합리적 부정기능을 지원하는 플래시 저장장치를 위한 새로운 플래시 명령어*

조건희○, 김명석†, 김지홍
서울대학교 컴퓨터공학부
†경북대학교 컴퓨터학부
{ghcho, jihong}@davinci.snu.ac.kr, †ms.kim@knu.ac.kr

New Flash Commands for Building Flash Storage Systems with Plausible Deniability

Geonhee Cho○, Myungsuk Kim†, and Jihong Kim
Department of Computer Science and Engineering, Seoul National University
†School of Computer Science and Engineering, Kyungpook National University

요 약

전통적인 암호화는 암호문의 존재를 숨기지 못하여, 복호화 키를 강압적으로 요구하는 공격자에 대응할 수 없다. 이를 해결하기 위해, 데이터의 존재를 부정할 수 있도록 하는 특성인 Plausible Deniability를 저장장치 측면에 적용하는 시도가 있어왔다. 최근에는 멀티스냅샷 공격에 대응 가능하도록 하는 플래시 펌웨어 수준 솔루션들이 제안되었으며, 이들은 암호문을 저장·수정한 흔적을 부정하기 위해 특수한 동작들을 펌웨어에 결합하였다. 하지만 이러한 특수동작의 존재로부터, Plausible Deniability 특성을 이용해 데이터를 숨기려는 의도가 쉽게 노출될 수 있다는 문제점을 가진다. 본 논문은 데이터 세니타이제이션 기능과 Plausible Deniability 특성을 동시에 지원하는 플래시 칩 수준 접근제어 커맨드셋을 제안하고, 이를 활용해 외부적으로 데이터 세니타이제이션을 지원하는 일반적인 플래시 저장장치의 형태를 띔으로써 Plausible Deniability 특성의 노출 가능성을 최소화한 솔루션 설계방식을 제시한다.

1. 서 론

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

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

초기 솔루션들은 파일시스템, 블록계층 등의 호스트 시스템 수준에서 구현되었다[1]. 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[2].

호스트 시스템 수준 솔루션의 문제점을 극복하기 위해, 플래시 펌웨어 수준에서의 솔루션들이 제안된 바 있다[2-4]. 이들은 플래시 매체에 대한 직접적인 제어권을 가지고 있는 플래시 변환 계층(Flash Translation Layer, FTL)에 Plausible Deniability 특성을 결합하여, 호스트 시스템 수준 솔루션의 보안 취약점을 보완하였다. 더불어, 여러 시간대에 걸쳐 확보한 물리적 디스크 스냅샷들을 비교·분석해 암호문 탐지를 시도하는 멀티스냅샷 공격에 대한 방어 기술의 근간이 되었다.

멀티스냅샷 공격에 대응하는 펌웨어 수준 솔루션들은 데이터의 존재를 부정할 수 있게 하는 한편, 데이터를 숨겼을 가능성에 대한 의심을 피하기는 어렵다. 이들은 숨겨진 암호문을 저장·수정함으로써 발생한 스냅샷 간 차이점이 마치 다른 이유로 인해 발생한 차이점인 것처럼 보이도록 돕는 특수동작들을 펌웨어에 결합한다. 이러한 동작들은 일반적으로 펌웨어에 포함되지 않는 동작이라는 점에서 쉽게 구분될 수 있으며, 동작 특성상 데이터를 숨기기 위한 동작이라는 의심을 받을 수 있다. 더욱이, 성능, 수명 등에 악영향을 미침에도 불구하고 결합된 특수동작이라는 점에서, 의심을 가중시킨다.

본 논문에서는 플래시 칩 수준에서의 접근제어 커맨드셋을 소개하고, 이를 활용해 Plausible Deniability 특성이 표면적으로 드러나지 않도록 하여 일반적인 플래시 저장장치의 형태를 가진 솔루션 설계방식을 보인다. 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

* 이 논문은 삼성전자 미래기술육성센터의 지원을 받아 수행된 연구임(SRFC-IT2002-06). (교신저자: 김지홍)

블록은 결과적으로 스냅샷 상에서 '0' 데이터로 보이게 되며, 공격자 입장에서는 단지 데이터 세니타이제이션의 결과로 해석돼 Plausible Deniability 특성을 지원하는 저장장치임을 알아채기 어렵게 된다.

본 논문의 구성을 다음과 같다. 2장에서는 현재까지 제안된 멀티스냅샷 공격에 대응가능한 펌웨어 수준 솔루션들에 대해 설명한다. 이어 3장에서는 그들의 문제점을 분석하고, 접근제어 기술을 기반으로 한 개선방안을 제시한다. 플래시 칩 수준 접근제어 기법 및 이를 활용한 Plausible Deniability 보장 방법에 대해 4장에서 설명한 후, 5장에서 결론을 맺는다.

2. 관련 연구: 펌웨어 수준 멀티스냅샷 방어 솔루션

2.1 더미 랜덤데이터를 활용한 솔루션

ECD[3]는 저장장치의 일정 공간을 더미 랜덤데이터로 채우고, 해당 영역을 고정된 크기의 세그먼트들로 나눈다. 그리고 여러 세그먼트 중 액티브 세그먼트를 선정해, 주기적으로 새로운 위치로 이주시킨다. 매 이주마다 암호문 저장·수정으로 인한 새로운 쓰기 데이터들을 액티브 세그먼트에 반영하며, 나머지 빈 공간은 새로운 더미 데이터로 채운다. 그림 1과 같이, 이주 전후 스냅샷들 사이에서 관찰되는 액티브 세그먼트 내 데이터의 변화가 1) 단순히 새로운 더미 랜덤데이터를 기록하여 생긴 변화인지, 2) 암호문 저장·수정을 위한 쓰기 데이터가 반영되어 변화한 것인지 알 수 없다는 점에서, 숨겨진 암호문의 존재를 부정할 수 있다.

2.2 Write-Once Memory (WOM) 코드를 활용한 솔루션

PEARL[4]은 2nd 부호워드 선택지가 두가지인 WOM 코딩 방식을 활용해, 일반 데이터 속에 암호문을 숨긴다. 그림 2처럼, 스냅샷 상의 2nd 부호워드는 1) 일반 데이터 속에 암호문을 인코딩하여 프로그램한 결과일 수도, 2) invalidate된 플래시 페이지에 일반 데이터만을 재-프로그램한 결과일 수도 있다. 기록된 2nd 부호워드가 어떤 과정을 거쳐 기록되었는지 공격자가 구분할 방법이 없다는 점에서, 숨겨진 암호문의 존재를 부정할 수 있다.

3. 문제점 및 개선방안

위의 두 솔루션들은 숨겨진 데이터의 존재를 부정할 수 있지만, 이를 위해 일반적인 플래시 저장장치의 동작과는 다른 특수한 동작들을 적용한다. ECD의 경우 주기적 세그먼트 이주 동작, PEARL의 경우 WOM 코딩의 사용이 특수동작에 해당한다. 이러한 특수동작들의 존재는 공격자에게 해당 저장장치가 무언가 수상쩍은 기능을 포함하고 있음을 암시해준다. 또한 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.

더욱이, 특수동작들을 결합하기 위해 성능, 수명 등을 희생시키므로, 다른 유용한 용도로써 결합되었다는 주장

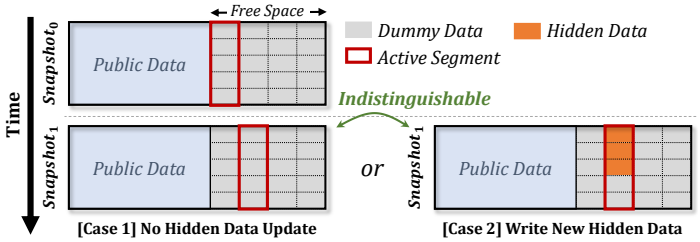

그림 1. ECD의 Plausible Deniability 개요

Original Data bits	1st Codeword	2nd Codeword	
		Hide 0	Hide 1
00	000	000	111
01	001	001	110
10	010	010	101
11	100	100	011

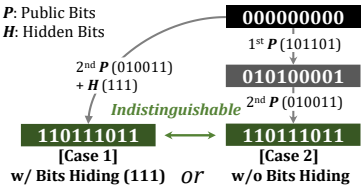

그림 2. PEARL의 Plausible Deniability 개요

이 어려워 의심을 가중시킨다. 예를 들어 ECD의 경우, 주기적인 세그먼트 이주 공간 확보를 위해 잦은 지우기 동작을 수행함에 따라 플래시의 수명을 빠르게 소모시킨다. 또한 PEARL의 경우, WOM 코드의 부호화율(실제 데이터 크기/부호화된 데이터 크기)이 낮아 쓰기·읽기 증폭이 발생해 I/O 처리량이 대폭 감소되며, 플래시에 대한 재-프로그램은 같은 셀을 공유하는 다른 페이지 혹은 인접 워드라인에 저장된 데이터의 신뢰성에 악영향을 주어 교정 불가능한 에러를 발생시킬 수 있다.

위의 문제점들을 해결하기 위해서는, 일반적인 플래시 저장장치에 최대한 가까운 형태의 솔루션이 필요하다. 즉, Plausible Deniability 특성에 대한 의심을 받을 만한 특수동작을 포함하지 않아야 하며, 수상한 기능이 포함되지 않은 일반적인 저장장치임을 합리적으로 주장 가능해야 한다. 이를 위해서는, 플래시 펌웨어의 기본적인 기능에 Plausible Deniability 특성을 표시나지 않게 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 커맨드 집합과, 이를 활용한 솔루션 설계를 제시한다.

4.1 접근제어를 지원하는 3D 낸드 플래시 칩 및 명령어

제안하는 새로운 플래시 커맨드셋은, 패스워드 기반 인증방식을 통해 특정 블록에 대한 접근을 금지하거나 허용한다. 이는 접근제어 방식의 데이터 세니타이제이션 기법을 제안한 논문인 Evanesco[5]의 플래시 칩 설계를 기반으로 확장되었다.

커맨드셋은 1) 특정 블록에 대한 접근을 금지시키는 커맨드인 *bHide*와, 2) 접근이 금지된 블록에 대해 다시 접근을 허용하는 커맨드인 *bExpose*로 구성된다. 그림 3(a), (b)는 각각 *bHide* 커맨드와 *bExpose* 커맨드의 동작 개요를 나타낸다. 주소 0x08에 해당하는 블록에 대한 접근을 제어하기 위해서는, *bHide* 커맨드를 통해 패스워

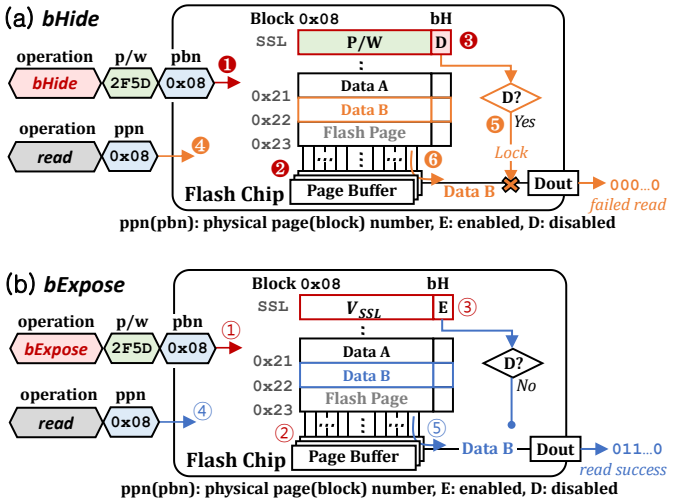

그림 3. *bHide* 커맨드 및 *bExpose* 커맨드의 동작 개요

드를 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 경로 차단 여부를 결정짓는 셀인 bH(Block Hiding) 셀을 프로그램 한다(❸). 이후 요청되는 *read* 커맨드(④)는 칩 외부로 데이터를 전달하는 경로가 차단되어 있기 때문에(⑤), 원본 데이터가 아닌 '0' 데이터를 돌려받는다(⑥). 반대로 그림 3(b)에서 보이듯, 주소 0x08에 해당하는 블록에 다시 접근을 허용하려면 *bExpose* 커맨드를 통해 패스워드를 전달한다(①). 칩 내부적으로, *bExpose* 커맨드와 함께 전달된 패스워드가 이전에 *bHide* 커맨드를 통해 설정되었던 패스워드와 일치하는지 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).

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 4는 접근제어를 활용해 Plausible Deniability를 제공하는 방법에 대한 개요를 나타낸다. 그림의 상단은 공격자가 확인가능한 스냅샷을, 하단은 실제 플래시 매체에 저장된 원본 데이터를 나타

표 1. 멀티스냅샷 방어 솔루션 간의 비교

	Plausibility 정도	성능 보존	수명 보존	신뢰성 보존	용량 보존
ECD [3]	낮음	X	X	O	X
PEARL [4]	낮음	X	X	X	X
Our Work	높음	O	O	O	O

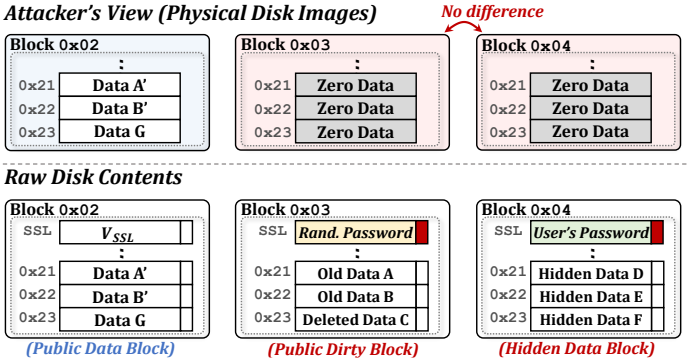

그림 4. 제안 솔루션의 Plausible Deniability 개요

낸다. 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 기존 솔루션들과, 본 논문에서 제시하는 커맨드셋 기반의 접근법 간의 비교 요약이다.

5. 결론 및 향후 연구

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. 향후 연구로는, 제시하는 커맨드셋을 유저가 활용 가능하도록 호스트-펌웨어 계층 사이의 인터페이스를 개발해 시스템 통합을 진행할 예정이다.

참 고 문 헌

[1] Anderson, R., et al., "The Steganographic File System," IH, pp. 73-82, 1998.
[2] Jia, S. et al., "DEFTL: Implementing Plausibly Deniable Encryption in Flash Translation Layer," CCS, pp. 2217-2229, 2017.
[3] Zuck, A. et al., "Preserving Hidden Data with an Ever-Changing Disk," HotOS, pp. 50-55, 2017.
[4] Chen, C et al., "PEARL: Plausibly Deniable Flash Translation Layer using WOM coding," Security, 2021.
[5] Kim, M. et al., "Evanesco: Architectural Support for Efficient Data Sanitization in Modern Flash-Based Storage Systems," ASPLOS, pp. 1311-1326, 2020.