情報保護學會論文誌
第20卷 第3號, 2010. 6

파일 조작에 따른 파일 시간 변화 분석*

방 제 완,† 유 병 영, 이 상 진‡
고려대학교 정보경영공학전문대학원

Timestamp Analysis of Windows File Systems by File Manipulation Operations*

Jewan Bang,† Byeongyeong Yoo, Sangjin Lee‡
Graduate School of Information Management and Security, Korea University

요 약

디지털 포렌식 수사에 있어 시간 정보는 중요한 요소이다. 윈도우즈의 NTFS(New Technology File System) 환경에서 획득할 수 있는 파일의 시간 정보는 생성, 수정, 접근, MFT entry 수정 시간이며 이는 파일의 복사나 이동, 이름 변경 등의 사용자의 행위에 따라 특징적으로 변경된다. 이러한 시간 변경 특징은 사용자의 데이터 이동 및 데이터 변경 등의 행위 분석에 활용할 수 있다. 본 논문에서는 윈도우즈 운영체제 별로 사용자의 행위에 따른 파일이나 폴더의 시간 변화를 분석하여 이를 바탕으로 시스템 분석시 사용자의 행위를 유추할 수 있도록 한다.

ABSTRACT

In digital forensics, the creation time, last modified time, and last accessed time of a file or folder are important factors that can indicate events that have affected a computer system. The form of the time information varies with the file system, depending on the user's actions such as copy, transfer, or network transport of files. Specific changes of the time information may be of considerable help in analyzing the user's actions in the computer system. This paper analyzes changes in the time information of files and folders for different operations of the NTFS and attempts to reconstruct the user's actions.

Keywords: Digital Forensic, File, Folder, Time, Windows, NTFS, Filesystem

I. 서 론

시간 정보는 사건 발생 순서를 재구성하는 과정에서 유용하게 쓰일 수 있으며[1], 수집한 데이터의 증거 인정 여부를 판가름하는 매우 중요한 부분이다. 또한 수사 시 필수적인 정보와 위장된 데이터를 구분할 수 있도록 하여 올바른 수사의 방향을 제시하는 기준으로도 사용된다[2]. EnCase[3]와 같은 디지털 포렌식 수사에 사용되는 도구들도 시간 정보를 기반으로 정렬하거나 검색 기능을 제공하며 수사관은 그 시간 정보를 기준으로 수사 범위를 설정할 수 있다. 윈도우 운영체제에서 사용되는 NTFS(New Technology File System)의 경우 MAC(Modified, Accessed, Created) 시간 이외에 MFT Entry modified 시간도 포함하고 있다[4]. 이는 여러 복합적이고 다양한 사용자의 행위에 따라 여러 형태의 시간 값의 변화가 일어날 수 있다는 뜻이며 반대로 이러한 시간 변화의 유형을 파악하여 사용자의 행위를 유추할 수도 있다[3]. 이에 본 논문에서는 여러 버전의 윈도우즈를 대상으로 NTFS에서 사용자의 행위에 따른 시간 정보의 변화를 정리하고 이를 통해 파일이나 폴더의 시간 정보를 통해 역으로 사용자의 행위를 유추할 수 있도

접수일(2009년 11월 18일), 수정일(1차: 2010년 1월 19일, 2차: 2010년 3월 18일), 게재확정일(2010년 3월 19일)
* 본 연구는 한국연구재단을 통해 교육과학기술부의 바이오연구개발사업으로부터 지원받아 수행되었습니다. (20090084147)
† 주저자, jwbang@korea.ac.kr
‡ 교신저자, sangjin@korea.ac.kr

록 한다.

본 논문의 구성은 다음과 같다. 2절에서 시간 정보 분석에 대한 기존 연구를 설명하고, 3절에서는 윈도우즈 파일 시스템과 시간 정보의 표현 방식을 설명한다. 그리고 4절에서는 파일과 폴더를 대상으로 윈도우즈 운영체제 별 행위에 따른 시간 변화를 분석하고, 5절에서 사례 분석을 통해 본 연구의 활용 방안을 제시한다.

II. 최근연구

K.P. Chow et al[5]은 NTFS 상의 MAC 시간을 기준으로 법칙을 세워 여러 가지의 적용 사례를 보여주었다. 그러나 NTFS에 있는 MFT entry 수정 시각을 고려하지 않았다. NTFS에서는 MFT entry 수정 시각은 파일의 이름을 변경하거나 내용을 수정할 경우 변경된다. MACE(Modified, Accessed, Created, Entry modified) 시간은 파일의 시간 정보를 수정할 수 있는 도구를 통해 변조가 가능하며 변경된 시간 정보 자체로는 그 변조 유무를 확인하기 어렵다[6]. 그렇기 때문에 전문 사용자의 시스템을 대상으로 시간 정보를 분석할 시에 변조 가능성을 파악할 수 없으므로 MACE 시간 정보 변화를 기반으로 사용자의 행위를 유추하는 K.P. Chow et al이 제안한 법칙은 신뢰성이 떨어진다. 하지만 NTFS의 파일이나 폴더는 $STANDARD_INFORMATION 속성 이외에 $FILE_NAME 속성에도 시간 정보를 담고 있다[6]. $FILE_NAME 속성의 시간 정보를 변조하는 방법이나 도구는 알려져 있지 않으며 변조를 위해서는 NTFS의 전 구조를 파악하여 직접 장치의 섹터를 접근하여 해당 데이터를 수정해야 한다. 이에 Jewan Bang et al[7]에서는 윈도우즈 XP를 대상으로 MACE 시간과 함께 행위에 따라 $FILE_NAME 속성의 시간 정보 변경에 따른 추가적인 분석을 수행하였다. 하지만 NTFS는 파일 데이터의 크기가 작은 경우 클러스터가 아닌 MFT entry 내에 파일 데이터를 포함하는 경우가 있으며 일부 다른 시간 변화의 양상을 보인다. 그리고 동일한 행위에도 운영체제에 따라서 다른 시간 변화를 가진다. 본 논문에서는 여러 버전의 윈도우즈를 대상으로 파일이나 폴더의 시간 변화의 차이를 분석하였고 데이터가 MFT entry 내에 존재하는 경우와 클러스터에 존재하는 경우에 발생하는 시간 변화의 차이를 분석하였다.

[표 1] 운영체제 별 NTFS 버전

NTFS 버전	운영체제
v1.0	Windows NT 3.1
v1.1 (3.5)	Windows NT 3.5
v1.2 (4.0)	Windows 3.51
v3 (5.0)	Windows 2000
v3.1 (5.1)	Windows XP
v3.1 (5.2)	Windows Server 2003
v3.1 (6.0)	Windows Vista
v3.1	Windows Server 2008
v3.1	Windows 7

III. 윈도우즈의 파일 시스템

3.1 NTFS - New Technology File System

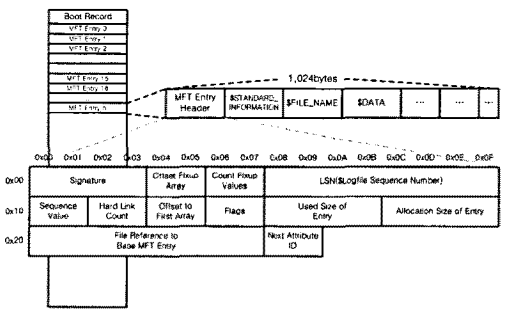
FAT 파일 시스템을 대체하기 위해 Microsoft사에서 개발한 NTFS는 1993년 윈도우즈 NT 3.1과 함께 공개되었다. 서버 환경을 고려하여 개발되었기 때문에 각 사용자의 디스크 사용량을 지정할 수 있는 디스크 쿼터 기능, 사용자 접근 권한 제한, 파일 암호화 기능, 압축 기능 등을 지원한다. 다른 파일 시스템과 가장 큰 차이는 정형화된 볼륨의 레이아웃이 없는 대신 파일 형태의 테이블로 정보를 관리한다. 이러한 특징으로 할당 테이블로 파일의 스트림을 관리하는 FAT 파일 시스템에 비해 파일 복구가 용이한 편이다. 2009년 10월 공개된 윈도우즈 7도 NTFS 3.1 버전을 사용하며 각 운영체제별 NTFS 버전은 [표 1]과 같다.

3.2 NTFS의 시간 정보

파일 시스템은 파일 데이터의 내용뿐만 아니라 파일의 이름, 시간, 크기 및 저장매체 상의 위치와 같은 데이터도 함께 생성하여 해당 파일 관리를 위해 사용한다. 이러한 데이터를 메타 데이터(Meta data)라 하며 이를 생성하고 관리하는 방법은 파일 시스템마다 다르다. 윈도우즈 환경에서 주로 사용되는 NTFS는 Master File Table(MFT)라는 구조를 이용하여 메타 데이터를 관리한다. NTFS에서는 파일이나 폴더의 메타 데이터 관리를 위해 1개 이상의 MFT entry를 사용하며 [그림 1]과 같이 MFT entry의 기본 정보를 관리하는 MFT entry 헤더와 파일 데이터를 관리하는 $DATA, 파일 이름을 관리하는 $FILE_NAME과 같이 '$'의 접두어를 가지는 각각의 속성으로 구성되어 있다[8].

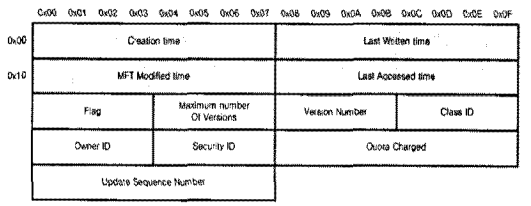
〔표 2〕 MFT entry의 속성의 종류(일부)[8]

속성번호	속성 이름	설명
16	$STANDARD_INFORMATION	접근, 생성 시간, 소유자 등의 기본 정보
32	$ATTRIBUTE_LIST	속성의 목록
48	$FILE_NAME	파일 이름
64	$OBJECT_ID	파일이나 폴더를 위한 고유한 값
80	$SECURITY_DESCRIPTOR	접근 제어와 보안 속성
128	$DATA	파일의 내용
144	$INDEX_ROOT	인덱스 트리의 루트 노드
160	$INDEX_ALLOCATION	인덱스 트리와 연결된 노드 정보
192	$REPARSE_POINT	소프트 링크를 위한 위치 정보

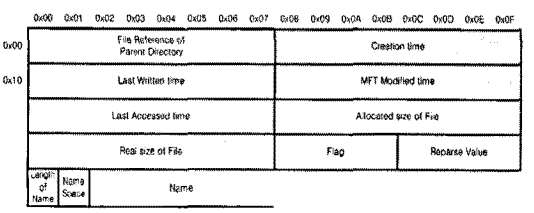


〔그림 1〕 MFT(Master File Table)의 구조[8]

파일이나 폴더의 시간은 MFT entry의 속성 중 $STANDARD_INFORMATION 속성과 $FILE_NAME 속성에 포함되어 있으며 각각 생성된 시간, 마지막으로 내용이 수정된 시간, MFT entry가 마지막으로 수정된 시간, 마지막으로 접근한 시간 정보와 같이 4가지 항목으로 존재한다.



〔그림 2〕 $STANDARD_INFORMATION 속성의 시간 정보 구조

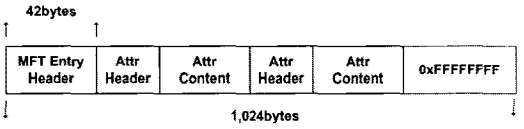


〔그림 3〕 $FILE_NAME 속성의 시간 정보 구조

실제 윈도우즈 탐색기 화면에서 확인할 수 있는 만든 날짜, 마지막 수정한 날짜, 마지막 접근한 날짜 정보는 $STANDARD_INFORMATION 속성의 정보를 기반으로 하며 마지막으로 MFT entry 수정 시간 정보는 EnCase[3]와 같은 디지털 포렌식 분석 도구를 통해 확인할 수 있다. $FILE_NAME 속성에도 $STANDARD_INFORMATION 속성의 시간 정보와 동일한 개수의 필드로 시간 정보를 담고 있다. MACE 시간 정보 모두 8바이트의 FILETIME[8]이라는 데이터 구조를 이용하여 정보를 표현하며 1601년 1월 1일부터 100나노 초 단위로 누적된 시간 값을 담고 있다.

3.3 NTFS의 데이터 스트림 구성

MFT entry는 MFT entry 헤더와 그 외의 속성들로 구성되어 있다. 속성의 종류에는 여러 가지가 있으며 MFT entry가 담고 있는 파일의 특성에 따라서 담기는 속성의 종류가 달라진다. 하나의 MFT entry의 크기는 1,024바이트이며 MFT entry 헤더의 크기는 42바이트이다. 남은 982바이트의 공간에 속성이 구성되게 되며 속성의 종류가 많아 982바이트 내에 표현할 수 없는 경우 MFT entry를 더 할당하여 표현하거나 별도의 클러스터를 할당하여 그 정보를 담도록 한다. 파일의 실제 데이터를 담고 있는 $DATA 속성의 경우 다른 속성들이 차지하고 남은 공간에 들어갈 수 있는 크기이면 MFT entry 내에 데이터를 기록한다. 즉, 〔그림 4〕와 같이 1,024바이트의 MFT entry에서 42바이트의 MFT entry 헤더와 파일의 속성 정보를 담고 있는 $STRANDARD_INFORMATION 속성, 파일 이름을 담고 있는 $FILE_NAME 속성을 제외하고 남은 약 750바이트의 크기에 데이터를 포함할 수 있는 경우 파일의 데이터는 클



〔그림 4〕 MFT entry 내의 속성의 구성

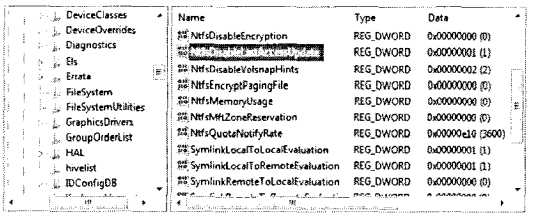
러스터에 할당되지 않고 MFT entry 내에 위치하게 된다〔8〕. 이러한 경우 클러스터에 데이터가 존재할 때와 다른 시간 변화를 보이며 이에 따른 분석이 필요하다.

3.4 레지스트리의 NtfsDisableLastAccessUpdate value

윈도우즈 Vista 이후 버전부터 시스템의 성능 향상을 위해 파일이나 폴더의 마지막 접근 시간을 갱신하지 않는 기능을 지원한다. 레지스트리의 HKLM\SYSTEM\CurrentControlSet\Control\FileSystem 위치에 REG_DWORD 타입의 NtfsDisableLastAccessUpdate value를 통해 기능 활성은 '1', 비활성은 '0'으로 설정할 수 있다. 윈도우즈 Vista, 2008, 7 버전은 기본 값이 활성화 상태로 되어 있으며 파일(폴더)의 열기, 복사, 접근과 같은 행위에 마지막 접근 시간이 변화하지 않는다. 이러한 특성 때문에 디지털 증거 분석시 운영체제의 해당 레지스트리 value의 설정 유무를 미리 확인하는 것이 필요하다.

IV. 윈도우즈 운영체제 별 행위에 따른 시간 변화 분석

각 윈도우즈 버전 별 행위에 따른 시간 정보 변화를



〔그림 5〕 레지스트리의 NtfsDisableLastAccessUpdate value

확인하기 위해 그림 〔표 3〕과 같이 6종의 환경을 구축하였으며 MFT entry의 $FILE_NAME 속성의 시간 값의 변화를 확인하기 위해서 EnCase〔3〕, The Sleuth Kit〔9〕와 The MFT Entry Parser〔10〕를 활용하였다.

4.1 NTFS의 파일 시간 변화

4.1.1 파일 생성

파일 생성시 $STANDARD_INFORMATION 속성과 $FILE_NAME 속성의 생성, 수정, 접근, MFT entry 수정 시각 모두 동일하게 파일 생성 시각으로 설정된다. 즉, 각 속성의 시간이 모두 동일할 경우 현재 위치하고 있는 폴더에서 생성되었다는 것을 유추할 수 있다. 아래 〔표 4〕의 M, A, C, E는 각각 마지막 수정 시각, 마지막 접근 시각, 생성 시각, MFT entry 수정 시각을 의미한다.

4.1.2 파일 복사

파일 복사는 "잘라내기 & 붙여넣기"와 "Ctrl+C & Ctrl+V"를 통한 복사 행위를 대상으로 실험하였

〔표 3〕 시간 변화 분석 환경

OS	◦ Microsoft Windows 2000 Service Pack 4 ◦ Microsoft Windows Server 2003 Service Pack 2 ◦ Microsoft Windows XP Professional Service Pack 3 ◦ Microsoft Windows Vista Service Pack 2 ◦ Microsoft Windows Server 2008 Service Pack 2 ◦ Microsoft Windows 7 Professional			
Tools	◦ Guidance Software EnCase 6.5.1.2 ◦ Brian Carrier's The Sleuth Kit 3.0.1 ◦ The MFT Entry Parser 1.5			
Volume	PhysicalDrive0		PhysicalDrive1	
	C:\	250GB	D:\	250GB

〔표 4〕 파일 생성시 시간 변화

속성 이름	모든 윈도우즈 버전
$STD_INFO	M == A == C == E
$FILE_NAME	M == A == C == E
수행 시각	M, A, C, E

〔표 5〕 파일 복사시 시간 정보 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상
$STD_INFO	M, E < C, A	M < C, A, E
$FILE_NAME	모두 복사 수행 시각으로 변경	모두 복사 수행 시각으로 변경
수행 시각	C, A	C, A, E

다. 동일한 볼륨 내에서 파일을 복사하였을 경우 Vista 미만 버전까지는 원본의 수정 시각과 MFT entry 수정 시각이 유지되며 나머지 시간 정보는 복사를 수행한 시각으로 변경된다. Vista를 포함한 이후 버전에서는 수정 시각만 원본의 시각으로 남으며 나머지는 복사를 수행한 시각으로 설정된다. 즉, 파일의 마지막 수정 시각과 MFT entry 수정 시각이 다른 시간 보다 이전일 경우 파일이 복사되었다고 유추할 수 있다. 다른 볼륨에서 파일을 복사하였을 경우 동일한 볼륨에서 복사를 수행하였을 경우와 시간 변화의 특징이 동일하다.

4.1.3 동일한 볼륨에서의 파일 이동

동일한 볼륨에서의 파일 이동의 경우 마우스를 통해 파일을 다른 폴더로 이동시키는 행위를 대상으로 실험하였다. 파일의 데이터가 MFT entry 내에 존재하는 경우와 클러스터에 할당된 경우가 다른 시간 변화를 보인다. Vista 이전 버전의 경우 MFT entry 내에 데이터가 존재하는 경우는 MFT entry 수정 시각만 이동을 수행한 시각으로 변경되며 나머지는 기존 시간이 유지된다. 하지만 클러스터에 데이터가 할당된 경우 마지막 접근 시각과 MFT entry 수정 시각 모두 복사를 수행한 시각으로 변경된다.

Vista 이상 버전의 경우 MFT entry 내에 데이터가 존재하는 경우는 MFT entry 수정 시각만 이동을 수행한 시각으로 변경되며 나머지는 기존

〔표 6〕 데이터가 MFT entry에 존재하는 경우의 시간 변화

속성 이름	모든 윈도우즈 버전
$STD_INFO	M, A, C < E
$FILE_NAME	기존 시각 유지
수행 시각	E[1]

〔표 7〕 데이터가 클러스터에 존재하는 경우의 시간 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상
$STD_INFO	M, C < A, E	M, C, E < A
$FILE_NAME	기존 시각 유지	기존 시각 유지
수행 시각	A, E	A

시간이 유지된다. 클러스터에 데이터가 할당된 경우 마지막 접근 시각이 이동을 수행한 시각으로 변경된다. 하지만 윈도우즈 7의 경우 데이터의 할당 위치에 상관없이 MFT entry 수정 시각만 이동을 수행한 시각으로 변한다.

4.1.4 다른 볼륨으로의 Drag & Drop을 이용한 파일 복사

볼륨끼리의 파일 이동을 위한 방법으로는 마우스를 이용해 "Drag & Drop"으로 이동하는 방식과 마우스 오른쪽 버튼 기능의 "잘라내기 & 붙여넣기"를 통해 이동하는 방식이 있다. "Drag & Drop" 방식을 이용하는 경우 Vista 미만 버전과 이상 버전의 시간 변화 차이가 보이며 클러스터 할당 유무와는 상관없이 시간 변화가 동일하다.

Vista 미만 버전의 경우 수정한 시각과 MFT entry 수정 시각이 유지되는 반면에 이상의 버전의 경우 수정한 시각만 이동시에 유지된다. Drag & Drop을 통한 파일 이동은 원본 파일을 유지시키기 때문에 유지되는 시간 정보를 이용하여 원본 파일을 유추할 수 있다.

〔표 8〕 Drag & Drop을 이용한 파일 복사시 시간 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상
$STD_INFO	M, E < C, A	M < C, A, E
$FILE_NAME	모두 복사 수행 시각으로 변경	모두 복사 수행 시각으로 변경
수행 시각	C, A	C, A, E

1) 윈도우즈 7은 데이터 할당 위치에 상관없이 MFT entry 수정 시각만 변경

〔표 9〕 잘라내기 & 붙여넣기를 이용한 파일 이동시 시간 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상
$STD_INFO	M, C, E < A	M, C < A, E
$FILE_NAME	모두 이동 수행 시각으로 변경	모두 이동 수행 시각으로 변경
수행 시각	A	A, E

4.1.5 다른 볼륨으로의 잘라내기 & 붙여넣기를 이용한 파일 이동

마찬가지로 Vista 버전을 기준으로 파일 이동시에 시간 정보 변화의 차이를 가진다. 다른 행위와의 차이점으로 원본 파일의 생성 시각이 파일 이동시에 유지된다.

"잘라내기 & 붙여넣기"를 통한 이동은 파일 이동 수행 후 원본 파일을 삭제하는 과정으로 이루어진다. 이동시에도 유지되는 생성 시각과 수정 시각을 이용하여 저장 매체의 미사용 공간 등에서 데이터 복원시 파일 데이터 비교를 통해 동일한 파일을 찾은 경우에 파일의 시간을 유추할 수 있다.

4.1.6 파일 속성 변경

파일의 '읽기 전용', '숨김' 속성에 변화를 주었을 때의 시간 정보 변화를 분석하였다. Vista 미만 버전의 경우 마지막 접근 시각과 MFT entry 수정 시각이

〔표 10〕 파일 속성 변경시 시간 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상
$STD_INFO	M, C < A, E	M, A, C < E
$FILE_NAME	기존 시각 유지	기존 시각 유지
수행 시각	A, E	E

〔표 11〕 NOTEPAD에서 내용 수정시 시간 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상
$STD_INFO	C < M, A, E	A, C < M, E
$FILE_NAME	기존 시각 유지	기존 시각 유지
수행 시각	M, A, E	M, E

속성을 변경한 시각으로 바뀌었으며 Vista 이상 버전의 경우 MFT entry 수정 시각만 변화 된다.

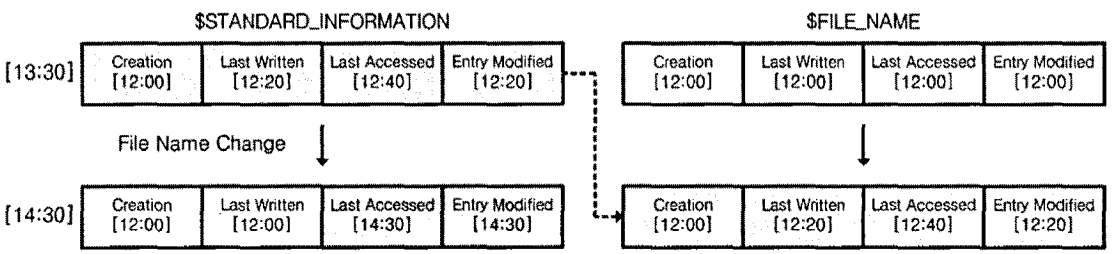
Vista 이상 버전의 경우 마지막 접근 시간을 업데이트하지 않는 옵션이 설정되어 있으므로 마지막 접근 시각이 변하지 않는 것을 확인하였다. 레지스트리의 NtfsDisableLastAccessUpdate value를 변경하면 마찬가지로 마지막 접근 시각도 변경된다.

4.1.7 파일 내용 수정

파일 내용 수정시의 시간 정보 변화 확인을 위해 윈도우즈의 NOTEPAD와 Microsoft Office WORD 2가지 응용 프로그램을 사용하여 문서 파일의 내용을 수정하고 저장하였다. 또한 파일 수정시 파일의 크기가 늘어나고 줄어드는 경우를 포함하여 파일의 크기가 늘어나 클러스터를 더 할당할 경우와 할당된 클러스터가 줄어드는 경우, 또는 MFT entry 내에 존재하는 데이터의 크기가 늘어나 클러스터에 할당되는 경우를 모두 조사하였다. 하지만 실험 결과 파일 할당 크기의 변화는 시간 정보 변경에 영향을 주지 않았다.

NOTEPAD의 경우 파일 내용 수정시 Vista를 기준으로 다른 시간 변화의 결과를 보이나 WORD의 경우 윈도우즈 버전에 상관없이 같은 결과를 보인다.

위의 실험 결과와 같이 응용 프로그램 마다 수정된 결과 저장시에 각기 다른 시간 변화를 보인다. 이러한 실험 결과는 일부 응용 프로그램은 데이터 수정시에



〔그림 6〕 이름 변경시 각 속성의 시간 정보 변화

〔표 12〕 WORD에서 내용 수정시 시간 변화

속성 이름	모든 윈도우즈 버전
$STD_INFO	C 〈 M, A, E
$FILE_NAME	C 〈 M, A, E
수행 시각	M, A, E

〔표 13〕 파일 이름 변경시의 시간 변화 (Vista 버전의 일부 경우 제외)

속성 이름	모든 윈도우즈 버전
$STD_INFO	M, A, C 〈 E
$FILE_NAME	변경 전 $STANDARD_INFORMATION 속성의 시각 정보
수행 시각	E

임시 파일을 생성하거나 파일을 재 생성하여 저장하기 때문이다. 응용 프로그램에 따라 다른 결과를 보일 수 있으므로 각 응용 프로그램별로 추가적인 분석이 필요하다.

4.1.8 파일 이름 변경

파일의 이름 변경시, 마지막 접근 시각과 MFT entry 수정 시각이 이름 변경 시간으로 변경된다. 마지막 수정 시각이 변경되지 않은 것은 실제로 데이터의 내용이 변경되지 않았기 때문이다. 이름 변경을 수행하였을 경우 파일 이름을 관리하는 $FILE_NAME 속성의 시간 정보도 이름 변경 시간으로 변경되어야 하지만 그렇게 변하지 않는다. 〔그림 6〕과 같이 이름 변경 이전의 $STANDARD_INFORMATION 속성의 MACE 시간이 $FILE_NAME 속성의 MACE 시간에 적용되는 것을 확인하였다.

파일 이름 변경시 Vista 미만 버전의 데이터가 클러스터 내에 존재하는 경우 마지막 접근 시각이 파일 변경을 시작한 시간으로 바뀌며 MFT entry 수정 시각은 이름 변경을 완료한 시각으로 바뀐다. 그 외의 경우에는 모두 마지막 접근 시각은 바뀌지 않으며 MFT entry 수정 시각만 이름을 변경한 시각으로 바뀐다.

〔표 14〕 Vista 미만에서 클러스터에 데이터 관리시 시간 변화

속성 이름	윈도우즈 Vista 미만
$STD_INFO	M, C 〈 A[2], E[3]
$FILE_NAME	변경 전 $STANDARD_INFORMATION 속성의 시각 정보
수행 시각	A, E

4.1.9 파일 덮어쓰기

파일 덮어쓰기의 경우 옮겨지는 파일(A)과 덮혀지는 파일(B)의 파일 크기가 동일하거나 크거나 작은 경우를 포함하여 실험하였다. 실험 결과 Vista 미만의 버전에서는 옮겨지는 파일(A)의 마지막 수정 시각과 MFT entry 수정 시각이 덮혀지는 파일(B)에 적용된다. Vista 이상 버전에서는 옮겨지는 파일(A)의 마지막 수정 시각만 덮혀지는 파일에 적용되는 것을 확인하였다.

〔표 15〕와 같이 Vista 미만의 버전에서는 마지막 접근 시각을 덮어쓴 시각으로 볼 수 있고 Vista 이상에서는 MFT entry 수정 시각을 덮어쓰기 시간이라고 할 수 있다.

4.2 NTFS의 폴더 시간 변화

4.2.1 폴더 생성

폴더도 파일과 마찬가지로 $STANDARD_INFORMATION 속성과 $FILE_NAME 속성 모두에 시

〔표 15〕 파일 덮어쓰기에 따른 시간 정보 변화

속성 이름	윈도우즈 Vista 미만[4]	윈도우즈 Vista 이상[5]
$STD_INFO	C 〈 A	A, C 〈 E
$FILE_NAME	기존 시각 유지	기존 시각 유지
수행 시각	A	E

〔표 16〕 폴더 생성시 시간 변화

속성 이름	모든 윈도우즈 버전
$STD_INFO	M == A == C == E
$FILE_NAME	M == A == C == E
수행 시각	M, A, C, E

2) 파일 이름 변경 시작 시각
3) 파일 이름 변경 완료 시각
4) 마지막 수정 시각과 MFT entry 수정 시각은 옮겨지는 원본 파일(A)의 시간 정보 유지
5) 마지막 수정 시각은 옮겨지는 원본 파일(A)의 시간 정보 유지

〔표 17〕 동일한 볼륨에서의 폴더 복사시 시간 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상
$STD_INFO	M == A == C == E	M < A, C, E
$FILE_NAME	M == A == C == E	M == A == C == E
수행 시각	M, A, C, E	A, C, E

간 정보를 가지고 있으며 폴더 생성시 모든 속성의 시각이 폴더 생성 시각으로 설정된다.

4.2.2 동일한 볼륨에서의 폴더 복사

폴더 복사시 Vista 미만 버전인 경우 모두 폴더의 복사 시간으로 변경된다. Vista 이상 버전의 경우 $STANDARD_INFORMATION 속성의 마지막 수정 시각은 원본 폴더의 시간으로 변경되며 나머지 시간은 폴더 복사 시각으로 변경된다.

Vista 미만 버전의 경우 폴더 복사시에 원본 폴더의 마지막 접근 시각이 복사를 수행한 시각으로 변경된다. 즉, 폴더의 생성 시각과 마지막 접근 시각을 비교하여 폴더의 원본 폴더를 유추할 수 있다. 동일한 볼륨에서의 복사 수행시 하위에 위치한 폴더는 모두 복사를 수행한 시각으로 변경되며 하위 파일은 Vista 미만 버전의 경우 원본 파일의 마지막 수정 시각과 MFT entry 수정 시각이 유지되며 Vista 이상 버전의 경우 파일의 MFT entry 수정 시각만 유지된다.

4.2.3 다른 볼륨에서의 폴더 복사

Vista 미만 버전의 경우 "Drag & Drop"과 "잘라내기 & 붙여넣기"를 모두 복사를 수행한 시각으로 변경한다. 또한 기존 볼륨에 남아있는 원본 폴더의 마지막 접근 시각이 이동을 수행한 시각으로 함께 변경된다.

하지만 Vista 이상 버전의 폴더 복사의 경우 "Drag & Drop"과 "잘라내기 & 붙여넣기"의 시간 변

〔표 18〕 다른 볼륨에서의 폴더 복사시 시간 변화

속성 이름	윈도우즈 Vista 미만
$STD_INFO	모두 복사 수행 시각으로 변경
$FILE_NAME	모두 복사 수행 시각으로 변경
수행 시각	M, A, C, E

〔표 19〕 Vista 이상 버전에서의 폴더 복사시 시간 변화

속성 이름	Drag & Drop 복사	잘라내기 & 붙여넣기 복사
$STD_INFO	M < A, C, E	M, C < A, E
$FILE_NAME	모두 복사 수행 시각으로 변경	모두 복사 수행 시각으로 변경
수행 시각	A, C, E	A, E

화에 차이가 있다. 〔표 19〕와 같이 "Drag & Drop "의 경우 기존 폴더의 마지막 수정 시각이 유지되며 " 잘라내기 & 붙여넣기"의 경우 마지막 수정 시각과 생성 시각이 유지된다. 복사 수행시 하위에 위치한 폴더와 파일의 시간 정보 변화는 동일한 볼륨에서의 폴더 복사와 동일하게 변경된다.

4.2.4 동일한 볼륨에서의 폴더 이동

대부분의 시간 변화가 Vista를 기준으로 차이를 보이는 것과는 달리 동일한 볼륨 내에서의 폴더 이동은 다른 변화를 보인다. 윈도우즈 XP와 2003의 경우 마지막 접근 시각과 MFT entry 수정 시각이 이동을 수행한 시각으로 바뀌며 나머지 윈도우즈 2000, Vista, 2008, 7의 경우 MFT entry 수정 시각만 이동 수행 시각으로 바뀐다.

또한, $FILE_NAME 속성의 경우 파일 이름 변경 시와 같이 이동 전의 $STANDARD_INFORMATION의 각 시간 값이 이동 후의 $FILE_NAME 속성에 반영된다. 즉, 이 정보를 이용하여 이동 전의 폴더의 시각 정보를 유추할 수 있다. 동일한 볼륨에서의 폴더 이동시 하위 파일(폴더)의 시간 변화의 경우 윈도우즈 2000, Vista, 2008, 7은 모두 기존의 시간이 유지된 상태로 이동된다. 하지만 윈도우즈 XP와 2003의 경우 하위 폴더는 마지막 접근 시각이 이동 시각으로 변한다. 하위 파일의 경우 다른 버전의 윈도우즈는 파일 시각이 유지되지만 윈도우즈

〔표 20〕 동일한 볼륨에서의 폴더 이동시 시간 변화

속성 이름	윈도우즈 XP, 2003	윈도우즈 2000, Vista, 2008, 7
$STD_INFO	M, C < A, E	M, A, C < E
$FILE_NAME	폴더 이동 전 $STANDARD_INFORMATION 속성의 시각 정보	폴더 이동 전 $STANDARD_INFORMATION 속성의 시각 정보
수행 시각	A, E	E

〔표 21〕 폴더 이름 변경시 시간 변화

속성 이름	모든 윈도우즈 버전
$STD_INFO	M, A, C < E
$FILE_NAME	변경 전 $STANDARD_INFORMATION 속성의 시각 정보
수행 시각	E

〔표 22〕 폴더 속성 변경시 시간 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상
$STD_INFO	M, C < A, E	M, A, C < E
$FILE_NAME	기존 시각 유지	기존 시각 유지
수행 시각	A, E	E

XP의 경우 MFT entry 수정 시각이 파일을 이동한 시각으로 변경된다.

4.2.5 폴더 아름 변경

$STANDARD_INFORMATION 속성의 정보는 운영체제 버전에 상관없이 MFT entry 수정 시각만 이름을 변경한 시각으로 변경되며 $FILE_NAME 속성 정보는 파일 이름 변경과 동일하게 기존 $STANDARD_INFORMATION 속성의 시간 값으로 변경된다.

4.2.6 폴더 속성 변경

파일과 마찬가지로 폴더의 '읽기 전용', '숨김' 속성을 변경하였을 때 Vista 미만 버전의 경우 마지막 접근 시각과 MFT entry 수정 시각이 속성을 변경한 시각으로 바뀌었으며 Vista 이상 버전의 경우 MFT entry 수정 시각만 바뀐다.

Vista 이상 버전의 경우 파일과 마찬가지로 레지스트리의 NtfsDisableLastAccessUpdate value를 변경하면 파일 속성 변경시에 마지막 접근 시각도 변경된다.

4.2.7 폴더 내부 파일(폴더) 생성

모든 버전의 운영체제가 폴더 내부에 폴더나 파일을 생성한 경우 마지막 수정 시각, 마지막 접근 시각, MFT entry 수정 시각이 파일을 생성한 시각으로 변경된다.

4.2.8 폴더 내부 파일(폴더) 수정

폴더 내부의 파일의 데이터를 수정한 경우 Vista 미만 버전의 경우 폴더의 마지막 접근 시각과 MFT entry 수정 시각이 수정된 시각으로 변경된다. Vista 이상 버전의 경우 MFT entry 수정 시각만 변경이 된다. 하지만 윈도우즈 2008의 경우는 특이하게 모든 시간 정보가 변경되지 않는다.

〔표 23〕 폴더 내부 파일(폴더) 생성시 시간 변화

속성 이름	모든 윈도우즈 버전
$STD_INFO	C < M, A, E
$FILE_NAME	기존 시각 유지
수행 시각	M, A, E

〔표 24〕 폴더 내부 파일(폴더) 수정시 시간 변화

속성 이름	윈도우즈 Vista 미만	윈도우즈 Vista 이상 (2008 제외)
$STD_INFO	M, C < A, E	M, A, C < E
$FILE_NAME	기존 시각 유지	기존 시각 유지
수행 시각	A, E	E

이름	생성시간	수정시간	접근시간	Entry 수정시간	$FILE_NAME 생성시간	$FILE_NAME 수정시간	$FILE_NAME 접근시간	$FILE_NAME Entry 수정
A04P202.D	2003-08-12 08:59:56	2003-08-12 09:03:32	2007-11-17 08:58:00	2003-08-12 12:20:57	2003-08-12 08:59:56	2003-08-12 09:03:32	2003-08-12 09:03:32	2003-08-12 12:20:37
A04p203o.d	2003-08-12 09:03:34	2003-08-12 09:06:55	2007-11-17 08:58:00	2003-08-13 12:50:53	2003-08-12 09:03:34	2003-08-12 09:06:55	2003-08-13 12:49:36	2003-08-12 12:20:57
A04P204.D	2003-08-12 09:06:59	2003-08-12 09:10:39	2007-11-17 08:58:00	2003-08-12 12:20:57	2003-08-12 09:06:59	2003-08-12 09:10:39	2003-08-12 09:10:39	2003-08-12 12:20:37
A04P205.D	2003-08-12 09:10:41	2003-08-12 09:14:16	2007-11-17 08:58:00	2003-08-12 12:20:57	2003-08-12 09:10:41	2003-08-12 09:14:16	2003-08-12 09:14:16	2003-08-12 12:20:37
A04P206.D	2003-08-12 09:14:17	2003-08-12 09:17:54	2007-11-17 08:58:00	2003-08-12 12:20:57	2003-08-12 09:14:17	2003-08-12 09:17:54	2003-08-12 09:17:54	2003-08-12 12:20:37
A04P207.D	2003-08-12 09:17:56	2003-08-12 09:21:27	2007-11-17 08:58:00	2003-08-12 12:20:57	2003-08-12 09:17:56	2003-08-12 09:21:27	2003-08-12 09:21:27	2003-08-12 12:20:37
A04P208.D	2003-08-12 09:21:29	2003-08-12 09:25:06	2007-11-17 08:58:00	2003-08-12 12:20:57	2003-08-12 09:21:29	2003-08-12 09:25:06	2003-08-12 09:25:06	2003-08-12 12:20:37
A04P209.D	2003-08-12 09:25:07	2003-08-12 09:28:43	2007-11-17 08:58:00	2003-08-12 12:20:57	2003-08-12 09:25:07	2003-08-12 09:28:43	2003-08-12 09:28:43	2003-08-12 12:20:37
A04P210.D	2003-08-12 09:28:44	2003-08-12 09:32:15	2007-11-17 08:58:00	2003-08-12 12:20:57	2003-08-12 09:28:44	2003-08-12 09:32:15	2003-08-12 09:32:15	2003-08-12 12:20:37

〔그림 7〕 대상 폴더의 각 속성별 시간 정보 (일부)

〔표 25〕 폴더 내부 파일(폴더) 삭제시 시간 변화

속성 이름	모든 윈도우즈 버전 (2003 제외)
$STD_INFO	C 〈 M, A, E
$FILE_NAME	기존 시각 유지
수행 시각	M, A, E

4.2.9 폴더 내부 파일(폴더) 삭제

폴더 내부의 파일이나 폴더를 삭제한 경우 윈도우즈 2003을 제외한 나머지 버전에서는 마지막 쓰기 시각, 마지막 접근 시각, MFT entry 수정 시각이 모두 파일(혹은 하위 폴더)를 삭제한 시각으로 변경된다. 윈도우즈 2003 버전에서는 모든 시간 정보가 변경되지 않는다.

4.2.10 폴더 덮어쓰기

폴더를 덮어쓰기 하는 경우 시간의 변화는 발생하지 않는다. 하지만 Vista 미만 버전의 경우 덮어쓰기 위해 이동된 원본 폴더의 마지막 접근 시간이 폴더 덮어쓰기를 수행한 시각으로 변경된다.

V. 사례 분석

본 사건은 약품 성분 검사 장비가 생성한 분석 결과 데이터 조작에 관한 내용으로 조작 시점과 원본 데이터의 위치를 파악하는 것이 주된 분석 내용이다. 사건의 분석 대상인 데이터를 생성하는 장비는 윈도우 운영체제가 설치된 컴퓨터에 의해 제어되며 분석된 결과를 해당 하드디스크에 자동적으로 저장하는데, 이러한 분석 결과는 샘플의 순서에 따라 순서적인 폴더를 생성하고 분석된 데이터를 폴더 내부에 파일로 저장한다. 하나의 분석된 결과를 하드디스크에 저장하는 시간은 약 3분에서 5분이 소요되며 저장하는 순서는 지정된 샘플 명칭에 따라 순차적으로 진행된다. 따라서 의뢰된 하드디스크상의 데이터가 해당 장비를 통해 자동적으로 저장되어 있다면 시간 정보가 순차적이어야 한다. 하지만 임의로 폴더나 파일을 대상으로 복사, 이동 등의 행위를 하면 시간 정보의 변화를 확인할 수 있으며, 본 논문에서 확인한 조작에 따른 시간 정보 분석을 이용해 사용자의 행위를 파악할 수 있다.

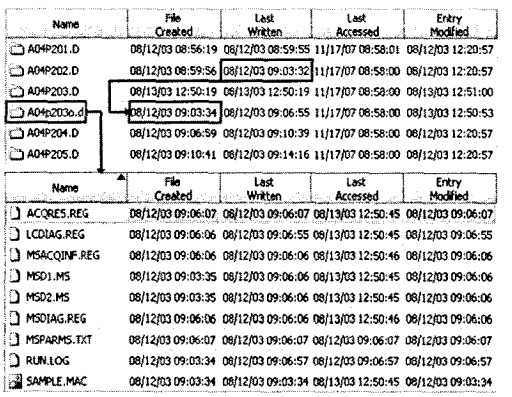
〔표 26〕 폴더 덮어쓰기시 시간 변화

속성 이름	모든 윈도우즈 버전
$STD_INFO	기존 시각 유지
$FILE_NAME	기존 시각 유지
수행 시각	Vista 미만 버전의 경우 원본 폴더의 마지막 접근 시각이 수행 시각으로 변경

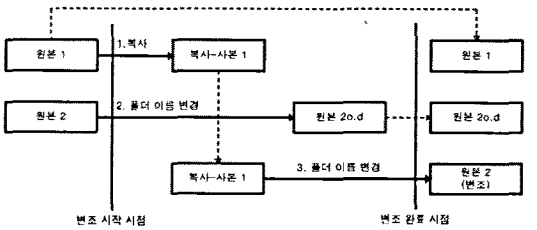
4.2.11 변조 흔적 여부 확인

대상 시스템은 윈도우즈 2000을 사용하는 것으로 분석되었으므로 본 논문에서 분석한 윈도우즈 Vista 미만 버전의 분석 내용을 적용한다. 순차적으로 폴더를 생성하여 분석된 데이터를 저장하는 장비의 폴더 이름과 시간 구성 법칙에 어긋나는 A04p203o.d 폴더를 발견하였으며 해당 위치의 폴더를 대상으로 $STANDARD_INFORMATION 속성의 시간 정보와 $FILE_NAME 속성의 시간 정보를 기반으로 분석을 수행하였다.

〔그림 7〕과 같이 분석 데이터가 저장 되는 위치의 모든 폴더의 $FILE_NAME 속성 시간이 모두 동일하지 않다. 폴더의 $FILE_NAME 속성의 시간 정보가 모두 일치하지 않는 경우는 이름을 변경하거나 동일 볼륨 내에서 이동했을 경우이다. 이름을 변경하거나 동일 볼륨 내에서 이동했을 경우 $STANDARD_INFORMATION 속성의 MFT entry 수정 시각은 해당 행위 수행 시각으로 변경된다. 즉, 2003년 8년 12월 12일 20분 57초에 일괄적으로 이름을 변경하거나 폴더를 이동 시켰다고 볼 수 있다. 시간 순서대로 생성되는 폴더 명의 규칙에서 벗어나는 A04P203.D의 $FILE_NAME 속성의 시간 정보를 살펴보면 MACE 시간 정보 모두 2003년 8월 13일 12시 50분 19초이다. 즉 동일 볼륨 내에서 복사해 왔다는 것을 알 수 있다.



〔그림 8〕 A04P203o.d 폴더의 변조 흔적 확인



〔그림 9〕 A04P203.D 폴더의 변조 과정

반면 A04P203.D 폴더는 A04P202.D의 마지막 쓰기 시각과 A04P204.D의 생성 시각 사이에 존재하지 않기 때문에 원본 폴더가 아니라고 볼 수 있다. 또한 A04P201.D, A04P202.D, A04P204.D, A04P205.D 폴더의 MFT entry 수정 시각이 2003년 8월 13일 오후 12시 20분 57초로 모두 동일하다는 것은 4개의 폴더가 동일 볼륨에서 일괄적으로 이동해왔음을 보여준다. 마지막 접근 시각의 경우 폴더의 접근만으로 변경되기 때문에 변조 시점의 시간이 유지되지 않는다는 점을 유의해야 한다.

분석 결과, 원본 A04P203.D를 포함한 총 5개의 폴더가 일괄 이동했으며 변조하고자 하는 폴더의 사본을 2003년 8월 13일 오후 12시 50분 19초에 생성했다. 그리고 〔그림 10〕과 같이 실제 원본 폴더의 이름을 A04p203o.d으로 8월 13일 오후 12시 50분 53초에 변경한 뒤, 사본의 이름을 A04P203.D로 8월 13일 오후 12시 51분 00초에 변경한 것으로 사건을 재구성할 수 있다. 이 외에 변조한 흔적이 총 3개의 폴더, 43개의 샘플에서 발견되었으며 변조한 데이터와 그 변조 시점을 확인할 수 있었다.

VI. 결 론

본 논문에서는 여러 버전의 윈도우즈를 대상으로 행위에 따라 파일과 폴더의 시간 정보 변화를 확인하였다. NTFS의 경우 시간 정보를 $STANDARD_INFORMATION 속성과 $FILE_NAME 속성으로 관리하며 두 속성의 시간 정보를 비교하여 더 많은 행위를 유추할 수 있다. 또한, MFT entry 내에 데이터 스트림을 관리하는 경우와 클러스터에 데이터 스트림을 할당하여 관리하는 경우에 행위에 따라 다른 시간 변화를 보이는 것을 확인하였다. 이 분석 결과를 통해 파일과 폴더의 시간 정보를 기반으로 사용자의 행위와 그 시점을 유추할 수 있다. 또한 디지털 포렌식 수사를 진행함에 있어 수사관이 용의자의 컴퓨터 시스템으로부터 사건과 관련된 증거를 수집할 때에도 도움이 될 것이다.

참 고 문 헌

〔1〕 G. Palmer, "A Road Map for Digital Forensic Research," technical report DTR-T001-0, Utica, New York, Nov. 2001.

〔2〕 C. Boyd and P. Forster, "Time and Date issues in forensic computing - a case study," Digital Investigation, vol. 1, no. 1, pp. 18-23, Feb. 2004.

〔3〕 Guidence Software, Inc. "EnCase," http://www.guidencesoftware.com

〔4〕 New Technology File System "NTFS," http://www.ntfs.com

〔5〕 K.P. Chow, F.Y.W. Law, M.Y.K. Kwan and P.K.Y. Lai, "The Rules of Time on NTFS File System," SADFE, pp. 71-85, Mar. 2007.

〔6〕 M. Geiger, "Evaluating Commercial Counter-Forensic Tools," Digital forensic research workshop, New Orleans, LA, pp. 39-41, Aug. 2005.

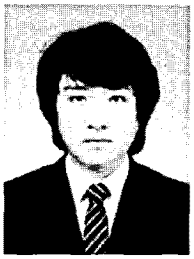
〔7〕 J.W. Bang, B.Y. Yoo, J.S. Kim, and S.J. Lee, "Analysis of Time Information for Digital Investigation," 5th International Joint Conference on INC, IMS and IDC, vol. 5, no. 1, pp. 1858-1864, Aug. 2009.

〔8〕 B. Carrier, File System Forensic Analysis, Addison-Wesly, Mar. 2005.

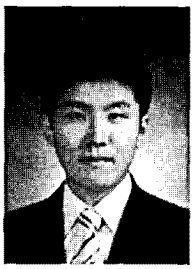
〔9〕 B. Carrier, "The Sleuth Kit," http://www.sleuthkit.org

〔10〕 J.W. Bang, "The MFT Entry Parser," http://www.forensic.or.kr

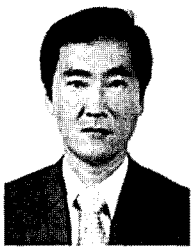
〈著者紹介〉



방 제 완 (Jewan Bang) 정회원
2007년 2월: 한세대학교 정보통신공학과 졸업
2007년 3월~현재: 고려대학교 정보경영공학전문대학원 석박사통합과정
〈관심분야〉 디지털 포렌식, 소프트웨어 역공학 분석, 임베디드 시스템



유 병 영 (Byeongyeong Yoo) 학생회원
2009년 2월: 상명대학교 컴퓨터과학과 졸업
2009년 3월~현재: 고려대학교 정보경영공학전문대학원 석사과정
〈관심분야〉 디지털 포렌식



이 상 진 (Sangjin Lee) 정회원
1987년 2월: 고려대학교 수학과 졸업
1989년 2월: 고려대학교 수학과 이학석사
1994년 8월: 고려대학교 수학과 이학박사
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~현재: 고려대학교 정보경영공학전문대학원 교수
〈관심분야〉 디지털 포렌식, 모바일 포렌식, 심층 암호, 해쉬 함수

부 록. 파일과 폴더 조작에 따른 파일 시간 변화

파일	속성	윈도우즈 Vista 버전 미만	윈도우즈 Vista 버전 이상
파일(폴더) 생성	모두	모두 생성 시각으로 변경	
복사[1]	$SI[2]	M, E < C, A	M < C, A, E
동일 볼륨 이동[3] (MFT entry 할당)	$SI	M, A, C < E[4]	
동일 볼륨 이동[3] (클러스터 할당)	$SI	M, C < A, E	M, C, E < A
다른 볼륨 이동[1] (Drag & Drop)	$SI	M, E < C, A	M < C, A, E
다른 볼륨 이동[1] (잘라내기 & 붙여넣기)	$SI	M, C, E < A	M, C < A, E
속성 변경[3]	$SI	M, C < A, E	M, A, C < E
NOTEPAD 내용 수정[3]	$SI	C < M, A, E	A, C < M, E
WORD 내용 수정	모두	C < M, A, E	
파일(폴더) 이름 변경	$SI	M, C < A, E[5]	M, A, C < E
	$FN[6]	변경 전 $STANDARD_INFORMATION 속성의 시각 정보	
덮어 쓰기[3]	$SI	C < A[7]	A, C < E[8]
폴더	속성	윈도우즈 Vista 버전 미만	윈도우즈 Vista 버전 이상
동일 볼륨 복사[1]	$SI	모두 복사 수행 시각으로 변경	M < A, C, E
다른 볼륨 이동[1] (Drag & Drop)	$SI	모두 복사 수행 시각으로 변경	M < A, C, E
다른 볼륨 이동[1] (잘라내기 & 붙여넣기)	$SI	모두 복사 수행 시각으로 변경	M, C < A, E
동일 볼륨 이동	$SI	M, C < A, E[9]	M, A, C < E[10]
	$FN	이동 전 $STANDARD_INFORMATION 속성의 시각 정보	
속성 변경[3]	$SI	M, C < A, E	M, A, C < E
내부 파일(폴더) 생성[3]	$SI	C < M, A, E	
내부 파일(폴더) 수정[3]	$SI	M, C < A, E	M, A, C < E[11]
내부 파일(폴더) 삭제[1]	$SI	C < M, A, E[12]	
덮어 쓰기[2]	$SI	모두 기존 시각 유지[13]	

1) $FILE_NAME 속성의 모든 시각 정보는 해당 행위를 수행한 시각으로 변경
2) $STANDARD_INFORMATION 속성의 시간 정보
3) $FILE_NAME 속성의 모든 시각 정보는 기존 그대로 유지
4) 윈도우즈 7은 데이터 할당 위치에 상관없이 MFT entry 수정 시각만 변경
5) Vista 버전 미만에서 클러스터 할당된 경우에만 마지막 접근 시각과 MFT Entry 수정 시각이 변경되고 나머지 경우에는 MFT entry 수정 시각만 변경
6) $FILE_NAME 속성의 시간 정보
7) 마지막 수정 시각과 MFT entry 수정 시각은 옮겨지는 원본 파일의 시간 정보 유지
8) 마지막 수정 시각은 옮겨지는 원본 파일의 시간 정보 유지
9) 윈도우즈 XP, 2003 버전
10) 윈도우즈 2000, Vista, 2008, 7 버전
11) 윈도우즈 2008 버전에서는 모든 시간 정보가 변경되지 않음
12) 윈도우즈 2003 버전에서는 모든 시간 정보가 변경되지 않음
13) Vista 미만 버전의 경우 원본 폴더의 마지막 접근 시각이 덮어쓰기를 수행한 시각으로 변경