Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.6, Dec. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.6.1439

MS 워드의 RSID 분석을 통한 문서파일 이력 추적 기법 연구*

전 지 훈,† 한 재 혁, 정 두 원, 이 상 진‡
고려대학교 정보보호대학원

Study on History Tracking Technique of the Document File through RSID Analysis in MS Word*

Jihun Joun,† Jaehyeok Han, Doowon Jung, Sangjin Lee‡
Institute of Cyber Security & Privacy (ICSP), Korea University

요 약

MS 워드를 포함한 다양한 전자 문서파일은 계약서 위조, 영업기밀 유출 등의 각종 법적 분쟁에서 주요 쟁점이 되고 있다. MS 워드 2007 이후부터 사용되는 OOXML(Office Open XML) 포맷의 파일 내부 메타데이터에는 고유의 RSID(Revision Identifier)가 저장되어 있다. RSID는 문서의 내용을 생성/수정/삭제 후 저장할 때마다 해당 단어, 문장, 또는 문단에 부여되는 고유한 값으로, 내용 추가/수정/삭제 이력, 작성 순서, 사용된 문서 어플리케이션 등의 문서 이력을 추정할 수 있다. 본 논문에서는 사용자의 행위에 따른 RSID의 변경 사항으로 원본과 사본 구별, 문서파일 유출 행위 등을 조사하는 방법론을 제시한다.

ABSTRACT

Many electronic document files, including Microsoft Office Word (MS Word), have become a major issue in various legal disputes such as privacy, contract forgery, and trade secret leakage. The internal metadata of OOXML (Office Open XML) format, which is used since MS Word 2007, stores the unique Revision Identifier (RSID). The RSID is a distinct value assigned to a corresponding word, sentence, or paragraph that has been created/modified/deleted after a document is saved. Also, document history, such as addition/correction/deletion of contents or the order of creation, can be tracked using the RSID. In this paper, we propose a methodology to investigate discrimination between the original document and copy as well as possible document file leakage by utilizing the changes of the RSID according to the user's behavior.

Keywords: Revision Identifier, Document forensics, OOXML, MS Word

I. 서 론

종이문서와는 달리 전자문서는 쉽게 수정, 복사가 가능하여 문서 유출, 저작권 침해, 표절, 계약서위조 등 다양한 부정행위가 발생할 수 있다. 또한 악성 프로그램을 문서 파일에 넣거나, 기밀 사항을 은닉하여 서로 통신할 수가 있으므로 전자 문서는 디지털포렌식에서 중요한 분석 대상 중의 하나이다[1].

원본과 사본을 구분하는 대표적인 선행 연구로 속성 정보 비교를 통해 부정행위를 조사하는 방법이 있

Received(08. 21. 2018), Modified(10. 12. 2018), Accepted(11. 14. 2018)
* 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2018-0-01000, 디지털 포렌식 통합 플랫폼 개발)
† 주저자, max.joun@gmail.com
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

다[2]. 하지만, 속성 정보를 변조하는 방법은 알려져 있으며, 시간값을 변경하는 여러 도구가 이미 존재하여 쉽게 변조할 수 있다. 본 논문에서 제시하는 RSID(Revision Identifier)를 이용하면, 시간값, 수정 횟수, 수정시간 등을 변조하는 안티포렌식 행위 여부를 확인할 수 있다.

RSID는 문서파일 내부의 고유한 값으로, 다른 속성정보를 이용하지 않아도 문서 자체, 혹은 문서 내 일부 내용의 복사 여부를 알 수 있고, 파일 작성 순서 및 이력을 알 수 있다. 파일 작성 이력으로는 작성자가 해당 문서파일을 작성하면서 사용한 문서 편집 프로그램, 작성된 내용의 출처(다른 MS 워드 문서 또는 다른 어플리케이션에서 복사, 직접 작성 여부 구별)를 알 수 있다. RSID만을 비교하여 많은 자료에서 관련된 문서파일만을 빠르게 찾을 수 있고, 문서를 열람하지 않고도 조사할 수 있어 개인정보를 보호할 수 있다. 또한, MS 워드 뿐만 아니라 다양한 문서파일에서도 RSID가 사용되고 있으며 특히, 리눅스에서 기본 문서 프로그램인 LibreOffice에서는 MS 워드와 비슷한 규칙으로 RSID가 생성되기 때문에 복사 여부, 문서파일의 이력 등을 알 수 있어 문서파일 포렌식에서 주요하게 사용될 수 있다.

II. 관련 연구 및 배경 지식

MS 오피스 2007 버전 이후의 MS 오피스는 OOXML을 기본 포맷으로 사용하고 있다. 속성 정보와 본 논문에서 사용하는 RSID가 포함되어 있는 OOXML의 구조는 Fig. 1과 같다. docProps 폴더 내 core.xml과 app.xml 파일에는 해당 파일의 작성시간과 수정시간, 작성자, 수정횟수 등의 메타데이터가 저장되어 있다. 하지만 이 속성 정보는 쉽게

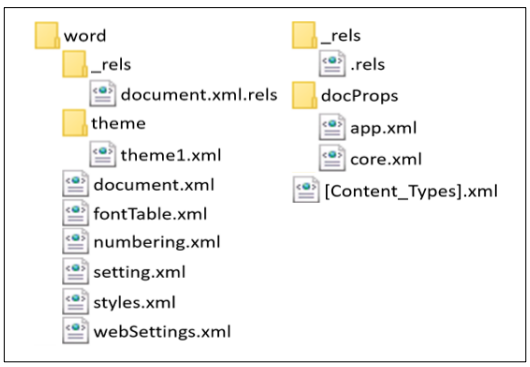


Fig. 1. OOXML Structure

변경할 수 있다. 폴더 내 "documents.xml"과 "settings.xml" 파일에는 다양한 종류의 RSID가 있다[3]. 이를 통해서 내용의 변경 사항, 복사 여부 등을 알 수 있다.

Fu Z[2]는 OOXML 포맷을 대상으로 두 가지 포렌식 분석 방법을 제시하였다. 첫 번째는 기존에도 사용되고 있는 방법인 MS 워드파일의 내용과 내부 속성 정보(파일의 크기, 만든 날짜, 수정한 날짜, 액세스한 날짜, 만든이, 마지막으로 저장한 사람, 수정 횟수)를 확인하는 것이다. 하지만 도구를 이용하거나, 직접 수정하여 쉽게 속성 정보를 변경할 수 있어 악용할 수 있다는 한계가 존재한다. 두 번째 방법은 RSID를 이용한 포렌식 분석 방법으로, 저자는 파일 복사 시에 "document.xml" 파일 내 rsidR, rsidRPr과 "settings.xml" 파일 내 RSID를 이용한 두 파일 간의 복사 여부를 확인하는 방법을 제시하였다. 하지만, rsidR과 rsidRPr 이외에 다른 RSID의 종류와 생성 규칙에 대한 설명이 없어 작성 이력을 알 수는 없다. 또한, 한컴 오피스를 이용하여 MS 워드 문서 파일을 작성한 경우에는 하나 이상의 RSID가 항상 같게 나오는 특징이 있어, 저자가 제시한 방법에는 오탐 가능성이 존재함으로 본 논문에서는 이를 보완한 MS 워드파일 포렌식 분석 방법을 제시한다.

Espen Didriksen[4]는 OOXML 포맷을 대상으로 하는 포렌식 방법을 좀 더 심층적으로 연구 하였다. 저자는 속성정보를 이용한 분석방법 이외에도 RSID의 정의와 종류별 정의를 설명하고 다양한 포렌식 분석 방법을 제시하였다. 하지만 저자의 논문에는 RSID 종류별 생성 규칙에 대한 자세한 설명이 없어, 파일 이력에 대한 조사에는 어려움이 있다. 하지만, 본 논문에서는 사용자의 행위에 따른 RSID 생성 규칙을 정리하고, 구체적인 복사 이력과 파일 편집 이력 분석 방법을 설명하고, 실무적 활용을 위한 효율적인 조사 절차를 제시한다.

RSID를 이용하여 실제 케이스를 분석한 논문[5]에서는 노르웨이의 오슬로와 우튜야에서의 공격 당일에 유포된 테러리스트 매뉴얼을 검토하여 테러리스트의 행위로 체포된 용의자의 주장과 일치하는지 알아보고, 또다른 저자가 존재하는지 분석하였다. 논문의 저자는 RSID를 이용하여 해당 테러리스트 매뉴얼이 다른 출처에서 저장되었다는 것을 확인하였다. 저자는 RSID 중에서 rsidR만을 이용하면 분석을 진행하였는데, 본 논문에서 설명할 rsidR,

rsidRDefault, rsidRPr, rsidP를 이용하여 어떠한 내용이 다른 출처에서 복사되어 저장되었는지를 확인하고, 작성 이력을 추적하여 더 자세한 결과를 얻을 수 있다.

또한, 악성코드가 은닉되어 있는 문서 파일의 Database에 수집해두고, 외부에서 받은 파일을 열람하기 전에 RSID를 확인하여 악성코드를 차단시키는 방법이 있다[6]. RSID를 이용하면 악성코드에 감염되지 않는 장점뿐만 아니라, 문서 파일 조사시에 파일의 내용을 열람하지 않고도 관계성을 찾을 수 있다는 장점을 가지고 있다.

MS 워드 파일 이외에 PDF 파일과 MS 엑셀 파일에서도 작성 이력을 확인할 수 있다. Hyunji Chung[7]은 PDF 파일 수정 후에도 남아있는 데이터를 확인하는 방법을 제시하였다. 수정하기 전의 데이터를 추적하여 PDF 파일의 작성 이력을 확인할 수 있으며, 이 영역을 사용하여 데이터를 은닉할 수도 있다고 설명하였다. Yoon Mi, Lee[8]는 MS 엑셀의 메타데이터를 이용하여 포렌식 조사방법 및 작성 순서를 추적하는 방법을 설명하였다.

MS 워드 파일의 작성 이력을 알 수 있는 논문으로는 임시파일을 이용하여 분석하는 방법이 존재하지만[9], 임시파일이 존재해야만 가능하다는 한계점을 가지고 있어 분석 가능한 파일이 단일 파일일 경우에는 조사가 어렵지만, RSID를 이용하면 하나의 MS 워드파일로도 파일 이력을 알 수 있다.

위에 언급한 논문들과 같이 문서 파일 종류별로 파일 이력을 추정하는 연구가 꾸준히 나오고 있는 것처럼 문서 파일 포렌식은 범죄 조사시에 유용하게 쓰이고 있다. 하지만, 실제 문서 파일 포렌식 케이스를 진행하면서 시간값이나 수정횟수와 같은 메타데이터가 고의로 변조되어 있는 경우가 있어, 파일 포렌식 분석에 어려움을 겪은적이 있다. 이 문제를 해결하기 위해 본 논문에서 제시할 RSID를 사용하면 파일이 변조되었다는 사실을 입증할 수 있다. 또한, RSID에 관련된 기존 논문에는 두 파일간의 복사이력에만 활용할 수 있지만, 본 논문에서는 서론에서 언급한 RSID를 이용한 다양한 활용 발법을 제시한다.

III. RSID (Revision Identifier)

RSID는 문서 내용을 변경하고 저장할 때마다 파일 내부에 할당되는 32bit의 고유한 값이다. RSID는 8자리 중 앞의 두 자리가 "00"으로 고정되어 있고, 뒤의 여섯 자리는 랜덤인 16진수로 구성되어 있다. 이론상으로 RSID가 랜덤하게 같은 값으로 중복되는 확률은 $1/16^6 = 1 / 16,777,216$ 이다. 파이썬으로 개발한 도구로 인터넷에서 임의로 docx 파일을 크롤링하여 획득한 1,000개의 파일의 RSID를 비교한 결과, 716,221개의 RSID가 존재하였고, 이 중 중복되는 RSID가 없는 것으로 확인하였다.

3.1 RSID 종류별 차이점 비교

RSID는 Table 1과 같이 문장(run), 문단(paragraph), 표(table), 구역(section)과 같이 글을 구성하는 요소의 범위에 따라 7가지로 구분된다. 기존 논문에서 이에 대한 종류는 정의되어 있으나[3], 각 RSID가 의미하는 수정범위와 생성규칙에 대한 연구는 추가적으로 필요하다.

rsidR는 문서를 처음 만들 때 기본적으로 생성되며, 이후에는 문서 내에 새로운 단어, 문장, 문단이 생성될 때 생성된다. 즉, 기존에 존재하던 문장에 이어서 새로운 내용을 작성하거나 복사하면 생성된다.

Table. 1. Types of RSID

Type	Paragraph	Run	Table	Section
rsidR	w:p w:rsidR	w:r w:rsidR	w:t w:rsidR	w:sectPr w:rsidR
rsidRDfault	-	w:r w:rsidRDfault	-	-
rsidRPr	w:p w:rsidRPr	w:r w:rsidRPr	w:t w:rsidRPr	w:sectPr w:rsidRPr
rsidDel	w:p w:rsidDel	w:r w:rsidDel	w:t w:rsidDel	w:sectPr w:rsidDel
rsidP	w:p w:rsidP	-	-	-
rsidTr	-	-	w:t w:rsidTr	-
rsidSect	-	-	-	w:sectPr w:rsidSect

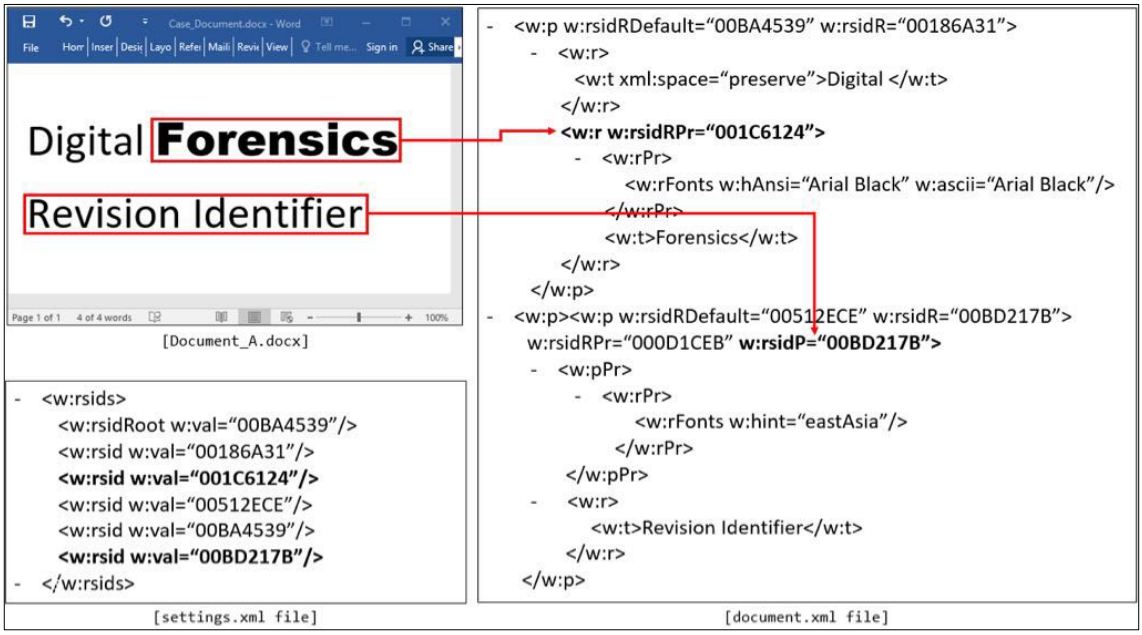


Fig. 2. document.xml and settings.xml in RSID_document.docx

rsidRDefault는 문서를 처음 만들 때 기본적으로 생성되며, 새로운 문단이 생성될 때만 생성된다. 즉, 줄 바꿈을 하고 문서를 작성하거나, 줄 바꿈이 포함된 문단을 복사하면 rsidR과 함께 생성된다.

rsidRPr은 글꼴에 변경이 일어났을 때 생성된다. 글꼴 변경될 경우는 크게 두 가지로 나눌 수 있는데, 첫 번째는 이미 작성된 글자의 글꼴을 변경하였을 때이며 두 번째는 다른 파일에서 글자의 글꼴이 해당 파일과 다른 경우, 이 글자를 복사하여 붙여넣었을 때이다. 즉 기존에 존재하던 내용의 글꼴을 변경하거나 MS 워드가 아닌 다른 문서 애플리케이션(해당 파일의 글꼴이 아님)으로 작성한 내용을 복사할 때 생성된다. Fig. 2와 같이 "Digital Forensics"에서 "Forensics"를 Arial Black 글꼴로 변경하면 새로운 rsidRPr가 "document.xml" 파일에 할당되고 해당 RSID가 "settings.xml" 파일에도 할당된다.

rsidP는 MS 워드, 또는 다른 문서 애플리케이션에서 새로운 문단을 복사할 때 생성된다. 새로운 문단은 줄 바꿈(enter)을 포함한 문단을 말한다. Fig. 2와 같이 "Revision Identifier"를 다른 MS 워드파일에서 복사하여 붙여넣으면 새로운 rsidP가 "document.xml" 파일에 할당되고 해당 RSID가 "settings.xml" 파일에도 할당된다.

rsidDel은 본문의 내용 중 문단이 통째로 삭제된 경우에 생성되는 RSID이다. 하지만, 문단을 지운다고해서 항상 생성되지 않고 특정한 경우에만 생성되는데, 이 부분은 추후 연구가 필요하다. **rsidTr**은 테이블이 수정되었을 때 생성되는 RSID이며, **rsidSect**은 레이아웃 (용지 크기, 용지 방향, 여백 등)을 변경하였을 때 생성되는 RSID이다. Espen Didriksen[4]의 논문에서는 해당 RSID가 MS 워드 2010 이후에는 존재하지 않는다고 설명하였지만, 최신버전에서도 존재하는 것을 확인하였다.

다음으로, 파일 편집시 이력을 추정하기 위해서 RSID의 생성 방식을 다음과 같은 순서로 설명한다.

1) 원본 파일 복사
2) 복사한 파일의 내용에 추가/수정/삭제
3) 원본 파일의 내용 복사
4) 문서 어플리케이션별 RSID 특성

3.2 RSID 특징을 파악하기 위한 실험 결과

본 논문은 윈도우 10에서 MS 워드 2016으로 진행되었다. MS 워드 2003, 2007, 2010, 2013, 2016에서도 큰 특이사항 없이 같은 규칙으로 RSID가 생성된다. 또한, 윈도우 XP, Vista, 7, 8, 10, MAC OS에서도 내용을 생성/수정/삭제했을 때도

같은 규칙으로 RSID가 할당되고 MS 워드파일이 운영체제별로 이동하더라도 RSID가 변하지 않는다.

① 새로운 MS 워드파일을 복사하는 경우 MS 워드파일을 복사하는 방법으로는 1) 원본 파일을 자체를 "복사-붙여넣기"하는 것과 2) 원본 파일에서 "다른 이름으로 저장" 기능을 사용하는 것이 있다. 1)의 방법으로 파일을 복사할 경우, RSID가 원본 파일과 모두 일치한다. 하지만 2)의 방법을 사용할 경우에는 원본 파일이 가지고 있는 RSID에 하나의 RSID가 추가적으로 생성되며 그 값은 "settings.xml"에 기록된다. 원본 파일의 RSID와 사본 파일의 RSID의 목록을 함께 비교함으로써 복사된 방법을 알 수 있다.

② 복사한 MS 워드파일에 내용을 추가/수정/삭제하는 경우 새로운 문서에 문장을 생성하거나, 기존의 존재하는 문장에 내용을 추가하고 저장하면 추가된 위치에 새로운 rsidR과 rsidRDefault가 "document.xml"과 "settings.xml" 파일에 해당 RSID가 추가된다. 문서 내용을 수정/삭제 시 이미 RSID를 할당받은 내용 중 최소 한 글자 이상 남아있으면 "document.xml" 파일에 기존의 RSID가 남아있다. 하지만 내용을 전부 삭제하면 기존에 존재하던 RSID가 모두 사라진다. 하지만 "document.xml" 파일 내의 RSID가 전부 사라져도, "settings.xml" 파일 내에는 한번 저장된 RSID는 모두 그대로 남아있다. 또한, 삭제 시 생성되는 새로운 RSID가 "settings.xml"에 추가로 기록된다.

생성/수정/삭제했을 때 변경되는 RSID의 규칙을 이용하면 원본과 복사본 혹은 유출한 파일이라고 의심되는 MS 워드파일이 존재할 때, 만약 유출자가 파일을 수정하거나 내용을 지워도 "settings.xml" 파일 내에 기존 RSID가 모두 기록되어 있어 해당 파일과 원본 파일의 관련성을 입증할 수 있다.

③ MS 워드파일의 내용을 다른 문서로 복사하는 경우 내용 복사를 설명하기에 앞서, 본 논문에서 언급하는 기본 글꼴이란, MS 워드에서 파일 작성 시에 글꼴의 크기, 색깔, 굵기, 기울임 등의 글꼴 변경을 하지 않고 작성한 상태이다. 즉, 한국판에서는 맑은 고딕, 영문판에서는 "Calibri" 글꼴로 작성한 상태를 말한다. **복사할 내용이 기본 글꼴일 경우,** 기본 글꼴로 작성된 내용을 복사하여 새로운 문서 또는 다른 문서에 붙여넣으면 기존의 파일에 할당된 RSID가 복사한 파일에 복사되지 않는다. 즉, 글꼴 및 스타일을 변경하지 않은 기본 글꼴로 작성된 내용은 복사하더라도 새로운 RSID가 복사한 내용에 할당된다. **복사할 내용의 글꼴이 기본 글꼴이 아닌 경우** 기본 글꼴을 복사한 경우와는 다르게, 글꼴을 수정하면 rsidRPr이라는 새로운 RSID가 수정된 내용에 할당되고, 새로운 문서에 복사할 경우, 원본의 rsidRPr이 복사된다. 문서 내용 일부의 글꼴을 변경(크기, 스타일, 색깔, 글꼴 종류)하였을 때, 변경된 내용 중 최소 하나의 글자가 포함된 원본의 내용을 새로운 문서에 복사하면 RSID가 그대로 복사되어 저장된다. 글자 스타일에 따라 rsidRPr이 생성되는 경우와 생성되지 않는 경우가 있는데 이는 Table 2에 정리하였다. 하지만, 다른 MS 워드파일에서 복사하여 붙여넣을 때, "원본 서식 유지" 옵션이 아닌 "텍스트만 유지" 옵션을 선택하여 붙여넣으면 rsidRPr이 생성되지 않으며 RSID도 원본의 값과 다른 새로운 값이 할당된다.

Table 2. Types of style that create rsidRPr

Type	rsidRPr created
Bold/Italic	O
Array	X
Color	O
Font Type	O
Font Size	O
Listing	X
Style	O
Chart	X
Picture	X
Text highlight	O

3.3 문서 편집기별 RSID 분석결과 비교

사용자는 MS 워드에서만 작성하지 않고, 다른 문서 편집기를 이용하여 작성하거나, 다른 운영체제에서 수정하는 경우가 있다. 본 소절에서는 MS 워드파일을 다른 워드 편집기에서 작성 및 수정하였을 때의 변경 사항을 설명한다.

RSID는 MS 워드에서만 생성되는 것이 아니라 다른 문서 편집기에서도 생성되는 경우가 있다. 같은 규칙으로 생성되는 프로그램도 있지만, 존재만 할 뿐, 디지털 포렌식적으로 사용할 수 없는 경우도 있다. 문서 편집기별로 RSID의 존재 여부는 Table 3과 같다. 각각의 문서 편집기로 생성/수정한 문서 내

Table 3. Existence of RSID according to application

Application type	RSID existed
MS Word	O
MS Office Online	O
Office 365	O
Google Document	O
Mobile MS Office	O
Libre Office	O
Open Office	X
Naver Office	X
Hancom Office	O

의 RSID를 통해 사용자가 사용한 편집기의 종류를 추정할 수 있다.

MS 오피스 온라인 워드의 경우에는 새로운 문서파일을 생성하면 문서 내 “document.xml” 대신에 “document2.xml” 파일이 생성되고 이 안에 RSID가 존재한다. 기존의 MS 워드에서는 처음 두 자리 8bit는 “00”으로 고정되어 있고, 뒤의 6자리 24bit는 임의로 생성되지만, MS 오피스 온라인 워드에서 문서를 새로 만들었을 경우, “5A4C71E5”와 같이 처음 두 자리 8bit가 “00”이 아닌 값이 할당된다. MS 오피스 온라인 워드에서 생성한 후에 MS 워드로 수정할 경우, “document2.xml” 파일이 다시 “document.xml” 파일로 변경되고, 수정한 부분은 기존 MS 워드와 동일하게 RSID가 생성된다. 이외의 RSID 생성 규칙은 MS 워드와 같다. 따라서, 파일 내에 “document.xml”이 존재하지 않고 “document2.xml”이 존재하거나, RSID의 처음 두 자리 8bit가 “00”이 아닌 다른 16진수로 할당된 경우는 MS 오피스 온라인 워드에서 처음 생성/수정한 것이다.

오피스 365로 생성한 파일에는 RSID가 MS 워드와 같은 규칙으로 생성되지만, “settings.xml” 파일에 “document.xml” 파일에는 존재하지 않은 RSID가 2개 더 생성되어 있다. 한컴 오피스는 해당 값이 고정되어 있지만 오피스 365는 랜덤하게 변하는 특징이 있다.

구글 도큐먼트(google document)는 새로 만들거나, MS 워드에서 생성한 문서를 구글 도큐먼트에서 저장하면 모든 RSID가 “00000000”으로 할당되어 있다. 따라서 파일 내 최소 한 개의 RSID가 “00000000”이면 구글 도큐먼트로 생성/수정한 것이다.

모바일용 MS 오피스는 기존 MS 워드와 동일한 규칙으로 RSID가 생성된다. IOS용 MS 오피스에서 생성한 문서파일을 Windows OS로 이동시켜도 RSID는 아무런 변화가 없고, MS 워드에서 문서 편집할 때와 동일한 규칙으로 생성된다.

리브레 오피스(libre office)의 경우 RSID가 “document.xml” 파일이 아닌 “content.xml” 파일에 존재하고 “settings.xml” 파일에도 존재하지만, RSID 생성 규칙이 MS 워드와 다르다. 리브레 오피스를 사용하여 문서를 편집할 경우에는, “content.xml”에 “paragraph-rsid” 와 “officeooo:rsid”라는 tag에 RSID가 할당되어 있고, “officeooo:rsid”를 포함하고 있는 단어, 문단은 다른 문서로 복사되었을 때 글꼴 변경 유무에 관계없이 항상 복사되어 저장된다.

오픈 오피스(open office)와 네이버 오피스(naver office)에서는 RSID가 생성되지 않고 이미 작성된 MS 워드파일을 수정하면 RSID가 없어지는 것을 실험으로 확인하였다.

한컴오피스 2018을 기본 환경으로 설치하면 “한컴오피스 2018 한워드”가 생성되는데, 마우스 우측 버튼을 클릭하여 워드 파일을 생성할 때 자동으로 한컴 오피스 워드로 생성된다. 이 행위는 한컴 오피스 2018을 설치하면 기본 세팅으로 설치되기 때문에, 대부분의 사용자가 마우스 우측 버튼을 클릭하여 워드 파일을 생성하면, 한컴 오피스 워드로 작성된다. 이 경우에는 RSID 규칙이 기존 MS 워드로 작성할 때와 다르다. 한컴 오피스로 작성할 경우 “document.xml” 파일의 가장 첫 rsidR이 항상 고정값으로 할당된다. 한컴 2018 버전에는 첫 rsidR가 “002764DB”로 고정되어 있고, “settings.xml” 파일에 rsidRoot가 “00506824”로 고정되어 있으며 이 값은 두 번 반복되어 저장된다. 이 값은 문자열을 모두 삭제하여도 남아있어 위의 고정된 RSID가 파일 내에 존재하면, 해당 문서가 한컴 오피스를 사용하여 생성한 문서임을 알 수 있다. 따라서 문서파일들의 RSID를 비교하여 관계성을 찾을 때에는 “002764DB”와 “00506824”를 제외하여야 한다.

IV. RSID 분석을 통한 문서파일 조사 방안

MS 워드에서 RSID 분석을 통해 유출행위를 확인하거나 원본과 사본을 구별할 수 있다. 특히, 파일 내 내용들의 작성 순서를 파악할 수 있기 때문에 표

절 여부를 판단할 수 있다. RSID를 활용한 다양한 분석 방법이 있지만, 다른 메타데이터와 마찬가지로 변조될 수 있다는 한계점이 있다. 하지만, “document.xml”과 “settings.xml”의 RSID를 변조하면 시간값이 “1980-01-01 오전 12:00:00”에서 현재 시간으로 변경되어 변조 여부를 확인할 수 있다[4]. 또한, 기존의 메타데이터 조사 방법과는 다르게 RSID를 변경하는 도구는 아직 존재하지 않으며, RSID를 흔적 없이 변조시키기 위해서는 본 논문에서 설명한 RSID의 규칙을 알아야한다. 기존의 RSID 관련 논문에 명시되어 있는 RSID의 규칙으로는 파일 내의 RSID를 임의로 지우거나, 변경할 경우에는 오히려 RSID 규칙에 어긋나 변조한 흔적을 찾아낼 수 있다.

분석을 위한 가장 첫 단계는 두 MS 워드파일의 “document.xml” 파일에서는 rsidR, rsidRDefault, rsidP, rsidRPr를 추출하고 “settings.xml” 파일에서는 rsidRoot 와 모든 RSID를 추출한여 두 파일에서 나온 RSID를 비교한다. Fig. 3은 과정을 도식화한 그림이다.

4.1 문서파일의 외부 유출 및 표절 여부 조사

MS 워드파일의 유출 여부를 조사하기 위해서는 “document.xml” 파일과 “settings.xml” 파일에

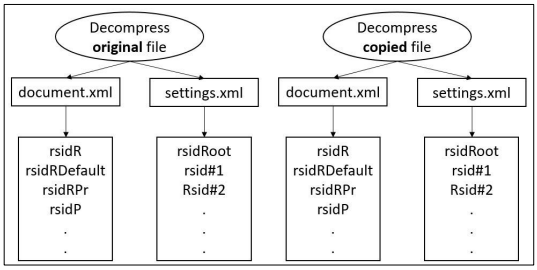


Fig. 3. Basic investigation process of document file leaking or plagiarism

존재하는 RSID를 구분하여 비교하여야 한다. 원본 문서의 모든 RSID를 추출하고, 검색하고자 하는 저장 매체 내 MS 워드파일 중 RSID를 추출하여 동일한 값을 갖는 파일을 먼저 조사하면 정확하고, 시간을 단축하여 확인할 수 있다. Fig. 3과 같이 기본 조사 과정을 완료하면, 심화 조사 과정을 진행한다. 심화 과정은 다음 설명과 같으며, Fig. 4는 도식화한 그림이다.

각 파일의 RSID를 비교하고, 같은 RSID가 없으면 두 파일은 관계성이 없는 것이며, 같은 RSID가 존재한다면, 두 파일의 “document.xml” 내의 RSID가 전부 같은지 확인한다. “document.xml” 과 “settings.xml” 내의 모든 RSID만 같으면 원본을 파일 복사한 상태이다. 만약, “settings.xml”에 하나의 RSID가 더 추가되어 있다면, 이는 원본 파일

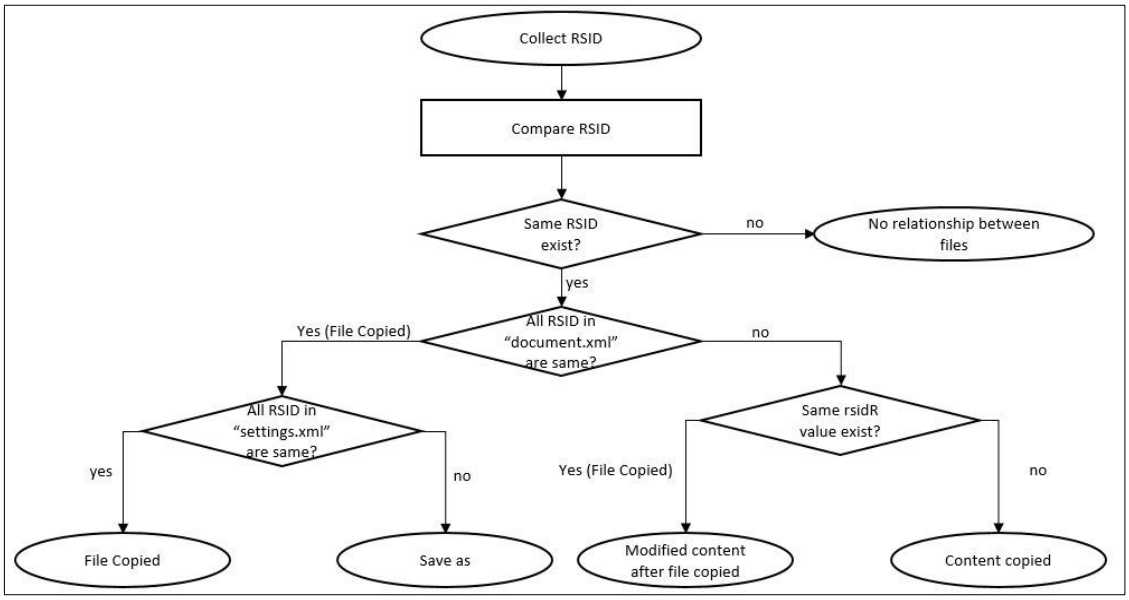


Fig. 4. Advanced investigation process of document file leaking or plagiarism

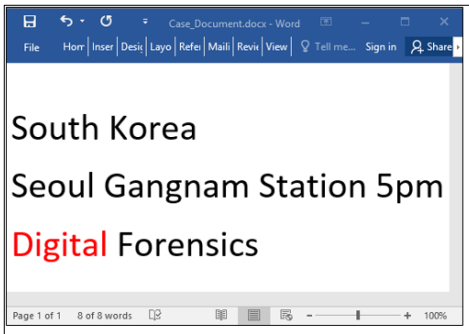
을 다른 이름으로 저장한 파일이다. “document.xml” 내 일부의 RSID가 같은 경우에는 두 파일 내 rsidR이 같은 경우가 하나 이상 존재하는지 확인하여 같은 값이 있으면 원본 파일을 복사한 후 수정한 상태이다. 하지만 “document.xml” 내에 같은 rsidR이 존재하지 않지만, rsidP와 rsidRPr이 같으면 rsidR을 가지고 있는 파일에서 일부 내용을 복사한 경우이다. 두 파일의 rsidR, rsidRDefault가 같으면, “settings.xml” 파일에 RSID의 개수가 더 적은 파일이 원본파일이며, 많은 파일이 사본파일이다.

4.2 문서파일의 내용 수정 이력 조사

문서파일 내 RSID와 메타 데이터를 이용하면 문서파일의 작성 순서를 추적할 수 있다. Fig. 5의 “Case_Document.docx” 파일은 임의의 순서로 내용이 작성된 문서파일이며, 내용만 봐서는 작성 순서를 알 수 없다. 이러한 경우에 RSID를 사용하면 사용자의 문서 작성 순서 및 다양한 정보를 알 수 있다.

먼저, “settings.xml” 파일의 rsidRoot가 “00B45129”이며 이 값이 “document.xml” 파일의 rsidRDefault와 같으므로 MS 워드에서 “새로 만들기“로 생성된 파일임을 알 수 있다. 따라서 작성자는 “South Korea”라는 단어를 가장 처음 작성하고 줄바꿈 후 저장하였다. 그 이유는 “document.xml” 파일 내 “Seoul Gangnam Station 5pm”이라는 문장에 할당된 rsidR과 첫 번째 문장에 할당된 rsidRDefault이 “00B46129”로 같으므로, 이는 첫 번째 문장을 작성한 후 줄 바꿈을 하고, 그 줄 바꾼 곳에서 작성하면 다음과 같이 첫 번째 문장의 rsidRDefault가 다음 문자 rsidR에 남아있게 된다. 하지만, “Seoul Gangnam Station 5pm”이라는 단어는 두 번째 행위가 아닌 마지막 행위이다. “Seoul Gangnam Station 5pm” 문장의 〈w:p〉〈/w:p〉 안을 보면 〈bookmarkStart〉, 〈bookmarkEnd〉가 있는데, 이는 마지막으로 생성/수정한 부분에 해당 태그가 남아있는 것을 확인하였다. 따라서 이 문장은 가장 마지막에 작성된 것이다.

두 번째 행위로 생성한 단어는 “Digital Forensics” 이다. 해당 단어를 포함하고 있는 태그를 보면 “Digital”이라는 단어에는 rsidRPr이 다른 값으로 존재하는데 이는, “Digital Forensics”라는 단어를 작성하고 저장한 후에, “Digital”이라는 단어를 빨간색으로 글꼴을 수정한 것을 알 수 있다. 그리고 위에 언급한 것과 같이 작성자는 마지막 행위도 두 번째 줄에 “Seoul”을 작성하고 “Gangnam



[Case_Document.docx]

```
- <w:rsids>
    <w:rsidRoot w:val="00B45129"/>
    <w:rsid w:val="00084837"/>
    <w:rsid w:val="003B1206"/>
    <w:rsid w:val="0052617C"/>
    <w:rsid w:val="00B45129"/>
    <w:rsid w:val="00DD44C3"/>
- </w:rsids>
```

[settings.xml file]

```
- <w:p w:rsidRDefault="00B45129" w:rsidR="00DD44C3">
  - <w:r>
      <w:t>South Korea</w:t> (1)
    </w:r>
  </w:p>
- <w:p w:rsidRDefault="003B1206" w:rsidR="00B45129">
  - <w:r>
      <w:t xml:space="preserve">Seoul </w:t> (4)
    </w:r>
  - <w:r w:rsidRPr="003B1206">
      <w:t>Gangnam Station 5pm</w:t> (5)
    </w:r>
    <w:bookmarkStart w:id="0" w:name="_GoBack"/>
    <w:bookmarkEnd w:id="0"/>
- </w:p>
- <w:p w:rsidRDefault="0052617C" w:rsidR="0052617C">
  - <w:r w:rsidRPr="00084837">
    - <w:rPr>
        <w:color w:val="FF0000"/>
      </w:rPr>
      <w:t>Digital</w:t> (3)
    </w:r>
  - <w:r>
      <w:t xml:space="preserve">Forensics</w:t> (2)
    </w:r>
  </w:p>
```

[document.xml file]

Fig. 5. document.xml and settings.xml in Case_Document.docx

Station 5pm"라는 문장을 복사하여 저장하였다. "Gangnam Station 5pm" 문장의 RSID를 확인하면 rsidRPr이 할당된 것을 보아 이는 다른 곳에서 복사해온 것이다. 만약 해당 문장을 감싸고 있는 태그에 〈w:b/〉, 〈w:color〉 등의 글꼴이 변경되었을 때 생성되는 태그가 존재하지 않는다면 글꼴이 변해 rsidRPr이 생성된 것이 아니라 다른 곳에서 복사한 것을 알 수 있다. 이 파일의 행위 순서를 정리하면 첫 번째 줄에 "South Korea"을 작성, 줄바꿈 후 저장하고, 줄을 바꾼 후에 세 번째 줄에 "Digital Forensics" 작성 후 저장하였다. 그 후에 "Digital Forensics"에서 "Digital"을 적색으로 변경 후 저장하고, 두 번째 문장에 "Seoul"를 작성 후 "Gangnam Station 5pm"를 다른 곳에서 복사하여 저장하였다.

4.3 적용 사례 소개 : 인터넷 게시글과 분석대상 파일의 문서내용 작성 시점 파악

포렌식 조사팀은 표절행위로 작성되었다고 의심되는 "파일 A"와 원본파일로 의심되는 인터넷 블로그에 작성되어있는 "게시글 B"을 확보하였다. 용의자는 자신이 직접 파일 A를 작성하였다고 주장하고 있으며, 시간값을 확인하면 "게시글 B"가 인터넷에 업로드된 날짜보다 일찍 작성되었다고 주장한다. "게시글 B"는 2018년 5월 10일에 작성되어있고, 파일 A의 만든 시간, 수정된 시간, 접근 시간은 모두 2018년 3월 5일로 되어있으며 수정횟수 또한 0으로 되어있는 상태이다.

조사관은 RSID를 이용하여 용의자의 안티포렌식 행위를 찾고, 표절행위를 조사하였다. 먼저, 파일 A의 "document.xml" 파일에 저장되어 있는 RSID는 총 3개이다. 이는 "파일 A"의 수정 횟수와는 다르게, 최소 3회 이상 수정하고 저장한 것을 알 수 있다. 두 번째로, "'파일 A"와 "게시글B"는 내용이 같은 부분이 다수 존재하며, 파일 A에 인터넷에 올라온 파일의 내용이 조금 수정되거나 추가된 상태이다. 파일 A의 "document.xml" 파일 내 rsidP가 있는 것으로 보아 이는 적접 작성한 것이 아닌 다른 소스에서 복사했다는 증거가 될 수 있으며, "글 B"와 다른 내용에는 rsidR 두 개가 할당되어 있음을 확인하였다.

결론적으로, 용의자의 주장과 "파일 A"의 정보와는 다르게, 용의자는 인터넷 게시글에서 복사하고, 일부분을 수정하였고 또한, 시간 값을 과거로 변경하고 수정 횟수를 변조하였음을 RSID를 이용한 조사방법으로 알아내었다.

V. 결 론

본 논문에서는 문서 작업 중에 MS 워드 파일 내에 기록되는 RSID를 이용한 디지털 포렌식 분석 기법을 연구하였다. RSID는 문서를 작성함에 따라 생성되는 고유한 값으로 이를 활용하여 문서 자체의 복사 여부뿐만 아니라 문서 내 내용의 수정 및 복사 여부를 알 수 있다. 작성 이력을 추적하여 안티포렌식 행위, 표절, 유출, 무결성 등의 다양한 조사방법으로 활용될 수 있을 것으로 기대되며, 분석한 결과를 활용하기 위해서 RSID 규칙을 수집할 필요가 있다.

References

[1] B. Park, J. Park, and S. Lee, "Data concealment and detection in Microsoft Office 2007 files," Digital Investigation, vol. 5, no. 3-4, pp. 104-114, Mar. 2009.

[2] Z. Fu, X. Sun, Y. Liu, and B. Li, "Forensic investigation of OOXML format documents," Digital Investigation, vol. 8, no. 1, pp. 48-55, Jul. 2011.

[3] ECMA, "ECMA-376-1:2016 Office Open XML file format - fondamentals and markup language reference," ECMA International Publication, Oct. 2016.

[4] E. Didriksen, "Forensic analysis of OOXML documents," MS. Thesis, Gjøvik University College, 2014.

[5] H. Langweg, "OOXML file analysis of the July 22nd terrorist manual," 13th International Conference on Communications and Multimedia Security, Sep. 2012.

[6] S.L. Garfinkel and J.J. Migletz, "New XML-based files implications for forensics," IEEE Security and Privacy, vol. 7, no. 2, Mar-Apr. 2009.

[7] H. Chung, J. Park, and S. Lee,

"Forensic analysis of residual information in adobe PDF files," Communications in Computer and Information Science, vol. 185, 2011.
[8] Y.M. Lee and S. Lee, "A Study for Forensic Methods of MS Excel Files," MS. Thesis, Korea University, 2015.
[9] D. Jeong and S. Lee, "Study on the tracking revision history of MS Word files for forensic investigation," Digital Investigation, vol. 23, pp. 3-10, Dec. 2017.

〈저자소개〉

전 지 훈 (Jihun Joun) 학생회원
2016년 5월: Pennsylvania State University 사이버 보안학과 졸업
2017년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 디지털 포렌식, 파일시스템, 역공학

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

정 두 원 (Doowon Jeong) 학생회원
2011년 8월: 고려대학교 공과대학 산업경영공학과 공학사
2011년 9월~현재: 고려대학교 정보보호대학원 정보보호학과 석·박사통합과정
〈관심분야〉 디지털 포렌식, 정보보호, 빅데이터 분석

이 상 진 (Sangjin 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, 심층암호, 해쉬함수