ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 49, No. 2, pp. 120-125, 2022. 2
https://doi.org/10.5626/JOK.2022.49.2.120

합리적 부정기능을 지원하는 플래시 저장장치를 위한 새로운 플래시 명령어

(New Flash Commands for Building Flash Storage Systems with Plausible Deniability)

조 건 희 † (Geonhee Cho)　김 명 석 †† (Myungsuk Kim)　김 지 홍 ††† (Jihong Kim)

요 약 전통적인 암호화는 암호문의 존재를 숨기지 못해, 복호화 키를 강압적으로 요구하는 공격에 대응할 수 없다. 이를 해결하고자, 데이터의 존재를 부정할 수 있게 하는 Plausible Deniability 특성을 저장장치에 적용한 Deniable Storage 솔루션 연구가 있어왔다. 히든 볼륨은 타 메커니즘 대비 상대적으로 낮은 성능 오버헤드를 가져 널리 활용되고 있으며, 최근에는 멀티스냅샷 공격에 대응 가능하도록 발전하였다. 하지만 히든 볼륨 메커니즘은 근본적으로 암호문을 숨기기 위한 랜덤데이터 풀을 필요로 하는데, 이로부터 Plausible Deniability 특성이 노출되어 데이터를 숨기려는 의도를 내비칠 수 있다는 문제점을 가진다. 본 논문은 데이터 세니타이제이션과 Plausible Deniability 특성을 동시에 지원하는 플래시 칩 수준 접근제어 커맨드셋을 제안하고, 이를 활용해 랜덤데이터 없이도 Plausible Deniability 특성을 지원하는 솔루션을 제안한다.

키워드: 3D 낸드 플래시, 합리적 부정 기능, 히든 볼륨 메커니즘, 부정가능 저장장치

Abstract Traditional encryption cannot defend against coercive attackers who compel the user to hand over decryption keys as it cannot hide the existence of the ciphertext. To solve this problem, there have been studies on a deniable storage solution that applies plausible deniability, a characteristic that allows the user to deny the existence of sensitive data, to a storage device. The hidden volume mechanism is being used in various deniable storage solutions due to its relatively low-performance overhead compared to other mechanisms, and has recently evolved to defend against multiple-snapshot attacks. However, the existing hidden volume mechanism fundamentally requires a dummy random data pool to hide the ciphertext. Due to the existence of dummy random data stored in the storage device, the plausible deniability characteristic is exposed, which can reveal the intention to hide the data. This study proposes a flash chip-level access control command set that simultaneously supports data sanitization and plausible deniability, and using this, we propose a hidden volume-based deniable storage solution that supports plausible deniability characteristics without dummy random data.

Keywords: 3D NAND flash, plausible deniability, hidden volume mechanism, deniable storage

· 이 논문은 2021 한국연구재단의 재원으로 서울대학교 컴퓨터공학부 BK21 FOUR 지능형컴퓨팅사업단의 지원을 받아 수행된 연구임(4199990214639)
· 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2021R1H1A2093240)
· 본 연구는 삼성전자의 지원(IO201207-07809-01)을 받아 수행된 결과임
· 이 논문은 2021 한국컴퓨터종합학술대회에서 '합리적 부정기능을 지원하는 플래시 저장장치를 위한 새로운 플래시 명령어'의 제목으로 발표된 논문을 확장한 것임

† 비 회 원 : 서울대학교 컴퓨터공학과 학생
ghcho@davinci.snu.ac.kr
†† 비 회 원 : 경북대학교 컴퓨터학부 교수(Kyungpook Nat'l Univ.)
ms.kim@knu.ac.kr
(Corresponding author임)
††† 종신회원 : 서울대학교 컴퓨터공학부 교수(Seoul Nat'l Univ.)
jihong@davinci.snu.ac.kr
(Corresponding author임)

논문접수 : 2021년 9월 28일
(Received 28 September 2021)
논문수정 : 2021년 11월 11일
(Revised 11 November 2021)
심사완료 : 2021년 11월 18일
(Accepted 18 November 2021)

Copyright©2022 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.
정보과학회논문지 제49권 제2호(2022. 2)

1. 서 론

전통적인 암호화 기술은 암호문의 존재 자체를 숨길 수 없다는 점으로 인하여, 발견된 암호문에 대해 강압적으로 복호화 키를 요구하는 공격 하에서 데이터 기밀성을 지키기 어렵다는 한계를 가진다.

암호화의 한계점을 보완하고자, 저장장치에 Plausible Deniability 특성을 적용한 Deniable Storage 솔루션들이 제안되어왔다. 이들은 공격자가 저장장치를 분석해보더라도 숨겨진 암호문의 존재를 증명해낼 수 없도록 설계되어, 민감한 데이터의 소유를 부정할 수 있도록 하는 특성인 Plausible Deniability를 제공한다.

히든 볼륨 메커니즘은 Deniable Storage 솔루션을 구현하기 위한 메커니즘들 중, 상대적으로 단순하여 구현이 쉽고 성능 오버헤드가 적어, 유명 암호화 소프트웨어 도구들[1,2]은 물론 다양한 Deniable Storage 솔루션 연구에서 채택되어 왔다.

일부 히든 볼륨 메커니즘 기반 Deniable Storage 솔루션들은 파일시스템, 블록계층 등의 호스트 시스템 수준에서 구현되었다[3,4]. 하지만 이들은 호스트 계층에서 직접 물리적 매체를 제어할 수 있는 저장장치 시스템(예: Hard Disk Drive, HDD)을 가정하고 설계되어, 이러한 가정이 성립하지 않는 플래시 저장장치 시스템(예: Solid State Drive, SSD)에서는 보안 취약점이 발생해 숨겨진 암호문의 존재가 노출될 수 있다[5].

호스트 시스템 수준 솔루션의 문제점을 극복하기 위해, 최근에는 플래시 펌웨어 수준에서의 솔루션들이 제안된 바 있다[5-7]. 이들은 플래시 매체에 대한 직접적인 제어권을 가지고 있는 플래시 변환 계층(Flash Translation Layer, FTL)에 히든 볼륨 메커니즘을 결합하여, 플래시 저장장치 시스템에서도 숨겨진 데이터의 존재를 부정 가능하게 하였다.

하지만, 숨겨진 데이터의 존재를 부정할 수 있는 것과 별개로, 히든 볼륨 메커니즘은 근본적으로 기기 소유주가 데이터를 숨기려는 의도를 가지고 있음을 부정하기 어렵다는 단점을 가진다. 메커니즘의 동작을 위해서는 저장장치 내에 충분한 양의 더미 랜덤데이터를 저장해야 하는데, 일반적인 플래시 저장장치는 다량의 더미 랜덤데이터를 저장하고 있을 이유가 없다는 점으로 인하여 Plausible Deniability 특성에 대한 의심받을 수 있다. 데이터를 숨기려는 의도를 파악한 이상, 공격자는 (숨겨진 데이터의 존재를 증명할 수 없는 상황이라도) 잠재적으로 존재할 수 있는 숨겨진 데이터에 대한 강압적 요구를 멈추지 않을 것이고, Deniable Storage 솔루션은 Plausible Deniability 특성의 효력을 잃게 된다. 따라서, Deniable Storage 솔루션이 강력한 Plausible Deniability를 갖추기 위해서는 숨겨진 데이터의 존재에 대한 부정은 물론, Plausible Deniability 특성 자체에 대한 부정 역시 가능해야 한다.

본 논문에서는 플래시 칩 수준에서의 접근제어 커맨드셋을 소개하고, 이를 활용해 Plausible Deniability 특성이 표면적으로 드러나지 않도록 하여 일반적인 플래시 저장장치의 형태를 가진 (히든 볼륨 메커니즘 기반의) Deniable Storage 솔루션을 제시한다. 제안하는 커맨드셋은 (블록 단위의) 패스워드 기반 접근제어를 지원하며, 접근이 금지된 블록은 다시 접근을 허용하기 전까지 읽기 요청에 대해 항상 원본 데이터가 아닌 '0' 데이터를 반환한다. 이는 랜덤 패스워드를 입력해 데이터 세니타이제이션 목적으로 활용될 수도, 유저 패스워드를 입력해 암호문을 숨기는 목적으로 활용될 수도 있다. 어느 목적으로 활용되든, 접근이 금지된 블록은 결과적으로 스냅샷 상에서 '0' 데이터로 보이게 되며, 공격자 입장에서는 단지 데이터 세니타이제이션의 결과로 해석돼 Plausible Deniability 특성을 지원하는 저장장치임을 알아채기 어렵게 된다.

또한 접근제어 커맨드를 호스트 시스템 수준에서 쉽게 활용할 수 있도록, 호스트 시스템과 저장장치 기기 간의 통합을 진행하였다. 저장장치 펌웨어 수준에 접근제어 커맨드를 기반으로 한 히든 볼륨 메커니즘을 설계·구현하였으며, 이를 쉽게 활용할 수 있는 직관적인 인터페이스를 제공함으로써 호스트 시스템이 접근제어 커맨드에 대한 구체적인 지식이 없이도 민감한 데이터를 숨김·열람할 수 있도록 하였다.

본 논문의 구성을 다음과 같다. 2절에서는 배경지식 및 선행 연구로서, 히든 볼륨 메커니즘과 이를 적용한 최신 솔루션들에 대해 설명한다. 이어 3절에서는 기존 솔루션들의 문제점을 분석하고, 접근제어 기술을 이용한 개선방안을 제시한다. 플래시 칩 수준 접근제어 커맨드 및 이를 활용한 Plausible Deniability 보장 방법에 대해 4절에서 설명한 후, 커맨드 활용을 위한 시스템 통합 설계를 5절에서 제시하며, 6절에서 결론을 맺는다.

2. 배경지식 및 최신 선행 연구

2.1 히든 볼륨 메커니즘

히든 볼륨은 Deniable Storage 솔루션을 구현하기 위한 다양한 메커니즘 중 하나이다. 상대적으로 복잡한 알고리즘을 필요로 하는 Steganography[8] 및 Oblivious RAM[9] 메커니즘과 달리 저장장치에 적은 오버헤드를 부과한다는 장점으로 인하여, 유명 암호화 소프트웨어와 다양한 연구에서 채택되어왔다. 그림 1은 히든 볼륨 메커니즘이 적용된 저장장치의 레이아웃을 나타내며, 다음과 같이 동작한다. 초기화 과정에서 전체 디스크를 더미

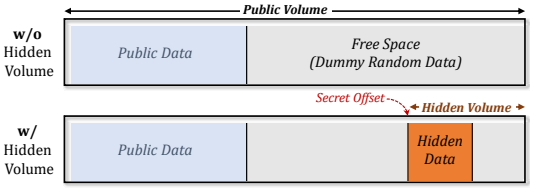

그림 1 히든 볼륨의 논리적 레이아웃

Fig. 1 Logical disk layout of the hidden volume

랜덤데이터로 채우고, 디스크 전체를 퍼블릭 볼륨[1]으로 사용한다. 민감한 데이터를 숨겨야 할 경우, 디스크의 빈 공간 중 secret offset을 정하여 히든 볼륨[2]을 생성한다. 그럼 디스크의 빈 공간은 더미 랜덤데이터와 secret key로 암호화된 데이터로 구성되는데, 두 데이터는 공통적으로 decoy key를 이용한 복호화가 불가능하여 모두 쓰레기 값으로 해석되며, 더미 랜덤데이터와 숨겨진 데이터 간에 구분이 불가능하다. 따라서, 공격자는 숨겨진 데이터가 실제로 존재하는지에 대해 증명할 수 없게 된다. 결과적으로 기기 유저는 강압적인 공격 하에서 decoy key만을 공개하고, decoy key로 복호화 가능한 데이터 외에 추가적으로 데이터를 보유하고 있지 않음을 주장함으로써 강압적인 공격에 대해 방어할 수 있게 된다.

하지만, 이와 같은 기본적인 메커니즘만으로는 여러 시간대에 걸쳐 확보한 디스크 스냅샷들을 비교·분석해 숨겨진 데이터의 탐지를 시도하는 멀티스냅샷 공격에 대한 방어가 불가능하다. 시간 t_0에 스냅샷 S_0가 확보된 뒤로, 다음 스냅샷 S_1이 확보되는 시간 t_1사이에 민감한 데이터를 저장·수정할 경우, 스냅샷 S_0과 S_1의 쓰레기 값 간에 차이가 발생하게 되는데, 이는 곧 데이터를 숨기기 위한 쓰기 동작이 일어났음의 증거가 되기 때문이다. 때문에 기본적인 메커니즘만으로는 오직 싱글스냅샷 공격만 방어할 수 있어, 이를 극복하기 위한 최신 연구들(2.2절)이 등장하였다.

2.2 멀티스냅샷 공격에 대응하는 펌웨어 수준 솔루션

MDEFTL[6]은 기본적인 히든 볼륨 메커니즘(2.1절)을 그대로 펌웨어에 구현한 DEFTL[5]을 발전시켜, 멀티스냅샷 공격에 대한 방어를 가능하게 만든 솔루션이다. 순차 주소 쓰기 방식이 아닌 랜덤 주소 쓰기 방식을 적용하여, 그림 2와 같이 퍼블릭 볼륨, 더미 랜덤데이터 풀, 그리고 히든 볼륨이 전부 뒤섞인 저장장치 레이아웃을 가진다. 또한 민감한 데이터의 저장·수정 동작이 발생했음을 부정하기 위해, 퍼블릭 볼륨에 쓰기 요청이 발생할 때 혹은 저장장치 유휴시간마다 펌웨어가 자체적으로 더미 랜덤데이터 쓰기를 추가적으로 실행한다. 그림 2에서

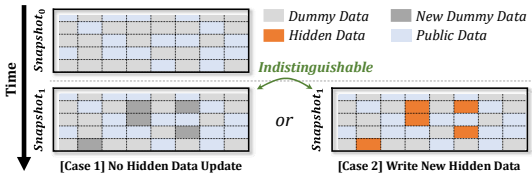

그림 2 MDEFTL의 레이아웃 및 동작 개요

Fig. 2 Disk layout and operational overview of MDEFTL

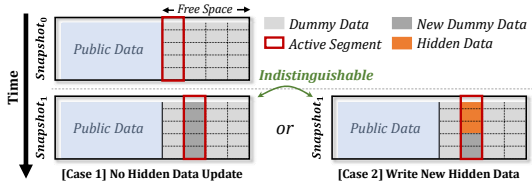

그림 3 ECD의 레이아웃 및 동작 개요

Fig. 3 Disk layout and operational overview of ECD

보이듯, 두 스냅샷들 사이에서 관찰되는 더미 랜덤데이터 풀 내의 데이터 변화가 1) 펌웨어가 자체적으로 실행한 더미 랜덤데이터 쓰기로 인한 변화인지, 2) 민감한 데이터를 저장·수정하며 발생한 변화인지 알 수 없다. 결과적으로, 멀티스냅샷 공격자는 스냅샷 간에 발생한 쓰레기 값의 차이를 데이터를 숨기기 위한 쓰기 동작이 일어났음의 증거라고 주장하기 어려워져, 기기 유저는 숨겨진 암호문의 존재를 부정할 수 있게 만든다.

ECD[7]는 저장장치를 두 영역으로 나누어 분할하고, 그 중 민감한 데이터를 숨길 파티션을 더미 랜덤데이터로 채운 뒤, 해당 영역을 고정된 크기의 세그먼트들로 나눈 저장장치 레이아웃을 가진다. 그리고 여러 세그먼트 중 액티브 세그먼트를 선정해, 담겨있는 데이터를 주기적으로 다음 세그먼트 위치로 이주시키고, 이주한 세그먼트를 액티브 세그먼트로 재설정한다. 매 이주마다 민감한 데이터의 저장·수정으로 인해 발생하는 새로운 쓰기 데이터들을 액티브 세그먼트에 반영하며, 민감한 데이터를 담은 공간 외의 빈 공간은 새로운 더미 랜덤데이터로 채운다. 그림 3과 같이, 이주 전후 스냅샷들 사이에서 관찰되는 액티브 세그먼트 내 데이터의 변화가 1) 단순히 이주 과정에서 새로운 더미 랜덤데이터를 기록하면서 생긴 변화인지, 2) 암호문 저장·수정을 위한 쓰기 데이터를 반영하며 생긴 변화한 것인지 알 수 없다. 때문에 MDEFTL과 마찬가지로, 기기 유저는 숨겨진 암호문의 존재를 부정할 수 있게 된다.

3. 문제점 및 개선방안

MDEFTL과 ECD는 물론, 기존의 모든 히든 볼륨 메커니즘 기반의 Deniable Storage 솔루션들은 각자의 방

1) 공개 가능한 데이터를 decoy key로 암호화해 저장하는 볼륨
2) 민감한 데이터를 secret key로 암호화해 저장하는 볼륨

식을 이용하여 숨겨진 데이터의 존재를 부정할 수 있지만, 공통적으로 모두 데이터를 숨기기 위한 더미 랜덤데이터 풀을 필요로 한다. 이러한 특수한 데이터[3]의 존재는 공격자에게 해당 저장장치가 무언가 수상쩍은 기능을 포함하고 있음을 암시해준다. 또한 멀티스냅샷 공격을 방어하기 위해 특수한 저장장치 레이아웃이 적용되고, 특수동작(예: 주기적인 세그먼트 이주 동작 등)이 결합되는 경우, 플래시 메모리의 성능, 수명 등의 필수적인 요구사항들을 희생[4]시키므로, 저장장치가 수상한 기능을 포함하고 있다는 의심이 가중된다. 더욱이 정황상의 근거까지 더해진다면, 공격자는 Plausible Deniability 특성을 확신하기에 충분하다. 유저가 데이터를 숨기려는 의도를 가지고 있음을 알게 된 이상, 숨겨진 데이터의 존재를 부정하는 주장은 효력을 잃게 될 것이다.

위의 문제점들을 해결하기 위해서는, 일반적인 플래시 저장장치에 최대한 가까운 형태의 솔루션이 필요하다. 즉, Plausible Deniability 특성에 대한 의심을 받을 만한 요소들을 포함하지 않아야 하며, 수상한 기능이 포함되지 않은 일반적인 저장장치임을 합리적으로 주장 가능해야 한다. 이는 기존에도 지적된 문제이며[10], 이를 해결하기 위해서는 Plausible Deniability 특성을 표시나지 않게 저장장치에 결합하는 방식으로 솔루션을 설계해야 할 필요가 있다.

4. 접근제어 기술 및 이를 활용한 Plausible Deniability

본 장에서는, 일반적인 플래시 저장장치 기능 중 하나인 데이터 세니타이제이션 기능을 지원하는 동시에 Plausible Deniability 특성을 제공할 수 있는 플래시 커맨드셋과, 이를 활용한 솔루션 설계를 제시한다.

4.1 커맨드 동작 개요

제안하는 새로운 플래시 커맨드셋은, 패스워드 기반 인증방식을 통해 특정 블록에 대한 접근을 금지하거나 허용한다. 이는 접근제어 방식의 데이터 세니타이제이션 기법을 제안한 논문인 Evanesco[11]의 플래시 칩 설계를 기반으로 확장되었다.

커맨드셋은 1) 특정 블록에 대한 접근을 금지시키는 커맨드인 bHide와, 2) 접근이 금지된 블록에 대해 다시 접근을 허용하는 커맨드인 bExpose로 구성된다. 그림 4(a), (b)는 각각 bHide 커맨드와 bExpose 커맨드의 동작 개요를 나타낸다. 주소 0×08에 해당하는 블록에 대한 접근을 제어하기 위해서는, bHide 커맨드를 통해 패스워드를 전달한다(❶). 커맨드를 입력 받은 플래시 칩은 내부적으로 이미 설정된 패스워드가 있는지 확인한다(❷). 이미 설정된 패스워드가 없는 상태라면, 지정된 블록의 SSL(String Select Line)에 전달받은 패스워드를 프로그램 한다. 또한 칩 데이터 전송 경로[5]의 차단 여부를 결정짓는 셀인 bH(Block Hiding) 셀을 프로그램 한다(❸). 이후 요청되는 read 커맨드(❹)는 칩 외부로 데이터를 전달하는 경로가 차단되어 있기 때문에 (❺), 원본 데이터가 아닌 '0' 데이터를 돌려받는다(❻). 반대로 주소 0×08에 해당하는 블록에 대한 접근을 허용하기 위해서는, 그림 4(b)에서 보이듯 bExpose 커맨드를 통해 패스워드를 전달한다(①). 칩 내부적으로, bExpose 커맨드와 함께 전달된 패스워드가 이전에 bHide 커맨드를 통해 설정되었던 패스워드와 일치하는 지 검사하고 (②), 일치하는 경우 SSL을 지워 기존에 설정되었던 패스워드 및 bH 셀을 초기화한다(③). bH 셀이 지워져 데이터 경로 차단이 해제되었으므로, 이후 요청되는 read 커맨드(④)에 대해서는 원본 데이터가 칩 외부로 전송될 수 있다(⑤).

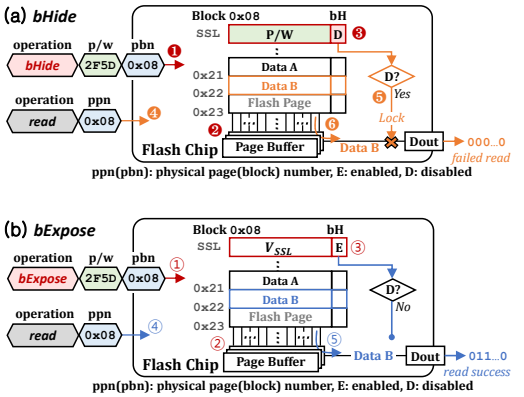

그림 4 bHide 커맨드 및 bExpose 커맨드의 동작 개요
Fig. 4 Operational overview of bHide and bExpose

4.2 접근제어 특성을 활용한 솔루션 설계 접근법

제안한 커맨드셋은, 랜덤 패스워드로 접근을 금지해 재접근이 불가능하게 함으로써 데이터 세니타이제이션 기술로 활용할 수도, 암호문 저장 후에 유저 패스워드로 접근을 금지하고 추후 재접근이 가능하게 할 용도로 활용할 수도 있다. 두 경우 모두 결과적으로 스냅샷 상에서 '0' 데이터로 보이게 되며, 단지 데이터 세니타이제이션의 결과로 보일 뿐이다. 그림 5는 접근제어를 활용해 Plausible Deniability를 제공하는 방법에 대한 개요를

3) 일반적인 플래시 저장장치라면, decoy key로 해석가능한 최신 데이터(valid page)와 과거 데이터(invalid page)만으로 구성되며, 다량의 더미 랜덤데이터가 존재할 이유가 없다.

4) 더미 랜덤데이터 쓰기 동작(MDEFTL), 주기적인 세그먼트 이주 동작(ECD)으로 인해 저장장치 대역폭 일부가 소모되어 성능이 저하되며, 유휴 공간을 불필요하게 지속적으로 소비하여 잦은 지우기 동작을 발생시킴에 따라 플래시 수명에 악영향을 미친다.

5) 페이지 버퍼와 데이터 출력 핀 사이의 데이터 입출력 경로

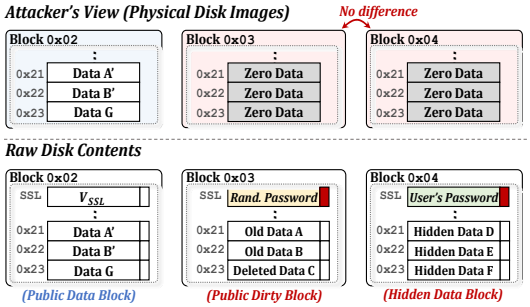

그림 5 제안 솔루션의 Plausible Deniability 개요

Fig. 5 Overview of how the proposed solution provides plausible deniability

나타낸다. 상단은 공격자가 확인가능한 스냅샷을, 하단은 실제 플래시 매체에 저장된 원본 데이터를 나타낸다. 일반 데이터가 저장된 블록(좌측)에는 접근제어가 적용되지 않으며, 가비지 컬렉션된 블록(중앙)과 암호문을 숨긴 블록(우측)에는 bHide 커맨드를 이용해 접근을 제어한다. 접근이 금지된 블록들(중앙, 우측)은 read 커맨드에 대해 '0' 데이터를 반환하기 때문에, 공격자에게는 세니타이제이션된 블록으로 보이게 된다. 결과적으로, 공격자의 입장에서 해당 저장장치는 단지 데이터 세니타이제이션 기능을 지원하는 일반적인 플래시 저장장치로 보이게 된다. 또한 암호문을 추가적으로 저장·수정하더라도, 스냅샷에는 그 흔적이 남지 않는다는 점에서 멀티스냅샷 공격에 대해서도 쉽게 방어 가능하다. 표 1은 기존에 제안된 (히든 볼륨 메커니즘 기반의) 펌웨어 수준 솔루션들과, 본 논문에서 제시하는 커맨드셋 기반의 접근법 간의 비교 요약이다.

5. 시스템 통합

접근제어 커맨드는 저장장치 펌웨어 수준에서만 사용가능하기 때문에, 호스트 시스템 수준에서 직접적으로 활용할 수 없다. 따라서 접근제어 커맨드를 이용한 히든 볼륨 관리 동작들을 추상화해 호스트 시스템이 쉽게 활용할 수 있는 인터페이스로 제공한다. 그림 6은 히든 볼륨 관리를 위한 모듈과, 히든 볼륨 메커니즘을 호스트 시스템에 제공하기 위한 인터페이스를 보인다. 호스트 시스템은 인터페이스를 통해 접근제어 커맨드를 기반으로 하는 히든 볼륨 메커니즘을 사용할 수 있다. Create를 통해 유저 패스워드로 초기화된, 즉 민감한 데이터를 숨기기 위한 영역을 (블록 단위로) 할당 받을 수 있고, Mount를 통해 유저 패스워드 인증절차를 거쳐 해당 영역에 대한 접근권한을 획득한다. 이후 기존하던 입출력 인터페이스를 통해 데이터를 숨기거나, 이미 숨겨져 있던 데이터를 읽어올 수 있다. 숨겨진 데이터 쓰기/읽기를 마친 뒤에는 Unmount를 통해 다시 해당 영역에 대한 접근을 차단할 수 있으며, 숨겨놓은 데이터를 파기하고 싶은 경우 Destroy를 통해 해당 영역을 할당 해제할 수 있다.

인터페이스를 통한 호스트 시스템의 요청을 처리함과 더불어, 펌웨어는 Plausible Deniability 특성 확보 및 데이터 세니타이제이션을 위해 가비지 컬렉션된 블록에 대해서도 접근을 차단한다. 가비지 컬렉션된 블록에 대해 랜덤 패스워드로 bHide 커맨드를 추가적으로 적용해야 하기 때문에 가비지 컬렉션 지연시간을 증가시킬 수 있지만, bHide 커맨드는 유효 페이지 복사에 소모되는 지연시간 대비 매우 미미한 추가 지연시간을 부과하기 때문에, 무시해도 될 정도의 성능저하를 발생시킬 것이다. 또한 추가적인 지우기 동작 등을 발생시키지 않아, 수명과 신뢰성에 대해서도 전혀 악영향을 미치지 않는다.

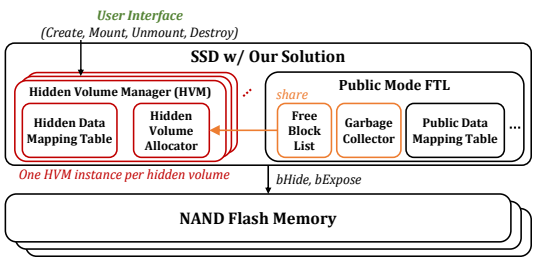

그림 6 제안하는 Deniable Storage 솔루션의 구조

Fig. 6 An organizational overview of the proposed solution

표 1 멀티스냅샷 방어 솔루션 간의 비교

Table 1 Comparison of multi-snapshot resistant solutions

	Plausibility	Performance	Lifetime	Reliability	Capacity
MDEFTL [6]	Low	X	X	X[6)]	O
ECD [7]	Low	X	X	O	X[7)]
Our Work	High	O	O	O	O

6) 랜덤쓰기 방식을 적용함에 따라, 모든 블록들이 쓰기데이터를 기록가능한 상태로 유지되어야 하는데, 이는 3D 낸드 플래시에서 신뢰성 저하 이슈를 발생시킨다. (Open block problem [11])

7) 저장장치를 두 파티션으로 나누고 하나의 파티션만 퍼블릭 볼륨으로 사용하므로, 일반 데이터를 저장할 수 있는 용량이 크게 감소된다.

6. 결 론

본 논문에서는 기 제안된 플래시 펌웨어 수준에서의 멀티스냅샷 방어 솔루션들은 데이터를 숨기려는 의도가 쉽게 노출됨을 지적하였다. 그리고 이러한 문제점을 개선하기 위해, 플래시 칩 수준에서의 접근제어 커맨드셋을 제시하고, 이를 활용해 일반 플래시 저장장치에 가까운 형태를 갖춤으로써 데이터를 숨기려는 의도의 노출 가능성을 최소화한 솔루션 설계 방안을 제시하였다. 또한 제시한 히든 볼륨 메커니즘을 호스트 시스템이 쉽게 사용할 수 있도록 추상화된 인터페이스를 설계하여, 저장장치 펌웨어 간의 시스템 통합을 진행하였다. 결과적으로, 제안된 솔루션은 기존 기법들 대비 더욱 강력한 Plausible Deniability 특성을 제공하며, 저장장치의 성능, 수명 그리고 신뢰성 등에 악영향을 주지 않는다는 장점을 가진다.

References

[1] TrueCrypt, "https://www.truecrypt71a.com/documentation/plausible-deniability/hidden-volume/".

[2] VeraCrypt, "https://veracrypt.eu/en/docs/hidden-volume/".

[3] Yu, X., et al., "Mobihydra: Pragmatic and Multi-level Plausibly Deniable Encryption Storage for Mobile Devices," *ISC*, pp. 555-567, 2014.

[4] Chang, B., et al., "User-friendly Deniable Storage for Mobile Devices," *Comput. Secur.*, pp. 163-174, 2018.

[5] Jia, S. et al., "DEFTL: Implementing Plausibly Deniable Encryption in Flash Translation Layer," *CCS*, pp. 2217-2229, 2017.

[6] Jia, S., et al., "MDEFTL: Incorporating Multi-Snapshot Plausible Deniability into Flash Translation Layer," *TDSC*, 2021.

[7] Zuck, A., et al., "Preserving Hidden Data with an Ever-Changing Disk," *HotOS*, pp. 50-55, 2017.

[8] Chang, B., et al., "Mobiceal: Towards Secure and Practical Plausibly Deniable Encryption on Mobile Devices," *DSN*, pp. 454-465, 2018.

[9] Chakraborti, A., et al., "DataLair: Efficient Block Storage with Plausible Deniability against Multi-Snapshot Adversaries," *PETS*, pp. 175-193, 2017.

[10] Davis, J. C., "Plausible Plausible Deniability with an Epistemological Gap," 2018.

[11] Kim, M., et al., "Evanesco: Architectural Support for Efficient Data Sanitization in Modern Flash-Based Storage Systems," *ASPLOS*, pp. 1311-1326, 2020.

조 건 희

2019년 아주대학교 소프트웨어학과 학사 2019년~현재 서울대학교 컴퓨터공학과 석박통합과정. 관심분야는 낸드 플래시 저장장치, 임베디드 시스템, 시스템 소프트웨어

김 명 석

2003년 연세대학교 전기전자공학과 학사 2005년 연세대학교 전기전자공학과 석사 2020년 서울대학교 컴퓨터공학과 박사 2005년~2020년 삼성전자 메모리사업부 플래시 개발실 근무. 2021년~현재 경북대학교 컴퓨터공학부 조교수. 관심분야는 임베디드 시스템 및 소프트웨어, 지능형 메모리 시스템

김 지 홍

1986년 서울대학교 계산통계학과 학사 1988년 University of Washington 컴퓨터과학과 석사. 1995년 University of Washington 컴퓨터과학 및 공학과 박사. 1995년~1997년 미국 Texas Instruments 선임연구원. 1997년~현재 서울대학교 컴퓨터공학과 교수. 관심분야는 낸드 플래시 저장장치, 저전력 시스템, 임베디드 소프트웨어, 컴퓨터 구조