情報保護學會論文誌
第18卷 第3號, 2008. 6

Microsoft Office 2007 파일에의 정보 은닉 및 탐지 방법*

박 보 라†, 박 정 흠, 이 상 진‡
고려대학교 정보보호기술연구센터

Information Hiding and Detection in MS Office 2007 file*

Bora Park†, JungHeum Park, Sangjin Lee‡
Center for Information Security Technologies, Korea University

요 약

정보 은닉 기술은 최근 들어 주목받고 있는 기술이다. 정보 은닉 기술을 보유하고 사용하는 것은 보안이 요구되는 통신 환경에서 경쟁력이 되기 때문이다. 본 논문에서는 2007년 초에 출시된 Microsoft Office 2007 파일에 정보를 은닉하는 것이 가능함을 보이고자 한다. Microsoft Office 파일 형식(format)이 Open XML을 따름을 고려할 때, Open XML의 특징을 이용하여 Microsoft Office 파일에 정보를 숨길 수 있다. Open XML 형식에서는 파트(Part)와 각 파트간의 관계를 사용자가 정의할 수 있는데, 이러한 사용자 정의의 파트와 사용자 정의의 관계가 Microsoft Office 2007 파일에 정보를 숨기는 데에 핵심적인 역할을 한다. Microsoft Office 응용 프로그램으로 만들어진 파일에는 이러한 사용자 정의의 파트와 파트간 관계가 존재하지 않으므로 이러한 요소의 존재 여부로, 정보가 은닉되었는지의 여부를 판단할 수도 있다.

ABSTRACT

Information hiding is a very important technology recently. Having this technology can be a competitive power for secure communication. In this paper, it will be showed that hiding data in MS Office 2007 file is possible. Considering Microsoft (MS) Office 2007 file format is based on Open XML format, the feature of Open XML format makes it possible to hide data in MS Office 2007 file. In Open XML format, unknown XML files and their relationships can be defined by user. These parts and relationships are used to hide data in MS Office 2007 file. Considering unknown parts and unknown relationships are not in normal MS Office 2007 file, the hidden data can be detected by confirming of unknown parts and unknown relationships.

Keywords : MS Office 2007, Open XML, unknown relationshi, unknown part, Data hiding

Ⅰ. 서 론

정보 은닉은 안티포렌식의 중요한 수단이며, 탐지하는 것이 현실적으로 상당히 어렵다. 대부분의 정보은닉은 영상 파일과 같은 대용량 멀티미디어 파일에 감시자가 알아볼 수 없는 형태로 데이터를 삽입하는 방식인

접수일 : 2007년 11월 7일; 수정일 : 2008년 2월 5일;
채택일 : 2008년 2월 25일
* 본 연구는 과학재단 디지털 정보 획득 기반기술 연구(M10740030004-07N4003-00410)의 지원으로 수행되었습니다.
† 주저자, danver123@korea.ac.kr
‡ 교신저자, sangjin@korea.ac.kr

스테가노그라피가 일반적이다. 하지만 감시하는 측에서는 탐지의 어려움 때문에 멀티미디어 파일이 전송되면 데이터의 은닉 여부와 상관없이 원래의 의미를 훼손하지 않으면서 멀티미디어 파일을 변조하여 전송함으로써 데이터 은닉 자체를 불가능하게 한다.

따라서 원본 데이터를 훼손할 수 없는 대상에 데이터를 은닉할 필요성이 있다. 사실 거의 모든 디지털 파일에 데이터를 은닉할 수 있다. 정보를 은닉하고자 하는 자는 감시하는 자가 알지 못하는 방식을 사용해야 하며, 감시하는 자는 삽입할 수 있는 방법을 미리 파악하여 중요한 정보가 외부로 빠져나가는 것을 방지해야 할 것이다.

각종 워드프로세서로 작성된 파일 역시 데이터 은닉의 대상이 될 수 있으며, 'Microsoft Office 파일'(이하 'MS Office 파일')의 경우 2003 이전 버전에 데이터를 은닉할 수 있음이 알려져 있다[2]. 기업에서 기밀 유출 방지책으로 이메일의 내용을 단순히 열람하고 있으나, Word 파일에 데이터를 은닉하여 첨부물 형태로 유출한다면 현재까지는 별다른 대책이 없다. 그러므로 산업 기밀 유출 방지 차원에서 MS Office 파일에 데이터를 은닉할 수 있는 방법과 은닉된 데이터를 탐지할 수 있는 방법에 대해 연구하는 것은 매우 중요하다.

본 논문에서는 MS Office 2007 버전의 문서 파일에 데이터를 은닉하고, 탐지하는 방법을 설명한다. 최근 발표 된 MS Office 2007 버전의 문서 파일은 기존 버전(MS Office 2003 이하)과 완전히 다른 새로운 형식으로 데이터를 저장한다. MS Office 2007은 데이터 관리의 효율을 높이기 위해 Open XML 파일 형식에 기반한 포멧을 만들어서 Word(docx), PowerPoint pptx), Excel(xlsx) 문서 파일에 적용하였다. 본 논문은 MS Office Word 2007 파일(docx)를 대상으로 하고 있으나, PowerPoint, Excel에도 Word와 유사한 방식이 적용 가능하다.

본 논문은 다음과 같은 순서로 구성되어 있다. 2절에서 관련 연구로 이전 버전의 MS Office 파일 혹은 XML 파일등에 데이터를 은닉하는 것에 대하여 다루고, 3절에서는 Open XML 형식에 기반을 둔 MS Office 2007 파일 형식에 대하여 설명한다. 4절에서 Office Open XML 형식을 이용한 정보 은닉 방법에 대해 다루고, 5절에서 은닉된 정보를 탐지하는 방법에 대해 다루며, 탐지 툴로 은닉된 데이터를 탐지해 본다. 마지막으로 6절에서 결론을 내린다.

Ⅱ. 관련 연구

디지털 포렌식 관점에서는 응용 프로그램으로 확인할 수 없는 데이터를 확인하는 것이 매우 중요하다. 이러한 데이터는 문서를 작성한 사람의 이름, 문서의 요약 정보와 같은 사용자에게 무해한 내용일 수도 있고 비도덕적이거나 위법적인 컨텐츠를 포함한 것일 수도 있다. 포렌식 수사관의 입장에서는 유해한 내용뿐 아니라 무해한 모든 내용의 데이터를 이용하여 수사를 진행해야 하기 때문에 응용 프로그램으로 확인할 수 없는 데이터를 추출해낼 수 있어야 한다[8]. 실제로 문서 파일을 이용한 데이터 은닉 및 통신이 많이 이루어지고 있으며 Byers가 [8]에서 제시한 바에 따르면 100,000개의 MS Word 문서 파일을 무작위로 수집했을 때, 각각의 파일에 은닉된 텍스트가 존재하였다. 이러한 은닉된 데이터를 삭제하기 위한 도구 역시 출시되었으나(Payne Group(2005)[9], Soft Experience (2005)[10], Workshare(2005)[11] 등) 존재하는 모든 은닉 데이터를 삭제하지는 못하며 이러한 도구를 이용하여 은닉된 데이터를 삭제했다고 하더라도 여전히 또 다른 은닉된 데이터가 존재하는 경우가 많다. MS Office 제품군은 전 세계적으로 널리 사용되고 있는 프로그램이며 MS Office 문서 파일에 데이터를 은닉하는 기술은 메타데이터를 삽입하기 위한 목적 이외에도 꾸준히 연구되고 있다. MS Office 문서 파일에 존재하는 은닉된 데이터를 탐지하는 기술 또한 디지털 포렌식 목적으로 연구되고 있다. 하지만 지금까지의 연구는 복합 문서 형식을 사용하는 2003 이하 버전의 MS Office 파일을 대상으로 한 것이며, Open XML 형식을 사용하는 MS Office 2007 문서 파일부터는 데이터 은닉 및 탐지 방법에 있어 새로운 연구가 진행되어야 한다. 다음 문단에서는 MS Office 2003 이하 버전과 2007 버전의 차이점을 간단히 언급하고, MS Office 2003 이하 버전의 파일에 대하여 연구된 데이터 은닉 기술을 간단히 소개한다.

2003 이하 버전의 MS Office 응용 프로그램은 데이터를 저장하기 위하여 복합 문서 형식(Compound File Format)을 사용하였다[3]. 2008년 1월까지 MS Office 응용 프로그램이 복합 문서 형식으로 파일을 저장하는 특성을 이용하거나, 복합 문서 형식의 특성을 이용하여 데이터를 은닉하는 많은 방법이 제시되고 있다. 우선

2003이하의 MS Office 파워포인트 응용 프로그램에서 파일을 복합 문서 형식으로 저장할 때 효율적으로 저장하기 위한 '빠르게 저장하기' 옵션 때문에 생길 수 있는 잉여정보를 이용하여 데이터를 은닉할 수 있음이 제시되었다[1]. '빠르게 저장하기' 옵션이 선택되어 있으면 파일을 저장할 때, 삭제되거나 수정되기 이전의 슬라이드를 삭제하는 것이 아니라 해당 스트림에의 오프셋만 변경하는 식으로 파일을 저장한다. 즉, 복합 문서 형식을 재구성하지 않고 새로운 스트림들을 추가하고 오프셋만 변경하기 때문에 문서를 저장하는 속도를 향상시킨다. 따라서 파워포인트 응용 프로그램으로 확인할 수 없는 영역이 파일 내에 존재한다. 또 다른 은닉 방법으로는, 복합 문서 형식이 파일 시스템의 구조와 유사성을 가지는 특징을 이용하여 파일 시스템처럼 복합 문서 역시 슬랙(Slack) 공간을 가지며 이러한 슬랙 공간을 이용하여 데이터를 은닉할 수 있는 방법이 제시되었다[2]. 또한 MS Office 응용 프로그램에서 인식할 수 없는 스트림을 생성하여 데이터를 은닉하는 것이 가능함을 보인 논문도 발표되었다[6]. 한편, MS Office 2007 응용 프로그램은 2003 버전에서 사용했던 복합 문서 형식(Compound File Format)이 아닌 Open XML 형식을 이용하여 데이터를 저장한다. MS Office 응용 프로그램에서는 복합 문서 형식을 가지는 파일들에 대한 호환성을 유지하고 있지만, 문서 형식이 완전히 달라졌기 때문에 복합 문서 형식을 가진 MS Office 파일들에 데이터를 은닉하는 방법들을 Open XML 형식을 사용하는 MS Office 2007 파일에는 더 이상 사용할 수 없다. 따라서 Open XML 형식을 사용하는 MS Office 2007 파일에 데이터를 은닉하기 위해서는 새로운 방법이 필요하다. [7]에서는 XML 파일에 데이터를 은닉하는 방법을 제시하고 있다. 즉, XML 스키마를 이용하여 데이터를 은닉하는 것이 가능함을 보이고 있다. 따라서 Open XML 형식을 사용하는 MS Office 2007 파일에 데이터를 은닉하려면, Open XML 형식과 해당 형식에서 사용하는 스키마를 분석하고 Open XML 형식의 파일을 MS Office 2007 응용 프로그램에서 읽을 때 어떠한 특성을 가지는지 분석해야 한다.

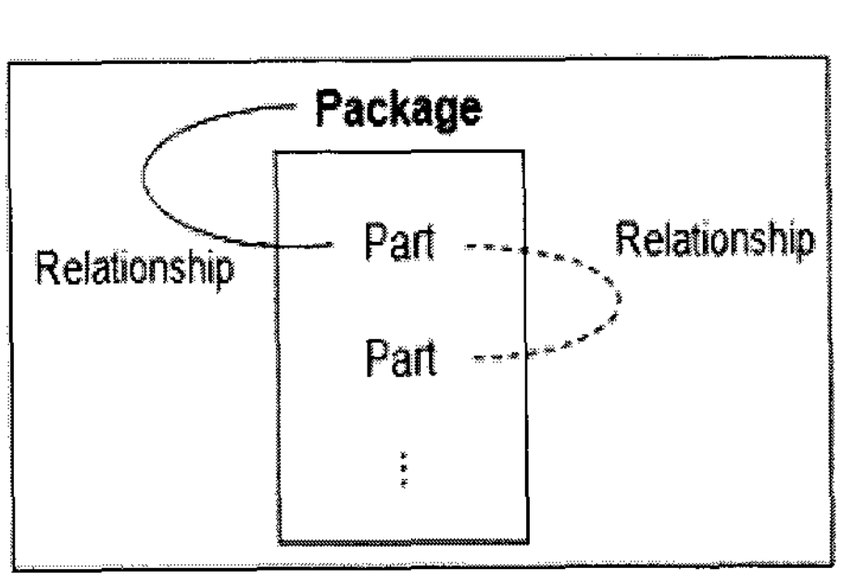
Ⅲ. MS Office 2007 파일 형식(MS Office 2007 File Format)

Open XML 형식을 사용하는 파일은 패키지(Package)라는 컨테이너 안에 저장된 여러 파트(Part)들과 그 파트들 간의 관계로 이루어진다. Office Open XML 파일 형식의 패키지 구조는 관계에 따라 크게 좌우된다. 사용자 혹은 응용 프로그램이 패키지 내에 파트를 만들 때, 각 파트를 서로 간의 관계를 바탕으로 패키지에 적절히 연결하지 못하는 경우, MS Office 2007 응용 프로그램이 각 파트를 인식하지 못하게 된다. 각 파트는 자신을 포함하는 패키지와의 관계 또는 다른 파트와의 관계를 가지고 있어야 하기 때문이다. 이러한 관계는 관계 패키지와 최상위 파트 간의 관계를 정의하는 '패키지-관계 파트'와 한 패키지 내에서 두 개의 파트 간 상/하위 관계를 정의하는 '파트 관계'가 있다[4].

MS Office 2007 파일은 여러 개의 압축된 파일과 압축되지 않은 파일들로 구성된다. Open XML 형식을 가지는 파일은 개방형 패키징 관례(OPC)로 인해 기본 파일 형식이 문서의 구체적 내용에 관계없이 동일하다. 개방형 패키징 관례(Open Packaging Convention, OPC)에서는 문서를 완전히 나타내기 위해 XML과 이미지, 메타데이터 등과 같은 다양한 유형의 내용을 저장하는 방법을 제공한다. 이를 위하여 포함 내용과 관계를 나타내는 논리적 모델이 설명되어 있다. OPC에서 권장하는 구현방식에서는 ZIP 파일 형식을 이용한다. 그러므로 ZIP 뷰어를 사용하여 Open XML 구조를 볼 수 있다[5].

[그림 1]에는 Open XML 파일을 구성하는 패키지와 파트, 그리고 관계가 나타나 있으며 이들에 대한 정의는 다음과 같다.

- 패키지 : ZIP 아카이브
- 파트 : ZIP 아카이브 내의 파일(들)
- 관계 : 파트와 패키지 간의 관계 혹은 파트와 파트 간의 관계



〔그림 1〕 Open XML 파일 구성 요소

3.1 패키지(Package)

패키지는 OPC에 명시된, 파트라고 불리는 문서 구성 요소들을 포함하고 있는 하나의 ZIP 아카이브이다. 즉, 사용자 관점에서는 하나의 MS Office 2007 파일 자체가 패키지라고 할 수 있다[5].

3.2 파트(Part)

파트는 패키지 내의 파일을 의미한다. 예를 들어, MS Office 2007 응용 프로그램에서 만들어진 Open XML 형식을 가지는 파일의 압축을 풀어보면 여러 개의 XML 파일과 기타 형식의 파일이 있는 것을 확인할 수 있는데 이러한 파일들을 파트라고 할 수 있다[5].

각각의 파트는 서로 다른 콘텐츠 유형을 가진다. MS Office 2007 응용 프로그램에서 만들어진 Open XML 형식을 가지는 파일을 구성해내기 위해서 각 파트들은 XML 형식으로 기술되어 있는데, 이 때 각 파트들은 특정 XML 스키마를 따르고 있다. 이 스키마들은 MS Office 2007 파일의 특성이나 구성요소들의 형식을 정의하기 위해 사용되고 있다. MS Office 2007 파일에서 사용될 수 있는 XML 스키마는 모두 문서화되어 있다[4].

3.3 관계(Relationship)

관계 파트는 각 파트가 하나의 파일을 만들어 내는 파트 조합 방법을 명시한 파일이다. 관계 파트에서는 실제 소스(source) 파트가 참조하는 타깃(target) 파트를 명시함으로써 해당 파트에서 특성 요소(예를 들어 텍스트 혹은 이미지)가 가지는 속성 등을 대응시키는 역할을 한다. 관계 파트도 XML 형식으로 이루어져 있으며 패키지-관계 파트는 패키지 내의 _rels\.rels 로서 존재한다[5].

관계를 사용하는 것은, 하나의 파트가 다른 파트와 어떻게 연결되는지를 알 수 있게 한다. 이는 각 파트의 특정 타깃에 해당 관계 파일 내에서 유일한 ID를 부여하는 형식으로 가능하다. 이러한 방법은 실제로 타깃의 콘텐츠 타입을 일일이 확인하지 않더라도 소스 파트와 타깃을 연결시킬 수 있다는 장점이 있다. 이러한 ID는 Office Open XML 스키마와는 독립적으로 존재할 수 있으며, 하나의 관계 파트 내에 같은 아이디를 가진 타깃이 존재할 수는 없다. 관계는 다음과 같은 형식으로 명시된다.

```
<Relationship Id= "ID"
        Type="relationshipType"
        Target="targetPart"
        (Targetmode="Internal/External") />
```

여기서 ID 값은 문자열(string)로 표현할 수 있으며 이는 해당 관계 파트에서 유일한 것이어야 한다. 타입(Type) 은 해당 콘텐츠의 타입을 명시하는 것으로 이는 Open XML 형식에 제시된 스키마를 지정하고 있다. 타깃(Target)은 소스 파트에서 참조하고자 하는 타깃의 경로를 지정하고 있다. 타깃 속성은 해당 타깃이 파일 내부에 있는지 외부에 있는지에 따라 다른 형태로 표현될 수 있다. 마지막으로 Target mode는 앞서 제시된 타깃이 패키지 내부에 있는지 혹은 패키지 외부에 있는지를 표현하는 값이다. 'Internal' 혹은 'External'값을 가질 수 있다. ID, 타입, 타깃은 필수적인 요소이며 Target mode는 생략되어도 무관하다. Target mode가 생략된다면 기본 값인 'Internal'로 설정된다[4].

관계는 '명시적 관계(Explicit relationship)'과 '묵시적 관계(Implicit relationship)'로 분류될 수 있다. 명시적 관계의 경우, 소스 파트의 ID 값과 관계 파일의 ID 값을 참조함으로써 소스 파트에서 참조하고자 하는 타깃을 직접적으로 찾아갈 수 있게 된다. 예를 들어, 하이퍼링크를 가진 문서 파일에서 문서 파트(document part)는 그 하이퍼링크의 ID 속성이 관계 파트에 정의되었을 때 관계 파일에서의 하이퍼링크 요소 이름을 참조하여 해당 타깃을 찾아갈 수 있게 된다. 명시적 관계가 아닌 모든 관계를 묵시적 관계라고 한다.

예를 들어 Office Open XML 형식을 가진 파일에 존재하는 관계를 살펴보자. MS Office 2007 워드 파일에 JPEG 형식을 가진 이미지 파일이 존재한다고 가정하자. 이 때 소스 파트가 되는 document.xml에 다음과 같은 XML 코드가 존재한다.

```
<w:pict>
        <v:image data r:Id="rId4"/>
</w:pict>
```

이 XML 코드는 이 문서파일이 rId4라는 ID를 가진 이미지 파일을 타깃으로 가진다는 의미이다. 응용 프로그램이 문서를 여는 도중 이 코드를 만나면 응용 프로

그램은 ID가 rId4인 타깃을 찾기 시작한다. 관계 파트 내에는 모든 타깃에 대한 ID 정보가 기술되어 있다. 이 같은 경우, 다음과 같은 XML 코드가 관계 파트에 기술되어 있다.

```
<Relationships ... >
    <Relationship Id="rId4"
    Type = "/relationship/image"
    Target = "media/image1.jpeg"/>
</Relationships>
```

응용 프로그램은 이 관계 파트 내에서 ID가 rId4인 타깃을 찾을 수 있고 이에 해당하는 타깃을 문서 파일 내에 삽입할 수 있게 된다. 이는 명시적 관계의 간단한 예이며, 묵시적 관계인 경우는 참조가 좀 더 복잡하게 이루어지게 된다.

Ⅳ. MS Office 2007 파일에의 정보 은닉 방법

OPC는 Office Open XML 문서 파일을 나타내기 위한 여러 가지 규칙을 명시하는 것과 동시에 사용자가 임의로 파일 형식을 확장하여 사용할 수 있도록 추가적인 기능을 제공하고 있다[4]. 특히 사용자 정의의 파트(Unknown Part)와 사용자 정의 관계(Unknown Relationship)가 사용자에 의한 문서 파일의 임의 확장을 위해 존재하는 대표적인 요소이다. 이러한 요소를 이용하여 MS Office 2007 파일에 정보를 은닉하는 것이 가능하다.

4.1 사용자 정의 파트(Unknown Part)

사용자 정의 파트란, 관계 파일에서 유효한 타깃으로 지정되지 않은 모든 파트를 의미한다. 사용자 정의 파트는 MS Office 2007 응용 프로그램에서 확인하지 않으므로 문서 파일을 열 때 무시된다. 즉, 사용자 정의 파트는 파일 내에 정상적으로 존재하지만 해당 응용 프로그램으로는 확인할 수 없는 부분이 된다.

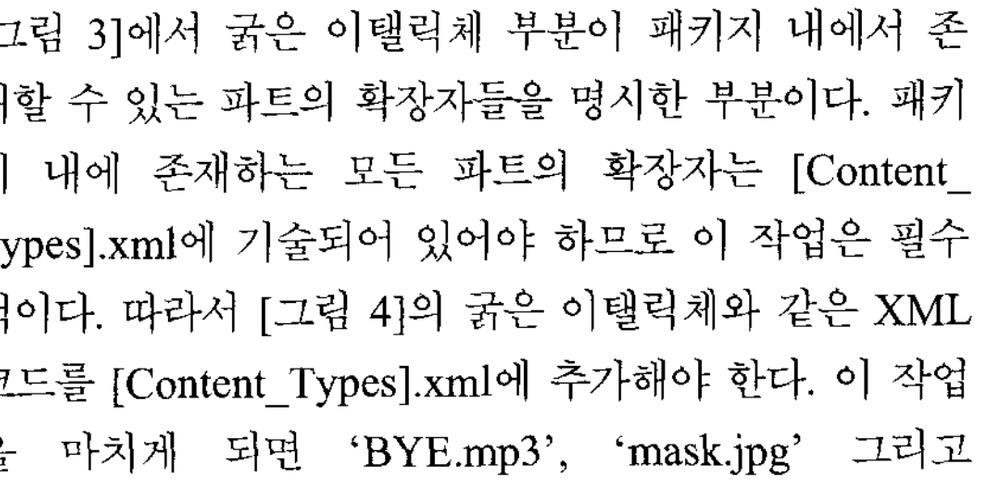
4.2 사용자 정의 관계(Unknown Relationship)

'ECMA 376'표준에 정의되지 않은 모든 관계는 사용자 정의 관계이다. 사용자 정의 관계가 OPC에 명시된 대로 정의된다면 이는 유효한 것으로 간주된다[4]. 유효한 사용자 정의 관계를 가진 파트 역시, 응용 프로그램으로 정상적으로 열리며, 이러한 관계가 정의된 문서를 '다른 이름으로 저장'하더라도 이러한 관계를 가진 파트들은 그대로 남아있게 된다. 만약 이러한 관계가 정의된 문서를 복합 문서 형식으로 저장하게 되면 즉, MS Office 2003 이하 버전으로 저장하게 되면 문서 형식이 재구성되면서 은닉된 데이터는 삭제되고, 유효한 데이터만 남는다.

4.3 사용자 정의 파트와 사용자 정의 관계를 이용한 정보 은닉 방법

이 절에서는 사용자 정의 파트와 사용자 정의 관계를 이용하여 MS Office 2007 파일에 데이터를 은닉하는 방법을 제시하고 간단한 예를 보인다.

우선 [그림 2]와 같이 패키지 내에 은닉하고자 하는 데이터를 삽입한다. 그 다음으로 최상위 폴더에 있는 [Contents_Types].xml에서 은닉한 파일 즉, 은닉한 사용자 정의의 파트들의 확장자를 명시해 주어야 한다. [그림 3]에서 굵은 이탤릭체 부분이 패키지 내에서 존재할 수 있는 파트의 확장자들을 명시한 부분이다. 패키지 내에 존재하는 모든 파트의 확장자는 [Content_Types].xml에 기술되어 있어야 하므로 이 작업은 필수적이다. 따라서 [그림 4]의 굵은 이탤릭체와 같은 XML 코드를 [Content_Types].xml에 추가해야 한다. 이 작업을 마치게 되면 'BYE.mp3', 'mask.jpg' 그리고 'sysinternals.zip'은 사용자 정의 파트가 된다. 이러한 파트들을 포함한 MS Office 2007 파일을 MS Office 2007 응용 프로그램으로 열면, 경고 창 없이 정상적으로 열린다. 이는 은닉하고자 했던 데이터가 완벽하게 은닉되었음을 의미한다. 이 때 이 은닉된 파트들에 대해 사용자 정의의 관계를 지정해 주면 이 파일들은 문서를



〔그림 2〕 패키지에 정보 은닉

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types xmlns="…">
<Override PartName="/word/footnotes.xml" ContentType="…"/>

<Default Extension="jpeg" ContentType="image/jpeg"/>
<Default Extension="rels"
ContentType="application/vnd.openxmlformats-
package.relationships+xml"/>
<Default Extension="xml" ContentType="application/xml"/>

<Override PartName="/word/document.xml" ContentType="…"/>
                    …
</Types>
```

〔그림 3〕 〔Content_Types〕.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Types xmlns="...">
<Override PartName="/word/footnotes.xml" ContentType="..."/>

<Default Extension="jpeg" ContentType="image/jpeg"/>
<Default Extension="rels"
ContentType="application/vnd.openxmlformats-
package.relationships+xml"/>
<Default Extension="xml" ContentType="application/xml"/>
<Default Extension="zip" ContentType="application/zip"/>
<Default Extension="mp3" ContentType="application/mp3"/>
<Default Extension="jpg" ContentType="application/jpg"/>

<Override PartName="/word/document.xml" ContentType="/>
                    …
</Types>
```

〔그림 4〕 수정한 〔Content_Types〕.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="…">
<Relationship Id="rId3"
Type="http://schemas.openxmlformats.org/officeDocument/2006/
relationships/extended-properties" Target="docProps/app.xml"/>
<Relationship Id="rId2"
Type="http://schemas.openxmlformats.org/package/2006/relation
ships/metadata/core-properties" Target="docProps/core.xml"/>
<Relationship Id="rId1"
Type="http://schemas.openxmlformats.org/officeDocument/2006/
relationships/officeDocument" Target="word/document.xml"/>
</Relationships>
```

〔그림 5〕 관계 파트

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns="…">
<Relationship Id="rId3" Type="…" Target="docProps/app.xml"/>
<Relationship Id="rId2" Type="…" Target="docProps/core.xml"/>
<Relationship Id="rId1" Type="…" Target="word/document.xml"/>

<Relationship Id="rId100"
Type="http://schemas.openxmlformats.org/officeDocument/2006/rela
tionships/a" Target="word/media/sysinternals.zip"/>

<Relationship Id="rId101"
Type="http://schemas.openxmlformats.org/officeDocument/2006/rela
tionships/b" Target="mask.jpg"/>

<Relationship Id="rId102"
Type="http://schemas.openxmlformats.org/officeDocument/2006/rela
tionships/c" Target="word/BYE.mp3"/>
</Relationships>
```

〔그림 6〕 수정한 관계 파트

수정하거나 '다른 이름으로 저장'하는 경우에도 사라지지 않고 그대로 남아있게 된다. 이후에는 이 사용자 정의 파트에 대하여 사용자 정의 관계를 부여하는 방법에 대해 기술한다.

[그림 5]에서 관계 파일 내부의 관계 요소(element)가 ID, 타입 그리고 타깃을 속성(attribute)으로 가지고 있음을 확인할 수 있다. [그림 6]에서 굵은 이탤릭체 코드는 사용자 정의의 파트들의 관계 즉, 은닉된 데이터의 관계를 정의하고 있는 부분이다. 예를 들어 'BYE.mp3'의 관계는 다음과 같다. 'BYE.mp3'의 ID는 rId102이며 이것의 타입은 "http://schemas.OpenXMLformats.org/officeDocument/2006/reltionships/c"이다. 이 ID 값은 해당 관계 파일에서 유일한 값이며, 해당 관계 파일 내에서 유일하기만 하다면 어떤 값이어도 상관없다. 또한 타입 'http://schemas.OpenXMLformats.org/officeDocument/2006/reltionships/c'은 'BYE.mp3'의 형식(format)을 명시하는 것이다. 이 때, 사용자가 Open XML 형식에서 사용하는 타입을 사용하게 되면 문서를 열 때, 'BYE.mp3'가 명시된 타입을 따르지 않아 경고창이 나타나므로 타입을 기술할 때는 Open XML 형식에서 사용하지 않는 임의의 문자열(예를 들어, [그림 6]의 경우 a, b, c)을 작성하여 넣도록 한다. 이 경우, 타입이 스펙에 명시된 경우가 아니므로 응용 프로그램이 해당 타입을 해석하지 않아 경고창이 나타나지 않는다. 즉, Open XML 형식을 사용하는 정상적인 MS Office 2007 파일의 경우 소스(Source) 파트에서 ID를 참조하여 타깃 파일을 찾을 수 있고, 그 타깃 파일을 주어진 타입에 따라 해석한다. 하지만 이 경우는 소스 파트에서 은닉하고자 하는 타깃의 ID가 명시되어 있지 않고 또한 MS Office 2007 응용 프로그램에서는 사용자 정의 파트나 사용자 정의의 타입이 지정된 사용자 정의 관계는 참조하지 않으므로 이러한 관계 설정이 가능하다.

[Content_Types].xml 파트와 관계 파트를 수정하고 나서 은닉된 데이터를 가진 문서파일을 응용 프로그램으로 열면, 경고 창 없이 정상적으로 열린다. 또한 사용자가 MS Office 2007 응용 프로그램을 이용하여 문서를 수정하거나 다시 저장하더라도 은닉한 데이터가 그대로 남아있다. 이러한 데이터 은닉 방법은 MS Office

2007 응용 프로그램이 문서를 열 때, 사용자 정의 파트와 사용자 정의 관계를 체크하지 않기 때문에 가능하다.

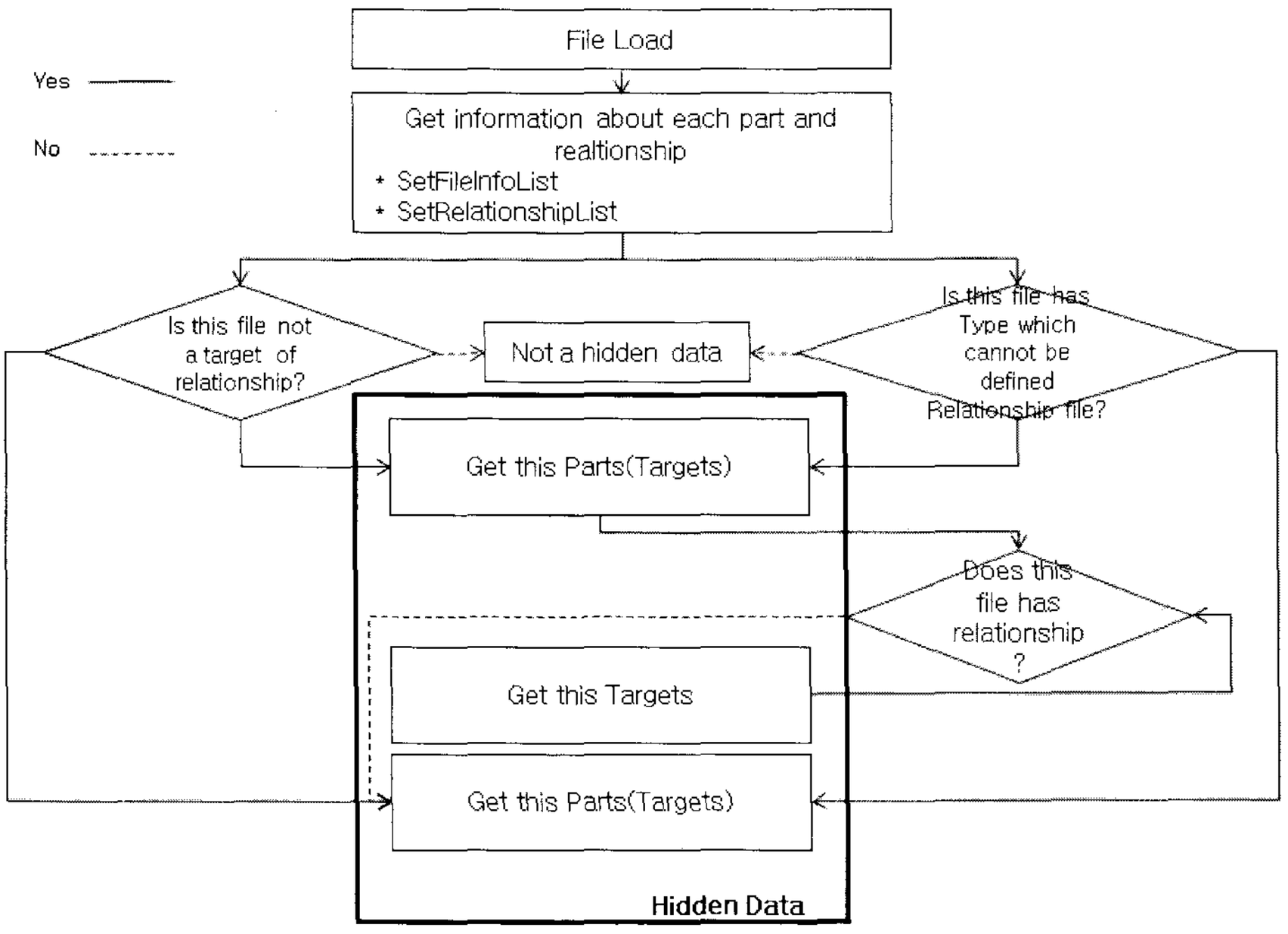
4.4 주석문을 이용한 데이터 은닉

XML 주석문을 이용하여 데이터를 은닉하는 것 또한 가능하다. MS Office 2007 응용 프로그램을 이용하여 정상적으로 만들어진 파일 내의 XML 문서에는 주석문이 존재하지 않는다. 이 방법은 MS Office 2007 파일에 데이터를 숨길 수 있는 가장 간단한 방법이라 할 수 있다.

V. 은닉된 데이터의 탐지 방법

이전 절에서는 MS Office 2007 파일에 데이터를 숨기는 방법에 대해 소개하였다. 이번 절에서는 은닉한 데이터를 탐지하는 알고리즘을 제시한다. 또한 그 이 알고리즘을 사용하여 은닉한 데이터를 탐지하는 도구를 제작한다. 앞 절에서 사용자 정의 파트와 사용자 정의 관계를 이용하여 MS Office 2007 파일에 정보를 숨기는 것이 가능함을 보였다. 따라서 사용자 정의 파트의 특징과 사용자 정의 관계의 특징을 이용하여 은닉된 데이터를 탐지하는 것이 가능하다. 탐지 알고리즘은 [그림 7]에 나타난 것과 같다.

은닉된 데이터가 존재하는 MS Office 2007 파일을 읽어 들인 후에 탐지 툴은 해당 파일 내에 존재하는 모든 파트와 관계에 대한 정보를 수집한다. 파트에 대한 정보는 SetFileInfoList 구조체에 저장된다. 이 구조체는 다음과 같다. 'IsUnknownPart'는 존재하는 파트가 사용자 정의 파트인지 아닌지를 명시하는 변수이다. 즉, 관계가 정의되어 있지 않거나 정의된 관계 파일 내에 [4]에 명시되지 않은 타입을 가지고 있는 파트인지를 알려주는 변수이다. 데이터를 은닉할 때, 사용자 관계를 정의해 주지 않더라도 이렇게 은닉된 데이터를 탐지하는 것이 가능하다. 'IsUnknownRel'은 이 파트가 사용자 정의의 관계 파일인지 아닌지를 명시하는 변수이다. 해당 관계 파일 내에 [4]에 명시되지 않은 타입이 정의되어 있으면 해당 파트를 사용자 정의의 관계 파일이라고 인식한다. 'HasComment'는 현재의 XML 파일이 주석문을 가지고 있으면 True, 그렇지 않으면 False를 명시하는 변수이다. 'DirPath'는 읽은 파트의 절대 경로를 의미하는 변수이고, 'FileName'은 읽은 파트의 파일이름을 의미하는 변수이다. 마지막으로 'comment'는 주석문이 존재할 때, 즉 HasComment가 True인 경우 파



〔그림 7〕 은닉 파트 탐지 알고리즘

일 내에 존재하는 주석문을 이 변수에 계속 추가하게 된다. 즉, 'comment'는 해당 XML 파일 내에 존재하는 주석문이 저장되는 변수이다.

```
struct FileInfo
{       public bool IsUnknownPart;
        public bool IsUnknownRel;
        public bool HasComment;
        public string DirPath;
        public string FileName;
        public string comment;          }
```

이러한 정보 수집과정이 끝나면 관계 파트에 대한 정보 수집을 위해서 다음과 같은 두 가지 작업을 한다.

첫째, 각각의 파트가 관계 파트 내에서 타깃으로서 존재하는지 확인한다. 만약, 타깃으로서 존재하지 않는 파트라면 이것은 은닉된 데이터이다. 만약, 이 은닉된 데이터가 관계 파트라면 이 관계 파트에 명시된 모든 타깃은 은닉된 데이터이다.

둘째, 타깃의 타입이 Open XML 표준에 명시된 타입이 아닌 경우, 이는 은닉된 데이터이다. 만약, 이 데이터가 관계 파트라면 이 관계 파트에 명시된 모든 타깃은 은닉된 데이터이다.

이 알고리즘에 대한 의사코드를 [그림 8~11]에 나타내었다. [그림 8]에서는 탐지 절차의 순서가 나타나 있다.

먼저, 문서 파일을 읽어 들인 후에 압축을 해제한다. 그 후에 압축을 해제하여 얻은 모든 파트와 관계 파일에 대한 정보를 수집한다. [그림 9~11]은 사용자 정의 파트와 사용자 정의 관계를 탐지하는 방법에 대한 의사코드이다. 우선 프로그램은 특정 파트가 관계 파일에 정의된 타깃인지 아닌지를 살펴본다. 만약, 타깃으로서 존재하는 파트가 아니라면 이는 은닉된 데이터로 결정된다. 게다가 이 은닉된 데이터가 관계 파트인지 아닌지를

```
FileLoad();

If(file != NULL)
{
    Detect.Initialization();

    Detect.ExtractZip(filepath);

    Detect.SetFileInfoList(subdir);
    Detect.SetRelationshipList();

    Detect.FindUnknownPart();
    Detect.FindUnknownRel("_rels₩₩₩.rels");
    Detect.FindXMLComment();
}
```

[그림 8] 의사 코드 1 : 탐지 순서

```
public static void FindUnknownPart( )
{
    for (int i = 0; i < Number of files in Package; i++)
    {
        Fill FileInfoList with this file;
        Get t_FileInfo structure of ;th file
        Get path of this file;

        foreach (t_Relationships tRelationships in RelationshipList)
        {
            string rels_path = tRelationships.rels_path;
            foreach (t_Relationship tRelationship in tRelationships.arrRelationships)
            {
                string absoluteTargetpath = GetAbsolutePath(rels_path, tRelationship.target);
                if (FilePath.Equals(absoluteTargetpath))
                {
                    this file is not unknown part
                    break;
                }
            }
        }

        if (this file is unknown part)
        {
            if (FilePath.LastIndexOf(".rels") != FilePath.Length - 5)
            {
                tFileInfo.IsUnknownPart = true;
            }
            else
            {
                if (IsNormalRels(tFileInfo.DirPath, tFileInfo.FileName) == false)
                    tFileInfo.IsUnknownPart = true;
            }

            Renew the FileInfoList
        }
    }
}
```

[그림 9] 의사 코드 2 : 사용자 정의 파트 탐지

```
private static void FindUnknownRel_from_Root(string rels_path)
{
    Fill the ReltionshipList( );

    for(int i = 0; i < reltaionship file in Package; i++)
    {
        Fill RelationshipList with this relationship file( );

        if (path of rel file in RelationshipList.Equals(rels_path))
        {
            for (int j = 0; j < tRelationships.arrRelationships.Count; j++)
            {
                tRelationship = (t_Relationship)tRelationships.arrRelationships[j];

                for (int k = 0; k < KnownRelationships_in_Word.Length/2; k++)
                {
                    if (KnownRelationships_in_Word[k, 0].Equals(tRelationship.type)
                       && KnownRelationships_in_Word[k, 1].Equals(tRelationship.target))
                    {
                        this relationship file is not unknown relationship
                        break;
                    }
                }

                if (unknown relationship file)
                {
                     if (has sub relationship file)
                     {
                          FindUnknownRel_from_Root(tmppath);
                     }
                }
                else
                     this relationship is not unknown relationship file.
            }
            break;
        }
    }
}
```

〔그림 10〕 의사 코드 3 : 사용자 정의 관계 탐지 1

```
private static void FindUnknownRel_from_UnknownPart()
{
     foreach (t_FileInfo tFileInfo  in FileInfoList)
     {
          if(tFileInfo.IsUnknownPart == true)
          {
               string FilePath = tFileInfo.DirPath + "₩₩" + tFileInfo.FileName;

               int index = FilePath.LastIndexOf("₩₩");
               string  tmpfilename = FilePath.Substring(index + 1, FilePath.Length - index - 1);
               string  tmppath = FilePath.Substring(0, index);
               tmppath = tmppath + "₩₩_rels₩₩" + tmpfilename + ".rels";

               if (HasRels(tmppath))
               {
                    SetIsUnknownRel(tmppath);
                    FindUnknownRel_from_Root(tmppath);
               }
               else
                    continue;
          }
     }
}
```

〔그림 11〕 의사 코드 4 : 사용자 정의 관계 탐지 2

살펴보고, 관계 파트라면 이 관계 파트에 명시되어 있는 모든 타깃을 은닉된 데이터라고 결정하게 된다. 하위에 관계 파트가 발견되지 않을 때까지, 이러한 작업이 계속된다.

그 다음으로 프로그램은 관계 파트 내에 기술되어 있는 각 타깃의 타입을 확인한다. 만약, 그 타입이 Open XML 표준에 이미 명시된 것이 아니라면 이는 사용자가 임의로 조작한 타입이라고 보고 이 타입을 가지는 타깃을 은닉된 데이터라고 결정한다. 예를 들어 MS Office 2007 워드 파일의 경우

"http://schemas.OpenXMLformats.org/officeDocument/2006/relationships/officeDocument",

"http://schemas.OpenXMLformats.org/officeDocument/2006/relationships/extended-properties",

"http://schemas.OpenXMLformats.org/officeDocument/2006/relationships/metadata/core-properties",

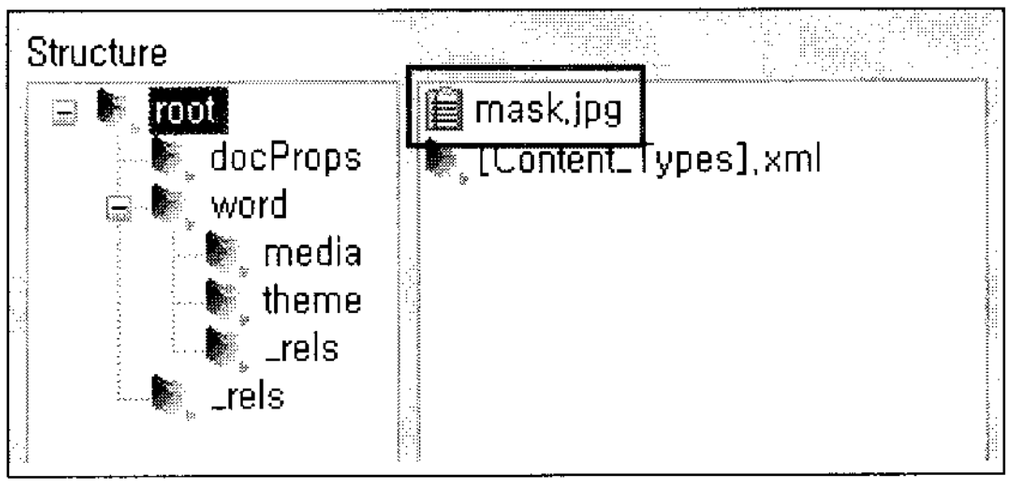
"http://schemas.OpenXMLformats.org/officeDocument/2006/relationships/digital-signature/origin",

"http://schemas.OpenXMLformats.org/officeDocument/2006/relationships/metadata/thumbnail",

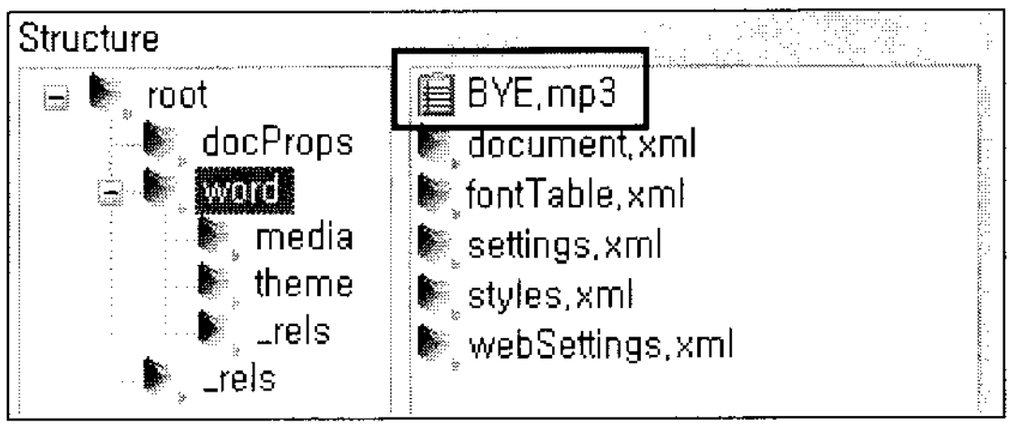
등과 같은 타입을 가질 수 있다.

[그림 12~14]에 위와 같은 알고리즘을 이용하여 은닉된 데이터를 탐지한 결과가 나타나 있다. [그림 5]에서 은닉하였던 'BYE.mp3', 'mask.jpg' 그리고 'sysinternal.zip'이 정상적인 파일들과는 다른 모양의 아이콘으로 표시되고 있음을 확인할 수 있다.

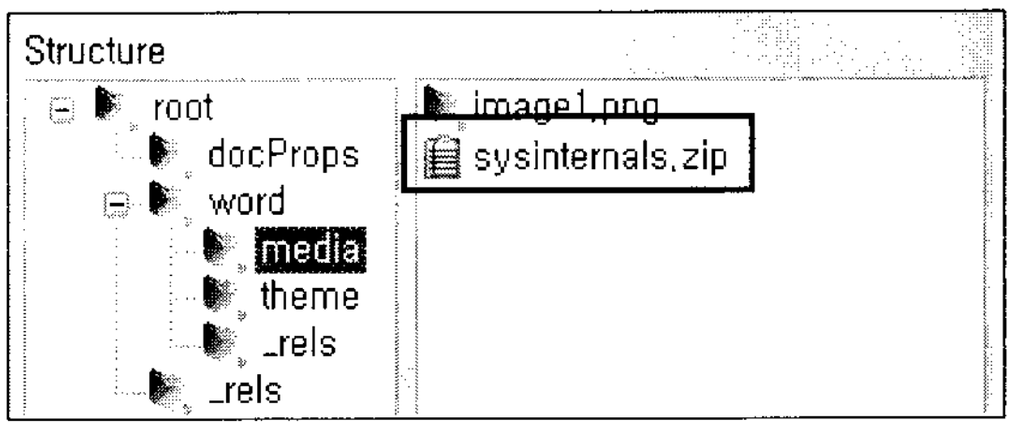
MS Office 2007 응용 프로그램에서 생성한 XML 파일은 주석문을 포함하지 않는다. 따라서 XML 주석문이 존재하는 경우, 사용자가 임의로 데이터를 삽입한 것으로 간주할 수 있다. XML 파일 내의 주석문을 검색하는 방법은 다음과 같다. 우선 각 XML 파일을 XML 객체에 로드하여 각 노드의 타입을 검사한다. 노드의 타입이 'Comment'인 경우에 해당 파일의 FileInfo 구조체의 HasComment를 True로 셋팅하고, 해당 주석문을 저장한다. 이 때, 하나의 노드는 여러 단계의 서브 노드를 가질 수 있기 때문에 재귀적으로 모든 노드를 탐색해야 한다. 결과적으로 각 XML 파일에 주석문이 존재하는 경우, FileInfo 구조체의 HasComment가 True가 되며, 존재하는 모든 주석문이 저장된다. [그림 15]는 주석문을 가지고 있는 XML 파일을 포함한 docx파일의 은닉



〔그림 12〕 탐지 결과 1



〔그림 13〕 탐지 결과 2



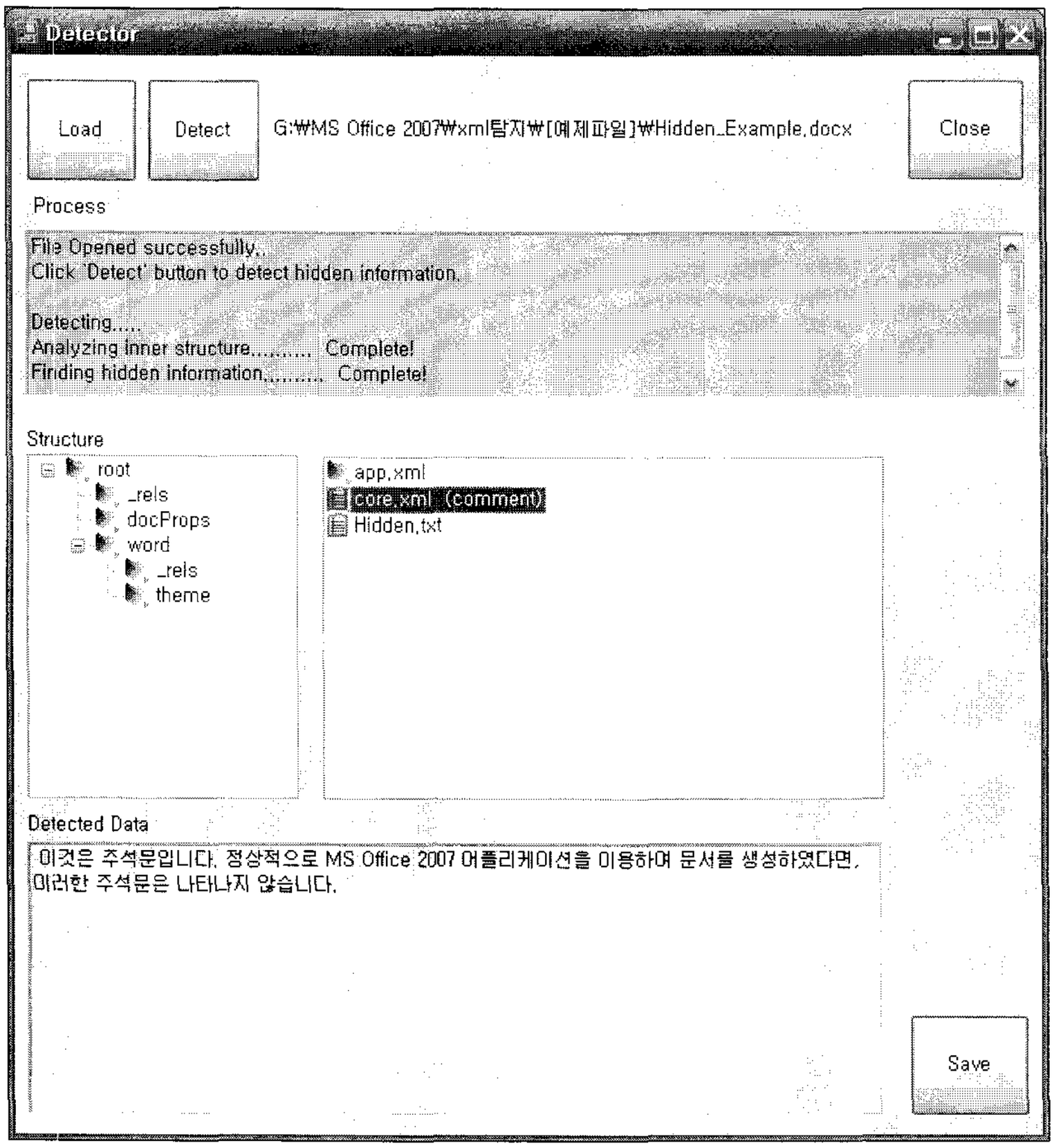
〔그림 14〕 탐지 결과 3

데이터 탐지 결과이다. 사용자 정의 파트인 'Hidden.txt'와 주석문을 가진 'core.xml'파일이 나타나 있다. 'Detected Data' 영역에는 'core.xml'에 존재하는 주석문의 내용이 표시된다.

Ⅵ. 결 론

현재까지의 정보 은닉은 파일시스템의 빈 공간을 사용하거나 멀티미디어 파일에 데이터를 삽입하는 형태로 이루어졌다. 그러나 파일시스템의 빈 공간은 쉽게 탐지 가능하며, 스테가노그라피는 전송되는 파일을 변조하면 쉽게 막을 수 있다. 반면에 일반 파일에 데이터를 삽입하는 방법에 대해서는 많이 주목하지 않았지만 본 논문에서 제시한 것처럼 일반 파일에도 데이터를 삽입할 수 있다.

따라서 기업 기밀 유출 방지를 위해서는 외부로 나가는 모든 파일을 대상으로 보내고자 하는 데이터 이외의



〔그림 15〕 은닉된 데이터 및 주석문 탐지 결과

것이 포함되어 있나 주의해야 하며, 본 논문에서 제시한 MS Word 2007 파일을 대상으로 하는 정보 은닉의 탐지 기법은 나름대로 의미가 있다고 판단된다. 또한 디지털 포렌식 수사에서도 단순히 문서 파일의 내용만을 분석하는 것이 아니라 각 파일에 정보가 은닉되어 있는지 검사해 볼 필요가 있다.

본 논문에서는 Open XML 형식을 사용하는 MS Office 2007 문서에 응용 프로그램이 접근하지 않는 영역을 생성하여 데이터를 은닉하는 방법을 제시하였다. 특히 MS Office 2007 파일 중 워드(docx) 파일을 중심으로 Open XML 형식을 사용하는 MS Office 2007 파일에 정보를 은닉하고 탐지하는 것이 가능함을 보였다. Open XML 형식을 가진, 엑셀(xslx)이나 파워포인트(pptx) 파일 역시 유사한 방법으로 데이터 은닉이 가능하고 은닉한 데이터의 탐지가 가능하다. 이와 같이 파일 형식을 분석하여 데이터를 은닉하는 방법과 탐지 방법을 연구하는 것은 디지털 포렌식 수사를 위해 꼭 필요하며, 이를 통해 문서 파일에 대한 조사를 보다 효과적으로 할 수 있을 것이다.

향후에는 현재의 은닉된 데이터 탐지 도구에 엑셀(xslx)과 파워포인트(pptx) 파일에 존재할 수 있는 은닉된 데이터를 탐지하는 기능을 추가할 것이며, MS의 Open XML 형식을 사용하는 파일과는 별개로 2006년 ISO 표준 (ISO 26300)으로 승인받은 Open Document 형식을 사용하는 파일에도 데이터를 은닉할 수 있는지에 대한 연구를 할 계획이다.

참고문헌

[1] Jung Heum Park, Bora Park, Sangjin Lee, Seokhie Hong, and Jong Hyuk Park, “Extraction of Residual Information in the Microsoft

PowerPoint file from the Viewpoint of Digital Forensics considering PerCom Environment," The 2nd International Workshop on Web and Pervasive Security, IEEE, pp.584-589, March 2008.

[2] A. Castiglione, De Santis, C. Soriente, "Taking advantages of a disadvantage : Digital forensics and steganography using document metadata", The Journal of Systems and software, vol 80, Issue 5, pp.750-764, May 2007

[3] Daniel Rentz, "Microsoft Compound Document File Format", http://sc.openoffice.org

[4] ECMA-376 1st Edition, "Office Open XML File Formats", December 2006

[5] Frank Rice, Microsoft Corporation, "Introducing the Office(2007) Open XML File Formats", May 2006, http://msdn2.microsoft.com/ko-kr/library/aa338205.aspx

[6] Hyukdon Kwon, Yeog Kim, Sangin Lee, Jongin Lim, "A Tool for the Detection of Hidden Data in Microsoft Compound Document File Format", International Conference on Information Science and Security ICISS 2008. p.141~146.

[7] Singo INOUE, Kyoko MAKINO, Ichiro MURASE, Osamu TAKIZAWA, Tsutomu MATSUMOTO, Hiroshi NAKAGAWA, "A Proposal on Information Hiding Methods using XML", http://takisawa.ne.jp/nlp_xml.pdf

[8] Byers, S., 2004. Information leakage caused by hidden data in published documents. IEEE Security Privacy 2 (2), 23-27.

[9] Payne Group, 2005. Metadata Assistant. Available from : http://www.payneconsulting.com/products.

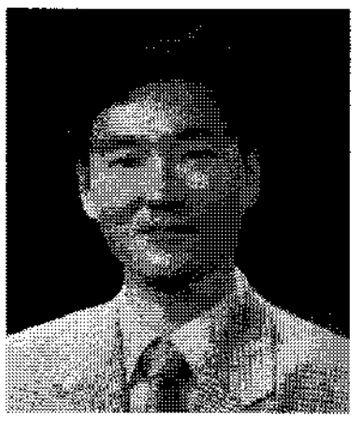
[10] Soft Experience, 2005. Catalogue. Available from : http://peccatte.karefil.com/software/Catalogue.

[11] Workshare, 2005. Trace. Available from : http://www.workshare.com

〈著者紹介〉

박 보 라 (Bora Park) 학생회원
2007년 2월 : 부산대학교 수학교육과 학사
2007년 3월~ : 고려대학교 정보경영공학전문대학원 석사과정
<관심분야> 디지털 포렌식, 정보 은닉 이론

박 정 흠 (Jung Heum Park) 학생회원
2007년 2월 : 한양대학교 정보통신대학 컴퓨터전공 학사
2007년 3월~ : 고려대학교 정보경영공학전문대학원 석사과정
<관심분야> 디지털 포렌식, 역공학, 시스템 보안



이 상 진 (Sangjin Lee) 종신회원
1989년 2월~1999년 2월 : 한국전자통신연구원 선임 연구원,
1999년 2월~2001년 8월 : 고려대학교 자연과학대학 조교수,
2001년 9월~현재 : 고려대학교 정보경영공학전문대학원 교수
<관심분야> 대칭키 암호. 정보은닉이론, 컴퓨터 포렌식