하드디스크의 물리적 섹터 접근 방법을 이용한 MFT기반 증거 파일 탐색 기법

김요식* · 최명렬* · 장태주* · 류재철**

요 약

대용량 하드디스크의 등장으로 많은 자료를 컴퓨터의 하드디스크에 저장할 수 있게 되었다. 하드디스크의 용량이 커지면서 저장되어 있는 파일 및 디렉토리가 증가하여 디지털 포렌식 분야에서도 탐색해야 하는 정보가 증가하게 되었다. 대용량 하드디스크에서 증거로 활용될 수 있는 파일 정보를 탐색하기 위해서는 윈도우 시스템에서 제공해주는 파일관리 함수군을 주로 이용한다. 하지만, 이 방법은 파일과 디렉토리의 수가 많을 경우 처리속도가 느리며, 파일 정보를 읽을 경우 파일의 접근시간이 변경된다. 또한, 운영체제 또는 응용 프로그램에 의해 이미 사용중인 파일의 경우 접근이 불가능하다는 단점이 있다. 본 논문에서는 대용량 하드디스크에 저장되어 있는 파일 및 디렉토리를 빠르게 탐색하기 위한 방법으로 하드디스크의 물리적 섹터에 접근하여 NTFS의 MFT 정보를 획득하고, 획득된 MFT 정보를 기반으로 증거파일을 탐색하는 방법을 제시하고 구현하였다.

MFT-based Forensic Evidence File Search Method Using Direct Access to Physical Sector of Hard Disk Drive

Yosik Kim* · Myeongryeol Choi* · Taejoo Chang* · Jaecheol Ryou**

ABSTRACT

According to the capacity of hard disk drive is increasing day by day, the amount of data that forensic investigators should analyze is also increasing. This trend need tremendous time and effort in determining which files are important as evidence on computers. Using the file system APIs provided by Windows system is the easy way to identify those files. This method, however, requires a large amount of time as the number of files increase and changes the access time of files. Moreover, some files cannot be accessed due to the use of operating system. To resolve these problems, forensic analysis should be conducted by using the Master File Table (MFT). In this paper, We implement the file access program which interprets the MFT information in NTFS file system.

* ETRI 부설연구소
** 충남대학교 정보통신공학부

We also extensibly compare the program with the previous method. Experimental results show that the presented program reduces the file access time then others. As a result, The file access method using MFT information is forensically sound and also alleviates the investigation time.

Key words : NTFS, Master File Table, Digital Forensic, Hard Disk

1. 서 론

디지털 포렌식에서 대용량의 하드디스크에 저장되어 있는 증거 데이터를 찾아내는 것은 중요한 요소중 하나이다. 오늘날 주요 저장매체로 활용되고 있는 하드디스크는 계속적으로 저장 용량이 증가하고 있는 추세이며, 이에 따라 포렌식에서 대용량의 하드디스크내에 저장되어 있는 파일을 종류별로 분류하고 내용을 탐색하여 증거를 찾아내는데 많은 시간이 필요하게 되었다.

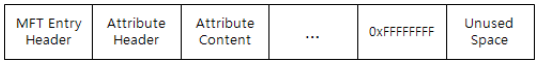
대용량 하드디스크에 저장되어 있는 파일 정보를 분석하기 위해서는 먼저 운영 중인 운영체제의 종류를 식별해야한다. 다양한 종류의 운영체제가 존재하지만, 현재에는 마이크로소프트사의 윈도우 운영체제가 주로 사용되고 있다. 동시에 디지털 포렌식 수사에서도 가장 많은 분석 대상이 되고 있다.

마이크로소프트사가 2006년 6월 윈도우 98 운영체제의 기술지원 서비스를 중단함으로써[1] 윈도우 2000 이후 버전의 운영체제가 주로 사용되고 있으며, 더불어 NTFS가 주요 파일시스템으로 자리잡게 되었다[2-4]. 본 논문에서는 물리적 섹터 접근 방식을 이용하여 NTFS의 MFT(Master File Table)정보를 물리적으로 획득하여 대용량의 하드디스크에 저장되어 있는 정보를 탐색하기 위한 효율적인 방법을 제안하고 구현하여 실험한다.

본 논문의 구성은 다음과 같다. 먼저, 제 2장에서는 NTFS의 MFT의 특징에 대해 간략히 살펴보고, 제 3장에서는 MFT를 기반으로 한 효율적인 증거 수집을 위한 제안된 기법을 기술하고, 제 4장에서는 제안된 기법을 구현하여 시험·분석하며, 제 5장에서는 결론을 제시한다.

2. 관련 연구

MFT는 NFTS에서 가장 중요한 정보로 볼륨에 존재하는 모든 파일과 디렉토리에 대한 정보를 담고 있는 테이블이다. 파일과 디렉토리들의 이름, 생성일자, 크기, 내용저장 위치, 소유자 등의 정보가 MFT에 저장된다. 윈도우 시스템에서 볼륨을 NTFS로 포맷할 경우 윈도우 시스템은 MFT의 초기 크기로 설정해 놓고 파일 및 디렉토리가 많아짐에 따라 MFT 엔트리의 수를 증가 시킨다. MFT 엔트리는 (그림 1)과 같이 MFT 엔트리 헤더와 속성(Attribute)로 나누어져 있다.



(그림 1) MFT 엔트리 구성

MFT 엔트리 헤더에는 MFT 엔트리의 정보와 상태를 저장하고 있으며, 크기는 42바이트로 고정되어 있다. 속성에는 파일 및 디렉토리가 <표 1>과 같이 속성타입에 따라 저장되어 있다.

〈표 1〉 주요 MFT 속성 종류

타입	이 름	내 용
16	$STANDARD_INFORMATION	최근 접근시간, 생성시간, 소유자, 보안 아이디 등
32	$ATTRIBUTE_LIST	속성 리스트
48	$FILE_NAME	파일 이름
96	$VOLUME_NAME	볼륨 이름 및 정보
112	VOLUME_INFORMATION	파일시스템 정보
128	$DATA	파일 내용

일반적으로 <표 1>에서 언급한 파일에 대한 이름, 내용, 생성/변경/삭제 시간 등의 정보를 추출하기 위해서는, 윈도우 시스템이 제공하는 파일관리 함수군[5]을 이용하여 볼륨에 존재하는 디렉토리 및 파일에 순차적으로 접근하여 정보를 얻어오는 방법을 사용한다. 하지만, 이 방법은 디렉토리와 파일의 수가 많아지게 되면 디렉토리와 파일을 탐색하는 함수의 호출 횟수가 많아져 오버헤드가 발생하게 된다.

뿐만 아니라, 운영중인 윈도우 운영체제의 시스템 프로세스 또는 일부 응용 프로그램이 사용 중인 파일에 대해서는 파일 정보를 추출할 수 없다는 단점을 가지고 있다.

본 논문에서는 파일관리 함수를 사용하여 파일 정보를 추출하는 대신, 하드디스크에 존재하는 MFT 정보를 물리적 섹터 단위로 접근하여 획득하고, 파싱하여 증거 파일을 탐색하고 파일 내용을 추출하도록 하였다.

3. MFT 기반 증거 파일 탐색 방법

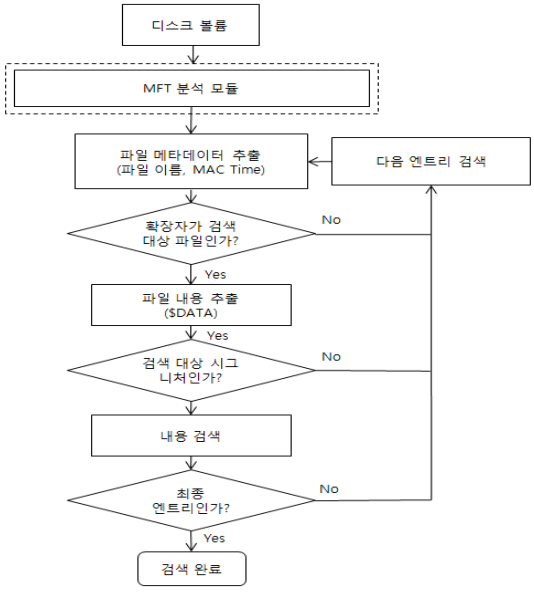
MFT는 볼륨에 존재하는 모든 파일과 디렉토리의 정보를 담고 있으므로 이 테이블을 분석하면 볼륨에 있는 모든 파일과 디렉토리에 대한 정보를 추출할 수 있다. 즉, MFT 엔트리 정보를 이용하면 운영체제가 실행중인 라이브 시스템 뿐만 아니라 DD 포맷[6]과 같이 포렌식 목적으로 획득된 디스크 이미지에 대해서도 인덱스 생성 및 증거 탐색을 수행할 수 있다.

3.1 증거 파일 탐색 절차

하드디스크의 섹터 정보를 읽기 위해서는 시스템에 설치되어 있는 물리적 하드 디스크에 접근해야 한다.

윈도우 운영체제는 CreateFile()의 인자로 \\.\\PHYSICALDRIVE를 사용하여 물리적 드라이브에 접근하여 섹터 정보를 읽을 수 있도록 지원하고 있다. 물리적 드라이브에 접근하여 시스템에 설치된 물리 드라이브의 개수를 판단 한 후, 각 물리 드라이브에 생성된 논리 드라이브를 식별하고, 식별된 논리 드라이브에서 MFT 정보를 획득한다.

획득된 MFT 정보를 기반으로 $STANDARD_INFORMATION, $FILE_NAME, $DATA 속성 정보를 이용하여 파일의 접근시간, 생성시간, 소유자, 파일 이름, 파일 내용을 추출한다. 모든 MFT 엔트리에 대해 탐색이 완료될 때까지 반복하여 탐색한다. 절차는 (그림 2)와 같다.



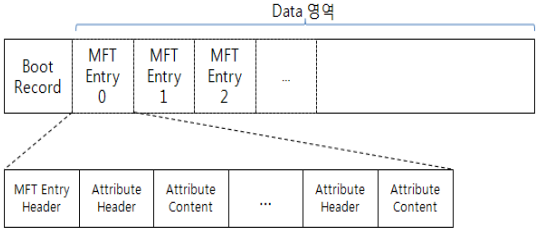
(그림 2) 증거 파일 탐색 절차

3.2 MFT 정보 획득 방법

본 논문에서 제시하는 물리적 MFT 정보 획득이라함은 MFT가 저장되어 있는 하드디스크의 섹터에 직접 접근하여 MFT 파일 정보를 획득하는 방법을 말한다. NTFS를 사용하는 윈도우 운영체제가 볼륨을 인식할때는 가장 먼저 부트 레코드(Boot

Record)[7]에 접근하는데, 이는 MFT가 어느 섹터에 위치하는지를 알아내기 위함이 주목적이라 할 수 있다. 윈도우 운영체제에서 볼륨을 NTFS로 포맷할 경우 윈도우는 MFT의 초기 크기를 작게 설정하고, 파일이 많아짐에 따라 윈도우는 MFT의 크기를 늘려간다. 따라서, 디렉토리나 파일의 수가 증가 할수록 MFT의 크기도 점점 더 커지게 된다.

부트 레코드내에는 MFT 엔트리 크기가 설정되어 있는 필드가 있는데 현재까지 사용되는 NTFS의 MFT 엔트리 크기는 1024바이트이다. 따라서 약 20,000개의 파일이 있다고 가정하면, 약 20여 메가 바이트정도 크기의 MFT 정보가 존재하게 된다. (그림 3)은 NTFS를 사용하는 일반적인 하드디스크의 구조를 나타낸다.



(그림 3) NFTS를 사용하는 하드디스크의 구조

NTFS를 사용하는 하드디스크로부터 MFT를 획득하기 위한 절차는 다음과 같다.

1) NTFS의 부트 레코드가 저장되어 있는 512바이트크기의 하드디스크의 0번째 섹터 정보를 획득한다.
2) MFT 엔트리 크기와 부트 섹터 오프셋 48~55번째에 저장되어 있는 MFT 테이블의 시작 클러스터 주소를 획득한다.
3) 메타 데이터 파일인 $MFT도 파일의 한 종류 이므로, $MFT 파일의 크기를 고려하여 Resident 또는 Non-resident 속성을 참조하여 $MFT 파일의 전체 내용을 획득한다.
4) MFT 엔트리 크기 단위(일반적으로 1024바이트)로 $MFT 파일을 분석한다.
5) 42바이트의 MFT 엔트리 헤더정보를 추출하고 남은 공간에서 파일이나 디렉토리의 속성정보를 추출하여 분석한다.
6) 4)~5)번의 과정을 반복하여 증거 파일 수집을 수행한다.

윈도우 운영체제의 경우에도 부트 섹터 48~55번째에 저장되어 있는 MFT 테이블의 시작 클러스터 주소를 이용하여 볼륨을 분석한다. 또한, 윈도우 운영체제 내부적으로는 메타 데이터 파일인 $MFT를 일반 파일과 동일하게 관리하고 있다.

4. 시험 및 분석

본 논문에서는 물리적 섹터 접근 방식을 이용하여 NTFS의 MFT 정보 획득 기능, MFT 엔트리 분석 기능, MFT 엔트리 정보 파싱기능, 파일 내용 정보 추출 기능을 구현하였으며, 성능 비교 실험을 위해 파일관리 함수군을 이용하여 볼륨내의 디렉토리 및 파일을 탐색하는 모듈도 구현하였다.

구현된 결과물에 대한 시험은 Intel Core2 Duo 2.4GHz의 프로세서와 4GB 메모리가 장착되어 있고 Windows XP Professional 운영체제 서비스팩 3이 설치된 시스템 환경에서 수행하였다. 시험에 사용된 디스크는 <표 2>와 같이 용량, 남은 사용공간, 사용기간이 서로 다른 총 6개 디스크를 이용하였다.

〈표 2〉 디스크 정보

구 분	크 기	RPM	파일 시스템	사용공간
HDD1	160G	7500	NTFS	115GB
HDD2	120GB	7500	NTFS	31GB
HDD3	500GB	7500	NTFS	51GB
HDD4	80GB	7500	NTFS	73GB
HDD5	250GB	7500	NTFS	105GB
HDD6	250GB	7500	NTFS	106GB

〈표 3〉 파일 처리 시험 결과

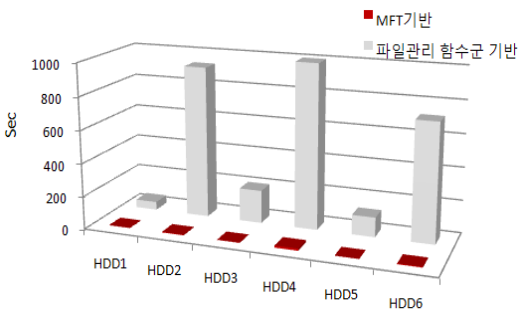
구 분	파일관리 함수군 기반 모듈						MFT 기반 모듈					
	HDD1	HDD2	HDD3	HDD4	HDD5	HDD6	HDD1	HDD2	HDD3	HDD4	HDD5	HDD6
	(80GB)	(250GB)	(250GB)	(250GB)	(250GB)	500(GB)	(80GB)	(250GB)	(250GB)	(250GB)	(250GB)	500(GB)
파일수/ MFT엔트리수	20,980	122,894	43,355	287,349	16,261	147,919	24,128	132,241	47,496	292,467	17,880	161,439
초기지연시간 (Sec)	-	-	-	-	-	-	0.922	2.547	1.0930	10.969	0.469	3.437
$MFT크기 (Byte)	-	-	-	-	-	-	24,739,840	135,446,528	48,676,864	299,532,288	18,350,080	165,347,328
전체 파일처리 시간(Sec)	49.484	926.734	201.406	2149.170	119.284	719.419	1.063	3.375	1.391	12.765	0.594	4.609
미처리수	0	110	0	2	3	4	0	0	0	0	0	0
삭제파일수	-	-	-	-	-	-	2,592	7	2,621	2,438	5	10,732
디렉토리수	2,688	11,974	5,218	31,919	1,077	11,305	3,278	12,205	5,577	32,206	1,129	11,885

가장 많은 시간이 소요될 것으로 예상되는 부분인 파일의 내용을 읽는 부분은 제외하였으며 파일에 대한 Open, Close 행위만을 수행하였다. 시험결과 파일관리 함수군 기반의 분석 모듈은 디렉토리 수와 파일 수에 비례하여 처리 속도가 급격하게 늦어지는 현상을 확인하였으며, 파일을 처리하는데 걸리는 시간은 파일 한 개당 0.000001042초~0.000008383초 가량 소요되었다. 반면 MFT를 기반으로 동작하는 모듈은 $MFT 파일의 크기에 따라 초반에 MFT 전체 정보를 메모리에 적재하기 위해 필요한 시간이 지연되었으며 파일을 처리하는데 걸리는 시간은 파일 한 개당 0.000000429~0.000000689초 가량 소요되었다. <표 2>의 6개의 하드디스크를 대상으로 시험한 결과는 <표 3>과 같다.

파일관리 함수군을 기반으로 한 분석 모듈에 사용된 함수는 FindFirstFile, FindNextFile, FindClose, CreateFile, CloseHandle이며, MFT 기반 모듈의 경우 획득된 $MFT 정보를 1024바이트씩 읽고 엔트리 헤더를 참조하여 $STANDARD_INFORMATION, $FILE_NAME, $DATA속성 정보를 추출하도록 하였다. 파일관리 함수군 기반 모듈은 볼륨에 존재하는 전체 파일수를 기반으로 파일 처리 소요시간을 계산하였으며, MFT 기반 모듈은 MFT 엔트리 수를 기반으로 계산하였다.

하드디스크를 포맷하지 않고 오랜 시간동안 사용하여 파일의 복사, 삭제, 이동이 많을 경우 $MFT 파일의 크기가 더 커지므로, MFT 기반 모듈의 경우 $MFT 파일 정보를 메모리에 적재하는데 걸리는 초기지연시간이 최대 10초 가량 소요되었다. 그리고, MFT 기반 모듈을 이용하여 탐색한 MFT 엔트리수가 파일관리 함수군 기반 모듈을 이용하여 탐색한 파일수 보다 많았음에도 불구하고, MFT 기반 모듈이 파일관리 함수군 기반 모듈보다 파일에 대한 Open, Close를 수행한 전체 파일 처리 시간이 98% 이상 단축되어 (그림 4)와 같이 높은 효율을 보임을 확인하였다.

또한, 파일관리 함수기반의 분석 모듈은 운영체제가 사용하는 파일에 대해 접근 및 내용을 획득할 수 없는 문제점으로 인해 미처리 건수가 발생하였으나, MFT 기반 모듈을 이용하여 탐색을 할 경우 운영체제가 사용하는 파일에 대해 파일을 열지 못하는 경우가 발생하지 않아 미처리 파일 수가 발생되지 않았다.



(그림 4) 파일처리 소요시간

5. 결 론

본 논문에서는 디지털 포렌식에 필요한 증거 탐색의 효용성 확인 위해 하드디스크의 물리적 섹터 접근 기법을 이용하여 MFT를 획득하고, 이를 기반으로 한 증거 파일을 수집하는 방법을 제안하고 시험하였다. 시험 결과 MFT 기반의 증거 탐색 및 수집 방법이 파일관리 함수군을 사용하는 방법보다 탐색속도와 파일 처리 건수에서 월등히 우수함을 알 수 있었고, 운영체제가 사용중인 파일의 경우도 검사 대상에 포함시킬 수 있으므로 효과적 방법임을 확인하였다.

향후 연구에서는 연구 결과물을 기반으로 하여 컴퓨터 포렌식 분야에서 필요한 검색엔진과 접목시켜 인덱스 생성, 키워드 검색의 성능을 개선하는 연구를 수행할 것이다.

참 고 문 헌

[1] 국가사이버안전센터, “윈도우98 보안 가이드 라인”, http://ncsc.go.kr, 2006.

[2] NTFS.com, “NTFS-New Technology File System designed for Windows Vista, XP, 2003, 2000”, http://www.ntfs.com.

[3] Microsoft Corporation, “How NTFS Works”, http://technet.microsoft.com/en-us/library/cc781134.aspx.

[4] Mark Russinovich, “Inside Win2k NTFS, Part1”, Microsoft Developer Network, 2008.

[5] Microsoft, “File Management Functions”, http://msdn2.microsoft.com/en-us/library/aa364232.aspx.

[6] Wikipedia, “DD(Unix)”, http://en.wikipedia.org/wiki/Dd_(Unix).

[7] FileRecovery.biz, “NTFS Boot Sector”, http://bootmaster.filerecovery.biz/appnote3.html.

김 요 식

1997년~1999년 (주)지란지교소프트 연구원
2000년~2004년 (주)케이사인 선임연구원
2005년 공주대학교 바이오정보학과 석사
2004년~현재 ETRI 부설연구소 선임연구원
2007년~현재 충남대학교 컴퓨터공학과 박사과정

최 명 렬

1991년 인하대학교 전자계산 공학과(공학사)
1993년 인하대학교 전자계산 공학과(공학석사)
1996년~1998년 국방정보체계 연구소 선임연구원
1999년~2000년 국방과학연구소 선임연구원
2003년 인하대학교 컴퓨터·정보공학과 박사과정 수료
2000년~현재 ETRI 부설연구소 선임연구원

장 태 주

1982년 울산대학교 전기공학과(학사)
1990년 한국과학기술원 전기 및 전자공학과(석사)
1998년 한국과학기술원 전기 및 전자공학과 (공학박사)
1982년~2000년 국방과학연구소 선임연구원
2000년~현재 ETRI 부설연구소 책임연구원

류 재 철

1985년 한양대학교 산업공학과 (학사)
1988년 Iowa State University 전산학(석사)
1990년 Northwestern University(전산학 박사)
1991년～현재 충남대학교 정보통신공학부 교수
1997년～현재 국가정보원 정보보호시스템 인증위원회 위원
2003년～현재 인터넷 침해대응기술연구센터장