Journal of The Korea Institute of Information Security & Cryptology(JKIISC) ISSN 1598-3986(Print)
VOL.23, NO.4, August 2013 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2013.23.4.655

포렌식 관점의 파티션 복구 기법에 관한 연구*

남 궁 재 웅,† 홍 일 영, 박 정 흠, 이 상 진‡
고려대학교 정보보호대학원

A research for partition recovery method in a forensic perspective*

Jaeung Namgung,† Ilyoung Hong, Jungheum Park, Sangjin Lee‡
Center for Information Security Technologies(CIST), Korea University

요 약

저장장치의 용량이 점차 대형화됨에 따라 대부분의 사용자들은 데이터의 저장 및 관리의 편의를 위하여 저장장치를 논리적으로 여러 개의 파티션으로 나누어 사용한다. 따라서 인위적인 파티션 은닉이나 파티션 손상 등으로부터 안정적으로 파티션을 복구해내는 것은 디지털 포렌식 관점에서 매우 중요한 문제이다.

본 논문은 파티션이 은닉되어 있거나 파티션 영역의 손상으로 인하여 파티션이 구분되지 않는 경우, 각 파일 시스템의 특징을 이용하여 안정적이고 효율적인 분석이 가능한 파티션 복구 알고리즘에 대해 제시한다.

ABSTRACT

As the capacity of storage devices becomes larger, most users divide them into several logical partitions for convenience of storing and controlling data. Therefore, recovering partitions stably which are artificially hidden or damaged is the most important issue in the perspective of digital forensic.

This research suggests partition recovery algorithm that makes stable and effective analysis using characteristics of each file system. This algorithm is available when partition is not distinguishable due to concealment of partition or damage in partition area.

Keywords: Digital Forensics, Partition recovery, Master boot record, Volume boot record, Super block, Boot record

I. 서 론

최근 저장장치들의 안전성 향상으로 인하여 물리적인 결함에 의한 저장장치의 손상은 점차 줄어들고 있지만 Disttrack, Dropper, Userinit 등과 같이 MBR을 주공격 대상으로 한 악성코드의 감염사례는 점차 늘고 있다. MBR을 공격 대상으로 한 악성코드들은 부트 코드에 악성코드를 삽입하여 좀비 PC로써 기능을 수행하도록 부팅하게 하거나 MBR의 모든 데이터를 삭제하여 부팅이 되지 않도록 하고 Partition Table을 통해 파티션을 나누지 못하도록 하여 데이터를 잃게 만드는 악성 행위를 수행한다. 일반적으로 MBR의 데이터를 다른 영역에 백업하지 않기 때문에 악성코드로 인하여 MBR의 Partition Table의 데이터가 손상되면 시스템에 존재하는 대부분의 데이터를 잃게 된다.

디지털 포렌식 관점에서 파티션의 존재 여부는 데이터를 분석함에 있어 큰 영향을 미칠 수 있다. 각 파티션에 존재하는 파일 시스템으로부터 각 데이터의 생

접수일(2013년 3월 18일), 수정일(2013년 6월 3일), 게재확정일(2013년 6월 24일)
* 본 연구는 산업통상자원부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음.[10035157, 실시간 분석을 위한 디지털 포렌식 기술 개발]
† 주저자. mgw_leader@korea.ac.kr
‡ 교신저자. sangjin@korea.ac.kr(Corresponding author)

성, 수정, 실행에 대한 시간 정보를 얻을 수 있으며 데이터가 담긴 파일명, 파일의 경로 등과 같은 메타 정보는 파일 시스템을 통해서 확인할 수 있기 때문에 파일 시스템을 포함하고 있는 파티션의 존재여부는 매우 중요하다.

따라서 파티션 정보의 손상으로 인하여 시스템이 파티션을 인식하지 못하는 경우에는 파티션을 복구하여 분석할 수 있도록 해야 한다. 또한 이전 시스템에 존재하였던 파티션이나 VM이나 백업 파일등에 존재하는 파티션을 복구하여 분석할 수 있는 환경을 제공하는 것도 디지털 포렌식 관점에서 상당히 큰 의미를 가질 수 있다. 하지만 아직까지 파티션을 복구하기 위한 연구가 많이 부족한 상태이며 파티션을 복구하기 위한 기존 도구들이 공개되어 있지만 포렌식 분석을 하는 관점에서 많은 한계를 가지고 있다.

본 논문에서는 MBR 영역의 손상으로 인한 파티션 손상이나 이전 시스템의 파티션 복구 및 현 시스템의 파일 내부에 존재하는 파티션(VM, 백업 이미지 등)을 복구하여 포렌식 관점에서 의미 있는 분석을 하기 위한 방안을 제시한다.

의미 있는 파티션 복구를 위한 방안을 제시하기 위하여 본 논문은 다음과 같이 구성된다. 2절에서는 관련된 배경지식에 대해서 설명하며, 3절에서는 기존의 파티션 복구 도구의 한계점을 설명한다. 4절에서는 각 파일 시스템의 특징을 통하여 파티션을 복구 및 검증 방법, 제안하는 알고리즘에 대해서 설명하며 제안하는 알고리즘에 대한 성능 평가를 5절을 통해 설명한 뒤, 6절에서 결론과 함께 향후 연구계획을 밝힌다.

II. 배경지식

일반적으로 파티션은 데이터 저장 및 관리 등의 편의를 위하여 사용자에 의해 생성되며 시스템에 의해 관리된다. 시스템은 생성된 파티션의 크기 정보와 파티션의 위치 정보 등을 MBR(Master Boot Record)의 Partition Table에 저장한다. Partition Table은 16 바이트씩 총 4개의 파티션 정보를 저장할 수 있도록 구성되어 있으며 시스템은 부팅 시, Partition Table에 저장된 각 파티션의 속성과 위치, 크기 정보를 통하여 파티션을 구분한다. 각각의 파티션은 사용자가 어떠한 파일 시스템으로 포맷했는지에 따라 해당 파티션의 시작위치에 VBR(Volume Boot Record)이 생성되기도 하고 Super Block이 생성되기도 한다. 일반적으로 FAT 파일 시스템과 NTFS는 파티션의 시작을 VBR로 시작하며 EXT 파일 시스템은 파티션의 시작을 Super Block으로 시작한다. 이와 같이 MBR과 VBR, Super Block 등은 파티션과 구조적으로 밀접한 관련을 가지고 있으며 포렌식 분석을 위하여 은닉 또는 삭제된 파티션을 복구해야 하는 경우에 중요한 단서로써 활용될 수 있다. 따라서 안정적으로 파티션을 복구하기 위해서는 MBR과 VBR, Super Block, 각 파일 시스템들의 대한 정확한 이해가 필요하다[1][2][3].

III. 기존 파티션 복구 도구의 한계

파티션 복구 도구는 시스템 상의 실제 손상된 파티션을 복구하기 위한 용도로 사용되는 복구 도구와 복구 대상의 이미지로부터 가상으로 파티션을 복구하여 분석을 하기 위한 용도의 도구로 분류할 수 있다.

실제 손상된 파티션을 복구하기 위한 용도의 도구로는 EaseUS의 Partition Recovery와 오픈소스로 제공되고 있는 TestDisk가 있으며, 이미지로부터 가상으로 파티션을 복구하여 파일 시스템을 분석하기 위한 용도의 포렌식용 도구로는 Guidance 사의 EnCase가 있다[4][5][6].

Partition Recovery와 TestDisk는 활성 시스템의 저장 장치 내에 존재하는 VBR 및 Super Block을 검색하여 사용자에게 그 결과를 보여준다. 사용자는 검색된 결과 중에서 복구를 원하는 파티션을 입력할 수 있으며 입력된 파티션은, MBR의 Partition Table에 저장되어 복구가 완성된다. 이러한 도구들은 활성상태에서 동작하기 때문에 무결성을 보장할 수 없으며 해당 파티션을 복구하기 위해서는 매번 새롭게 MBR의 파티션 테이블의 내용을 수정해야 한다. 따라서 포렌식 관점에서 이 도구들은 분석을 위한 용도로 활용하기가 힘들다는 단점이 있다.

포렌식 전문 도구인 EnCase는 다른 일반적인 복구 도구보다 분석에 더 적합한 기능을 제공한다. 포렌식 전문 도구는 파티션 복구 대상의 시스템을 이미지 형태로 읽은 후, 가상으로 파티션을 복구하여 분석을 시도할 수 있다. 따라서 포렌식 전문 도구를 사용하는 경우에는 무결성을 보장할 수 있다. 또한 파티션을 가상으로 복구하여 파일 시스템을 분석하기 때문에 자유롭게 복구 또는 복구 취소를 하며 분석할 수 있는 장점이 있다. 하지만 포렌식 전문 도구가 지원하는 파티션 복구 기능에도 한계가 존재한다. 우선, 포렌식 도구는 내장 기능인 파티션 검색을 통해 발견한 각 파티

션이 실제로 유효한 파티션인지 여부를 판단해주지 않는다. 또한 파일 시스템의 특징에 의해 존재하는 중복된 VBR 및 Super Block에 대하여 중복제거를 해주지 않는다. 마지막으로, EnCase는 사용자가 생성한 VM이미지나 백업 파일 안에 존재하는 VBR 및 Super Block을 정확하게 탐지하지 못하는 한계를 지니고 있다. 따라서 무결성을 보장하면서 분석의 편의와 다양한 기능을 제공할 수 있는 연구가 필요하다.

IV. 효과적인 파티션 복구를 위한 알고리즘 제안

본 절에서는 기존의 파티션 복구 도구의 한계점을 극복하기 위하여 저장장치 내에 다수의 파티션이 생성되는 이유에 대하여 상세히 알아보고 포렌식 관점에서 의미 있는 분석을 하기 위하여 각 파티션을 어떻게 효과적으로 복구 및 검증할 것인지에 대해서 제안한다.

4.1 VBR 및 Super Block의 생성 요인

저장장치 내에는 사용자나 시스템의 쓰임에 따라 다수의 VBR 및 Super Block이 존재할 수 있다. VBR은 저장장치 내에서 하나의 파티션을 구분하기 위해 생성되기 때문에 저장장치를 다수의 파티션으로 구분하는 경우, VBR이 파티션의 개수만큼 생성된다. 또한 Super Block은 VBR의 기능을 대부분 포함하고 있으며 EXT 파일 시스템에서 파티션을 구분하기 위한 역할을 포함하고 있기 때문에 다수의 파티션이 존재하는 경우, 다수의 Super Block이 존재할 수 있다. 〔표 1〕과 같이 일반적으로 VBR과 Super Block은 사용자가 시스템을 포맷하면 새롭게 생성되며 이전의 시스템에 존재하던 VBR이나 Super Block이 완전히 삭제되지 않아 남아 있는 경우도 있다. 윈도우 시스템의 경우에는 필요에 의해 파일단위의 VBR을 생성하는 경우도 있으며 사용자가 특정 용도를 위해 생성한 백업 이미지나 가상 이미지에 포함되는 경우도 존재한다. 〔그림 1〕은 EnCase를 사용하여 저장장치 내에 검색된 VBR 및 Super Block을 나타낸 것으로 하나의 저장장치 내에 얼마나 많은 VBR 및 Super Block이 생성되었는지 보여준다.

〔표 1〕 VBR 및 Super Block의 생성

생성 주체	VBR 및 Super Block의 생성 이유
시스템	pagefile.sys, boot.sdi의 활용을 위해 시스템이 생성하는 경우
	이전 시스템에 존재하였으나 완전히 삭제되지 않고 남아있는 경우
가상 시스템	가상 이미지 내, 시스템이 생성하는 경우
사용자	사용자가 편의를 위해 파티션을 나누는 경우
	백업 및 가상 이미지가 부팅을 위해 포함하고 있는 경우

Name	Comment
Unused Disk Area	EXT3: 72758272, Bookmark Start: 1048576...
Unused Disk Area	NTFS: 8369801, Bookmark Start: 92903086...
Unused Disk Area	NTFS: 8369801, Bookmark Start: 93143649...
Unused Disk Area	NTFS: 6173, Bookmark Start: 10285768704...
Unused Disk Area	NTFS: 6173, Bookmark Start: 10288939008...
Unused Disk Area	NTFS: 586108927, Bookmark Start: 104913...
Unused Disk Area	EXFAT: 1163280720, Bookmark Start: 1275...
Unused Disk Area	EXFAT: 0, Bookmark Start: 12752271872, B...
Unused Disk Area	FAT: 0, Bookmark Start: 12752365056, Bo...
Unused Disk Area	FAT: 2097152, Bookmark Start: 127523655...

〔그림 1〕 저장장치 내부에 존재하는 다양한 VBR 및 Super Block 예

4.2 파티션의 검색 및 검증

존재하는 모든 파티션의 복구를 위해서는 각 파티션의 정보를 갖고 있는 VBR 및 Super Block의 검색이 필요하다. FAT, NTFS, EXT 등의 파일시스템들은 각각 VBR, Super Block을 시작으로 하나의 파티션을 시작하기 때문에 VBR과 Super Block의 검색을 통하여 각 파티션의 시작위치를 찾을 수 있다. 하지만 단순 검색에 의해 찾은 VBR과 Super Block은 오래 전에 사용된 흔적이어서 파일 시스템이 완전하지 않은 경우가 존재한다. 또한 각 파일 시스템들의 특징으로 인하여 중복이 되는 경우도 발생하며 VBR 또는 Super Block과 비슷한 형식을 가진 데이터가 오탐으로 검색되는 경우도 발생할 수 있다. 따라서 오탐과 중복을 줄이고 완전한 파일 시스템으로 존재하는 VBR 및 Super Block을 검색하기 위해서는 VBR 및 Super Block의 내부적인 특징을 파악하여 검색하기 위한 시그니처로 활용해야 하며 각 파일 시스템의 특징을 파악하여 검색한 VBR 및 Super Block에 포함된 파일 시스템의 정상여부를 검증하는 절차가 필요하다.

4.2.1 FAT 파티션 검색 및 검증 기법

시스템 상에서 파티션을 구성하는 VBR은 섹터의

시작과 함께 VBR의 구조가 시작된다. VBR은 일반적으로 하나의 섹터크기인 512 바이트로 구성되어 있기 때문에 VBR의 특징을 활용하여 섹터 단위로 VBR을 검색할 수 있다. 〔표 2〕는 VBR 검색에 활용되는 특징 요소들을 나타내었다.

〔표 2〕 VBR 검색에 활용할 특징 요소〔2〕

오프셋	항 목	설 명	값
0	Jump code	부트 코드로 점프	0xEB
3-10	OEM Name	FAT16	MSWIN4.1
			MSDOS5.0
		FAT32	MSWIN4.1
			MSDOS5.0
		NTFS	NTFS
54-61	File System Type	FAT16	FAT16
82-89		FAT32	FAT32
510-511	Signature	시그니처 항목	0xAA55

VBR은 0번 오프셋에서 Jump Code를 통하여 부트 코드로 이동하는 과정을 거친다. 따라서 VBR의 0번 오프셋은 일반적으로 0xEB값을 갖는다.

VBR의 3~10 오프셋은 해당 VBR이 어떠한 파일 시스템으로 이루어져 있는지 직관적으로 확인할 수 있는 몇 가지 형태의 스트링을 제공한다. 일반적으로 FAT32와 FAT16 파일 시스템을 사용하는 VBR은 OEM Name으로 “MSDOS5.0” 또는 “MSWIN4.1”의 문자열을 갖는다. 따라서 이 영역에 저장된 문자열을 통하여 VBR이 사용하는 파일 시스템을 판단할 수 있다. 하지만 OEM Name 영역은 문자열을 통해 파일시스템의 정보를 제공하는 것 이외에는 어떠한 역할도 담당하지 않는다. 그래서 OEM Name 영역은 어떠한 수정이 가해져도 시스템 상으로 아무런 문제가 없다. 따라서 쉽게 위조도 가능할 수 있는 영역이며 전적으로 신뢰할 수 없는 영역이기도 하다. 하지만 정형화된 방식으로 문자열을 남기고 있기 때문에 VBR을 검색하는데 있어서 분명 유용하게 활용될 수 있는 문자열이다.

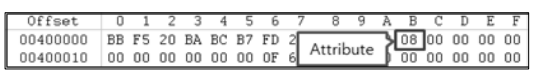
VBR의 55~62 오프셋과 83~90 오프셋은 파일 시스템의 타입을 정의하는 항목으로 〔표 2〕와 같이 각 파일 시스템에 따라 정해진 값을 갖는다. 따라서 이 영역에 정해진 값을 통하여 FAT16과 FAT32를 구분할 수 있으며 정해진 값을 제외한 다른 값이 오는 경우는 VBR로써 판단할 수 없다. 마지막으로 510~511 오프셋은 VBR의 Signature 항목으로 0xAA55의 정해진 값을 갖는다. 해당 위치에 존재하는 값이 Signature가 아닌 경우에 VBR에서 제외할 수 있는 근거로써 활용할 수 있다.

위와 같은 방법을 통하여 검색된 VBR은 해당 VBR이 사용하는 파일 시스템의 종류에 따른 검증 절차를 통하여 정상여부를 판단할 수 있다. 검색한 VBR이 FAT 파일 시스템으로 구성되어 있는 경우, 검증 시스템은 FAT 파일 시스템의 특징인 Root Directory와 FSInfo, VBR 사본 등을 통하여 파티션을 검증할 수 있다. 모든 FAT 파일 시스템은 기본적으로 Root Directory 영역을 가지고 있으며 모든 파일 및 서브 Directory는 Root Directory를 기준으로 관리된다. 〔표 3〕은 Root Directory의 위치를 계산하기 위해 필요한 VBR의 항목들을 나타낸 것으로 Root Directory의 위치는 VBR의 Reserved Sector Count와 Number of FATs, Root Directory Cluster 항목을 통해 계산할 수 있으며 계산 위치에 실제로 Root Directory가 존재하는지 판단하여 정상적인 VBR인지 여부를 검증할 수 있다.

〔표 3〕 Root Directory 계산을 위한 VBR 항목

오프셋	항 목	설 명
14-15	Reserved Sector Count	예약된 영역의 섹터 수
16	Number of FATs	볼륨에 있는 FAT 영역의 수
44-47	Root Directory Cluster	루트 디렉토리의 클러스터 값

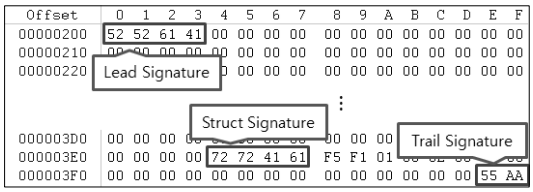
〔그림 2〕는 계산을 통하여 실제 Root Directory 영역을 찾아 나타낸 것으로 간단한 구조를 통해 Directory 및 파일을 관리한다. 〔그림 2〕의 Attribute 영역은 볼륨, Directory, 파일의 속성을 관리하는 항목으로 정상적인 값으로 16진수 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x0F 값이 유효한 값으로 사용된다. 따라서 유효하지 않은 값이 존재하는 경우에는 Root Directory 영역으로 볼 수 없으며 해당 VBR도 정상적인 VBR로 볼 수 없다.



〔그림 2〕 FAT 파일 시스템의 Root Directory

Root Directory는 생성된 하위 Directory 및 파일의 개수만큼 32바이트의 구조체가 연속적으로 저장된다. 따라서 각 노드의 Attribute 영역에 정상적인 값이 존재하는지 여부도 확인하여 정상 VBR 여부를 판단할 수 있다.

FAT32는 VBR의 FSInfo(File System Information) 항목을 통하여 FSInfo가 어디에 저장되어 있는지를 가리킨다. FSInfo는 FAT32를 효율적으로 사용하기 위하여 간단한 정보를 Reserved Sector 영역에 저장한다. 〔그림 3〕은 FAT32의 FSInfo를 나타내고 있다. FSInfo는 3개의 Signature(Lead, Struct, Trail)를 가지고 있으며 각각의 값들은 항상 16진수 0x41615252, 0x61417272, 0xAA550000 값을 갖는다. 따라서 FSInfo의 Signature를 통해서도 VBR의 정상 여부를 검증할 수 있으며 정상적이지 않은 값을 갖고 있는 경우에는 정상적인 파일 시스템이 아닌 것으로 판단할 수 있다.



〔그림 3〕 FAT32 파일 시스템의 FSInfo

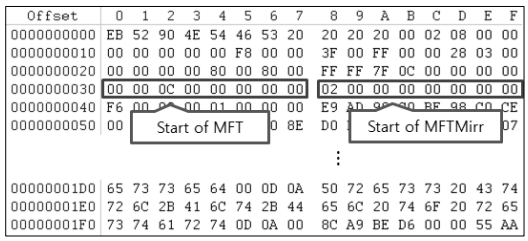
FAT32는 VBR의 사본을 생성하여 Boot Record Backup Sector 항목에 설정된 수만큼의 섹터 뒤에 복사본을 저장한다. 즉, 설정된 수만큼의 섹터 간격으로 같은 VBR이 존재한다. 따라서 Boot Record Backup Sector에 설정된 크기의 간격으로 같은 VBR이 나오는 경우에는 같은 파일 시스템을 사용하기 때문에 이 위치에서 발견된 VBR은 중복으로 표시하여 분석자의 편의성을 제공한다.

4.2.2 NTFS 파티션 검색 및 검증 기법

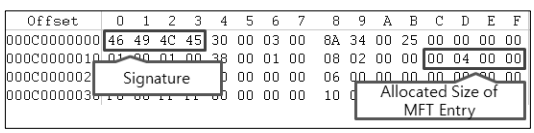
NTFS는 FAT 파일 시스템과 마찬가지로 VBR을 통하여 파티션을 구성한다. NTFS의 VBR은 FAT 파일 시스템의 VBR과 같이 0번 오프셋에서 같은 Jump Code를 사용하며 510~511 오프셋에서 같은 Signature를 활용한다. 그러나 OEM Name에서는 "NTFS " 라는 문자열을 사용하는 차이점을 갖는다. 따라서 이러한 차이를 통하여 VBR이 사용하는 파일 시스템을 구분할 수 있다.

위와 같은 방법을 통하여 검색된 VBR이 NTFS로 구성되어 있는 경우에는 NTFS의 MFT의 시작위치와 MFTMirr 위치에서 해당 섹터의 내용을 비교함으로써 파티션을 검증할 수 있다.

NTFS는 MFT(Master File Table)를 통하여 볼륨에 존재하는 모든 파일 및 Directory를 관리한다. MFT는 NTFS의 필수적인 메타 데이터 파일로써 VBR의 Start of MFT 항목을 통하여 위치를 확인할 수 있다. 그리고 MFT는 백업을 위한 사본을 MFTMirr 영역에 저장하며 MFTMirr는 VBR의 Start of MFTMirr를 통하여 그 위치를 확인할 수 있다. 〔그림 4〕는 VBR의 항목 중, Start of MFT와 Start of MFTMirr를 나타내고 있다. MFT는 〔그림 5〕와 같이 Signature와 MFT Entry의 크기 정보를 가지고 있으며 일반적으로 Signature는 "FILE" 이라는 문자열, MFT Entry 크기는 1024 바이트의 크기를 갖는다. 따라서 NTFS를 사용하는 시스템은 VBR의 정상 여부를 판단하기 위하여 Start of MFT 항목을 MFT로 이동한 후, Signature 항목과 Allocated Size of MFT Entry 항목에 올바른 값이 있는지 여부를 확인함으로써 가능하다. 그리고 MFTMirr 영역은 MFT의 내용과 같은 내용이 복사되어 있기 때문에 두 영역이 일치하는지 여부를 파악함으로써 VBR의 정상 여부를 검증할 수 있다.



〔그림 4〕 NTFS VBR의 MFT, MFTMirr 항목



〔그림 5〕 MFT의 Entry Header

4.2.3 EXT 3/4 파티션 검색 및 검증 기법

FAT 파일 시스템과 NTFS의 VBR과는 달리 EXT 파일 시스템은 EXT가 내부적으로 포함하고 있는 Super Block을 통하여 파티션의 검색 및 검증이 가능하다. 〔그림 6〕은 EXT4 파일 시스템의 레이아웃을 나타내는 것으로 각 Block Group의 첫 번째 Block에 Super Block이 위치하고 있으며 Super Block은 1 Block 크기의 공간을 차지한다. Block의 크기는 Super Block의 Block Size 항목을 통하여 확인 가능하며 Block Size는 0x00, 0x01, 0x02 중, 한 가지 값으로 Size의 크기를 정의한다. 〔표 4〕와 같이 Block Size의 값, 0x00은 1KB를 의미하고 0x01은 2KB를 의미하며 0x02는 4KB를 의미한다. 이와 같은 값들을 제외한 나머지 값은 존재할 수 없다. 따라서 0x00, 0x01 그리고 0x02 이외의 다른 값이 존재하는 경우에는 Super Block의 검색대상에서 제외할 수 있다. 또한 Super Block은 VBR과 마찬가지로 2 바이트의 시그니처를 가지고 있다. VBR의 시그니처 0xAA55와는 달리 Super Block은 56~57 오프셋에서 0xEF53 값을 갖는다. 이와 같이 〔표 4〕와 같은 Super Block의 구조를 이용하여 Super Block을 검색하기 위한 근거로 활용할 수 있다.

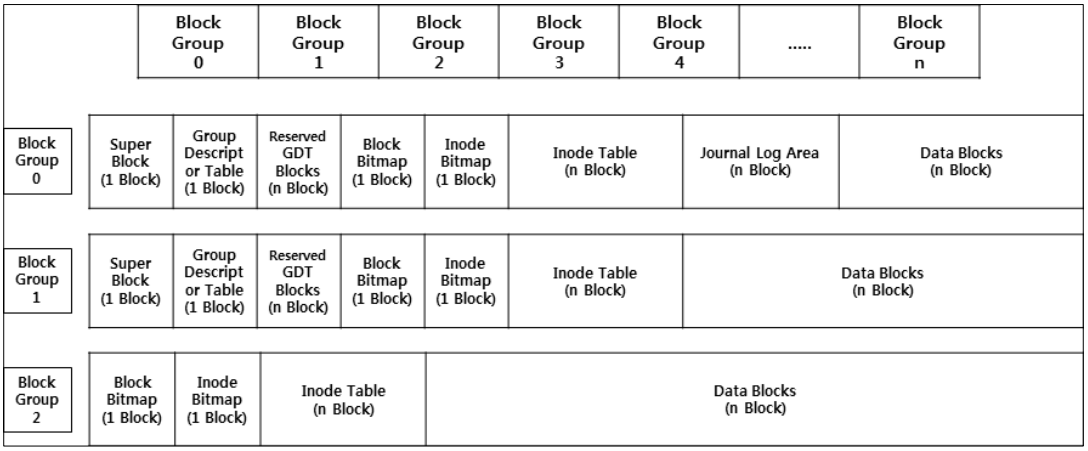
위와 같은 방법을 통하여 검색된 파티션의 검색결과가 Super Block인 경우에는 EXT 파일시스템의 특징인 Super Block의 위치 값, 저널로그 블록의 위치 값 등을 통하여 파티션을 검증할 수 있다.

EXT 파일 시스템은 Block Group 0을 포함하여 홀수 Block Group에 계속적으로 Super Block을 포함시킨다. 0번 Group을 포함한 모든 홀수 그룹의 첫 번째 블록에는 Super Block이 존재하며 각 Super Block에 포함된 Magic Signature의 확인을 통하여 파일시스템의 정상 여부를 검증할 수 있다. EXT 파일 시스템의 Super Block은 블록 그룹 0 위치에 최초 포함되어 있으며 이후, 홀수 그룹에서 Super Block이 반복적으로 나타난다. 따라서 홀수의 블록 그룹에 존재하는 Super Block에서 반복적으로 Magic Signature를 확인할 수 있으며 Signature가 다른 경우에는 잘못된 Super Block으로 판단할 수 있다. 또한 Super Block은 반복적으로 존재하기 때문에 같은 Super Block이 여러 개 존재할 수 있다. 하지만 포렌식 전문 도구는 Super Block의 특징을 고려하지 않고 Super Block을 검색하기 때문에 Super Block의 개수만큼의 중복이 발생할 수 있다. 따라서 Super Block을 포함하는 각 블록 그룹의 Super Block을 비교하여 같은 Super Block이 중복되지 않도록 해야 한다.

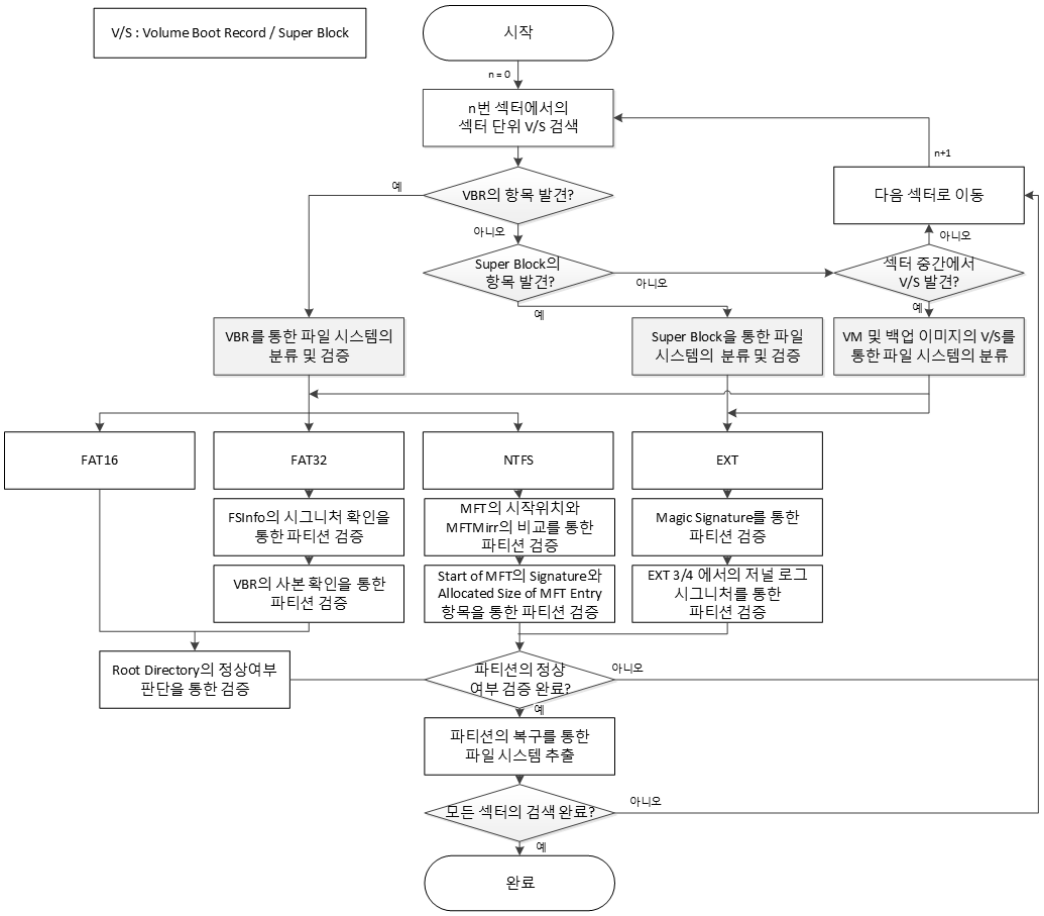
〔표 4〕 Super Block의 구조〔3〕

오프셋	항 목	설 명	값
24-27	Block Size	1KB	0x00
		2KB	0x01
		4KB	0x02
56-57	Signature	시그니처 항목	0xEF53

EXT3 파일 시스템과 EXT4 파일 시스템은 Super Block을 통해 지정된 블록 그룹에 저널 로그 블록을 포함하고 있다. 저널 로그는 0~3 오프셋에서 시그니처로 0xC03B3998 값을 가지고 있으며 이 시그니처를 통하여 정상적인 파일 시스템 여부를 판단할 수 있다.



〔그림 6〕 EXT4 파일 시스템 레이아웃



〔그림 7〕 효과적인 파티션 복구를 위한 제안 알고리즘

4.2.4 VM 및 백업 이미지 파티션 검색 및 검증 기법

VM 및 백업 이미지는 사용자가 지정한 특정 파티션 내부에 파일 형태로 저장되며 파일 내부적으로 하나의 가상 시스템을 이룬다. VM 및 백업 이미지는 내부의 가상 시스템을 통하여 하나의 파일 시스템을 가지며 파일 시스템의 종류에 따라 VBR 또는 Super Block을 포함한다. 따라서 앞서 설명한 검색 및 검증 기법을 통하여 VBR 및 Super Block의 검색 및 검증이 가능하다. 그러나 VM과 백업 이미지는 특정 파티션에서 파일 형태로 존재하고 있고 해당 파일을 컨트롤 하는 응용 프로그램의 특성에 따라 VBR 또는 Super Block이 섹터의 중간 영역에서 시작하는 경우가 존재한다. 즉, 일반적인 섹터 단위 검색으로는 파일 형태의 VBR과 Super Block을 검색할 수 없다. 따라서 VM 및 백업 이미지를 포함한 모든 파티션을 검색하고 검증 및 복구를 하기 위해서는 바이트 단위의 Signature 검색이 이루어져야 한다. 또한 VM 및 백업 이미지는 파일 내부에서 가상 시스템으로 존재하기 때문에 파일 내부의 VBR이 시작하는 위치를 논리적으로 0번째 섹터로 간주하여 복구해야 한다. 하지만 바이트 단위의 Signature 검색을 통한 파티션의 검증 및 복구는 저장장치의 용량 크기에 따라 소요되는 시간에 큰 차이가 발생할 수 있다. 따라서 효율적인 VM 및 백업 이미지의 파티션 복구를 위해서는 분석자의 필요에 따라 VM 및 백업 이미지의 파티션 복구 여부를 선택적으로 조절할 수 있도록 하는 알고리즘이 필요하다.

4.3 제안하는 알고리즘

〔그림 7〕은 본 논문에서 제안하는 파티션 복구에

대한 알고리즘을 도식화한 것이다. 본 논문의 파티션 복구 알고리즘은 앞서 제안한 각 파일 시스템에 대한 파티션의 검색 및 검증 기법을 포함함으로써 복구 대상 파티션 검색의 미탐 및 오탐을 줄일 수 있도록 하였다. 또한 기존의 파티션 복구 도구에는 없는 중복되는 파티션을 검사할 수 있도록 하여 효과적인 파티션의 복구 및 분석이 가능하도록 하였으며 VM 및 백업 이미지에 대한 파티션 검색 및 복구도 가능할 수 있도록 하였다.

디지털 포렌식 분석가는 제안된 알고리즘을 통해 삭제된 파티션의 검색 및 복구, 분석을 효과적으로 수행할 수 있으며 파티션 복구 후 남아 있는 비할당 영역에 대해서는 파일 카빙을 수행 할 수 있다.

V. 제안 알고리즘의 실험 및 결과

〔표 5〕는 본 논문에서 제안한 알고리즘과 각 파티션 복구 도구의 성능을 비교하기 위하여 구성한 테스트 이미지에 대한 정보다. 테스트 이미지는 총 8개의 파티션을 포함하고 있으며, 1번 파티션을 제외한 나머지 파티션은 모두 삭제되었다. 일반적인 파일 시스템과 같이 각 파티션은 백업 VBR을 포함하고 있기 때문에 파티션을 위해 존재하는 실제적인 VBR은 총 16개이다. 〔표 5〕의 3번 파티션은 2번 파티션 내부에 VM 파일로 포함되어 있었던 VM 파티션이며, 5번, 7번, 8번 파티션은 파티션 복구에 필수적인 정보인 파일 시스템의 메타영역이 손상되었기 때문에 복구가 불가능한 파티션이다.

본 논문에서는 〔표 5〕와 같이 구성된 테스트 이미지를 활용하여 제안한 알고리즘을 통해 만든 복구 도구와 각 파티션 복구 도구를 테스트하여 얻은 결과를 비교 분석하였다.

〔그림 8〕은 본 논문에서 제안하는 알고리즘을 통해 만든 파티션 복구 도구의 실험 결과로써, 테스트 이미지 내에 존재하는 모든 VBR을 검색하였으며 VBR 사본에 대한 중복처리와 VM 파티션 검색, 파티션 검증을 통하여 정상적인 파티션인지 여부를 표시하고 있다. 또한 NTFS의 경우에는 $MFT의 생성시간을 통하여 실제 파일 시스템의 생성시점을 추가적으로 파악할 수 있다. 제안한 알고리즘은 기존의 TestDisk와 Partition Recovery 도구와는 달리 실제로 마운트된 파티션뿐만 아니라, 파티션 이미지 파일에서도 동작할 수 있도록 하였기 때문에 디지털 포렌식의 증거 무결성 관점에서 더욱 의미가 있다.

VBR Search

No	Status	FileSystem	Start Sector	End Sector	Capacity	duplication	Verify	Format Time	Note
01	E	NTFS	56	20948759	9.99GB	-	Normal	2013/05/31	-
02	E	NTFS	-	-	-	D	-	-	duplicate
03	L	NTFS	20948816	52407599	15.00GB	-	Normal	2013/05/31	-
04	L	NTFS	35285119	-	9.98GB	-	-	-	VM
05	L	NTFS	-	-	-	D	-	-	VM dupl.
06	L	NTFS	-	-	-	D	-	-	duplicate
07	L	FAT32	52407656	56605919	2.00GB	-	Normal	-	-
08	L	FAT32	-	-	-	D	-	-	duplicate
09	L	FAT32	56605976	60804239	2.00GB	-	Damaged	-	Error
10	L	FAT32	-	-	-	D	Damaged	-	duplicate
11	L	FAT16	60804296	64988279	2.00GB	-	Normal	-	-
12	L	FAT16	-	-	-	D	-	-	-
13	L	FAT16	64988336	69172319	2.00GB	-	Damaged	-	Error
14	L	FAT16	-	-	-	D	Damaged	-	Error
15	L	NTFS	69172376	73370639	2.00GB	-	Damaged	-	Error
16	L	NTFS	-	-	-	D	-	-	Error

〔그림 8〕 제안하는 알고리즘을 통한 VBR 검색 결과

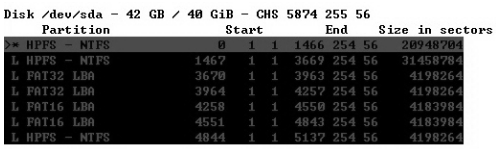
〔그림 9〕와 〔그림 10〕은 EaseUS의 Partition Recovery와 TestDisk의 VBR 검색 결과를 보여준다. 기본적으로 두 도구는 중복된 VBR을 검색결과에 반영하지 않는다. 그러나 두 도구는 손상된 파티션을 검증하지 못하며, 손상 여부를 확인하기 위해서는 모든 파티션을 직접 파티션을 복구 해보아야 하는 단점이 있다. 또한 두 도구는 VM파티션은 검색하지 못하였으며 파티션에 대한 어떠한 시간정보도 얻을 수 없다는 단점이 있다.

Status	Type	Flags	Start Sector	End Sector	Capacity	Free Space
Existing	NTFS	Primary	56	20948759	9,99 GB	6,36 GB
Lost	NTFS	Logical	20948816	52407599	15,00 GB	3,88 GB
Lost	FAT32	Primary	52407656	56605919	2,00 GB	1,99 GB
Lost	FAT32	Primary	56605976	60804239	2,00 GB	1,98 GB
Lost	FAT16	Primary	60804296	64988279	2,00 GB	1,97 GB
Lost	FAT16	Primary	64988336	69172319	2,00 GB	1,99 GB
Lost	NTFS	Logical	69172376	73370639	2,00 GB	1,99 GB

〔그림 9〕 EaseUS의 VBR 검색 결과

〔표 5〕 테스트 이미지 정보

No	상태	파일시스템	시작 섹터	볼륨 사이즈	손상여부	비고
1	정상	NTFS	56	9.99 GB	정상	-
2	삭제	NTFS	20948816	15.00 GB	정상	-
3	삭제	NTFS	35285119	9.98 GB	정상	VM 파티션
4	삭제	FAT32	52407656	2.00 GB	정상	-
5	삭제	FAT32	56605976	2.00 GB	손상	-
6	삭제	FAT16	60804296	2.00 GB	정상	-
7	삭제	FAT16	64988336	2.00 GB	손상	-
8	삭제	NTFS	69172376	2.00 GB	손상	-



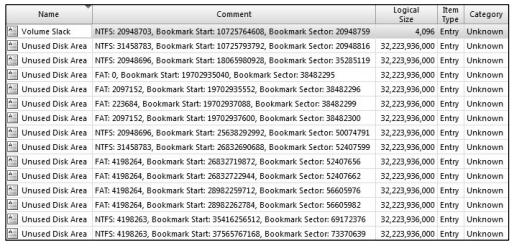
〔그림 10〕 TestDisk의 VBR 검색 결과

〔그림 11〕은 EnCase를 통한 VBR 검색 결과를 보여준다. EnCase는 〔그림 8〕과 〔그림 9〕의 두 도구에 비하여 더 많은 VBR 결과를 보여주고 있다. 하지만, 각 파티션들의 VBR 백업에 대한 중복 처리와 파티션 검증 등의 절차가 포함되지 않고 단순히 이미지에 존재하는 모든 VBR을 검색한 것이다. 따라서 정상적인 파티션 검사를 위해 검색된 모든 파티션을 분석가가 직접 복구를 수행하여 확인해야 된다는 불편함이 있으며, 본 논문에서 제안한 도구와는 달리 각 파티션들의 생성 시간 정보도 확인할 수 없다.

위 그림과 같이 본 논문을 통하여 제안한 알고리즘은 기존의 파티션 복구 도구가 지원하지 않는 무결성 보장과 검색 결과의 중복 체크 및 파일 시스템 검증, VM 내에 존재하는 파티션의 검색, 시간 정보 제공 등 디지털 포렌식 분석에서 효과적으로 활용 될 수 있는 기능들을 포함하기 때문에 단순히 복구만을 위한 기존의 도구들과는 많은 차이가 있다.

〔표 6〕은 본 논문에서 제안한 알고리즘으로 개발한 도구와 기존의 도구들의 실험 결과를 표를 통해 정리한 것이다. 하나의 디스크 내부에는 많은 수의 VBR이 존재하기 때문에 효과적인 복구 및 분석을 위해서는 단순히 존재하는 모든 VBR의 검색을 통한 파티션 복구가 아닌 디지털 포렌식 관점에서 본 논문에서 제안한 VBR 사본의 중복 확인, 파일 시스템 검증, VM 파티션 검색 등의 기능들이 필요하다.

실제로 〔표 6〕의 실험 결과는 VBR의 중복 확인과 파일 시스템 검증, VM 파티션 검색 등을 통하여 테스트 이미지 내에 존재하는 8개 파티션에 대한 16개의 VBR을 모두 검색하였으며 그 중, VBR 사본에 대한 중복 확인과 정상 파티션 검증 등을 통하여 복구 가능한 불필요한 복구를 진행하지 않기 위한 결과를 보여 주고 있다. 반면 Partition Recovery와 TestDisk는 유효한 파티션을 검증해주지 않기 때문에 불필요한 복구를 진행해야 VM 파티션의 경우는 검색을 제공하지 않기 때문에 검색 목록에 포함되지 못하였다. 또한, 두 도구는 활성 시스템에서 복구를 진행해야 하기 때문에 무결성을 보장할 수 없다.



Name	Comment	Logical Size	Item Type	Category
Volume Slack	NTFS: 20948703, Bookmark Start: 10725764608, Bookmark Sector: 20948759	4,096	Entry	Unknown
Unused Disk Area	NTFS: 31458783, Bookmark Start: 10725793792, Bookmark Sector: 20948816	32,223,936,000	Entry	Unknown
Unused Disk Area	NTFS: 20948696, Bookmark Start: 18065980928, Bookmark Sector: 35285119	32,223,936,000	Entry	Unknown
Unused Disk Area	FAT: 0, Bookmark Start: 19702935040, Bookmark Sector: 38482295	32,223,936,000	Entry	Unknown
Unused Disk Area	FAT: 2097152, Bookmark Start: 19702935552, Bookmark Sector: 38482296	32,223,936,000	Entry	Unknown
Unused Disk Area	FAT: 223684, Bookmark Start: 19702937088, Bookmark Sector: 38482299	32,223,936,000	Entry	Unknown
Unused Disk Area	FAT: 2097152, Bookmark Start: 19702937600, Bookmark Sector: 38482300	32,223,936,000	Entry	Unknown
Unused Disk Area	NTFS: 20948696, Bookmark Start: 25638292992, Bookmark Sector: 50074791	32,223,936,000	Entry	Unknown
Unused Disk Area	NTFS: 31458783, Bookmark Start: 26832690688, Bookmark Sector: 52407599	32,223,936,000	Entry	Unknown
Unused Disk Area	FAT: 4198264, Bookmark Start: 26832719872, Bookmark Sector: 52407656	32,223,936,000	Entry	Unknown
Unused Disk Area	FAT: 4198264, Bookmark Start: 26832722944, Bookmark Sector: 52407662	32,223,936,000	Entry	Unknown
Unused Disk Area	FAT: 4198264, Bookmark Start: 28982259712, Bookmark Sector: 56605976	32,223,936,000	Entry	Unknown
Unused Disk Area	FAT: 4198264, Bookmark Start: 28982262784, Bookmark Sector: 56605982	32,223,936,000	Entry	Unknown
Unused Disk Area	NTFS: 4198263, Bookmark Start: 35416256512, Bookmark Sector: 69172376	32,223,936,000	Entry	Unknown
Unused Disk Area	NTFS: 4198263, Bookmark Start: 37565767168, Bookmark Sector: 73370639	32,223,936,000	Entry	Unknown

〔그림 11〕 EnCase의 VBR 검색 결과

EnCase는 VM 파티션의 대한 VBR은 검색하였으나 각 파티션들의 사본 VBR에 대한 중복을 체크하지 않으며 VM 파티션에 포함된 Pagefile.sys 파일 내부에 존재하는 불필요한 VBR도 검색하였다. 또한 손상된 파티션도 모두 포함함으로 인하여 EnCase가 검색한 15개의 VBR 중에서 실제로 복구가 가능한 파티션은 3개에 불과하다는 것을 알 수 있다.

VI. 결론 및 향후 계획

최근 안티 포렌식 행위를 위하여 사용자의 시스템에 존재하는 파티션을 인위적으로 숨기는 사례가 증가하고 있다. 또한 악성코드를 통한 MBR의 해킹으로 인하여 MBR의 파티션 테이블 정보가 삭제되어 시스템 내에 존재하는 파티션을 인식하지 못하는 사례도 증가하고 있다. 이와 같이 사용자가 파티션을 인위적으로 숨기거나 악성코드에 의해 파티션 테이블이 삭제되어 시스템이 파티션을 인식하지 못하게 되는 경우에 해당 파티션 내에 존재하는 파일 시스템으로부터 메타정보 및 모든 데이터를 획득할 수 없게 된다. 파일 카빙 기법을 통하여 일부 데이터의 추출이 가능하지만 파일이 조각난 경우에는 해당 파일을 완전히 추출하는 것이 어렵고 메타 정보를 얻지 못하기 때문에 의미 있는 분석도 어렵다. 따라서 파티션을 복구하여 파티션 내의 파일시스템을 해석하는 것은 포렌식 관점에서 매우 중요한 사항이다.

기존의 포렌식 도구들은 저장장치 내에 존재하는 VBR 및 Super Block의 시그니처를 검색하여 검색된 위치로 파일시스템을 해석할 수 있는 기능을 제공한다. 그러나 그 기능이 극히 제한적이며 VBR 이나 Super Block이 많이 존재하는 경우에 분석을 위해서는 일일이 복구해야 하는 어려움이 존재 한다. 또한 VM과 백업 파일 상에 존재하는 VBR 및 Super Block의 경우에는 시작위치가 섹터의 시작 위치가 아닌 경우에 검색대상에서 제외된다. 따라서 특정 파일

〔표 6〕 테스트 이미지를 통한 각 도구의 실험결과

도구명	No	상태	타입	시작섹터	볼륨크기	중복여부	손상여부	포맷시간	비고	특이사항
제안 알고리즘	1	정상	NTFS	56	9.99GB	-	-	2013/05/13	-	VBR 사본에 대한 중복 처리
	2	정상	NTFS	-	-	중복	-	-	-	
	3	삭제	NTFS	20948816	15.00GB	-	-	2013/05/13		
	4	삭제	NTFS	35285119	9.98GB	-	-	-	VM파티션	
	5	삭제	NTFS	-	-	중복	-	-	VM파티션	
	6	삭제	NTFS	-	-	중복	-	-	-	
	7	삭제	FAT32	52407656	2.00GB	-	-	-	-	
	8	삭제	FAT32	-	-	중복	-	-	-	
	9	삭제	FAT32	56605976	2.00GB	-	손상	-	-	
	10	삭제	FAT32	-	-	중복	손상	-	-	
	11	삭제	FAT16	60804296	2.00GB	-	-	-	-	
	12	삭제	FAT16	-	-	중복	-	-	-	
	13	삭제	FAT16	64988336	2.00GB	-	손상	-	-	
	14	삭제	FAT16	-	-	중복	손상	-	-	
	15	삭제	NTFS	69172376	2.00GB	-	손상	-	-	
	16	삭제	NTFS	-	-	중복	손상	-	-	
EaseUS Partition Recovery	1	활성	NTFS	56	9.99GB	-	-	-	-	VM 파티션 미탐, 활성상태에서 사용가능
	2	삭제	NTFS	20948816	15.00GB	-	-	-	-	
	3	삭제	FAT32	52407656	2.00GB	-	-	-	-	
	4	삭제	FAT32	56605976	2.00GB	-	-	-	손상여부오탐	
	5	삭제	FAT16	60804296	2.00GB	-	-	-	-	
	6	삭제	FAT16	64988336	2.00GB	-	-	-	손상여부오탐	
	7	삭제	NTFS	69172376	2.00GB	-	-	-	손상여부오탐	
Test Disk	1	활성	NTFS	56	9.99GB	-	-	-	-	VM 파티션 미탐
	2	삭제	NTFS	20948816	15.00GB	-	-	-	-	
	3	삭제	FAT32	52407656	2.00GB	-	-	-	-	
	4	삭제	FAT32	56605976	2.00GB	-	-	-	손상여부오탐	
	5	삭제	FAT16	60804296	2.00GB	-	-	-	-	
	6	삭제	FAT16	64988336	2.00GB	-	-	-	손상여부오탐	
	7	삭제	NTFS	69172376	2.00GB	-	-	-	손상여부오탐	
EnCase	1	활성	NTFS	-	-	-	-	-	중복여부미탐	VM파티션, 중복여부미탐, 파티션 손상여부오탐, FAT16,FAT32 구분 불가능
	2	삭제	NTFS	20948816	15.00GB	-	-	-	-	
	3	삭제	NTFS	35285119	9.98GB	-	-	-	VM여부미탐	
	4	삭제	FAT	-	-	-	-	-	VM Pagefile (오탐)	
	5	삭제	FAT	-	-	-	-	-		
	6	삭제	FAT	-	-	-	-	-		
	7	삭제	FAT	-	-	-	-	-		
	8	삭제	NTFS	-	-	-	-	-	VM, 중복여부미탐	
	9	삭제	NTFS	-	-	-	-	-	중복여부미탐	
	10	삭제	FAT	60804296	2.00GB	-	-	-	손상여부오탐	
	11	삭제	FAT	-	-	-	-	-	중복여부미탐	
	12	삭제	FAT	64988336	2.00GB	-	-	-	손상여부오탐	
	13	삭제	FAT	-	-	-	-	-	중복여부미탐	
	14	삭제	NTFS	69172376	2.00GB	-	-	-	손상여부오탐	
	15	삭제	NTFS	-		-	-	-	중복여부미탐	

내부에 존재하는 파일시스템을 분석하기 위해서는 직접 파일을 읽어서 분석해야 하는 단점도 있다.

본 논문은 VBR과 Super Block, 그리고 각 파일 시스템들의 특징을 분석하여 복구 대상 파티션의 중복을 줄여 분석의 번거로움을 줄이고 파일 시스템의 검증을 통하여 손상된 파티션을 복구하기 위한 수고도 덜면서 할당 및 비 할당 영역 내에 존재하는 복구 가능한 모든 파티션을 복구하여 복구된 파티션 상에 존재하는 파일들로 하여금 의미 있는 분석이 가능하도록 하는 알고리즘을 제안하였다.

또한, 파일 시스템의 메타 정보에 해당하는 영역이 모두 손상 및 손실 되었을 경우에도 MFT 엔트리 등의 정보를 통해서 파티션을 복구 하는 등의 추가적인 연구를 진행할 계획이다. 디지털 포렌식 분야에서 복

구는 사용자가 삭제한 데이터를 복구, 분석한다는 관점에서 필수적이지만 파일 카빙 기법을 사용한 복구는 매우 많은 시간이 걸리며 파일이 조각난 경우에는 복구율이 떨어지는 등의 많은 한계점이 존재하였다. 하지만 본 논문에서 제안한 파티션 단위의 복구를 수행하면 해당 파티션 내부에 존재하는 모든 파일들을 정상적으로 한 번에 복구할 수 있기 때문에 디지털 포렌식 관점에서 많은 도움을 줄 수 있을 것이다.

참고문헌

〔1〕 Master boot record, http://en.wikipedia.org/wiki/Master_boot_record.

〔2〕 Volume boot record, http://en.wikipedia.org/wiki/Volume_Boot_Record.

〔3〕 Kevin D. Fairbanks, "An analysis of EXT4 for digital forensics," Digital Investigation, vol. 9, pp. 118-130, Aug. 2012.

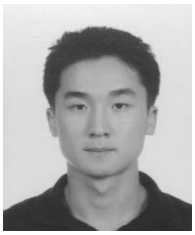
〔4〕 EaseUS, http://www.easeus.com/partition-recovery.

〔5〕 TestDisk, http://www.cgsecurity.org/wiki/TestDisk.

〔6〕 Guidance Software homepage, http://www.guidancesoftware.com.

〔7〕 Brian Carrier, "File System Forensic Analysis," Addison-Wesley Professional, 22 March 2005.

〈저 자 소 개〉



남 궁 재 웅 (Jaeung Namgung) 학생회원
2010년 2월: 세종대학교 전자정보공학대학 컴퓨터공학 공학사
2011년 3월~2013년 2월: 고려대학교 정보보호대학원 석사수료
〈관심분야〉 디지털 포렌식, 데이터베이스 포렌식

홍 일 영 (Ilyoung Hong) 학생회원
2000년 2월: 경북대학교 사회학과 졸업
2012년 2월: 고려대학교 정보보호대학원 석사
2012년 3월~현재: 고려대학교 정보보호대학원 박사과정
2006년 2월~현재: 대검찰청 디지털수사담당관실
(관심분야) 디지털 포렌식, 모바일 포렌식, 클라우드 컴퓨팅, 사이버 법률

박 정 흠 (Jungheum Park) 학생회원
2007년 2월: 한양대학교 정보통신대학 컴퓨터전공 공학사
2007년 3월~2009년 2월: 고려대학교 정보경영공학전문대학원 공학석사
2009년 3월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 디지털 포렌식, 안티-안티 포렌식

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