Techniques for Improving Security of Non-volatile Memories

Sparsh Mittal

IIT Hyderabad, India

Acronyms/terms

- NVM = Non-volatile memory
- NVMM = non-volatile main memory
- WL = wear-leveling
- WRE = write endurance
- Zeroing = writing zero to a entire block/memory
- CME = counter-mode encryption

Types of attacks

Stolen memory attack

Bus snooping attack

Write attack

Repeatedly write to a memory cell to reach its endurance to make it fail

Summary of attacks

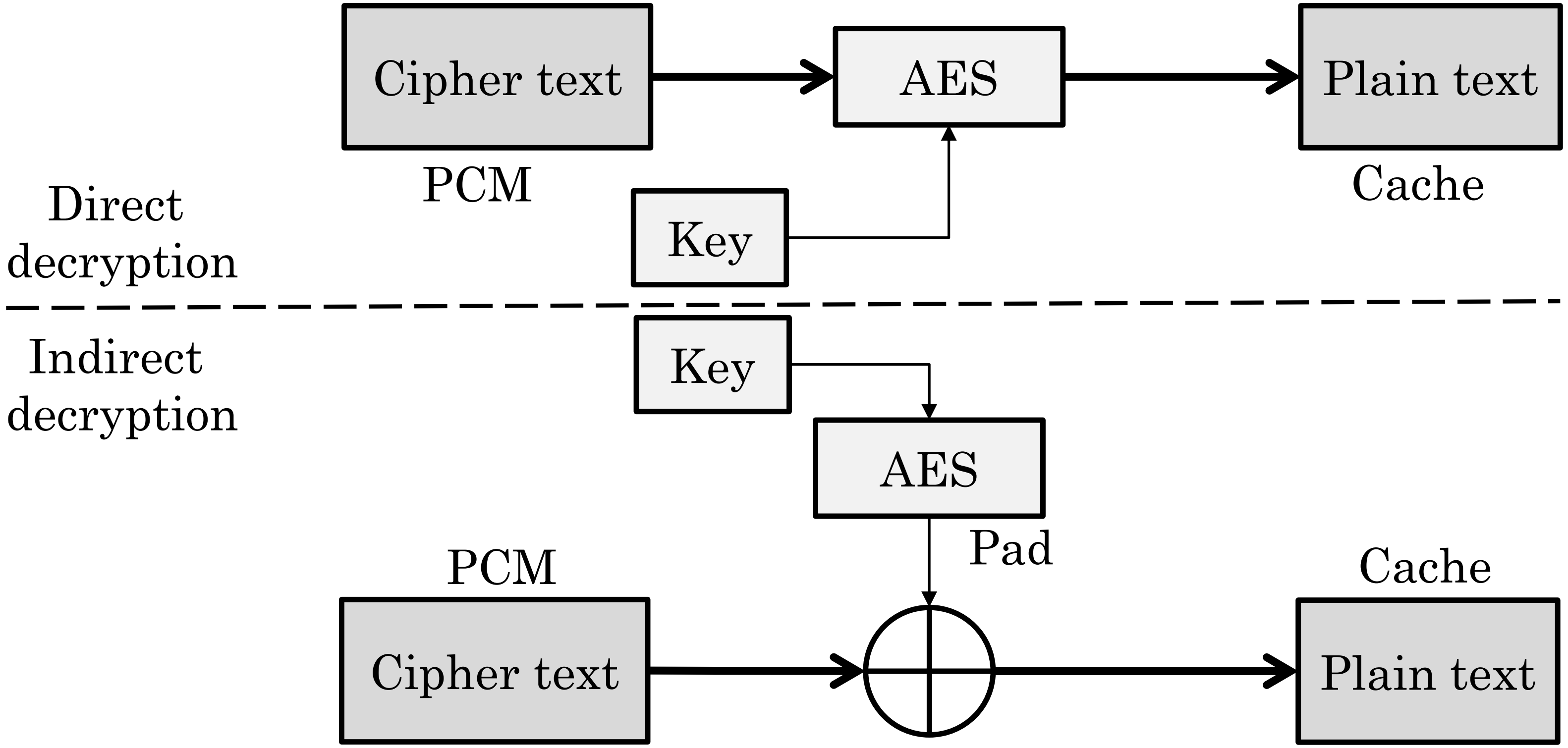
Name	Only in NVMs	Memory destroyed	Data-stolen	Mitigation
Stolen memory attack	No, but more severe in NVMs due to data-retention property	No	Yes	Encryption
Bus snooping attack	No	No	Yes	Encryption
Write attacks	Yes (due to limited WRE of NVMs)	Yes	No	WL and write-reduction

Data shredding

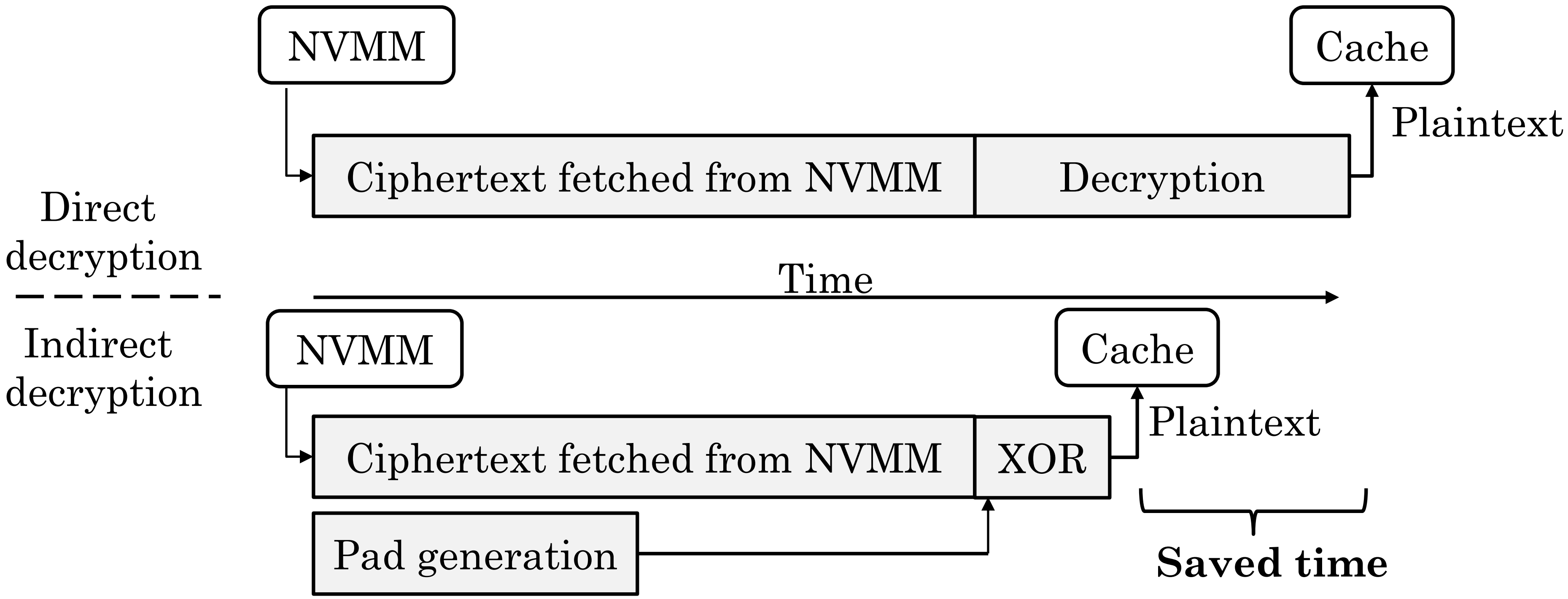
- Destroying contents of a physical page before allocating this page to another process
- It avoids data-leak between two processes or virtual machines
- Performed frequently => responsible for large number of memory writes
- Generally, data shredding is achieved by writing zero on each cell of the page

Background on Encryption and Decryption

Direct vs. indirect decryption



Direct vs. indirect decryption (latency impact)



Indirect decryption allows hiding the latency and hence, it is used widely.

Counter mode encryption (1 of 2)

Using identical key for all blocks allows adversary to compare encrypted lines to identify the lines storing the same value and then launch a dictionary-based attack.

Idea: use address of each line along with the key for doing encryption. Overall key becomes unique => thwart stolen memory attack.

Limitation:
successive writes to a line can still be monitored using the "bus snooping attack".

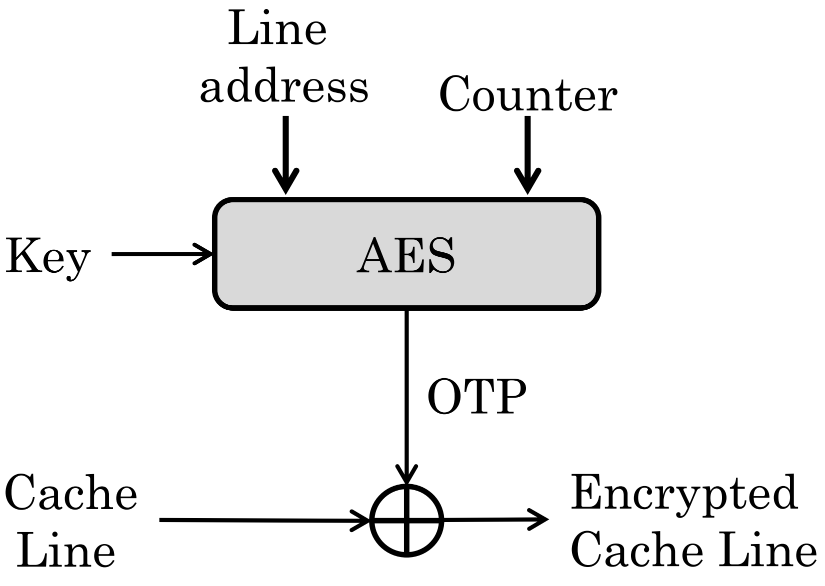
Counter mode encryption (2 of 2)

- Idea: To avoid this, use a per-line counter with key and line address for performing encryption.

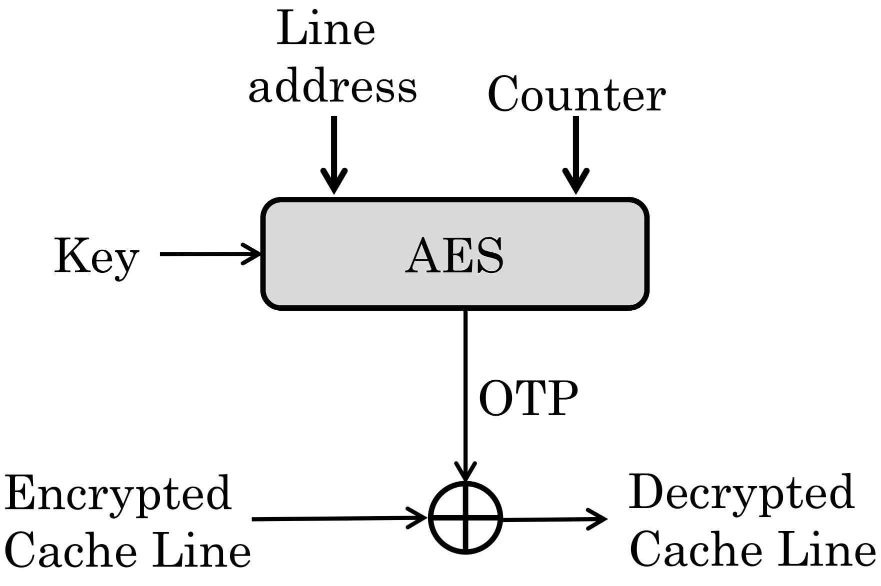
- Generate counter value from a function which produces distinct values over a long period.
- In practice, the counter is simply incremented by one on each write.

- This ensures uniqueness of the overall key for every write to every line
- => insulates the memory from both “stolen memory” and “bus snooping” attacks.

(Indirect) Counter-mode Encryption/decryption operations



(a) Encryption



(b) Decryption

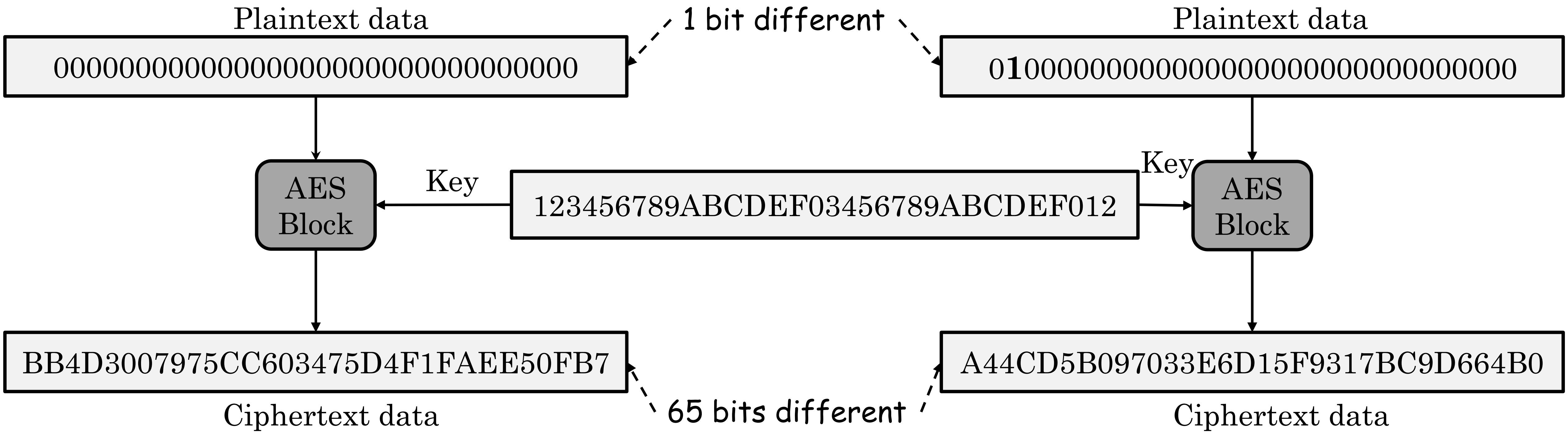
Avalanche effect of encryption

- A good encryption algorithm has the property of diffusion whereby even a small change in plaintext changes a large number of bits in ciphertext.
- Also called avalanche effect

- Diffusion property ensures that for two plaintexts with only minor difference, their ciphertexts have no relationship.

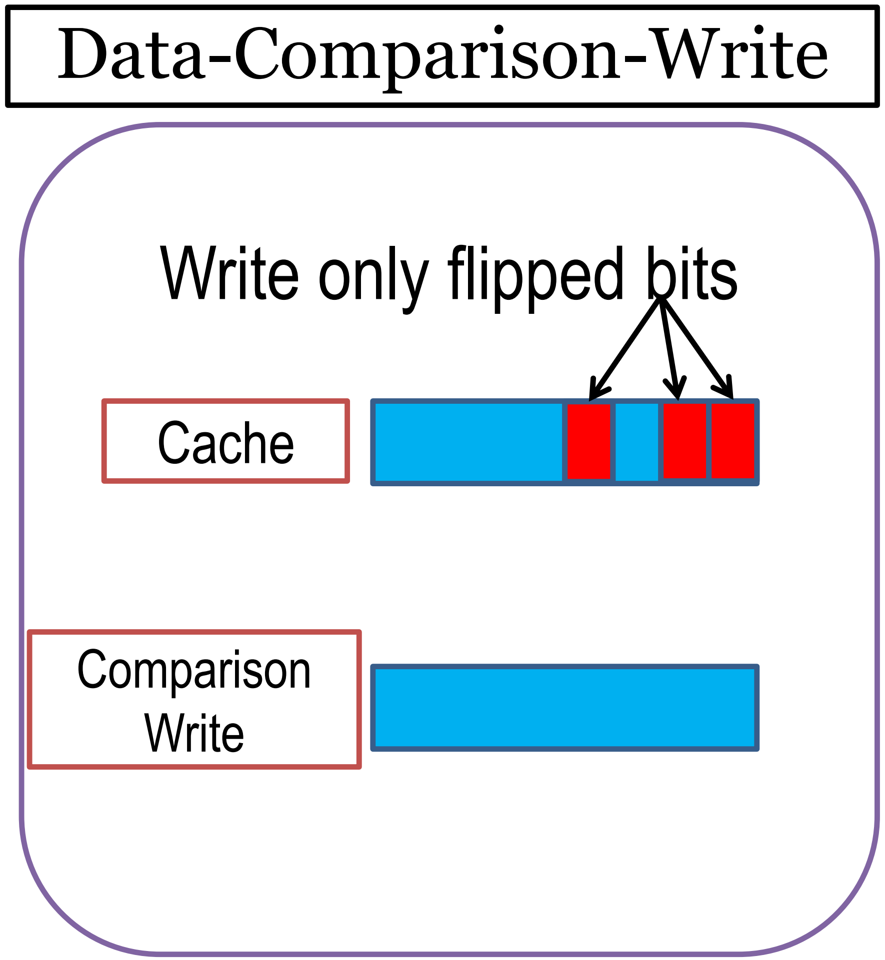
- => even a minor change in plaintext or update of the counter in CME changes the ciphertext completely

Avalanche effect of encryption



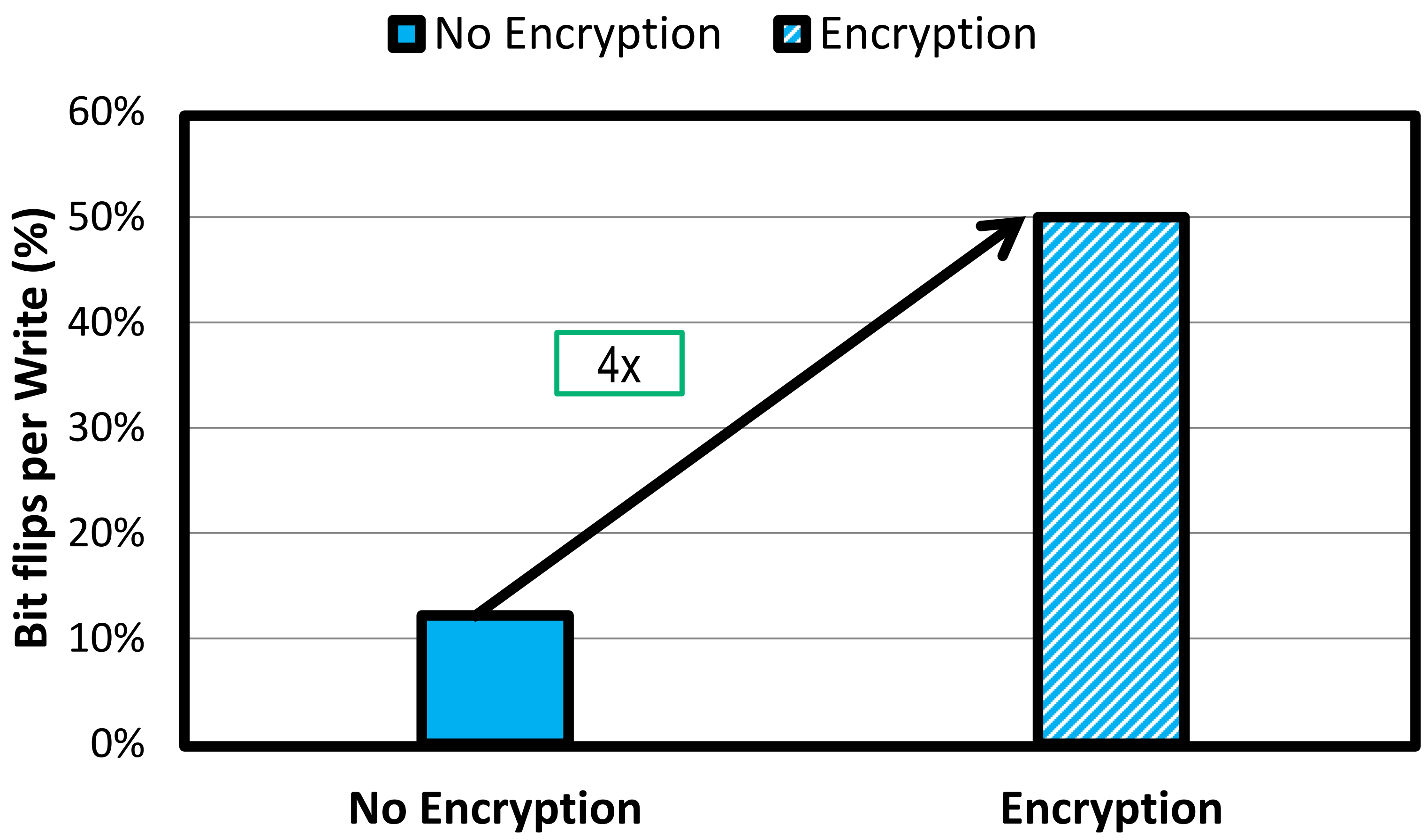
Challenges of Securing NVM

Typical Write optimizations in unencrypted PCM



It reduces bit-flips per write to 10-12%

Avalanche effect nullifies the advantage of data-comparison write

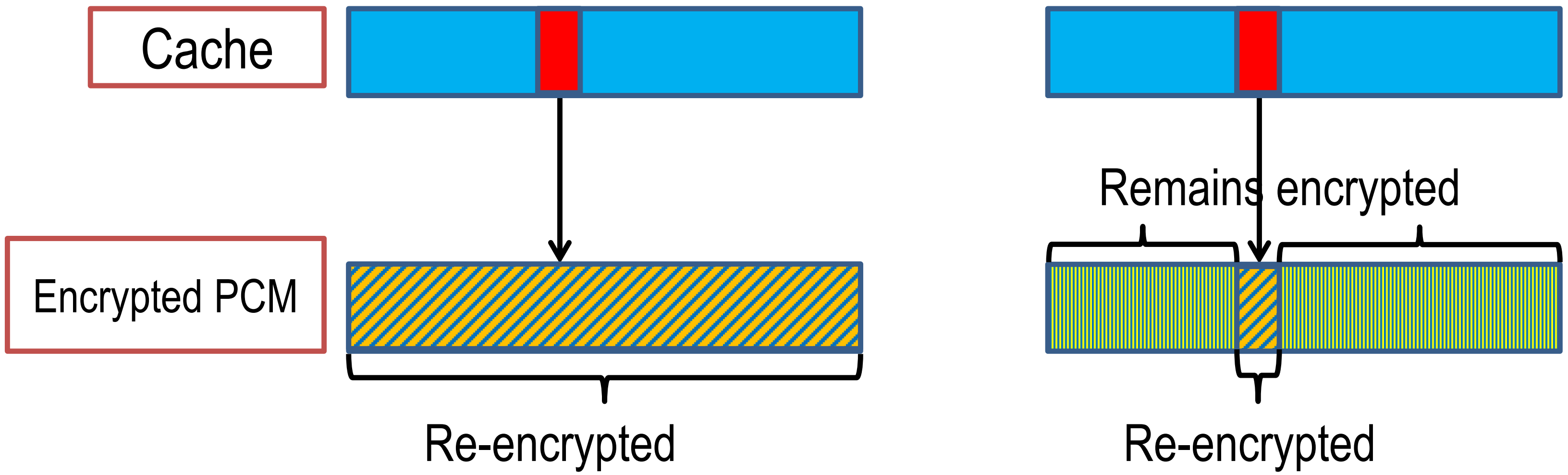


Encryption increases bit flips from 12% to 50% (4x!)

Techniques for Reducing Encryption Overhead

Reduce bit-flips due to avalanche effect of encryption

What if we re-encrypt only modified words?



Reduce bit flips by re-encrypting only modified words

18

Techniques for Mitigating Write Attacks

Wear-leveling (1 of 2)

- In an NVM which does not utilize WL, the LA-to-PA mapping remains fixed which allows the adversary to easily launch write attack
- WL techniques are used to distribute the write traffic uniformly to avoid early wear-out of few cells
- Idea: migration operations introduced by WL can also be leveraged to hide the actual location of a data-item from an outsider
- WL techniques proposed for improving NVM lifetime can be redesigned to improve NVM security also

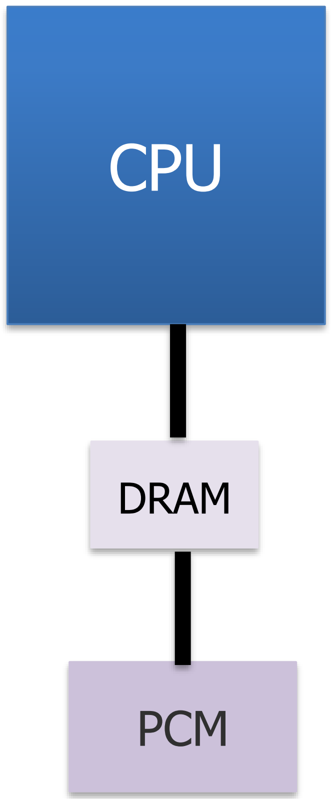
Wear-leveling (2 of 2)

- Limitation: Simple WL techniques remap the blocks in a systematic and predictable manner which can be inferred by the adversary.

- Idea: dynamically change the remapping relationship in WL over time.
- This makes it difficult for an adversary to infer location of a PA inside the memory
- It forces attacker to write to many cells which slows-down the attack.

Write-reduction

- Use DRAM as a cache before PCM
- Provides performance, energy and security advantage



Change rate of wear-leveling based on write-attack intensity

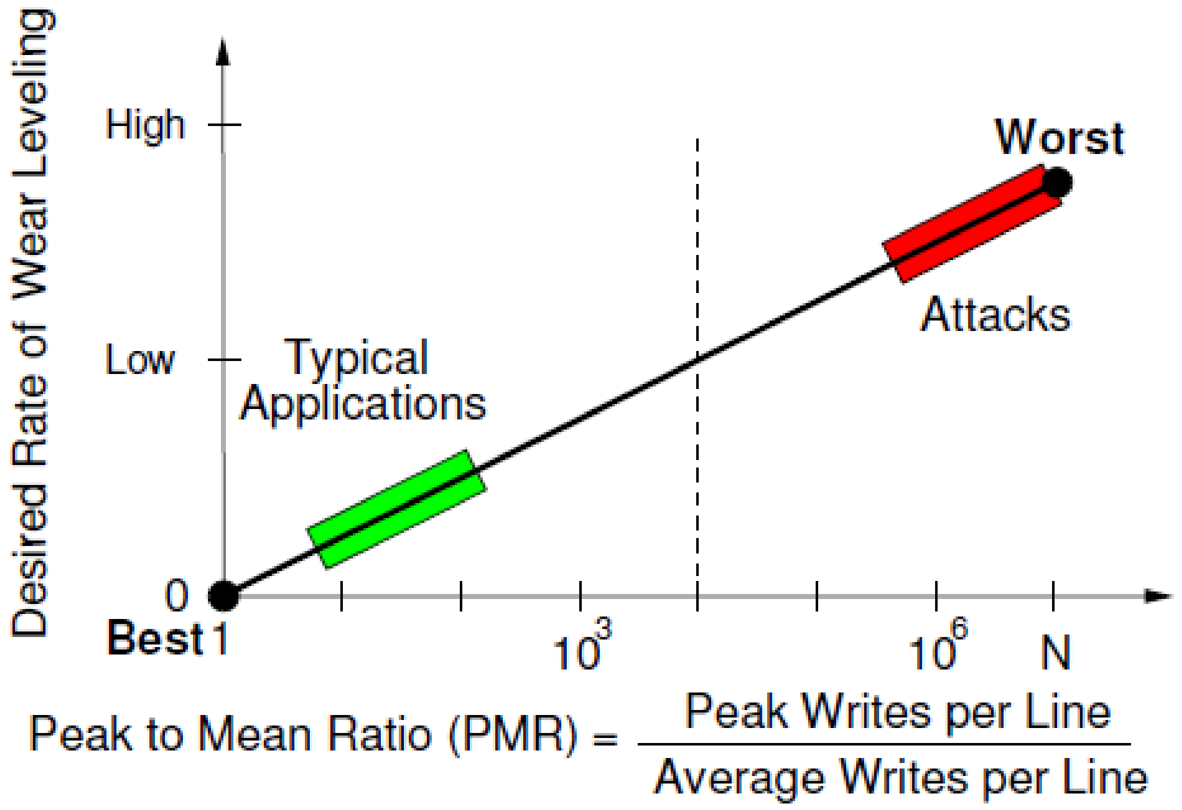


Figure 1. Rate of wear leveling for different types of write stream

Courtesy: M. Qureshi

Techniques for reducing overhead of data-shredding

Using encryption to write random data for data-shredding (1 of 2)

- Challenge: Data-shredding is especially costly for write-agnostic NVMs
- Observation: writing any random/unintelligible data to a page before its reuse has the same effect as zeroing it, since both ensure that no meaningful data can be read from the page
- In an un-encrypted memory, writing random data provides no advantage over writing zero-data
- However, in an encrypted memory, changing the encryption key from, say Key1 to Key2 ensures that decrypting the page leads to meaningless data

Using encryption to write random data for data-shredding (2 of 2)

- This allows initialization of a reused page with random-data without any overhead
- Due to the diffusion property of encryption, the new decrypted data has no correlation with the original data
- To change the key, just change the counter