Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.4, Aug. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.4.835

시나리오 기반 이미지 개발을 통한 파일 카빙 도구 검증 방안 연구*

김 해 니,† 김 재 욱, 권 태 경‡
연세대학교 정보대학원 정보보호 연구실

A Study of Verification Methods for File Carving Tools by Scenario-Based Image Creation*

Haeni Kim,† Jaeuk Kim, Taekyoung Kwon‡
Information Security Lab, GSI, Yonsei University

요 약

파일 카빙(File Carving)은 저장 매체가 포맷되거나 파일시스템이 손상되어 메타데이터가 없는 파일 복구를 시도하는 기법으로 일반적으로 파일의 특정 헤더/푸터 시그니처 및 데이터 구조를 찾는다. 그러나 파일 카빙은 오랫동안 단편화 (Fragmentation) 된 파일을 복구해내는 문제점에 직면하고 있으며, 디지털포렌식에서 중요한 대상의 파일(doc, hwp, xls 등)은 비교적 단편화되기 쉬우므로 이에 대한 해결방안 제시는 매우 중요하다. 이와 같은 한계점을 극복하기 위하여 다양한 카빙 기법 및 도구들이 지속적으로 개발되고 있으며, 기능 검증을 위하여 다양한 연구 및 기관에서 데이터셋을 제공한다. 그러나, 기존에 제공된 데이터셋은 환경적인 조건이 상당히 제한되어 도구를 검증하는데 있어 비효율적이다. 본 논문에서는 단편화된 파일 카빙의 중요성을 언급하고, 카빙 도구 검증을 위한 시나리오 기반의 16가지의 이미지를 개발한다. 개발된 이미지는 상용 카빙 도구로 잘 알려진 Foremost를 통하여 매체 별로 카빙률 및 정확도를 계산하여 나타낸다.

ABSTRACT

File Carving is a technique for attempting to recover a file without metadata, such as a formated storage media or a damaged file system, and generally looks for a specific header / footer signature and data structure of the file. However, file carving is faced with the problem of recovering fragmented files for a long time, and it is very important to propose a solution for digital forensics because important files are relatively fragmented. To overcome these limitations, various carving techniques and tools are continuously being developed, and data sets from various researches and institutions are provided for functional verification. However, existing data sets are ineffective in verifying tools because of their limited environmental conditions. Therefore, this paper refers to the importance of fragmented file carving and develops 16 images for carving tool verification based on scenarios. The developed images' carving rate and accuracy of each media is shown through Foremost which is well known as a commercial carving tool.

Keywords: Forensics, File Carving, Tool Testing, Recovery

Received(06. 19. 2019), Modified(07. 15. 2019), Accepted(07. 15. 2019)

* 이 논문은 2019년도 정부 (과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No. 2017-0-00380, 차세대 인증 기술 개발)과 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 지원을 받아 수행된 연구임(IITP-2019-2016-0-00304)

† 주저자, gosl4576@naver.com
‡ 교신저자, taekyoung@yonsei.ac.kr(Corresponding author)

I. 서 론

디지털 기기는 개인에 대한 상당한 범위의 기록을 데이터로 남기고 있다. 따라서, 최근 소송에서는 디지털 증거에 비중을 맞추고 있어 디지털포렌식의 중요성이 확대되고 있다. 디지털 기기를 대상으로 하는 조사에서 중요한 점으로 삭제된 데이터를 얼마만큼 복구할 수 있는지가 쟁점이 되고, 이러한 요구사항에 맞춰 파일 복구를 위한 포렌식 하드웨어 및 소프트웨어 도구들이 등장하였다. 파일 복구를 위한 기법으로는 일반적으로 메타데이터 기반 파일 복구와 카빙 기반 파일 복구로 구분한다. 카빙 기반 파일 복구는 파일이 삭제되거나 파일시스템이 포맷된 경우에 파일 메타정보들이 다른 데이터로 변경되거나 사라져 해당 파일에 접근할 수 없는 경우에 사용되는 복구 방법으로 증거인멸의 흔적을 찾아낼 수 있으므로 디지털포렌식에서는 필수 요소이다. 그러나, 파일 카빙은 저장매체의 기능적 혹은 구조적 차이에 따라서 성능의 차이가 발생할 수 있으며, 일반적인 PC 사용자들은 문서작업을 위하여 반복적인 파일 쓰기 작업을 수행하기 때문에 자연스럽게 파일들이 저장 매체로부터 단편화 (Fragmentation) 되어 저장되는 경우가 발생한다. 이처럼 도구들의 파일 카빙 기능은 다양한 경우에서의 차이가 발생할 수 있으므로 여러 가지 시나리오로부터 현실적인 상황이 반영된 환경에서의 도구 검증이 필요하며, 검증 결과에 따라 도구들의 한계점을 보완할 필요가 있다. 하지만 기존 연구 및 프로젝트에서는 다양한 상황이 반영된 파일 카빙 검증을 위한 이미지가 제공되고 있지 않으므로 도구에 대한 신뢰성을 검증하는 것이 어렵다.

본 논문에서는 HDD (Hard Disk) 및 SSD (Solid State Disk), MBR (Master Boot Record) 및 GPT (GUID Partition Table) 개발 환경과 파일이 단편화되지 않은 디스크(S1), S1에서 포맷된 디스크(S2), 파일이 단편화되어 저장된 디스크(S3) 및 S3에서 포맷된 디스크(S4)에 대한 4가지 시나리오가 반영된 16가지의 현실적인 파일 카빙 도구 검증을 위한 이미지를 제시한다. 2장에서는 카빙의 중요성과 기존의 도구 검증 이미지를 제공하는 프로젝트 및 연구를 소개하고, 이에 대한 한계점을 제시한다. 3장에서는 저장매체에 따른 카빙 도구에 미치는 영향력에 대하여 기술하며, 4장에서는 파일 단편화 및 삭제에 관한 시나리오에 대하여 기술한다. 그리고 5장에서는 제시된 카빙 검증 이미지를 이용하여 Foremost 도구를 검증한 결과를 기술한다.

II. 관련 연구 및 배경 지식

2.1 파일 카빙 도구 검증의 필요성

디지털 기기를 조사하는데 있어 삭제된 파일을 복구해내는 것은 중요한 과정 중 하나이다. 일반적으로 단순히 파일이 삭제되는 경우 메타데이터 영역이 삭제되는 것이 아닌 특정 플래그 값이 변경되고 파일에 대한 이름, 크기, 할당 위치 등의 정보가 유지되므로 실제 데이터의 위치를 알 수 있어 삭제된 파일을 원래 상태로 복구하는 것이 가능하다[1]. 그러나, 파일시스템이 존재하지 않거나 손상이 난 경우 혹은 더 많게는 저장 매체가 포맷되거나 메타데이터가 다른 값으로 덮어 씌워진 경우에 실제 파일 데이터의 위치를 알 수 없으므로 파일을 복구해내는 것이 어려워 파일시스템 메타데이터 기반의 복구 기법에는 한계가 존재한다. 이러한 한계를 해결하기 위하여 활용되는 기법인 카빙은 파일 형식에 대한 헤더 (Header)와 푸터 (Footer)의 데이터베이스를 사용하여 디스크 이미지의 파일시스템에 상관없이(오프셋/섹터의 참조 없이) 파일을 검색하는 것도 가능하다. 따라서 파일 카빙의 경우 파일시스템 구조를 이용하거나 혹은 파일시스템의 메타데이터가 파괴되더라도 파일을 복구하는 것이 가능하기 때문에 비교적 많은 양의 삭제된 파일을 획득할 수 있어 피 압수자의 증거 인멸의 흔적을 찾아내는 것이 효과적이다. 이러한 요구사항에 맞춰 다양한 기법의 파일 카빙 소프트웨어 도구들이 등장할 뿐만 아니라 기존 컴퓨터 포렌식 도구에 파일 카빙 기능이 임베딩되었다. 외관적인 기능 면으로만 봐서는 자칭 파일 카빙 도구들이 난립하였으나 도구 기능에 대한 신뢰성이 문제가 되면서 도구 기능에 대한 검증이 필요하게 되었다. 필요성에 따라 파일 카빙 도구 검증에 관한 다양한 연구가 진행되었으며, 테스트를 위한 이미지를 제공하고 있다.

2.2 관련 연구

미국의 국가표준기술연구소 (NIST, National Institute of Standards and Technology)의 CFTT (Computer Forensics Tool Testing) 프로젝트[2]는 디지털 포렌식에 사용되는 도구들의 요구사항을 정의하고 각 도구를 검증하는 방법 및 절차

를 수립하여 CFReDS (Computer Forensic Reference Data Sets) 프로젝트를 통하여 테스트 환경을 구축하였다. 파일 카빙 테스트를 위해서는 그래픽, 문서, 압축, 오디오, 비디오 파일과 6개의 서로 다른 수준의 단편화 시나리오를 제공한다. 속성별 파일이 한 개씩으로 구성되어 각 이미지의 총 데이터셋의 개수가 10개 이하이고, 이미지 크기 또한 최대 50MB로 이미지의 크기가 작다. 또한, 생성된 이미지는 인위적으로 만들어진 것으로 파일시스템 구조를 가지지 않으며, 이는 도구 검증에 있어 신뢰성이 떨어질 수 있다.

DFTTI (Digital Forensics Tool Testing Images)[3]는 상기 CFTT의 공공 기관에서 이루어지는 도구 검증 기법 연구와 민간단체에서 수행되는 검증 기법 연구와의 격차를 줄이기 위하여 2003년에 수행된 프로젝트이다. 파일 카빙 도구 검증을 위하여 FAT32와 EXT2 두 개의 파일시스템을 대상으로 하며, USB 플래시 드라이브를 mkfs.vfat, mkfs.ext2를 통하여 포맷한 이미지를 제공한다. 테스트 파일의 수는 15개 이하로 구성되어 도구를 검증하고 평가하기에 데이터셋이 매우 적다.

RDC (Real Data Corpus)[4]는 일반 사용자들이 사용한 저장 장치로부터 데이터를 삭제 혹은 삭제하지 않고 폐기한 기기들을 전 세계 2차 시장에서 구입하여, 데이터를 추출함으로써 실제로 발견되는 데이터를 밀접하게 모방하는 데이터 세트를 생성하고 제공한다. NPS 테스트 디스크 이미지는 컴퓨터 포렌식 도구를 테스트하기 위해 만들어진 디스크 이미지 셋으로 공개적으로 제공된다. 그 중 기본 파일 및 파편화된 파일에 대한 카빙을 검증할 수 있는 Canon 디지털카메라로 촬영한 이미지 데이터 세트에서는 일부는 단편화되었지만 파일이 JPG로 제한되어 있다.

Laurenson 및 Thomas[5]는 DFTTI의 Basic Data Carving Test #1 (11-carve-fat.dd)[6], DFRWS2006 Forensics Challenge 데이터셋 (dfrws-2006-challenge.img)[7] 그리고 직접 개발한 Baseline Carving 데이터셋 (bcds.raw)[8]을 활용하여 EnCase, FTK, WinHex, PhotoRec, Scalpel, Foremost 도구에 대한 카빙 성능을 비교·검증하였다.

Simson Garfinkel[9]는 2차 시장에서 획득한 드라이브를 분석하여 많은 단편화되어 저장된 파일을 발견하였으며, 단편화된 파일 복구에 대한 문제점을 제시하며 이를 해결하기 위한 카빙 알고리즘을 적용한 도구를 제시한다.

Table 1. Differences from proposed scenarios

Related work	Proposed scenario	Difference
CFTT	- Up to 50MB Artificial images (Non Filesystem) - Archive, Audio, Document, Graphic, Video files (Total 31) - Non fragmented files - Sequential fragmentation - Non sequential fragmentation - Missing fragments - Nested Files - Braided files	- 100MB Natural OS image (Windows 10) - NTFS Filesystem - Hard Disk Drive & Solid State Drive - MBR & GPT - Archive, Audio, Document, Graphic, Video, Email files (Total 241) - USB booting format - Files are not fragmented and the disk is not formatted - The disk is formatted - Files are fragmented without disk format - The disk is formatted after files are fragmented
DFTTI	- 32MB USB Flash Drive images - FAT32 & EXT2 - Audio, Document, Graphic, Video file (Total 25) - mkfs.vfat & mkfs.ext2 - Non fragmented files - Sequential fragmentation	
RDC	- 32MB SD card image (Flash memory card) - NTFS - Only JPG files (Total 51) - Non fragmented files - Sequential fragmentation - Non sequential fragmentation	

기존 연구와 같이 파일 카빙 도구 검증을 위한 다양한 이미지 개발 연구가 꾸준히 수행되고 있는 것처럼, 파일 카빙의 중요성과 이를 수행하는 도구에 대한 신뢰성을 검증하는 것이 매우 중요하다. 카빙 도구는 작은 성능 향상이 파일 복구에 직접적인 영향을 주기 때문에 데이터셋이 성능의 개선을 탐지할 수 있도록 크게 만들어야 하며, 도구를 정량적으로 분석하기 위하여 데이터셋은 변화를 감지할 수 있을 정도로 커야 한다. 또한, 이러한 검증은 주로 실험적으로 평가되어야 하지만 기존 연구로부터 개발된 제한적인 이미지로부터 테스트된 도구는 좋은 검증 결과를 나타내더라도 실제 조사에 투입되는 경우 SSD 사용, 파일의 다양화 및 단편화 등과 같은 더 복잡한 경우에 맞닥뜨리게 되며 도구의 성능 결과가 달라질 수 있어 기능성이 불분명하다. Table 1은 본 논문에서

제시하는 시나리오와 기존에 제시된 시나리오와의 차이점의 개요를 나타낸다.

2.3 파일 카빙 도구

다양한 기법에 따라 카빙 기능이 삽입된 포렌식 도구 또는 오직 파일 카빙의 기능을 제공하는 도구들이 다양하게 존재한다. 이러한 도구들은 제공하는 파일 유형 또는 기법이 서로 다르기 때문에 같은 환경에서의 도구별 파일 카빙 결과는 상이하게 나타난다. 잘 알려진 오픈소스 파일 카빙 도구로는 Foremost, Scalpel, Bulk extractor, Photorec 등이 존재한다. Bulk extractor은 zip, exif 파일 카빙이 가능하지만 대부분 문자열을 카빙하는데 활용되며 Photorec의 경우 그래픽 파일을 카빙하는 목적으로 사용된다. Foremost와 Scalpel은 문서, 오디오, 비디오, 압축, 그래픽 등의 다양한 파일을 지원하는 파일 카빙 도구이다. 본 논문에서는 다양한 파일 유형의 데이터셋으로 구성된 환경에서의 도구 검증을 위하여 Foremost v1.5.7[10] 도구를 사용하였다. 해당 도구는 데이터 복구를 위한 Linux 플랫폼 오픈소스 파일 카빙 도구로 널리 알려져 있으며 초기에 사용되어 헤더, 푸터 및 데이터 구조를 사용하여 파일을 복구할 수 있다. dd, Safeback, Encase 등으로 생성한 이미지 파일 혹은 드라이브에 직접 이미지 파일을 카빙할 수 있으며, 헤더, 푸터 및 데이터 구조는 configuration (config) 파일로부터 카빙 대상의 파일들을 지정하므로 복구하려는 특정 파일의 헤더 및 푸터의 정보를 입력함으로써 기존에 지원하지 않는 파일을 속성을 입력하여 선택할 수 있다. Foremost는 Scalpel[11]이라는 카빙 도구 개발에도 사용되어 졌으며, Scalpel은 현재 Foremost와의 코드를 공유하지만 불필요한 memory-to-memory 복사와 디스크 I/O을 줄일 수 있는 최적화된 방식 사용하여 성능이 비교적 우수한 것으로 연구되었다[12]. 하지만 Scalpel은 기본적으로 설정되어있는 파일 카빙을 위한 헤더/푸터 시그니처 및 데이터 구조가 Foremost보다 매우 다양하여 카빙할 수 있는 파일의 개수가 불필요하게 많아지기 때문에 하나의 시나리오로부터 4TB 이상의 파일이 복사되어 본 연구에서 실험 환경적인 한계가 존재한다. 따라서 Foremost를 통해 개발된 이미지로부터 검증한 결과를 제시한다.

III. 이미지 개발 환경

3.1 디스크 구조적 차이(MBR vs GPT)

마이크로소프트 윈도우 운영체제에서 데이터를 저장하기 위하여 디스크 드라이브를 사용 가능한 영역으로 분할하는 2가지의 아키텍처를 제공한다. BIOS (Basic Input/Output System) 방식의 시스템에서 사용되는 MBR과 EFI (Extensible Firmware Interface) 방식 시스템에서 사용되는 GPT 두 가지 방식의 아키텍처의 접근법의 차이는 논리적인 블록 번호에 대한 물리적 디스크 섹터의 맵핑을 추적하는 방법에 따라 달라 각 방식은 디스크를 관리하는데 구조적인 차이를 보인다. MBR 디스크 방식은 3개의 기본 파티션과 1개의 확장 파티션 생성이 가능하며, 확장 파티션은 여러개의 논리 파티션으로 나눌 수 있다. 하지만 디스크에서 인식할 수 있는 주소의 개수가 2^{32}개로 최대 크기가 2TB이다. 이러한 MBR 디스크 방식의 구조적인 한계로 인하여 새로운 디스크 형식의 GPT에서는 디스크 당 최대 128개의 파티션 생성이 가능하게 되었으며, 볼륨의 최대 용량이 18EB(Exabyte)로 확장되었다[13]. 또한 해당 사양의 하위 집합에는 DOS/MBR 파티션 테이블을 대체하기 위한 GUID (Globally Unique Identification) 파티션 테이블 또는 GPT 헤더가 포함되며 디스크 마지막 공간에 백업 데이터를 포함하고 있으므로 파일이 삭제된 경우 이를 활용하면 MBR 방식보다 파일 복구를 수행하는데 용이할 수 있다. 따라서, 파일 카빙 수행 이전에 디스크 인식 방식이 GPT로 파악된 경우 백업 데이터 분석을 선행적으로 하는 것이 유용할 수 있으므로 MBR과 GPT 두 가지 다른 구조의 디스크 인식 방식으로부터의 도구의 분석 절차가 달라져야 한다. 이는 카빙 도구에서도 마찬가지로 해당 복구 데이터 영역을 활용한다면 좀 더 정확하고 많은 파일을 카빙해낼 수 있을 것으로 보인다. 따라서, 이러한 디스크의 구조적 차이에서의 파일 카빙 도구 검증을 위하여 해당 시나리오를 포함한다.

3.2 디스크 하드웨어 차이(HDD vs SSD)

최근 시스템 드라이브로 HDD와 SSD 모두 많이 사용된다. HDD는 read/write 헤드가 자성기판을 통과함에 따라 데이터 비트는 0 또는 1으로 정렬된

다. 이러한 데이터 비트 집합들은 함께 바이트를 형성하고 일반적으로 섹터(보통 512Byte)로 그룹화한다. 그러나 SSD는 HDD와 달리 자기적이 아닌 전기적으로 데이터가 기록되며 삭제 명령이 주어질 때 데이터를 덮어쓰는 것이 아닌 TRIM 기능을 통해 데이터를 완전하게 삭제하고 빈 블록에 데이터를 기록하는 과정을 수행하기 때문에 삭제된 파일을 복구하기가 어렵다. SSD의 Wear-leveling 기능은 프로그램-삭제 (P/E Cycles) 횟수가 제한되어 있어 컨트롤러가 전체 블록에 대해서 P/E cycle이 골고루 분산되도록 쓰기를 실행한다. 예컨대, SSD는 크기가 서로 다른 페이지에 데이터를 저장한다. 그런 다음 이 페이지는 삭제 블록으로 그룹화되며, 물리적 주소를 기반으로 함께 영역화된다. 데이터는 순차적으로 페이지에 기록되지 않으며 삭제 블록에 걸쳐 스트라이프 (Stripe) 처리되어 Wear-leveling 컨트롤러에 의하여 관리된다. 디스크에 저장된 데이터가 수정되면 Wear-leveling 컨트롤러는 전체 블록을 새로운 위치로 이동시킨 후 원래 블록은 삭제하도록 예약하며 HDD와 다르게 덮어쓰기가 될 수 없다. 즉, SSD 사용자는 데이터가 작성되는 위치를 제어할 수 없으며 더욱 복잡한 방식으로 데이터가 단편화된다. 이처럼 기본적으로 쓰기가 가장 적은 블록에 데이터가 저장되는 등의 저장되는 패턴이 없는 SSD에서는 HDD와 비교하여 데이터를 복구하는데 어려울 수 있다[14].

현실 세계에서는 이처럼 구조적 혹은 기능적으로 차이를 보이는 저장 매체를 사용하기 때문에 다양한 환경에서의 카빙 기능에 대한 신뢰성을 검증할 필요가 있다.

IV. 이미지 개발 시나리오

파일 카빙 테스트를 위한 시험용 전체 이미지는 HDD와 SSD 그리고 MBR과 GPT를 사용하는 네 가지 저장 매체 환경에서 발생 가능한 다양한 경우를 반영한다. PC 환경은 Windows 10 운영체제와 100GB 크기의 NTFS 파일시스템을 동일하게 구축하였다. Windows 10의 경우 Trim 기능이 기본적으로 활성화되어 있으며, SSD에서 파일이 삭제될 때마다 Trim 명령을 처리한다. 따라서, SSD를 사용한 시나리오의 경우 Trim이 작동한 후이다. 또한, 4가지의 저장 매체로부터 데이터셋이 단편화되지 않고 저장된 경우(S1-선형적으로 저장된 파일 카빙 성능을 측정하기 위함), S1의 환경에서 저장 매체가 포맷된 경우(S2-삭제된 파일 카빙 성능을 측정하기 위함), 데이터셋이 단편화되어 저장 매체에 저장된 경우(S3-비선형적으로 저장된 파일 카빙 성능 측정을 위함), 그리고 S3의 환경에서 저장 매체가 포맷된 경우(S4-비선형적으로 저장된 후 삭제된 파일 카빙 성능 측정을 위함) 네 가지 시나리오로 구성하였다. S1와 S3의 경우에는 저장매체가 포맷되지 않았으므로 파일 카빙 도구가 파일시스템의 구조를 이용하여 데이터를 복사 및 복구하는 경우에서의 목적으로 활용이 가능하다.

Table 2. Scenario for File carving validation

Index	Scenario & Object
S1	Files are not fragmented and the disk is not formatted - To measure the performance of carving files stored linearly - To measure the file carving performance through file system
S2	The disk is formatted in S1 - To measure the performance of carving deleted files
S3	Files are fragmented without disk format - To measure the performance of carving files stored nonlinearly - To measure the file carving performance through file system
S4	The disk is formatted in S3 - To measure the performance of carving deleted files after being stored nonlinearly
MBR	To measure the performance of carving files in the master boot record structure
GPT	To measure whether the tool capable of GPT recognition and the ability to carve files in a GUID Partition Table structure.
HDD	To measure of performance of caving fragmented and deleted files in a relatively simple way
SSD	To measure the performance of carving fragmented files in a relatively complex way and deleted files with Trim

Table 2는 이미지 개발을 위한 네 가지 시나리오의 설명과 목적 그리고 3.1과 3.2 섹션에서 제시한 MBR, GPT, HDD, SSD의 목적에 관한 개요를 나타낸다. 결과적으로 4가지 저장매체와 4가지 시나리오로부터 카빙 도구 검증을 위한 16가지의 이미지가 생성되며, 하위 섹션에서는 단편화된 파일 카빙의 중요성과 이미지 개발을 위하여 실행한 파일 단편화 방법 및 결과를 설명한다.

4.1 파일 단편화

디지털포렌식에서 단편화된 파일을 카빙하는 것은

중요한 부분으로 자리 잡고 있다. 이는 최근 포렌식 조사에서 중요한 파일들이 다른 유형의 파일보다 단편화되기 쉽기 때문이다. 일반적인 운영체제인 Windows에서는 파일을 저장할 때 단편화되지 않도록 파일시스템으로부터 파일이 할당되는 공간을 찾으려 하지만 다음과 같은 경우에서는 파일이 두 개 이상으로 비 인접 파일 또는 분할된 파일로 저장될 수 있다.

- 저장 매체를 오랜기간 동안 사용하면서 용량이 거의 가득 차 있는 상태에서 여러 개의 파일을 추가 및 삭제하는 경우
- 기존에 저장된 파일에서 데이터를 추가한 경우에 파일 끝에 데이터가 저장될 섹터 공간이 부족한 경우
- 파일시스템 자체에서 연속적인 방법으로 특정 크기의 파일을 쓰기는 지원하지 않는 경우
- 다수의 프로세스가 번갈아가며 동기적 쓰기 작업을 수행하는 경우

이처럼 파일시스템 사용이 오래되거나 파일이 생성, 수정되고 삭제됨에 따라서 단편화 현상이 증가하며 큰 용량의 파일은 더 많이 단편화된다. 일상적으로 사람들은 업무로부터 문서작업을 위하여 다운로드한 파일로부터 편집하여 다른 이름으로 저장하거나, 혹은 작성 중인 파일을 여러 번 편집하여 덮어쓰기를 하여 저장한다. 대용량 파일의 경우 대부분이 고해상도의 스틸 이미지 혹은 오디오/비디오 파일이며, 이러한 파일들이 단편화되는 것은 드문 일이 아니나, 이미지나 영상 편집과 관련된 직업군의 사람들은 편집 프로그램을 통해 원본 파일을 편집하여 저장하는데 이러한 경우에 자연스럽게 파일들이 저장 매체에 단편화되어 저장될 수 있다. 또한, 단편화되어 저장된 파일은 선형방식 또는 비선형방식 두 가지 범주로 나눌 수 있다. 선형 단편화는 파일이 두 개 이상의 조각들로 분할되어 저장되었지만 조각들이 순서대로 데이터 세트에 있는 경우이며, 비선형 단편화는 원래의 파일과 다른 순서의 데이터셋으로 존재하는 경우이다.

4.1.1 파일 단편화 실험 방법

현실적인 환경을 반영하기 위하여 인위적으로 파일을 단편화시켜주는 도구를 사용하지 않았으며, 파일이 두 개 이상으로 단편화되는 경우 중 쉽게 단편화가 수행되는 경우인 기존에 저장된 샘플 파일로부터 큰 용량의 데이터를 삽입하는 과정을 선택하였다. 기존에 저장된 샘플 파일의 유형과 개수는 Table 3에 해당되며 유형별 파일들은 2KB ~ 55MB 사이의 크기로 다양하게 구성된다. 저장된 샘플 파일을 단편화시키기 위하여 해당 파일로부터 대용량 데이터를 삽입한 후 파일을 덮어쓰기로 저장 혹은 다른 이름으로 저장하는 방법을 수행한다. PDF를 제외한 문서 파일의 경우 텍스트, 비디오, 오디오, 그래픽, PDF 파일을 삽입하고, PDF 파일은 다른 PDF 파일과 merge 기능을 통해 파일을 덮어쓴다. 오디오 파일과 비디오 파일의 경우에는 편집 프로그램을 통해 다른 오디오 및 비디오 파일을 merge 하거나 줄이는 편집을 수행한 후 덮어쓰기 및 다른 이름으로 저장한다. 그래픽 파일의 경우 또한 편집 프로그램을 통하여 사진의 크기를 늘리고, 다양한 효과를 편집함으로써 파일을 덮어쓴다. 마지막으로 압축 파일의 경우 동영상, 오디오, 이미지, 압축 파일, PDF를 삽입시키고 파일을 덮어쓴다.

Table 3. Sample File type for Carving Verification

File	Extension	Number
Document	doc	10
	docx	10
	ppt	12
	pptx	11
	xls	10
	xlsx	10
	hwp	11
	pdf	10
Audio	mp3	13
	wav	10
Video	avi	11
	mp4	14
Graphic	bmp	10
	gif	10
	jpg	11
	png	10
Archive	7z	24
	rar	14
	zip	30
E-mail	eml	10
Total		**241**

4.1.2 파일 단편화 실험 결과

각 구성된 PC 환경에서 위와 같은 실험 방법으로 총 241개의 파일 중 파일 단편화를 위한 시나리오 수행을 완료한 결과 파일의 데이터 런리스트 (Runlist)를 통해 단편화된 파일의 수를 확인하였다. 파일

Table 4. Fragmentation results of files

# Fragments	HDD MBR	SSD MBR	HDD GPT	SSD GPT
	Number of files			
(None)	124	97	113	104
2	26	15	24	17
3	20	20	25	25
4	19	23	30	27
5 ~ 10	58	77	38	77
11 ~ 20	20	29	23	24
21 ~ 100	19	34	31	6
101 ~ 1000	14	5	15	20
1001 ~	0	0	1	0
Total Files				**300**

속성 내용이 MFT 엔트리의 크기보다 커 별도의 클러스터를 할당받아 저장하는 방식인 Non-resident 속성의 파일은 여유 공간이 없는 경우 대부분 비연속적으로 할당된다[15]. 이렇게 비연속적으로 할당된 클러스터들은 효과적으로 관리하기 위하여 클러스터 런이라고 하며 클러스터 런을 표현하는 것을 런리스트라고 한다. Table 4는 각 파일로부터 단편화된 파일의 수를 나타낸다. 많은 파일이 두 개 이상의 조각들로 단편화되어 저장되었음을 확인할 수 있으며, SSD가 HDD보다 파일의 단편화가 많이 이루어졌다.

4.2 파일 삭제

디지털 기기 조사 시에 파일이 삭제되는 경우는 대다수이며 삭제된 파일에 대한 복구는 포렌식에서 중요한 역할이다. 저장 매체로부터 데이터를 삭제하는데 있어서 소프트웨어 기반 삭제, 하드웨어 기반 삭제 (Degaussing), 물리적 기반 파괴로 분류할 수 있다[16]. 소프트웨어 기반 삭제는 HDD를 대상으로 개발된 방법으로 일반적으로 디스크의 각 섹터에 순차적 방식으로 특정 데이터 패턴을 기록하고 원래 데이터를 덮어쓴 후 복구할 수 없도록 한다.

하지만 SSD의 경우 Wear-leveling 컨트롤러에 의하여 제어되므로 소프트웨어가 데이터가 기록되는 특정 영역을 제어할 수 없어 아직 SSD의 데이터를 손상시키는데 적합한 솔루션은 아니나 데이터 삭제를 위하여 일반적으로 가장 많이 사용되는 방법이다. 소프트웨어 기반의 삭제의 대안으로 하드웨어 기반 삭제는 Degausser로 미디어를 통해 자기 펄스(Magnetic Pulse)를 보내는 방식으로 작용한다. 대부분의 경우 HDD를 작동 불능으로 만드는 빠른 방법이지만 SSD의 경우 데이터가 자력으로 작성되는 것이 아닌 전자적으로 저장되기 때문에 효과적인 방법이 아니다. HDD 및 SSD 드라이브에서 데이터를 삭제하는 가장 좋은 방법은 물리적 기반 파괴로 일반적으로 단일 칩을 매우 작은 조각으로 파쇄하는 과정이다. 우리는 디지털 조사과정에서 데이터 삭제를 위하여 일반적으로 발생 되는 USB 부팅 포맷을 통하여 저장매체의 데이터셋 파일 삭제를 수행하였다.

Table 5. Quality of the carving results

DataSet / Recovered	Yes	No
Yes	Positive	False positive
	Known False positive	
No	Supported False negative	-
	Unsupported False negative	

V. 파일 카빙 도구 검증

5.1 파일 카빙 결과

파일 카빙 도구를 검증하는데 있어 샘플 파일 집합에 대한 레이아웃에 대한 정보를 수집하였다. 이 정보에는 모든 파일 목록, 샘플 파일의 크기, 블록 범위 및 MD5로 구성된다. MD5는 암호 해시 함수를 사용하여 계산된 32자의 16진수를 나타내어, 특정 파일을 고유하게 식별하는데 사용할 수 있다. 도구로부터 파일을 카빙하는데 있어 결과는 크게 세 가지 유형으로 이어질 수 있다. Table 5는 세 가지 유형의 결과를 나타낸다. 카빙 결과에서 샘플 파일 집합의 MD5와 일치하는 경우는 올바르게 카빙된 파일로 판단되어 Positive를 의미한다. 하지만 MD5가 일치하지 않는 항목이 없는 경우 False positive라고 의미하지 않는다. 대부분 파일 형식은 시작과 끝 부분에 일정한 공간을 가지고 있다. 예를 들어, html 파일은 정확한 결과를 위해서는 카빙될 필요가 없는 비어있는 새로운 데이터와 함께 카빙되어 다른 MD5 값이 생성된다. 하지만 블록 범위가 동일하면 카빙 결과를 Positive라고 할 수 있으며 이러한 결과를 Known false positive라고 한다. 따라서, 우리는 파일 유사도 측정 도구인 ssdeep을 사용하여 파일 간의 유사도가 99% 이상의 결과를 나

Table 6. Quantification for File carving Verification

Index	Content
Rel (Relevance)	Dataset (Sample file)
Ret (Return)	The number of files carved by the tool(including Sample files)
RnR (Return & Relevance)	The number of files carved from the tool that correspond to the sample file.
Accuracy	Percentage of data in datasets (samples) that the tool has recovered successfully.
Carving Rate	Percentage of the sample file that is recovered from the tool

타내는 경우를 Positive와 동일하게 판단한다. 또한, Positive와 Known false positive 외의 결과를 False negative라고 할 수 있다. 즉, 샘플파일 집합에 포함되나 카빙이 되지 않은 경우를 False negative로 나타낸다. 또한 False negative는 도구가 샘플 파일을 지원하지 않아 카빙되지 않은 경우와 지원은 하지만 제대로 카빙 처리를 할 수 없는 경우로 나뉠 수 있다. 이를 각각 Supported false negatives와 Unsupported false negatives라고 한다. 도구 결과가 Unsupported false negatives보다 Supported false negative가 신뢰성을 떨어뜨리는 것이므로 나쁜 것으로 판단하여야한다[17].

파일 카빙 결과를 수치화하고 기능성의 기준을 결정하기 위하여 데이터셋(샘플파일)를 Rel (Relevance의 약자)로 표현하고, 도구로부터 복구된 파일의 개수(Positive, Known false positive)를 Ret (Return의 약자)로 표현하였다. 그리고 도구로부터 복구된 파일 중 샘플파일에 해당되는 파일의 개수를 RnR (Return & Relevance)로 정하였다. 이와 같은 수치로부터 도구의 성능을 결정하기 위하여 도구의 정확도(Accuracy)를 $RnR/Ret*100$로 계산하여 데이터셋(샘플파일) 중 도구에서 카빙에 성공한 데이터의 비율을 나타낸다. 카빙률 (Carving rate)은 $RnR/Rel*100$로 계산하여 도구에서 카빙한 데이터 중 데이터셋(샘플파일)의 비율을 나타낸다. Table 6은 파일 카빙 검증을 위한 수치화에 관한 개요를 나타낸다. 파일 단편화 시나리오를 통하여 생성된 매체의 경우 Table 3의 샘플파일 외에도 다른 이름으로 저장된 파일, 압축 파일 내에 포함된 파일 및 단편화를 위하여 삽입되거나 합병된 대용량 파일의 개수를 포함하여 결과를 계산한다.

5.1.1 카빙률

Figure 1과 2는 각각 MBR HDD, MBR SSD, GPT HDD, GPT SSD 저장 매체에서의 시나리오별 Foremost 도구의 카빙률과 정확도이다. 카빙률은 정확도보다 비교적 높은 결과를 나타내는 것을 알 수 있으나, 전체적으로 상당히 낮은 결과를 나타낸다. 우리는 Foremost의 configuration 파일로부터 카빙할 대상의 파일 정보를 직접 지정하지 않고 기존에 지원하는 방식을 사용하였다. 따라서, 카빙 결과가 낮게 나온 이유는 다음과 같다.

- mp4, eml 파일 시그니처를 지원하지 않음
- xls(x), ppt(x), doc(x), mp3, wav, avi, bmp, png 파일의 경우 헤더 시그니처는 존재하지만 푸터 시그니처가 존재하지 않음. 해당 도구에서는 이런 경우 임의의 최대 파일의 크기를 지정하여 정확하게 파일을 복구하는 것이 불가능함
- 샘플 파일의 크기가 임의로 지정된 최대 파일의 크기보다 큰 경우가 존재함

데이터셋이 단편화되지 않고 디스크가 포맷되지 않은 시나리오(S1)에서는 파일이 선형적으로 저장된 저장매체에서의 파일 카빙 성능을 측정하기 위함이다. 카빙률은 25.2%로 GPT SSD 저장 매체에서의 결과가 가장 좋았으며 차례대로 MBR SSD, GPT HDD, MBR HDD로 높은 성능을 나타낸다. 특히, HDD보다 SSD에서 그리고 MBR 보다 GPT에서 비교적으로 좋은 것으로 나타난다. 전반적으로 선형적으로 파일이 저장된 저장 매체로부터의 카빙 성능은 다른 시나리오에서보다 결과가 좋았으며 이는 해당 도구가 파일시스템을 무시하고 저장 매체의 첫 오프셋부터 데이터를 차례대로 읽어 파일의 헤더 및 푸터를 이용하여 복구해내는 기법이므로 선형적으로 저장된 파일을 복구해내는데 효과적이기 때문인 것으로 보인다.

S1의 저장매체가 포맷된 시나리오(S2)는 파일이 삭제된 경우 파일 카빙 성능을 측정하기 위함으로 GPT HDD에서 0.5%로 낮은 결과가 나타난다. 게다가 기타 세 가지 저장 매체에서는 모두 0%로 샘플 파일을 카빙하는데 실패하였다. 4.1에서 언급되었듯이 파일이 삭제될 때, 파일이 특정 영역에 단편화되어 기록되거나, 데이터가 지워질 수 있으므로 카빙

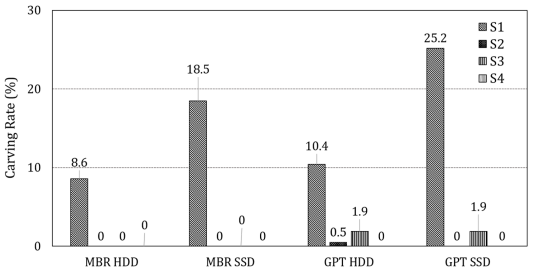

Fig. 1. Carving Rate of Foremost (Range Zomed:30%)

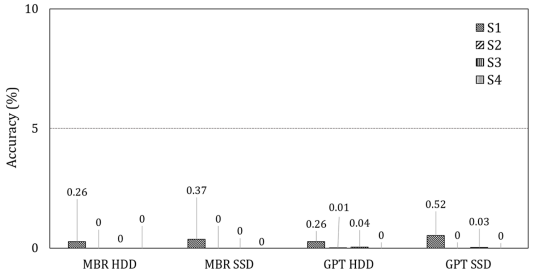

Fig. 2. Accuracy of Foremost (Range Zomed:10%)

성능이 낮아질 수 있다. S2와 같이 저장매체가 포맷되는 경우에는 Foremost로부터의 파일 카빙이 어렵지만 HDD가 Trim 동작을 수행하는 SSD보다 파일 카빙 성능이 좋을 것으로 보여진다.

데이터셋이 단편화되어 저장되고 디스크가 포맷되지 않은 시나리오(S3)는 비선형적으로 저장된 파일 카빙 성능 측정하기 위함으로 카빙률은 MBR 구조에서는 모두 0%의 결과가 나왔으며 GPT 구조에서 HDD와 SSD가 1.9%로 동일하였다. 그러나 해당 저장 매체로부터 복구된 파일들은 모두 단편화되지 않은 파일로 사실상 단편화되어 저장된 파일을 대상으로 하는 도구의 카빙률은 모두 0%로 샘플파일을 복구할 수 없는 것을 알 수 있다.

마지막으로 데이터셋이 단편화되고 디스크가 포맷된 경우(S4)에서는 파일이 단편화되어 저장된 후 삭제된 저장 매체로써 모든 저장 매체에서의 카빙률은 0%로 샘플파일 카빙에 실패하였다.

5.1.2 정확도

Figure 2는 각각 MBR HDD, MBR SSD, GPT HDD, GPT SSD 저장 매체에서의 시나리오별 Foremost 도구의 정확도로 전반적으로 모두 10% 이하의 결과로 정확도가 매우 낮다.

먼저, 데이터셋이 단편화되지 않고 디스크가 포맷되지 않은 시나리오(S1), 즉 저장 매체의 파일이 선형적으로 저장된 경우에는 높은 결과는 아니나 GPT SSD가 0.52%로 가장 좋았으며 카빙률과 동일하게 전반적으로 선형적으로 파일이 저장된 저장 매체로부터의 카빙 성능은 다른 시나리오에서보다 전체적으로 결과가 좋았다. HDD보다 SSD에서 그리고 MBR보다 GPT에서 비교적으로 좋은 것으로 나타난다.

S1의 저장매체가 포맷된 시나리오(S2)는 파일이 삭제된 경우 파일 카빙 성능을 측정하기 위함으로 정확도가 대부분 저장 매체에서 카빙에 실패한 반면 GPT HDD에서 0.01%로 압축파일 하나가 카빙되어 HDD가 SSD에서보다 파일이 카빙될 수 있는 확률이 높음을 나타낼 수 있다.

데이터셋이 단편화되어 저장되고 디스크가 포맷되지 않은 시나리오(S3)는 비선형적으로 저장된 파일 카빙 성능 측정하기 위함으로 정확도는 MBR 구조에서는 모두 0%의 결과가 나왔으며 GPT 구조에서의 정확도는 각각 0.04%, 0.03%로 아주 낮은 결과를 나타낸다. 그러나 카빙률에서와 마찬가지로 해당 저장 매체로부터 복구된 파일들은 모두 단편화되지 않은 파일로 사실상 단편화되어 저장된 파일을 대상으로 하는 도구의 정확도는 모두 0%로 복구할 수 없는 것을 알 수 있다.

마지막으로 데이터셋이 단편화되고 디스크가 포맷된 경우(S4)에서는 파일이 단편화되어 저장된 후 삭제된 저장 매체로써 모든 저장매체에서의 정확도는 0%로 샘플파일을 카빙하는데 실패하였다.

실험적 결과 Foremost의 성능은 상당히 낮으며, 파일의 헤더 및 푸터를 이용하여 복구해내는 기법의 카빙 도구이므로 파일이 단편화되지 않고 저장된 시나리오(S1)에서의 결과가 비교적으로 좋은 결과가 나타나며, 파일이 단편화된 시나리오(S3, S4)에서 단편화된 파일을 카빙하는데 실패하였다.

현실적으로 디지털포렌식에서 중요한 대상의 파일은 저장 매체에 단편화되어 저장되기 쉬우며, 악의적인 목적을 가진 사람의 경우, 저장 매체를 포맷하여 은닉하려는 행위가 다 반수이다. 이와 같은 다양한 시나리오가 반영된 저장 매체에서의 카빙 결과는 상당히 좋지 않은 것이 도구 검증에서 나타났으며 다양한 도구들이 Foremost 또는 Scalpel을 기반으로 파일 카빙을 수행하는데, 이는 결국 현실에서의 카빙

도구를 통하여 파일 복구를 수행하였을 때 문제를 해결하는데 도움이 되기 어려울 수 있다는 것을 알 수 있다. 같은 시나리오에서 Foremost의 파일 카빙은 HDD와 SSD 그리고 MBR과 GPT에서의 도구의 성능은 차이가 존재하며, 대부분의 카빙된 파일은 문서 및 이미지가 비교적으로 많으므로 이미지와 문서 파일을 복구하는데 더욱 도움이 될 수 있을 것으로 보인다. 또한, 해시값이 일치하는 경우보다 ssdeep을 통하여 99%이상 일치하는 파일이 상대적으로 많으므로 정확하게 동일한 파일이 복구되는 것이 어려울 수 있음을 알 수 있다.

VI. 결 론

본 논문에서는 파일 카빙 도구 검증을 위한 기존의 연구 및 프로젝트로부터 개발된 데이터셋의 한계점을 지적하고, 현실성 있는 환경 및 시나리오를 반영한 이미지를 개발하였다. 개발된 이미지의 환경은 MBR, GPT 그리고 HDD, SSD로 구성된 4가지 저장 매체 환경과, 파일이 단편화되지 않은 디스크 매체, 파일이 단편화되지 않고 저장된 후 포맷된 디스크 매체, 파일이 단편화되어 저장된 디스크 매체, 파일이 단편화되어 저장된 후 포맷된 디스크 매체로 총 네 가지 시나리오에 따른 이미지를 생성하였다. 결과적으로 개발된 16가지의 이미지 검증을 위하여 파일 카빙 도구로 잘 알려진 Foremost v1.5.7을 사용하여 결과를 수치화하였다. 오직 Foremost 도구를 대상으로 하여 제한되어 있다는 한계점이 있으나, 향후 연구에서는 시나리오 이미지를 더욱 세분화하여 효과적으로 카빙 도구 검증을 수행할 수 있도록 하며, 개발된 이미지를 활용하여 연구된 다양한 카빙 기법 및 도구를 검증하고 비교함으로써 더 나은 성능의 카빙 기법 및 도구 연구 및 개발하는데 기여할 수 있을 것으로 보인다.

References

〔1〕 Madril and Abedon, et al. "Metadata recovery in a disk drive." U.S. Patent No. 8,612,706. 17 Dec. 2013.

〔2〕 NIST CFTT, "Forensic Image for File Carving Image" https://www.cfreds.nist.gov/FileCarving/index.html, Jun. 2019.

〔3〕 Digital Forensics Tool Testing Images, "Digital Forensics Tool Testing Images" http://dftt.sourceforge.net/, Jun. 2019.

〔4〕 Digital Corpora, "Real Data Corpus" "http://digitalcorpora.ofg/corpora/disk-images/real-data-corpus, Jun. 2019.

〔5〕 Laurenson and Thomas. "Performance analysis of file carving tools." IFIP International Information Security Conference. Springer, Berlin, Heidelberg, 2013.

〔6〕 Basic Data Carving Test #1,"Digital Forensics Tool Testing Images" http://dftt.sourceforge.net/test11/index.html, Jun. 2019.

〔7〕 DFRWS 2006 Forensics Challenge File Image Layout, "DFRWS2006 Forensics Challenge Data Set" http://old.dfrws.org/2006/challenge/layout.shtml, Jun. 2019.

〔8〕 Baseline Carving Data Set, "Carving Data Set" https://github.com/thomaslaurenson/, Jun. 2019.

〔9〕 GARFINKEL and Simson L. "Carving contiguous and fragmented files with fast object validation." digital investigation, 4: 2-12. 2007.

〔10〕 Air Force Office of Sepcial Investigations and The Center for Information Systems Security Studies and Research, "Foremost" http://foremost.sourceforge.net/, Jun. 2019.

〔11〕 "Scalpel" https://github.com/sleuthkit/scalpel, Jun. 2019.

〔12〕 RICHARD III, Golden G.: ROUSSEV, Vassil. "Scalpel: A Frugal, High Performance File Carver." In: DFRWS. 2005.

〔13〕 NIKKEL and Bruce J. "Forensic analysis of GPT disks and GUID partition tables", Digital Investigation, 2009, 6. 1-2: 39-47.

〔14〕 GEIER and Florian. "The differences b

etween SSD and HDD technology rega rding forensic investigations.", 2015.
〔15〕 CHO and Gyu-Sang. "NTFS Directory Index Analysis for Computer Forensic s." In: 2015 9th International Confere nce on Innovative Mobile and Internet Services in Ubiquitous Computing. p. 441-446, IEEE, 2015.
〔16〕 WINTER and Robert. "SSD vs HDD - d ata recovery and destruction.", Netwo rk Security, 2013, 2013.3: 12-14.
〔17〕 KLOET, S. J. J., et al. "Measuring an d improving the quality of file carving methods.", Almere, Niederlande: Eind hoven University of Technology, 4-79, 2007.

〈저자소개〉

김 해 니 (Haeni Kim) 학생회원
2018년 2월: 경일대학교 사이버보안학과 졸업
2018년 3월~현재: 연세대학교 정보대학원 석사과정
〈관심분야〉 정보보호, 디지털 포렌식 등

김 재 욱 (Jaeuk Kim) 학생회원
2018년 2월: 세명대학교 정보통신학부 졸업
2018년 3월~현재: 연세대학교 정보대학원 석사과정
〈관심분야〉 정보보호, 디지털 포렌식, AML 등

권 태 경 (Taekyoung Kwon) 종신회원
1992년 2월: 연세대학교 컴퓨터과학과 학사
1995년 2월: 연세대학교 컴퓨터과학과 석사
1999년 8월: 연세대학교 컴퓨터과학과 박사
1999년~2000년: U.C. Berkely Post-Doc
2001년~2013년 8월: 세종대학교 컴퓨터공학과 교수
2007년~2008년: Univ. Maryland at College Park 교환교수
2013년 9월~현재: 연세대학교 정보대학원 교수
〈관심분야〉 암호프로토콜, Usable Security, 소프트웨어/시스템보안, 기계학습과보안 등