DEFTL: Implementing Plausibly Deniable Encryption in Flash Translation Layer

By Shijie Jia, Luning Xia, Bo Chen, Peng Liu

Presenter: Niusen Chen (PhD student)
Department of Computer Science
Michigan Technological University

Outline

- Background Introduction
- Attack Scenarios
- Design of DEFTL
- Evaluation

Features of Flash Memory

1. Read/Write on pages, but erase on blocks
2. Erase - before - write
3. Out - of - place update
4. Limited of program-erase(P/E) cycles

Special Functions in Flash

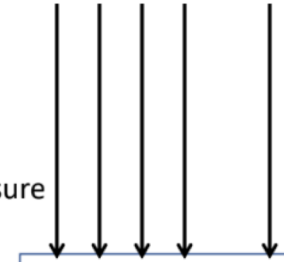
Garbage Collection: Blocks containing too many invalid pages will be reclaimed by copying valid data out of them, and the reclaimed blocks will be placed to free block pool to be re-used

Wear Levelling: Distribute writes/erasures evenly across flash memory by swapping hot and cold blocks

How to Program/Write Data to Flash

All ‘1’ initially: 1 1 1 1 1 1 1 1

Write 0x0b:
Rule:
1) 1 can be programed to 0
2) 0 cannot be programed to 1 except performing an erasure



0 0 0 0 1 0 1 1

Write 0x0f?

Need to erase to all ‘1’ first

0 0 0 0 1 1 1 1

Full Disk Encryption (FDE)

1. Everything on disk is encrypted
2. Totally transparent to users
3. Can defend against a passive attacker

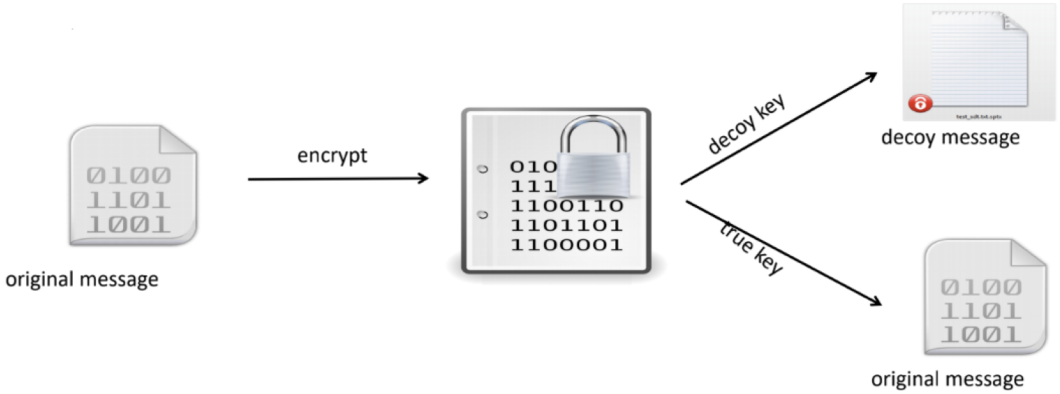
Coercive Attack:

An attacker forces the device owner to disclose the decryption key

FDE is vulnerable to a coercive attack

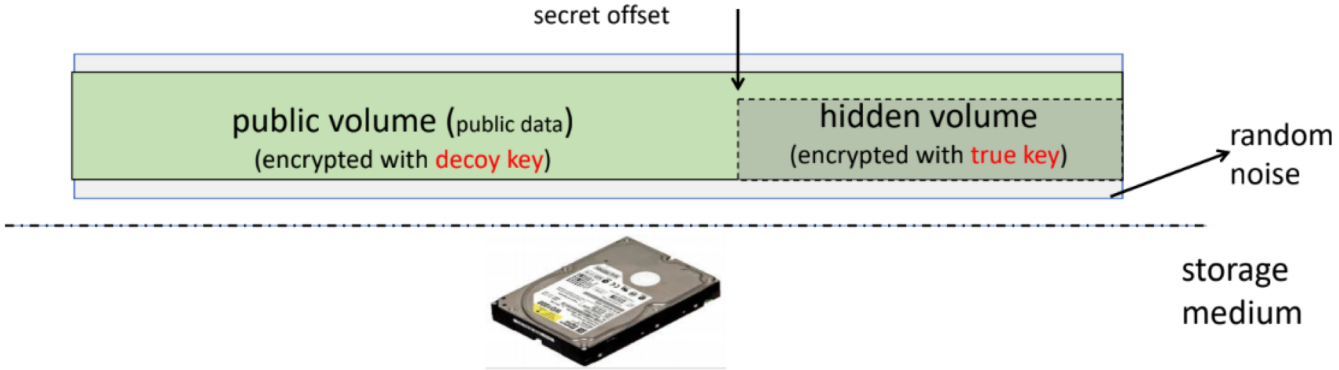
Plausibly Deniable Encryption (PDE):

- A crypto primitive designed for mitigating coercive attacks
- Disclose the decoy key
- Keep the true key secret



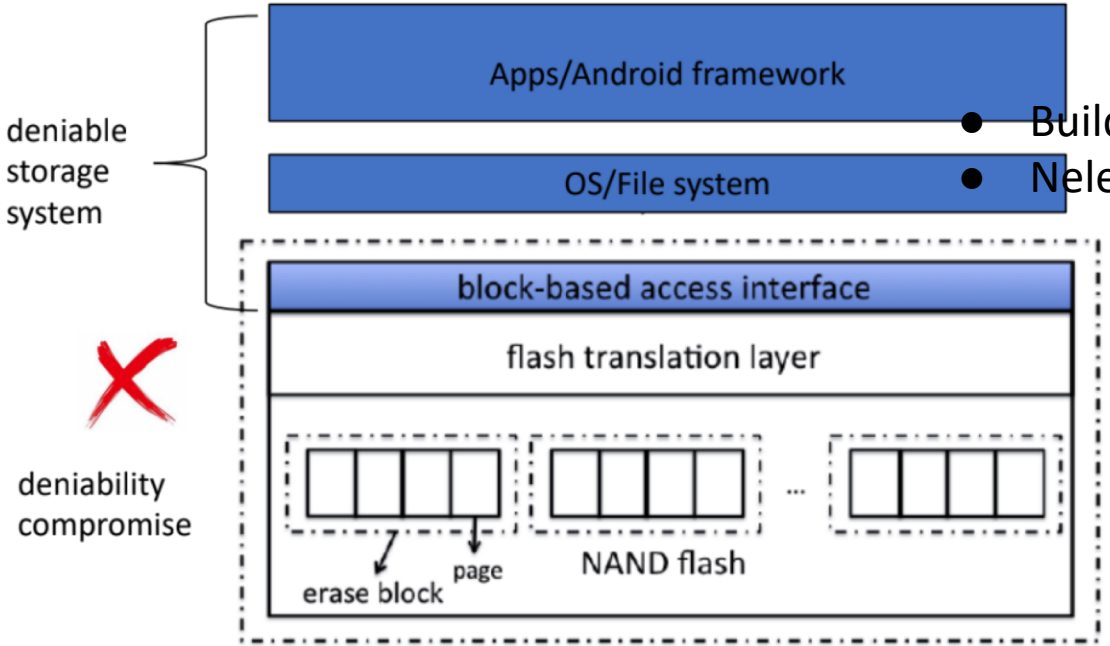
Hidden Volume-based PDE

- Initialize flash device with randomness
- Encrypt public volume with decoy key
- Encrypt hidden volume with true key



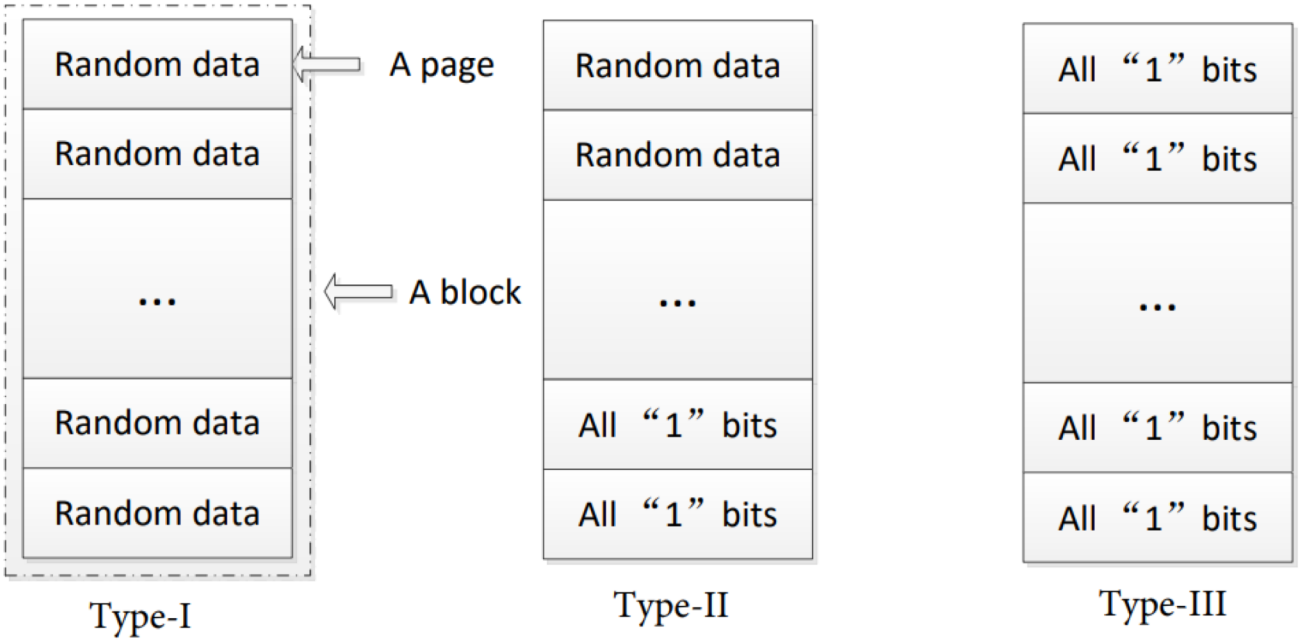
The encrypted hidden volume cannot be differentiated from randomness

Attack Scenario

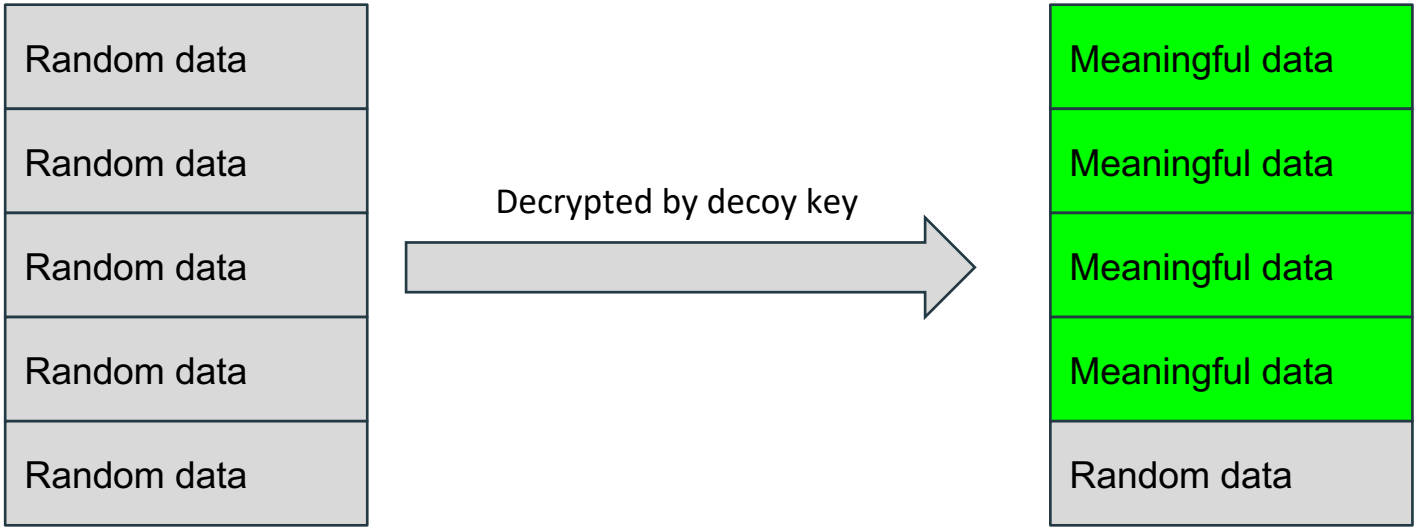


- Build on the block device layer
- Nelect the special nature of flash

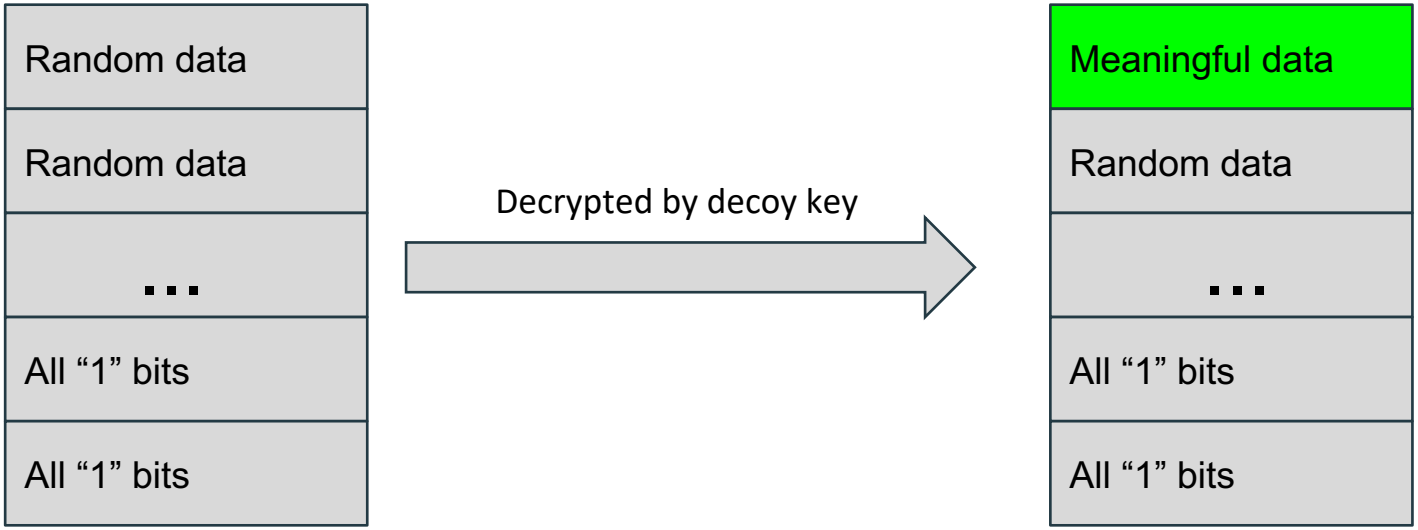
Attack Scenario (cont.)



Attack Type 1 Blocks



Attack Type 2 Blocks



Design of DEFTL

Overview

- How to prevent the sensitive data from being leaked to a coercive adversary ?
- How to prevent the hidden sensitive data from being overwritten by the non-sensitive data?

Four Block Types:

A: Do not store any valid public or hidden data

B: Do not store any valid public data, but store valid hidden data

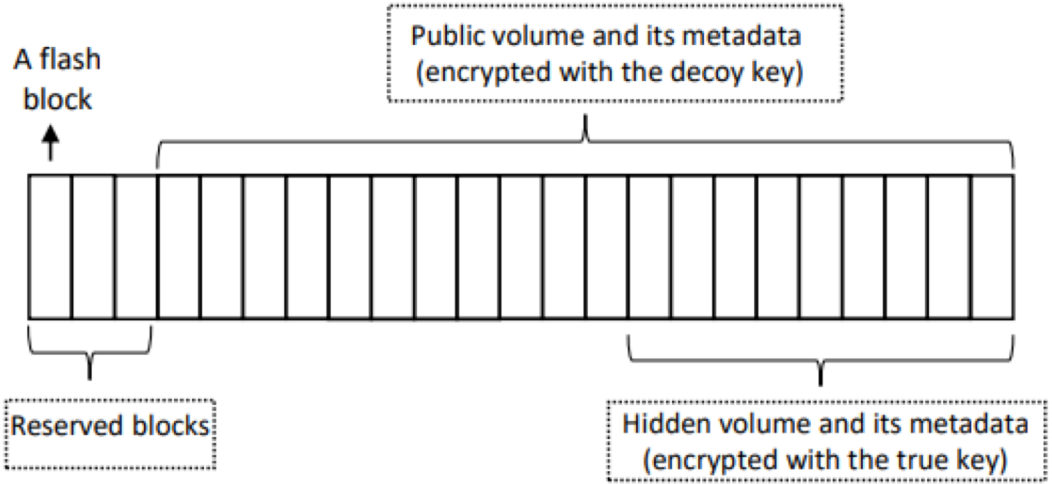
C: Contain both the valid public volume pages and the invalid public volume pages

D: Only contain valid public volume pages

Dirty Block Table: Stores the count of valid pages for each flash block, and organizes the blocks according to their counts in an increasing order

Initialization

- Filling the entire flash with randomness
- Initializing the public and the hidden volume

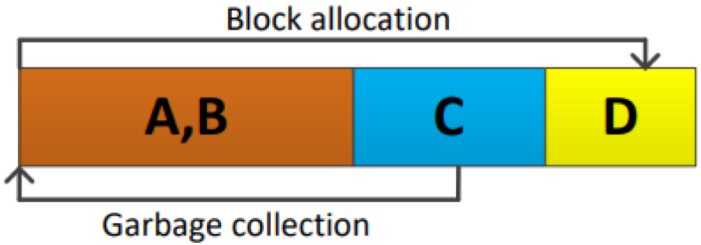


Public Mode

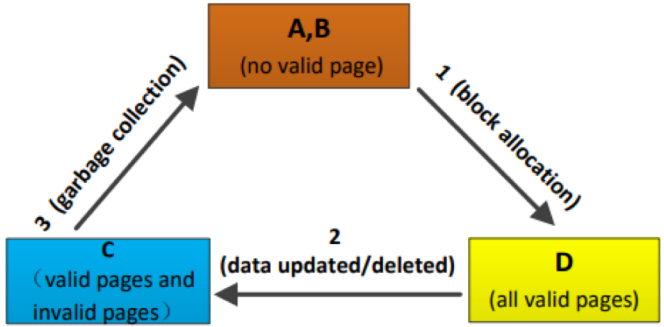
Block Allocation:

- Select the free blocks from the head of the dirty block table when a new write request comes

- Smartly manipulating the dirty block table of the public volume to ensure that it is more likely the blocks in state A will be allocated, rather than the blocks in state B



Dirty Block Table

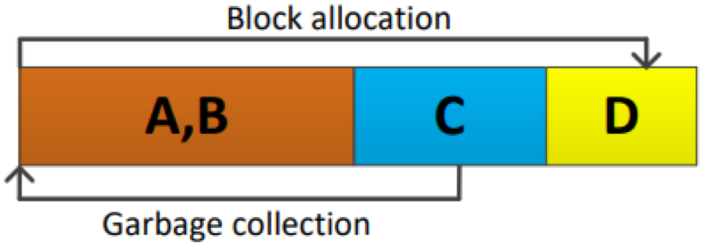


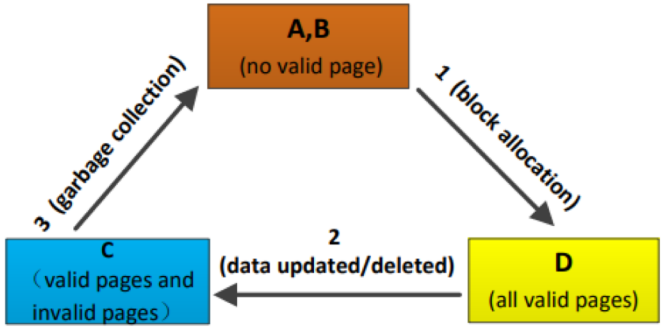
Public Mode (cont.)

Garbage Collection:

- Perform active garbage collection over blocks in state C

- Reclaim blocks in state C when threshold is reached and relocate them to the head of dirty block table





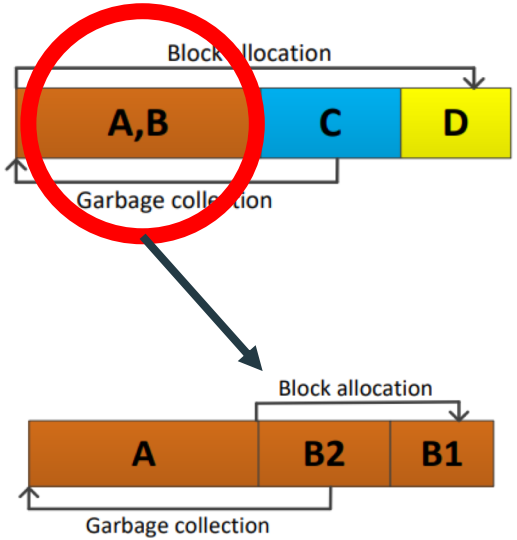
PDE Mode

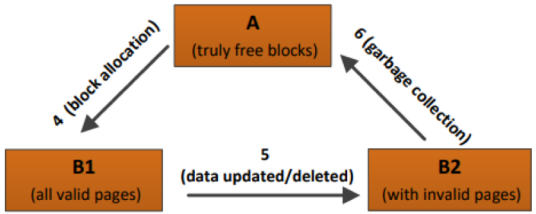
Block Allocation:

- Select free blocks from the dirty block table from the tail of the blocks in state A

B1: The blocks which only contain valid hidden data

B2: The blocks which contain both valid and invalid hidden data





PDE Mode (cont.)

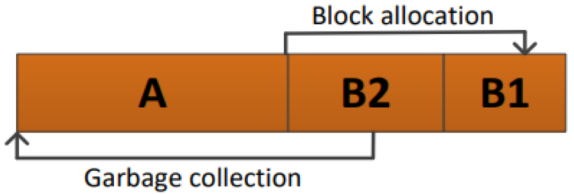
Garbage Collection:

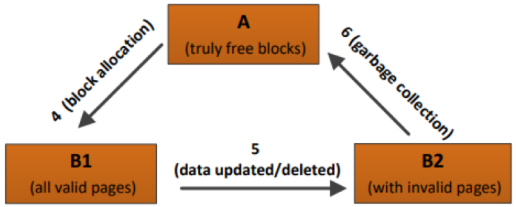
- Perform active garbage collection over blocks in state B2

- Reclaim blocks in state B2 when threshold is reached and relocate them to the head of dirty block table

B1: The blocks which only contain valid hidden data

B2: The blocks which contain both valid and invalid hidden data





- How to prevent the sensitive data from being leaked to a coercive adversary ?
  - Hidden Volume Technique
- How to prevent the hidden sensitive data from being overwritten by the non-sensitive data?
  - Public Mode: Use blocks from head
  - PDE Mode: Use blocks from tail

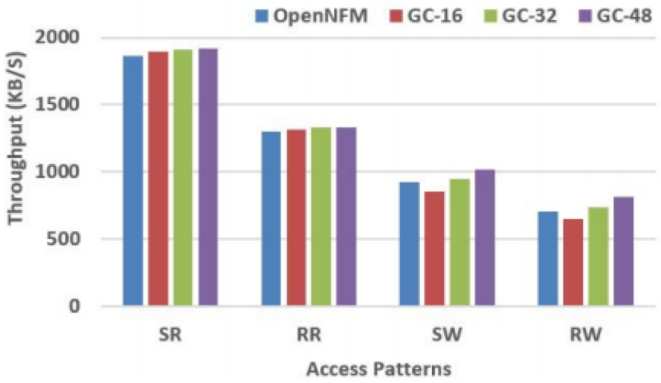
User Steps

Enter decoy password: Using the decoy password, DEFTL can derive the decoy key and use the decoy key to decrypt the public volume metadata

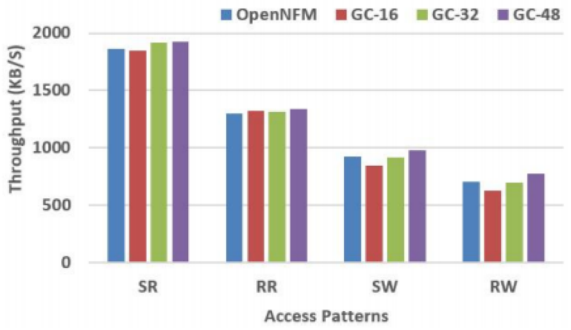
Enter true password: Using the true password, DEFTL can derive the true key and further localizes the hidden volume metadata, and decrypts them using the true key

Evaluation

Throughput:



OpenNFM vs. Public Mode



OpenNFM vs. PDE Mode

Evaluation (cont.)

Wear Leveling:

Wear Leveling Inequality (WLI): Calculating an appropriately normalized sum of the difference of each measurement to the mean. Small WLI indicates a better wear leveling performance.

Wear leveling threshold	Average erasures	WLI (%)
200	0.97	11.5
150	1.06	10.2
100	1.10	8.9
50	1.15	7.3

Questions