NAND 플래시 메모리에서 디지털 포렌식을 위한 파일 복구기법

(A File Recovery Technique for Digital Forensics on NAND Flash Memory)

신 명 섭 † 박 동 주 ††
(Myung-Sub Shin) (Dong-Joo Park)

요 약 최근 플래시 메모리가 디지털 기기의 저장장치로 널리 사용됨에 따라 플래시 메모리에서 디지털 증거를 분석하기 위한 디지털 포렌식의 필요성이 증가하고 있다. 이를 위해 플래시 메모리에 저장되어 있는 파일을 효율적으로 복구하는 것이 매우 중요하다. 그러나 기존의 하드 디스크 기반 파일 복구 기법을 플래시 메모리에 그대로 적용하기에는 너무나 비효율적이다. 덮어쓰기 불가능과 같이 플래시 메모리는 하드 디스크와 전혀 다른 특성을 가지기 때문이다. 본 논문에서 디지털 포렌식을 지원하기 위한 플래시 메모리를 잘 이해하는 파일 복구 기법을 제안한다. 첫째, 플래시 메모리 저장장치로부터 복구 가능한 모든 파일들을 효과적으로 검색하는 방법을 제안한다. 이것은 플래시 메모리의 쓰기 연산을 담당하는 FTL (Flash Translation Layer)의 메타데이터를 최대한 활용한다. 둘째, 복구 대상 파일들 중에서 특정 파일을 효율적으로 복구할 수 있는 기법을 제안하며, 이를 위해 FTL의 사상 테이블의 위치 정보를 이용한다. 다양한 실험을 통해 본 논문에 제안하는 기법이 기존의 하드 디스크 기반 파일 복구 기법보다 우수함을 보인다.

키워드 : 파일 복구, 디지털 포렌식, 플래시 메모리, 플래시 변환 계층

Abstract Recently, as flash memory is used as digital storage devices, necessity for digital forensics is growing in a flash memory area for digital evidence analysis. For this purpose, it is important to recover crashed files stored on flash memory efficiently. However, it is inefficient to apply the hard disk based file recovery techniques to flash memory, since hard disk and flash memory have different characteristics, especially flash memory being unable to in-place update. In this paper, we propose a flash-aware file recovery technique for digital forensics. First, we propose an efficient search technique to find all crashed files. This uses meta-data maintained by FTL(Flash Translation Layer) which is responsible for write operation in flash memory. Second, we advise an efficient recovery technique to recover a crashed file which uses data location information of the mapping table in FTL. Through diverse experiments, we show that our file recovery technique outperforms the hard disk based technique.

Key words : file recovery, digital forensic, flash memory, FTL

· 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2010-0013387)

† 학생회원 : 숭실대학교 컴퓨터학과
shinms88@ssu.ac.kr
†† 정 회 원 : 숭실대학교 컴퓨터학부 교수
djpark@ssu.ac.kr

논문접수 : 2010년 8월 19일
심사완료 : 2010년 9월 24일

정보과학회논문지: 데이타베이스 제37권 제6호(2010.12)

1. 서 론

최근 플래시 메모리는 디지털 기기의 저장장치로 많이 이용되어 디지털 증거를 분석하는 디지털 포렌식 수사가 진행된다. 디지털 포렌식 수사는 증거 수집, 증거 분석, 증거 제출의 절차로 이루어진다. 이 중에 증거 분석 방안으로 데이터 복구가 존재하고 디스크기반 복구 기법이 보편적이다.

파일시스템을 사용하면 서로 다른 특징을 지닌 파일이 발생한다. 본 논문은 이러한 파일을 구별하기 위해 네 가지로 분류한다. 첫 번째는 순수 삭제파일로 파일시

스템에서 일반적인 삭제처리만 된 파일이고 두 번째는 수정파일로 파일의 데이터가 수정된 형태이며 세 번째는 조각파일로 물리적으로 떨어진 영역에 나누어 저장된 파일이 삭제된 것이다. 마지막은 불순 삭제파일로 삭제된 파일의 데이터가 다른 파일에 의해 덮어쓰기 된 파일이다. 특히 불순 삭제파일은 일반적인 파일시스템의 사용으로 발생하기 드물며, 파일 손실을 위한 삭제기법이 사용되면 발생할 가능성이 높다. 결국 불순 삭제파일은 디지털 증거일 가능성이 높다. 그러나 디스크기반 복구기법은 조각파일과 불순 삭제파일의 복구가 불가능한 한계점이 있지만 플래시 메모리에서 사용된다.

플래시 메모리는 하드디스크와 다른 특성이 존재하여 디스크기반 복구기법의 한계점을 해결할 수 있다. 플래시 메모리는 제자리 덮어쓰기(In-Place Update)가 불가능하기 때문에 덮어쓰기를 하기 위해서는 소거연산 후 쓰기작업(Erase-Before-Write)을 한다. 소거연산은 읽기, 쓰기 작업에 비해 연산 비용이 크다. 따라서 제자리 덮어쓰기가 발생하면 다른 자리에 덮어쓰기(Out-of-Place Update)를 하고 사상정보를 유지하여, 소거 연산을 회피하는 시스템 소프트웨어인 FTL(Flash Translation Layer)[1]을 사용한다. 연구기반인 BAST FTL[2]의 경우, 소거 연산을 회피하기 위해 덮어쓰기 데이터인 최신 데이터를 로그블록의 다른 자리에 덮어쓰기 한다. 결국 이전 데이터는 덮어쓰기가 발생하지 않아 유지되고 사상정보로 인해 이전 데이터의 위치도 유지된다.

본 논문은 위와 같은 플래시 메모리의 특성을 고려한 효율적인 플래시 메모리기반 복구기법을 제안한다. 파일시스템이 파일의 삭제, 생성, 수정작업으로 메타데이터를 수정하면, BAST FTL이 덮어쓰기 데이터인 최신 메타데이터를 로그블록에 기록해서 이전 메타데이터가 유지된다. 따라서 복구 대상 파일을 최신 메타데이터와 이전 메타데이터를 비교하여 검색할 수 있다. 또 BAST FTL로 인해 불순 삭제파일의 데이터는 장치 내 영역에 분산된다. 따라서 불순 삭제파일은 사상정보를 이용하여 이전 데이터의 위치를 추적한 후 복구한다.

본 논문의 구성은 다음과 같다. 2장은 배경지식 및 관련연구인 BAST FTL과 디스크기반 복구기법에 대해서 기술하고 3장은 효율적인 플래시 메모리기반 복구기법에 대해서 기술하며 4장은 실험 및 실험결과에서는 효율적인 플래시 메모리기반 복구기법과 디스크기반 복구기법, 파이널데이터 2.0을 비교 실험한 결과를 기술하고 5장은 결론 및 향후계획을 기술한다.

2. 배경지식 및 관련연구

2.1 플래시 메모리

플래시 메모리는 구성하는 반도체 메모리의 셀(Cell) 종류에 따라 크게 노어(NOR)형, 낸드(NAND)형 두 가지 종류로 분류된다. 낸드형 플래시 메모리의 소 블록 구조인 경우, 장치는 다수 개의 블록(Block)으로 구성되고 한 블록의 크기는 16KB이며, 하나의 블록은 32 개의 페이지(Page)로 구성되고 하나의 페이지는 데이터가 기록되는 크기가 512Byte인 섹터(Sector)와 크기가 16Byte인 여유 영역(Spare Area)으로 구성된다[3]. 낸드형 플래시 메모리는 페이지 단위의 읽기, 쓰기작업을 하며 서로 속도가 상이하다. 또한 제자리 덮어쓰기가 불가능하고 블록 단위의 소거연산이 존재한다. 만약 특정 블록의 한 페이지에 제자리 덮어쓰기가 발생하면 특정 블록을 소거(Erase)하고 유효한 페이지를 다시 기록하는 연산을 수행해야 한다. 게다가 소거연산은 다른 연산에 비해 연산 비용이 크다. 따라서 제자리 덮어쓰기가 발생하면 위와 같은 연산을 효율적으로 변환하는 소프트웨어인 FTL을 사용한다.

2.2 BAST FTL

FTL은 상위 계층의 명령인 논리적주소의 읽기, 쓰기연산을 낸드형 플래시 메모리의 물리적주소의 읽기, 쓰기, 소거연산으로 변환한다. 만약 제자리 덮어쓰기가 발생하면 비용이 큰 소거연산을 회피하기 위해 다른 자리에 덮어쓰기를 한다. 또 주소 사상을 위해 사상테이블(Mapping Table)을 관리한다. FTL은 사상방안이나 소거연산의 회피방안에 따라 분류되며 본 논문은 연구기반으로 BAST FTL를 사용한다. BAST FTL은 데이터를 기록하기 위한 데이터블록과 다른 자리에 덮어쓰기를 하기 위한 공간인 로그블록으로 플래시 메모리의 블록을 나누어 사용한다. 데이터블록은 사상단위가 블록인 블록 사상기법[4,5]를, 로그블록은 섹터 사상기법[6]을 사용한다. 따라서 데이터블록을 사상하는 블록 사상테이블과 로그블록을 사상하는 로그 사상테이블을 지닌다. 로그블록은 제자리 덮어쓰기가 발생한 해당 데이터블록에 하나가 할당되고, 한정된 개수로 설정된다. 만약 로그블록이 모두 할당되어 있을 시에 빈 로그블록이 필요하면, 그 중 하나를 희생블록으로 선정하고 병합작업을 한다. 병합작업은 로그블록과 데이터블록에서 유효한 데이터만 취합하여 하나의 데이터블록으로 합치는 작업이다. 또한 본 논문의 연구기반인 BAST FTL은 무효한 데이터를 수집하는 가비지 컬렉터(Garbage Collector)[7]를 사용한다.

그림 1은 블록 내 페이지 수를 네 개로 가정한 플래시 메모리에서 BAST FTL의 작업을 표현한 그림이다. 그림 1과 같이 LSN(Logical Sector Number) 0번에 제자리 덮어쓰기가 발생하면 로그블록인 PBN(Physical Block Number) 90번을 할당받아 덮어쓰기 데이터인 LSN 0번을 로그블록에 기록한 후 로그 사상테이블에 덮어쓰기 내역을 기록한다.

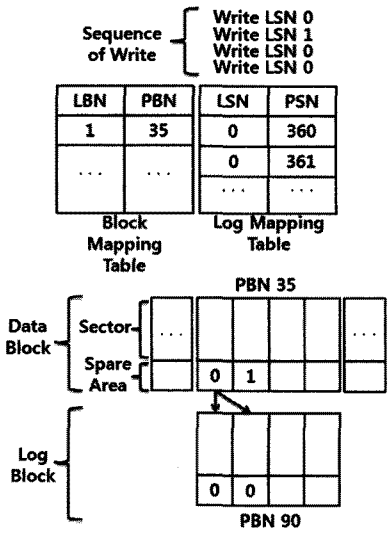

그림 1 BAST FTL

2.3 FAT32

파일시스템은 파일 생성, 삭제, 수정 등의 파일을 관리하기 위한 시스템이며 많은 분류와 종류를 지닌다. 본 논문은 연구기반으로 디스크 파일시스템 중에 대표적이고 많이 사용되는 FAT32[8]를 연구하였다. FAT32는 관리영역인 MBR(Master Boot Record), FAT(File Allocation Table) 영역, 디렉토리 엔트리(Directory Entry) 등과 데이터영역으로 구분된다. 데이터영역은 최소 할당 크기인 클러스터(Cluster)로 나누어져 있다. FAT 영역은 클러스터 총 개수만큼 클러스터의 할당 정보가 기록된 FAT 엔트리(FAT Entry)를 지닌다. FAT32에서 파일 삭제연산이 발생하면 그림 2와 같이 디렉토리 엔트리를 검색하여 삭제할 파일을 검색하고 파일의 시작 클러스터로 할당정보 추출한다. 그 후 해당되는 FAT 엔트리 5번과 6번을 할당해제한 후 디렉토리 엔트리의 파일명을 'E5h'로 표기하여 삭제연산을 처리한다. 결국 삭제파일은 데이터 영역에 잔존하게 된다[8].

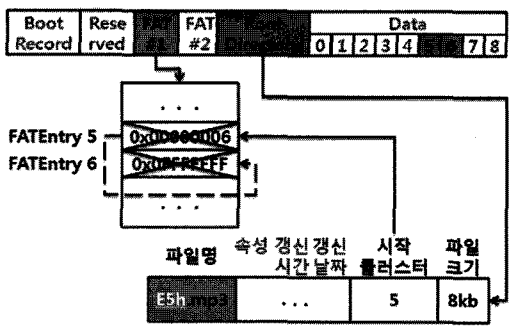

그림 2 FAT32의 파일 삭제 연산

2.4 데이터 복구기법

본 절에서는 디지털 포렌식 수사에서 증거분석 절차의 데이터복구 방안 중 디스크기반 복구기법에 대해서 기술한다.

2.4.1 디렉토리 스캐닝

디렉토리 스캐닝은 2.3절과 같이 삭제파일이 잔존되는 것을 이용한다. 그림 3과 같이 검색 시에는 장치 내의 모든 디렉토리 엔트리에서 삭제 플래그 'E5h'로 삭제된 파일을 검색한다. 복구 시에는 삭제파일의 시작 클러스터 5번으로 관련 FAT 엔트리를 분석한 후, FAT 영역에서 관련된 FAT 엔트리 5번부터 연이어진 비 할당 FAT 엔트리인 6번까지 삭제파일로 인식하고 복구한다. 디렉토리 스캐닝은 파일시스템의 특성에 의존하기 때문에 복구가 불가한 경우가 존재한다. 먼저 삭제된 디렉토리 엔트리가 덮어쓰기 되면 손실되어 복구가 불가하다. 또한 물리적으로 떨어진 클러스터에 할당된 파일이 삭제된 경우에는 할당정보가 삭제되어 파일의 데이터 위치를 추출할 수 없기 때문에 복구가 불가하다. 그리고 삭제파일의 데이터 영역이 다른 파일에 의해 덮어쓰기 될 때, 하드디스크의 경우 데이터가 손실되기 때문에 복구가 불가하다.

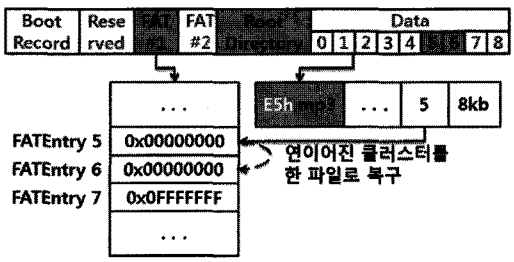

그림 3 디렉토리 스캐닝

2.4.2 섹터 스캐닝

섹터 스캐닝은 파일의 내부적 특징을 이용한 복구기법이다. 파일의 내부적 특징이란 파일의 종류에 따라 특정 섹터, 바이트, 위치에 특정 값, 구조 등을 말한다. 파일의 첫 섹터에 존재하는 특징은 파일 헤더(File Header), 마지막 섹터에 존재하는 특징은 파일 푸터(File Footer)라 한다[9,10]. 섹터 스캐닝은 그림 4와 같이 전 영역의 섹터를 검색하여 파일 헤더와 파일 푸터를 검출하고 검출된 파일 헤더부터 파일 푸터까지 한 파일로 인식하여 복구한다. 섹터 스캐닝은 전 영역을 검색하기 때문 복구가 가능한 대상이 많은 장점이 있다. 하지만 모든 파일을 복구하기 위해서는 모든 파일의 내부적 특징을 알아야 한다. 또 파일이 물리적으로 떨어져 저장된 경우, 복구 파일 중간에 파일에 상관없는 데이터가 삽입되어 제

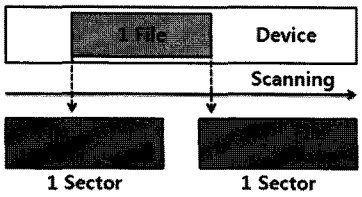

그림 4 섹터 스캐닝

대로 복구가 되지 않는다. 그리고 삭제파일의 데이터가 덮어쓰기가 되면 복구가 불가하다.

3. 효율적인 플래시 메모리기반 복구기법

2장에서 살펴본 디스크기반 복구기법은 삭제파일의 데이터가 덮어쓰기 되었거나, 물리적으로 떨어진 영역에 나누어 저장된 파일이 삭제되면, 복구가 불가한 한계점을 보인다. 이것은 FAT32로 인해서 서로 다른 특징을 지닌 파일, 데이터가 발생하기 때문이다. 본 논문은 이러한 파일을 구별하기 위해 네 가지로 분류한다. 첫 번째는 순수 삭제파일로 FAT32의 삭제처리인 메타데이터의 수정으로 데이터영역에 삭제파일이 잔존하는 파일이다. 두 번째는 수정파일로 FAT32의 수정작업인 다른 클러스터에 재 할당작업의 대상인 파일이다. 세 번째는 조각파일로 물리적으로 떨어진 클러스터에 할당된 파일이 삭제된 경우이다. 네 번째는 불순 삭제파일로 삭제파일의 데이터가 다른 데이터에 의해 덮어쓰기 된 파일이다.

특히 불순 삭제파일은 일반적인 FAT32의 작업으로 발생하기 드물다. 이것은 FAT32가 클러스터의 할당을 처음부터 마지막까지 순차적으로 할당하여 최근에 삭제처리 된 파일의 데이터가 다시 덮어쓰기 되는 경우가 적기 때문이다. 따라서 최근에 발생한 불순 삭제파일은 FAT32의 시스템에 의해 발생하지 않고 인위적인 정보은폐 작업인 파일 손실을 위한 삭제기법이 사용된 가능성이 높다. 결국 디지털 포렌식 수사 관점에서 충분히 복구할 필요가 있는 디지털 증거일 가능성이 있으며 본 논문은 불순 삭제파일에 초점을 둔다.

최근 디지털 기기에서 기존 기법을 사용하면 디스크장치가 아닌 플래시 메모리에 수행된다. 그러나 플래시 메모리는 기존 기법의 한계점을 해결할 수 있는 특징이 있어 기존 기법을 사용하는 것은 비효율적이다. 따라서 플래시 메모리기반 복구기법이 필요하다.

3.1 아이디어

이번 절에서는 효율적인 플래시 메모리기반 복구기법의 아이디어가 되는 FAT32와 BAST FTL의 사용으로 발생하는 특징들을 기술한다.

3.1.1 파일시스템의 메타데이터 수정

파일시스템에서 파일의 삭제, 생성 작업은 메타데이터

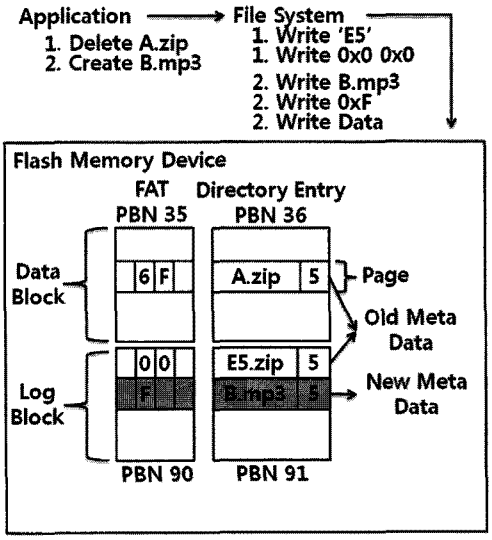

그림 5 이전 메타데이터

의 덮어쓰기가 발생하여, 플래시 메모리에는 제자리 덮어쓰기가 발생한다. BAST FTL은 소거연산을 회피하기 위해 로그블록의 다른 자리에 덮어쓰기를 한다. 따라서 삭제파일이 발생할 때 삭제파일의 메타데이터는 덮어쓰기가 발생하지 않아 유지된다. 또 다른 자리에 덮어쓰기 된 메타데이터도 존재하게 된다. 본 논문은 복구대상 파일과 관련된 메타데이터를 이전 메타데이터라고 정의하고 다른 자리에 덮어쓰기 된 메타데이터를 최신 메타데이터로 정의한다.

예시로 그림 5에서 불순 삭제파일이 발생하였을 때 이전 메타데이터의 발생을 보여준다. 그림 5와 같이 데이터블록 PBN 35번, 36번에 'A.zip'의 메타데이터가 존재하고 있을 때 'A.zip'의 삭제가 되면 파일시스템은 삭제작업으로 디렉토리 엔트리의 덮어쓰기와 FAT 엔트리 덮어쓰기를 수행된다. 플래시 메모리에서는 제자리 덮어쓰기가 발생하여 최신 메타데이터의 내용인 'E5', '0×0 0×0' 를 로그블록 PBN 90번, 91번에 다른 자리에 덮어쓰기 한다. 이후 삭제파일의 데이터에 다른 파일에 의해 덮어쓰기가 발생하는 파일생성 작업이 수행되면 플래시 메모리에는 제자리 덮어쓰기가 발생한다. 마찬가지로 최신 메타데이터의 내용인 'B.mp3', '0xf', 파일의 데이터를 로그블록 PBN 90번, 91번에 다른 자리에 덮어쓰기 한다. 결국 'A.zip' 파일은 디스크기반 복구기법으로 복구가 불가한 불순 삭제파일이 된다. 그림 5와 같이 로그블록에는 최신 메타데이터가 기록되고 두 개의 이전 메타데이터가 유지된다.

3.1.2 BAST FTL의 병합작업

불순 삭제파일은 삭제파일의 데이터가 덮어쓰기 되어

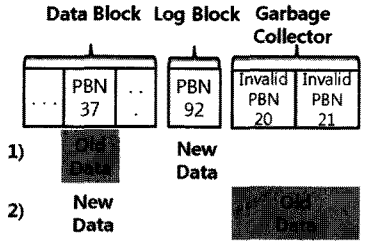

그림 6 이전 데이터의 위치

메타데이터 외에도 파일의 데이터가 로그블록에 기록된다. 그런데 덮어쓰기 데이터인 최신 데이터의 크기가 로그블록의 크기보다 크면 최신 데이터의 모든 부분을 로그블록에 적재할 수 없다. 빈 로그블록이 없는 경우에는 병합작업이 발생하여 최신 데이터의 일부는 로그블록에, 다른 일부는 병합되어 다른 영역에 분산된다. 따라서 최신 데이터의 병합작업에 포함되는 불순 삭제파일의 데이터, 이전 데이터는 로그블록, 데이터블록, 가비지 컬렉터 영역에 분산될 수가 있다.

그림 6은 최신 데이터가 크기가 큰 경우의 데이터의 위치를 표현한 그림이다. 그림 6의 1)번처럼 최신 데이터가 로그블록에 위치하고 있으면 로그블록의 병합작업이 발생하지 않아 이전 데이터는 데이터블록에 존재한다. 만약 병합작업이 발생하면 그림 6의 2)번처럼 최신 데이터는 병합되어 데이터블록에, 이전 데이터는 가비지 컬렉터 영역에 존재한다. 대 부분의 경우 그림 6의 1)번 경우와 2)번 경우가 혼재되어 있다. 반대로 최신 데이터가 크기가 작다면 로그블록에 존재하게 되므로 이전 데이터는 로그블록이나 데이터블록에 존재한다.

3.2 파일 복구시스템의 구조

플래시 메모리는 3.1절에서 기술한 내용과 같이 하드디스크와 다른 특성이 존재하여 디스크기반 복구기법의 한계점을 해결할 수 있다. 본 논문은 플래시 메모리의 특성을 고려한 효율적인 플래시 메모리기반 복구기법을 제안한다. 그림 7과 같이 제안하는 파일복구 시스템의 구조는 플래시 메모리, BAST FTL, FAT32와 검색기, 복구기로 구성된다. 플래시 메모리는 BAST FTL를 사용함으로써 데이터블록과 로그블록과 가비지 컬렉터로 세 영역이 존재한다. BAST FTL의 블록 사상테이블은 이전 메타데이터가 유지된 영역인 데이터블록을 사상하고 로그 사상테이블은 최신 메타데이터가 기록된 영역인 로그블록을 사상한다.

파일 검색기는 3.1.1과 같이 장치 내에 이전 메타데이터와 최신 메타데이터가 존재하는 현상을 이용한다. 두 메타데이터를 비교하여 상이한 정보를 추출하여 복구 대상 파일을 검색한다. 검색된 파일은 복구 대상 파일리스트로 구성하여 복구에 필요한 파일명, 크기, 확장자,

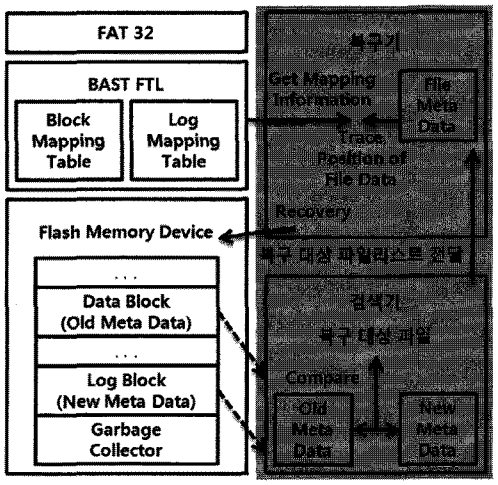

그림 7 파일복구 시스템의 구조

논리적주소 등의 정보를 유지하고 복구기에 전달된다. 파일 복구기는 전달된 리스트에서 사용자가 선택한 파일을 복구한다. 3.1.2절과 같이 파일의 데이터는 장치의 데이터블록 영역, 로그블록 영역, 가비지 컬렉션 영역에 분산된다. 따라서 이전 데이터의 위치를 추적해야 한다. 먼저 선택한 파일의 논리적 위치와 BAST FTL의 사상정보를 이용해 파일의 데이터가 데이터블록, 로그블록, 가비지 컬렉션 영역 중 어느 영역에 존재하는지 수집한다. 그 후에 최신 데이터의 위치를 추출하고 수집된 영역에서 대응되는 이전 데이터의 위치를 추적한다. 추적에 성공하면 추적된 데이터를 파일로 복구한다.

3.3 파일 검색방안

네 분류의 복구 대상 파일은 모두 제자리 덮어쓰기가 발생하므로 모두 이전 메타데이터가 유지되고 최신 메타데이터가 기록된다. 그러나 각각 이전 메타데이터가 유지되는 형태와 최신 메타데이터가 기록되는 형태가 다르다. 따라서 두 메타데이터를 비교 또는 분석하면 복구 대상 파일의 분류하여 검색할 수 있다.

순수 삭제파일과 수정파일은 디스크기반 복구기법으로 복구가 가능한 파일이다. 따라서 본 논문은 디렉토리 스캐닝 기법을 플래시 메모리의 특성을 이용하여 변경하는 것을 제안한다. 제안하는 순수 삭제파일과 수정파일의 검색방안은 2.4.1절과 같이 디렉토리 스캐닝기법을 사용할 시 발생하는 디렉토리 엔트리가 덮어쓰인 경우의 한계점을 보안한다. 본래 기법이 검색하는 로그블록뿐만 아니라 이전 메타데이터가 기록된 데이터블록까지 검색하는 방안이다. 여기서 순수 삭제파일의 경우 이전 디렉토리 엔트리와 최신 디렉토리 엔트리에서 삭제플래그인 'E5'가 존재하는 디렉토리 엔트리를 검색한다. 수정파일의 경우, 재 할당 작업을 하기 때문에 이전 디렉

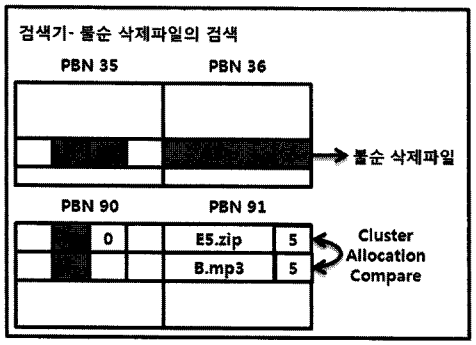

그림 8 불순 삭제파일의 검색

토리 엔트리와 최신 디렉토리 엔트리에서 할당영역만 변경된 디렉토리 엔트리를 검색한다.

불순 삭제파일의 경우 삭제파일의 데이터영역이 덮어쓰기 된 상태이므로 이전 디렉토리 엔트리와 최신 디렉토리 엔트리의 할당영역이 같은 특징이 있다. 따라서 이전 디렉토리 엔트리와 최신 디렉토리 엔트리를 비교하여 같은 영역을 지니는 디렉토리 엔트리를 검색한다. 그림 8은 불순 삭제파일의 검색을 위한 예시이며 PBN 35와 PBN 36은 데이터블록으로 PBN 90과 PBN 91은 로그블록으로 가정한다. 그림 8과 같이 PBN 91번에서 할당된 영역 5번으로 같은 디렉토리 엔트리, 'E5.zip'와 'B.mp3'가 존재하면 'A.zip' 파일은 불순 삭제파일로 검색할 수 있다. 불순 삭제파일을 검색한다면 파일명, 크기, 논리적주소 등의 복구 정보를 수집한다. 특히 논리적주소의 경우 그림 8과 같이 'A.zip'의 덮어쓰기 순서를 계산하였을 때, PBN 35번의 이전 FAT 엔트리와 관련이 있으므로 'A.zip' 파일의 데이터는 클러스터 5번, 6번이며 해당되는 논리적주소를 알 수 있다.

마지막으로 조각파일의 검색은 물리적으로 떨어져 기록된 특징을 이용한다. 따라서 파일 검색 시에 이전 FAT 엔트리를 분석한다. FAT 엔트리가 연이어져 있지 않으면 해당 파일은 조각파일로 검색할 수 있다. 조각파일로 분류되면 해당 FAT 엔트리 정보를 기본 복구 정보 외에 추가로 수집한다. 검색방안으로 수집된 정보는 복구 시 사용하고 사용자에게 보고하기 위한 목적으로 리스트화 한다.

3.4 파일 복구방안

복구 대상 파일의 복구는 세 가지 복구방안이 존재한다. 순수 삭제파일, 수정파일의 복구방안의 경우 디스크기반 복구기법으로 복구가 가능하여 이 두 파일의 복구방안으로 디스크기반 복구기법과 같은 기법을 사용한다. 그러나 2.4.1절의 디렉토리 엔트리가 덮어쓰인 경우의 한계점에 해당하는 파일도 검색하여 기존 기법보다 더 많은 량의 복구가 가능하다. 그리고 조각파일의 경우 디스크기반 복구기법에서 복구가 불가능한 이유는 FAT 엔트리가 손실되었기 때문이다. 본 논문의 3.3절과 같이 조각파일의 검색방안에서 FAT 엔트리를 수집하여 FAT 엔트리를 알 수 있다. 따라서 수집된 FAT 엔트리에 해당하는 데이터를 복구한다.

불순 삭제파일의 경우, 3.1.2절과 같이 불순 삭제파일은 로그블록, 데이터블록, 가비지 컬렉션 영역에 분산될 수 있다. 따라서 불순 삭제파일을 복구하기 위해서 파일의 데이터 위치를 추적해야 한다. 일단 불순 삭제파일의 논리적주소와 사상정보로 이전 데이터가 포함되는 영역을 검출한다. 그 후 사상정보로 최신 데이터의 위치를 추출하고 검출된 영역에서 3.1.2절과 같이 대응되는 이전 데이터를 추적하여 복구한다.

예시 그림 9는 한 블록에 네 개의 페이지로 구성된 플래시 메모리로 가정한 불순 삭제파일의 복구 예시 그림이다. 먼저 그림 9와 같이 클러스터 5번, 6번이 속한 LBN(Logical Block Number) 3번이 어떤 영역에 존재하는 지 분석한다. BAST의 사상정보로 LBN 3번은 데이터영역은 PBN 37번, 로그영역은 PBN 92번, 가비지 컬렉터 영역은 PBN 20번으로 세 영역에 모두 존재하는 것을 알 수 있다. 만약 클러스터 5번만 덮어쓰기 되었다는 가정을 하고 이 불순 삭제파일을 복구한다면 세 가지 추적방안이 존재한다. 첫 번째, 3.1.2절의 그림 6의 1번과 같이 클러스터 5번의 최신 데이터가 로그블록인 PBN 92번에 존재하면, 아직 병합작업이 발생하지 않은 것이다. 따라서 클러스터 5번의 이전 데이터는 데이터블록인 PBN 37번에 존재한다. 두 번째, 3.1.2절의 그림 6의 2번과 같이 클러스터 5번의 최신 데이터가 데이터블록인 PBN 37번에 존재하면, 병합작업이 발생하였다. 따라서 클러스터 5번의 이전 데이터는 가비지 컬렉션 영역인 PBN 20번에 존재한다. 세 번째, 3.1.2절의 그림 6의 1번, 2번의 혼재된 경우로 클러스터 5번의 최신 데이터 일부가 데이터블록인 PBN 37번에 존재하고 나머지는 로그블록인 PBN 92번에 존재하면, 클러스터 5번의 이전 데이터는 PBN 20번과 PBN 92번에 존재한다. 이렇게 불순 삭제일의 데이터 추적에 성공하면 추적된 위치정보를 토대로 데이터를 복구한다.

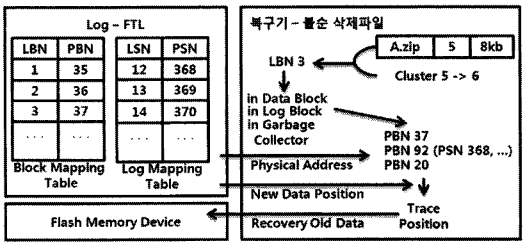

그림 9 불순 삭제파일의 복구

4. 실험 및 평가

본 장에서는 효율적인 플래시 메모리기반 복구기법으로 복구 대상 파일을 복구하는 실험을 보여 디스크기반 복구기법보다 우수함을 보인다. 비교대상은 디렉토리 스캐닝기법을 사용하는 복구 프로그램과 디렉토리 스캐닝기법과 섹터 스캐닝기법을 사용하는 상용프로그램 파이널데이터2.0이다.

실험 환경은 플래시 메모리 시뮬레이터인 Flasim[11]을 사용하는 환경이다. 가상 플래시 메모리는 총용량이 32MB에 파일시스템으로 FAT32를 사용하고 4KB 클러스터로 설정하였다. 실험 데이터는 총 50개의 파일로 음악파일, 사진파일, 문서파일이며 50개 파일의 총 크기는 32MB이다. 음악파일은 6개이고 파일크기는 365KB~2.6MB사이이며 사진파일은 20개이고 46KB~1MB사이이며 문서파일은 24개이고 1KB~1.2MB사이이다.

본 논문에서 제안하는 기법은 디스크기반 복구기법의 한계점을 해결하는 목표를 지닌다. 따라서 복구가 불가능한 불순 삭제파일을 검색하고 복구하는 두 가지 실험을 하였다. 검색실험은 복구 대상 파일의 검색 개수와 불순 삭제파일의 검색 개수를 측정하였다. 이것은 불순 삭제파일을 검출할 수 있는지 측정한다. 복구실험에서는 검색된 불순 삭제파일의 복구율을 측정하였다. 이것은 본 기법이 어느 정도의 성능을 보이는지 측정한다. 그리고 실험대상이 일반적으로 발생하지 않는 불순 삭제파일이기 때문에 발생비율을 임의적으로 조정하여 실험하였다. 삭제파일의 비율을 100%, 50%, 30%로, 불순 삭제파일의 비율을 100%, 50%, 30%, 0%로 변경하며 실험하였다.

그림 10과 같이 불순 삭제파일의 검출은 본 기법만이 가능하다. 따라서 기존 기법의 복구 대상 파일 검색이 되었으나 검출하지 못한 불순 삭제파일은 복구에 실패한다. 그림 11은 그림 10에서 검색된 불순 삭제파일을 본 기법으로 복구한 결과를 측정한 것이다. 삭제비율이 50%, 30%인 장치에서 복구율이 25%, 100%로 측정되었다. 따라서 기존 기법보다 본 기법이 불순 삭제파일의 복구에서 우수하다는 것을 보여준다. 하지만 그림 11과 특정 비율의 경우 불순 삭제파일의 복구를 실패하였다. 이 경우에 불순 삭제파일 크기의 총합이 8MB이상인 공통점을 발견하였다. 이것은 검색에 이용된 메타데이터의 일부가 병합되어 로그블록에 존재하지 않기 때문이다. 결국 여러 개의 이전 메타데이터가 가비지 컬렉션 영역에 존재할 수 있다. 본 논문에서는 이러한 경우에 복구 대상 파일의 연관성을 계산하기는 어려워 다루지 않았기 때문이다. 그러나 불순 삭제파일의 특성상 최근에 발생할 확률이 높기 때문에 그림 11의 특정 비율처럼 삭제파일의 대부분이 불순 삭제파일일 경우가 적다. 따라서 최근에 발생한 불순 삭제파일은 본 기법으로 복구할 가능성이 있다.

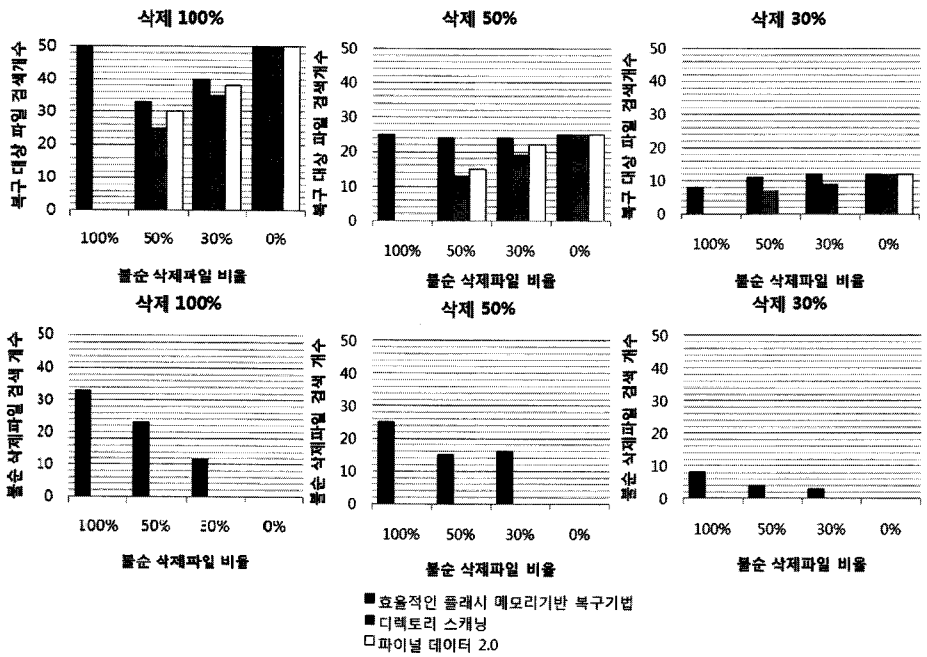

그림 10 삭제 비율 변화에 따른 검색 비율

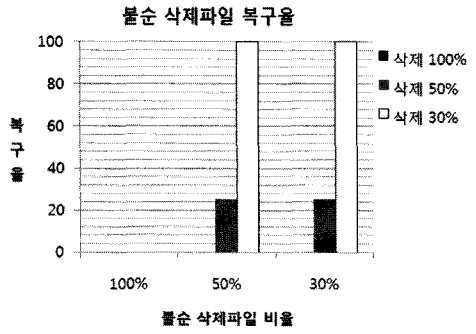

그림 11 불순 삭제파일의 복구 비율

5. 결론 및 향후계획

플래시 메모리의 수요가 늘어 플래시 메모리에서 디지털 포렌식 수사가 수행될 가능성이 높다. 하지만 현재의 데이터 복구기법은 디스크기반 복구기법을 사용하여 복구가 불가한 불순 삭제파일이 존재한다. 본 논문에서는 플래시 메모리상에서 불순 삭제파일을 검색, 복구하는 효율적인 플래시 메모리기반 복구기법을 제안했다. 디스크기반 복구기법과 다르게 본 기법은 BAST FTL을 이용하여 불순 삭제파일의 복구가 가능하다. 실험결과, 본 기법으로 불순 삭제파일은 이전 메타데이터가 로그블록에 존재하고 이전 데이터가 장치 내에 존재할 때 복구가 가능하다. 따라서 본 기법은 기존 기법보다 우수하여 디지털 포렌식 수사에 도움을 줄 것을 기대한다.

향후 연구계획으로는 본 논문에서 효율적인 플래시 메모리기반 복구기법의 한계점인 가비지 컬렉터 영역에 이전 메타데이터가 존재할 때에도 불순 삭제파일의 데이터를 검색할 수 있도록 이전 메타데이터와 이전 데이터의 연관성을 계산하는 역추적 알고리즘을 고려하고 있으며 로그블록의 활용률이 높은 FTL 기법에 효율적인 플래시 메모리기반 복구기법을 적용하는 연구계획이 있다. 또한 복구에 적합한 FTL 기법의 연구도 고려하고 있다.

참 고 문 헌

[1] Intel Corporation, "Understanding the flash Translation layer(FTL) specification," http://www.intel.com, 1998.

[2] J. Kim, J. M. Kim, S. H. Noh, S. L. Min, Y. Cho, "A Space-Efficient Flash Translation Layer for CompactFlash Systems," *IEEE Transactions on Consumer Electronics*, vol.48, no.2, pp.366-375, 2002.

[3] Samsung Electronics, "NAND Flash Spare Area Assignment Standard," http://www.samsung.com/, 2005.

[4] A. Ban, "Flash file system optimized for page-mode flash technologies", United States Patent. no. 5,937,42, 1999.

[5] T. Shinohara, "Flash memory card with block memory address arrangement," United States Patent, no. 5,905,993, 1999.

[6] A. Ban, "Flash File System," United States Patent. no.5,404,485, 1995.

[7] A. Kawaguchi, S. Nishioka, H. Motoda, "A Flash-Memory based File System," *Proceedings of 1995 USENIX Technical Conference*, pp.155-164, 1995.

[8] Microsoft Corporation, "FAT: General Overview of On-Disk Format," Version 1.02, May 5, 1999.

[9] S. L. Garfinkel, "Carving contiguous and fragmented files with fast object validation," Digital Investigation, 2007.

[10] G. G. Richard III, V Roussev, "Scalpel: A Frugal, High Performance File Carver," 2005 DFRWS Published by Citeseer, 2005.

[11] H. Y. Choe, S. H. Kim, S. W. Lee, S. W. Park. "FlaSim : A FTL Memory Emulator using Linux Kernel Modules," *Journal of KIISE : Computing Practices and Letters*, vol.15, no. 11, pp.836-840, Nov.2009. (in Korean)

신 명 섭

2008년 평생교육진흥원 멀티미디어학과(학사). 2010년 숭실대학교 컴퓨터학과(석사). 2010년~현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 플래시 메모리, 데이터베이스 등

박 동 주

1995년 서울대학교 컴퓨터공학과(학사) 1997년 서울대학교 컴퓨터공학과(석사) 2001년 서울대학교 전기전자컴퓨터공학부(박사). 2001년~2003년 삼성전자 책임연구원. 2004년~2005년 숭실대학교 컴퓨터학부 전임강사. 2006년~2009년 숭실대학교 컴퓨터학부 조교수. 2010년~현재 숭실대학교 컴퓨터학부 부교수. 관심분야는 데이터베이스, 멀티미디어 데이터베이스, 임베디드 데이터베이스 등