DeadDrop-in-a-Flash

INFORMATION HIDING IN SSD NAND FLASH MEMORY

Overview

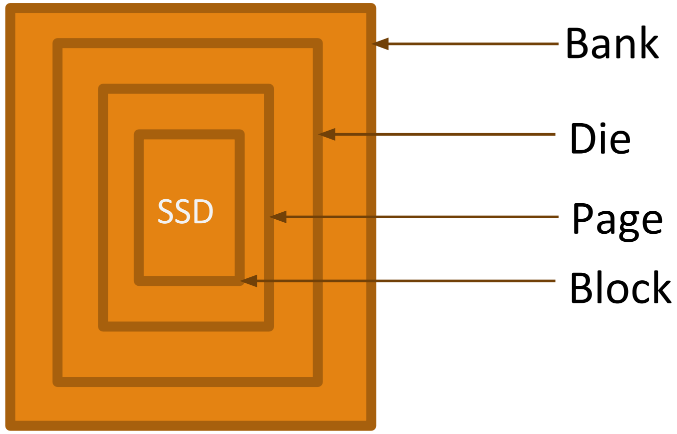
- Introduction
- SSD background
  - Flash Translation Layer
  - Reference SSD and Open SSD
- Design
- Implementation
- Experiment and results

Introduction

- ❑ A novel attempt at information hiding at the physical layer of a Solid State Drive (SSD).
- ❑ Proposed solution is filesystem and OS-independent.
- ❑ Our framework takes advantage of the design complexity in SSDs and uses them as covert channel.
- ❑ Proposed method is resistant to firmware updates and transparent to the OS and the user.
- ❑ The only attempt to hide information in flash (Thumb drives) was by Wang et. Al. They were able to hide up to 64 MB of data on 32GB thumb drive. While utilizing our framework one can hide up to 2 GB of information in a 32 GB SSD.

SSD Background

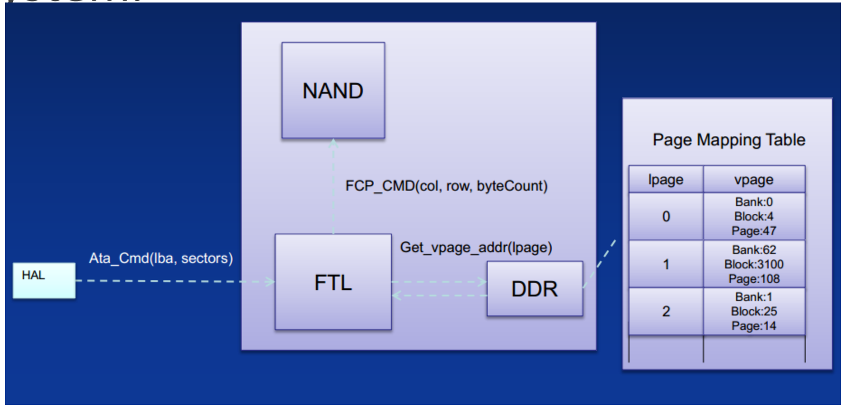
- ❑ A data storage device using integrated circuits and assemblies.
- ❑ Better in performance than traditional hard disk drives.
- ❑ Most current SSD’s utilize NAND flash cells.
- ❑ Basic read and write unit is page.
- ❑ Basic erase unit is a block.



- ❑ A page has to be erased before write and hence the block holding the page has to be erased.
- ❑ Should support the sector and track based access of the operating system.

Flash Translation Layer(FTL)

- ❑ Takes care of the logical to physical mapping for the operating system.
- ❑ Controls the performance boosters
  - ❑ Bad block management.
  - ❑ Wear Leveling
  - ❑ Garbage collection



Page Mapping Table

lpage	vpage
0	Bank:0 Block:4 Page:47
1	Bank:62 Block:3100 Page:108
2	Bank:1 Block:25 Page:14

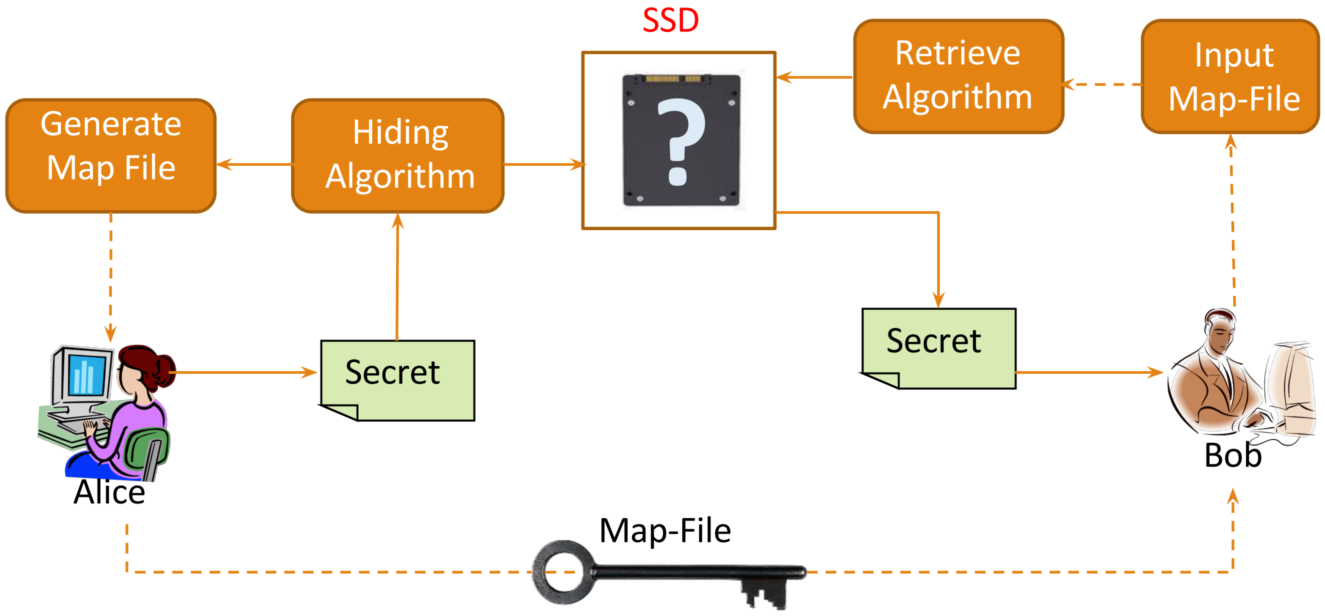
- ❑ Over provisioning:
  - ❑ To support the inbuilt performance boosters, the solid state drives are manufactured with a size greater than advertised size.
  - ❑ The additional space is never visible to the normal user.

Reference SSD and Open SSD

- Reference SSD
  - Thirty GB OCZ Vertex series SATA II 2.5” SSD.
  - Utilizes Indilinx Barefoot controller.
  - Holds 8 flash memory packages with 2 dies in each, each die contains 4096 blocks with 128 pages in each block.

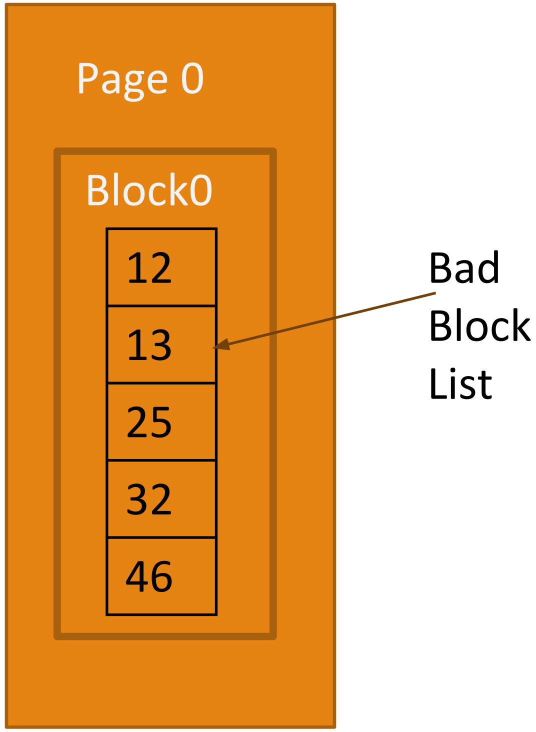
- Open SSD
  - Created by Sungkyunkwan University in Suwon, South Korea in collaboration with Indilinx, to promote research and education on SSD technology.
  - Provides the firmware implementation for Indilinx Barefoot controller utilized by several SSD manufacturers including OCZ, Corsair, Mushkin and Runcore IV.

Design


SSD
Generate Map File
Hiding Algorithm
Retrieve Algorithm
Input Map-File
Secret
Secret
Alice
Bob
Map-File

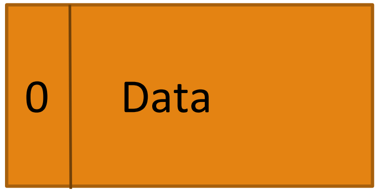
Design

- Framework to hide and retrieve information.
- Utilizes Bad Block Management, which is part of firmware.
- Bad block management is handled through a bad block list, holding the list of blocks identified to be bad.
- Add a set of blocks to the existing bad block list and utilize them to hide information.
- Problem: Can’t exceed the maximum allowed bad blocks.



Implementation

- Command line interface with functionalities to
  - Allocate bad blocks prior to hiding and retrieving information.
  - Uniquely identify the blocks added by us with the erase count added to the first byte of each added block.
  - Generate Map file to uniquely identify the hidden data.
  - Retrieve information with or without erasing the data.
- Significance of Erase count
  - Differentiates blocks added by us from original bad blocks.
  - Blocks wear out and become unusable over continuous erase and write.
  - Increment the erase count before write as the blocks are erased before write.
  - Pick the block with least erased count each time when information is to be hidden.



Bad Block

Research implications

- Bad block management is part of overprovisioning.
- Adding bad blocks will never result in change in the logical size of the SSD.
- The information can be retrieved with the mentioned changes to the SSD:
  - Firmware reinstallation
  - NTFS format on the drive
  - Partitioning the drive
  - Populating the drive to its full capacity
- None of the existing Anti-forensics can reveal the existence of hidden information tested with WinHex and FTK imager.

Questions?

Thank You