 문자열을 가지고 있는지 여부를 판단 할 수 있게 되는데 그 값은 0x0FA0, 0x0FA8이 된다. 해당 값이 0x0FA0일 때는 뒤에 따라오는 레코드의 내용이 UTF16LE 로 인코딩 된 문자열(본문 내용)이 있다는 것을 의미하고, 0x0FA8인 경우 UTF8로 인코딩 된 본문 내용이 있다는 의미를 가진다. 따라서 본 복원기는 파일이 완전히 손상되어 복원 할 수 없더라도 이러한 파일의 구조를 기반으로 파일 전체에서 해당 레코드를 추출하여 텍스트를 최대한 복원해내는 기술을 탑재하였다.

Fig.7. 은 PDF 파일이 완전 손상되어 열 수 없는 파일이지만, 내부 Object 중 Contents Object와 Image Object를 카빙하여 추출 후 내용을 얻어 낼 수 있었다.

PDF 파일의 경우 각각의 요소(Object)가 obj ~ endobj 태그 사이에 존재하게 된다. 따라서 이미지 또한 해당 태그 사이에 존재하며, 압축되어 있다면 해당 오브젝트에 압축 방식에 대한 설명이 모두 존재하기 때문에 압축 해제 후 이미지 파일이라면 추출 해내는 방식으로 PDF 의 이미지를 추출 하였다. PDF 또한 텍스트를 추출 해 낼 수 있는데, 이미지

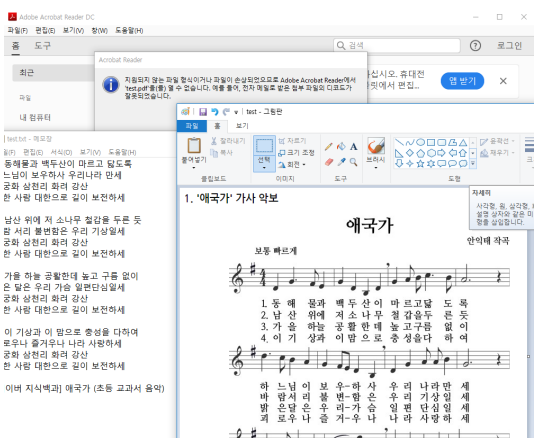

Fig. 7. Extract text and images from corrupted files

추출과 마찬가지로 obj를 찾고, 해당 object 가 Contents 라면 압축을 해제 하여 문자열을 추출 할 수 있도록 하였다.

이러한 문서 복원 기술과 문서 내용 추출 기능을 통해 4724개의 손상 파일 중 복원 가능한 파일 2325개(완전히 다른 파일로 대체된 경우를 제외한)에서 78.36%인 1822개의 파일을 복원 할 수 있었다.

3.3 손상 파일 분류기 및 복원기 시스템 구성

본 논문에서 연구 및 개발한 시스템의 구성은 Fig.8.과 같으며, 손상파일의 자동 구분과 복원은 다음과 같은 순서로 진행된다.

- 관리자 또는 자동화된 시스템에서 파일 서버에 손상이 예측되는 전자기록물을 전송한다.
- 손상파일 구분 엔진은 해당 기록물을 열어 해당 기록물의 파일 형태를 분석한다.
- 파일 형태에 따라 손상여부를 검사 후 손상 코드에 따라 파일서버에 디렉터리를 생성하여 저장한다.
- 손상파일 복원 엔진은 이렇게 구분 된 전자기록물에서 복원이 가능한 파일을 찾아 열어본다.
- 해당 파일을 원본으로 복원 할 수 있다면 원본 그대로 복원하여 파일서버에 전송한다.
- 원본으로 복원이 불가능 하다면, 파일의 내용 추출을 위해 카빙하여 추출된 내용과 추출된 이미지를 파일서버에 전송한다.

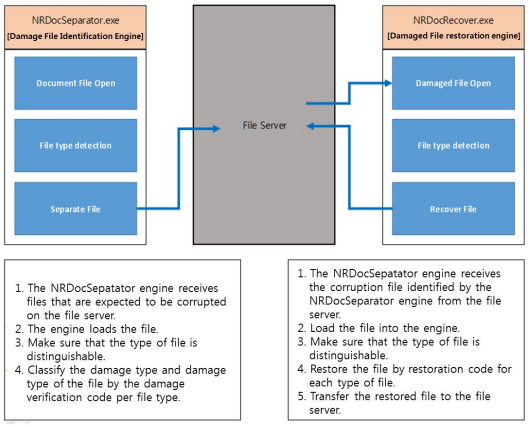

Fig. 8. System configuration of corrupted file separator and recover

Fig.8. 에서 제안 한 시스템 구성을 통해 전자기록물 관리자는 손상된 파일을 빠르고 정확히 구분 할 수 있으며, 생산 기관으로부터 손상된 전자기록물을 재전송 받을 수 없는 경우라도 복원기를 통해 최대한 복구된 전자기록물을 유지 할 수 있게 된다.

IV. 실험 결과 및 성능

본 연구의 결과물인 손상 파일 분류기를 통해 Table 5.와 같이 4724개의 손상 파일 중 정상파일 505개를 제외한 약 89.3%에 달하는 4219개의 파일을 손상 파일로 구분 할 수 있었다.

Table 5. Performance of corrupted file separator

Code	Count
0000(nomal)	505
S	973
L	0
B	2839
E	257
U	150

4.1 손상 파일 복원기의 성능.

본 연구의 결과물인 손상 파일 분류기를 통해 분류된 손상 파일중 복원이 가능한 파일은 2325개이다. 나머지는 파일이 완전히 손상되거나 완전히 다른 문자열로 치환된 파일이기 때문에 실제 복원이 불가능하다.

현재까지 개발된 복원기는 2325개의 복원 가능한 파일 중 1822개의 파일을 복원하여 복원율은 78.36%에 달한다.

4.2 타 연구와 비교.

Karl Wust[8]는 Limitations에서 해당 연구 결과의 한계를 서술하는데, 해당 한계 중 Large Corruption에 대해 해결하기 위해 조각난 파일(완전히 손상된 파일)에서 의미 있는 내용 및 이미지를

Table 6. Performance of corrupted file recover

Code	Count
R	1822

추출하여 하도록 하였으며, 이를 통해 원본 파일로 복원을 할 수 없는 전자 기록물의 내용을 알아 볼 수 있도록 하였다.

특히 완전히 손상되어 복원이 불가능한 경우에도 본문이 있을 수 있는 레코드타입을 탐색하여 최대한 본문 내용을 추출 할 수 있도록 함으로써 전자기록물의 내용 보전에 상당히 기여할 것으로 생각된다.

V. 결론 및 향후 계획

본 연구를 통해 손상된 파일의 상당 부분을 손상된 전자 기록물로 구분 할 수 있었으며, 복원기를 통해 손상된 파일 또한 78.36%에 달하는 복원율로 복원 할 수 있었다.

기존에 진행된 연구는 단순히 Header가 있지만, Footer가 없는 경우, 혹은 확장자가 맞지 않는 경우에 대한 단순한 연구만 진행 되었으나 정확한 손상 파일 구분을 위해서는 OLE 파일의 구조상 올 수 없는 값을 검사, XML 파일의 Well-Form 검사 등 다양하고 깊이 있는 방식으로 파일의 손상 여부를 검사 하여야 한다.

따라서 본 논문에서는 손상된 문서 파일을 분류하기 위해 OLE 파일의 구조분석 및 XML Well-Form 체크기 등을 탑재 하는 등 손상 파일 진단율을 높이고, 또한 손상 된 파일의 복원을 위해 구조적인 문제를 해결 할 수 있는 엔진을 개발하였다.

향후 OLE 파일 혹은 XML, PDF 등 문서의 구조적인 문제뿐만 아니라 각 문서 제작 도구의 구조상의 문제(테이블의 크기가 너무 큰 등)까지 검출 해 낼 수 있도록 하기 위해 한글 문서, 마이크로소프트 문서들의 내부 구조까지 연구하여 적용한다면 손상된 파일로 구분되지 않던 505개의 파일까지 구분해 낼 수 있을 예정이며, 복원 또한 조금 더 정밀하게 진행 할 수 있을 것으로 판단된다.

References

[1] RMS main function : http://www.archives.go.kr/archivesdata/upFile/palgan/1404206260657.pdf

[2] Object Linking and Embedding (OLE) Data Structures, https://msdn.microsoft.com/en-us/library/dd942265.aspx

[3] Extensible Markup Language (XML) 1.0 (Fifth Edition), https://www.w3.org/TR/xml/

[4] HWP File Format, https://www.hancom.com/etc/hwpDownload.do

[5] Portable document format – Part 1: PDF 1.7, https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf

[6] Jeewon Jang, "A Recovery Technique of PDF File in the Unit of Page", http://www.ndsl.kr/ndsl/commons/util/ndslOriginalView.do?cn=JAKO201710758143462&dbt=JAKO&koi=KISTI1.1003%2FJNL.JAKO201710758143462

[7] [MS-PPT]: PowerPoint (.ppt) Binary File Format https://msdn.microsoft.com/en-us/library/office/cc313106(v=office.12).aspx

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

〈저자소개〉

김 지 훈 (Jihun Kim) 정회원
2006년 2월: 경일대학교 컴퓨터 공학과 학사
2013년 2월: 동국대학교 정보 보호학과 석사
〈관심분야〉 디지털 포렌식, 악성코드 분석, 해킹, 정보 보호

최 원 혁 (Wonhyok Choi) 정회원
1997년 2월: 경일대학교 컴퓨터공학과 학사
2001년 2월: 동국대학교 정보 보호학과 석사
2015년 2월: 동국대학교 정보통신공학과 컴퓨터전공 박사과정 수료
〈관심분야〉 디지털 포렌식, 악성코드 분석, 해킹, 정보 통신, 정보 보호