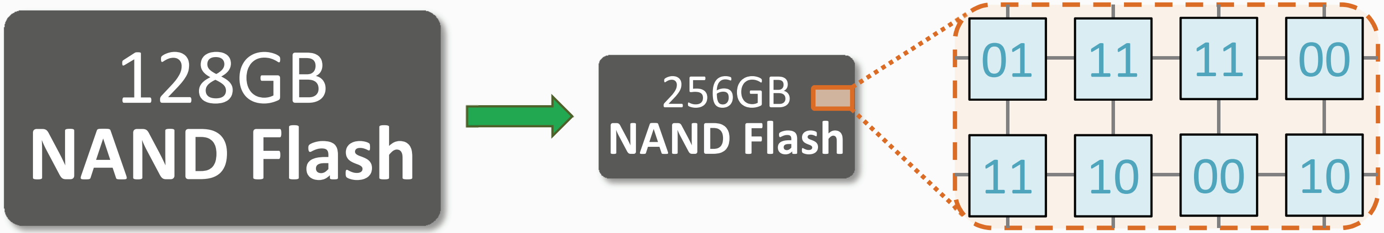
# Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

Carnegie Mellon University SEAGATE ETHzürich SAFARI

## HPCA Session 3A – Monday, 3:15 PM, Salon F



NAND flash scaling: **shrink size** of each flash cell, **store *two bits*** per cell

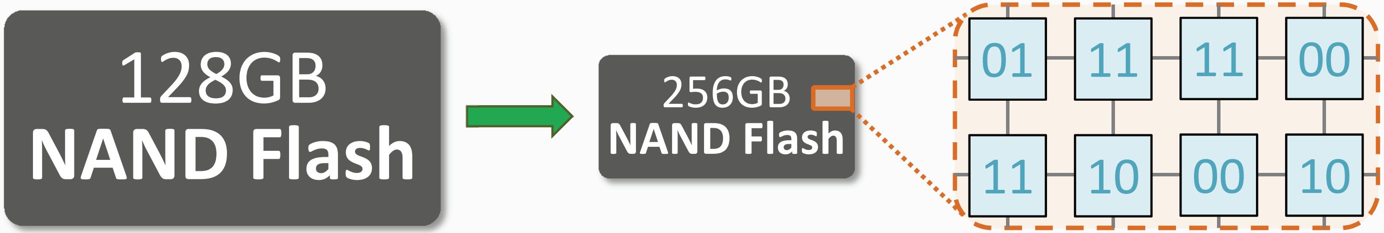
# Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

**HPCA Session 3A – Monday, 3:15 PM, Salon F**

Carnegie Mellon University

SEAGATE

ETH zürich SAFARI

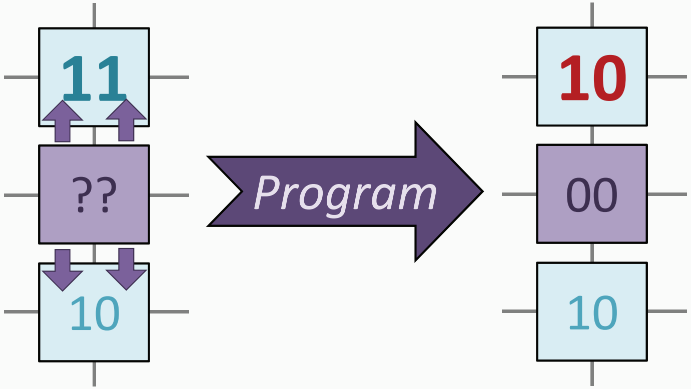


128GB **NAND Flash**

256GB **NAND Flash**

| | | | |
|---|---|---|---|
| 01 | 11 | 11 | 00 |
| 11 | 10 | 00 | 10 |

NAND flash scaling: **shrink size** of each flash cell, **store *two bits*** per cell



11

??

10

*Program*

**10**

00

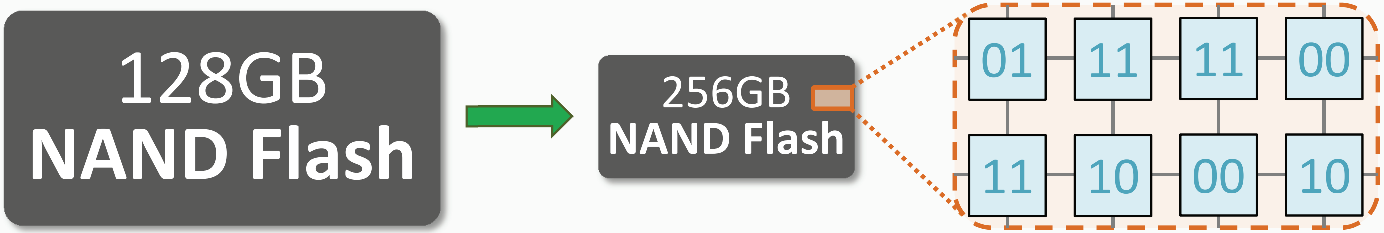
10

As the cells become smaller, they ***interfere*** with each other during **programming**…

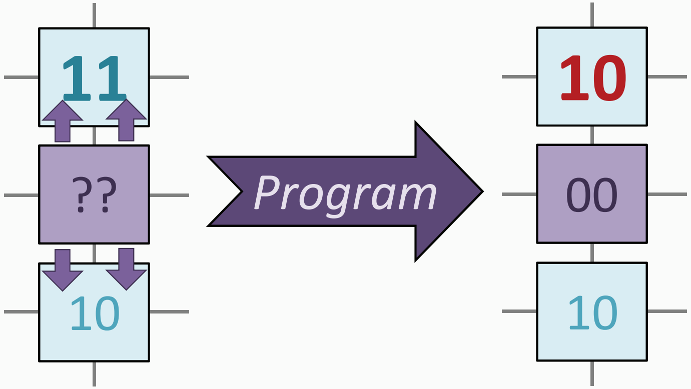
# Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

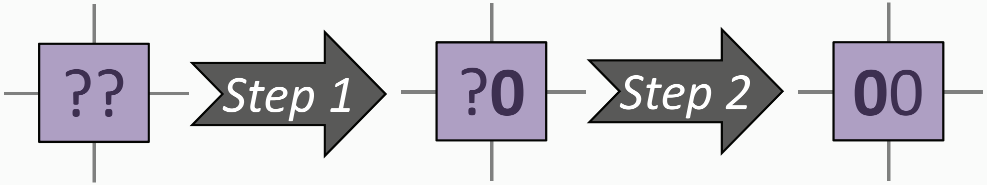
## HPCA Session 3A – Monday, 3:15 PM, Salon F

Carnegie Mellon University
SEAGATE
ETHzürich
SAFARI



NAND flash scaling: **shrink size** of each flash cell, **store *two bits*** per cell





As the cells become smaller, they ***interfere*** with each other during **programming**…

…to reduce interference, today's MLC NAND flash chips use ***two-step programming***

# Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

**HPCA Session 3A – Monday, 3:15 PM, Salon F**

Carnegie Mellon University

SEAGATE

ETH zürich

SAFARI

Using **real MLC NAND flash chips**,
we show that two-step programming introduces
**new reliability and security vulnerabilities**

# Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

## HPCA Session 3A – Monday, 3:15 PM, Salon F

Carnegie Mellon University

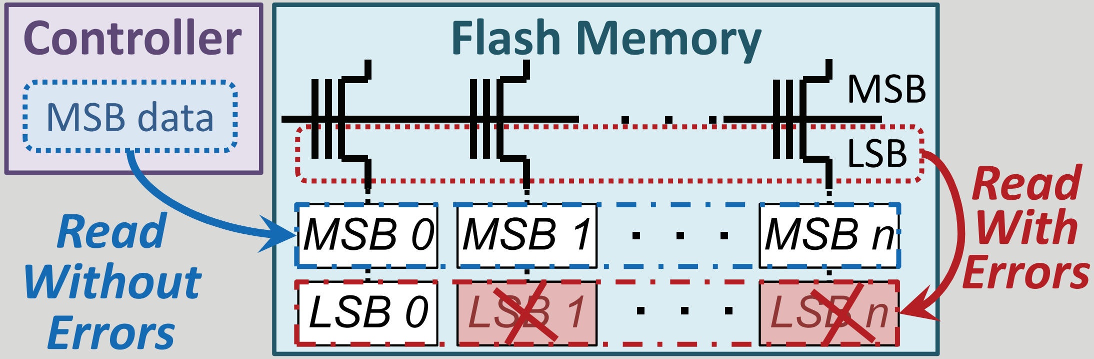
SEAGATE

ETH zürich

SAFARI

Using **real MLC NAND flash chips**, we show that two-step programming introduces **new reliability and security vulnerabilities**

We find that **cells with only one bit programmed** are **more vulnerable** to **interference during reads and writes** than fully-programmed cells



# Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

**HPCA Session 3A – Monday, 3:15 PM, Salon F**

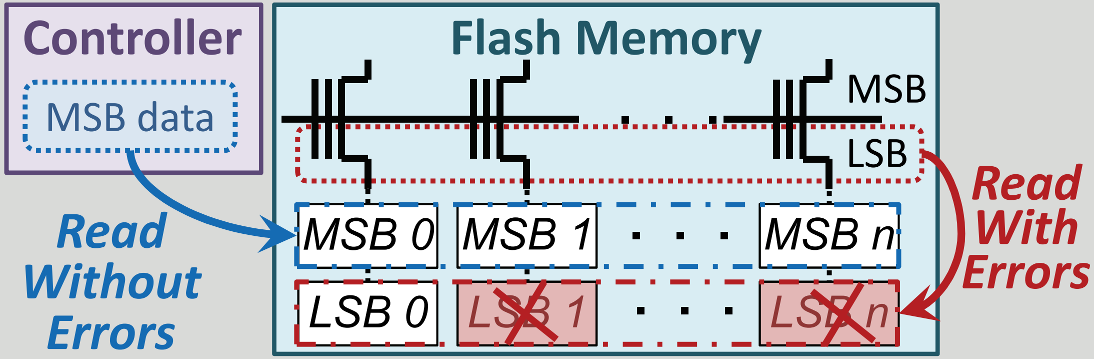
Carnegie Mellon University

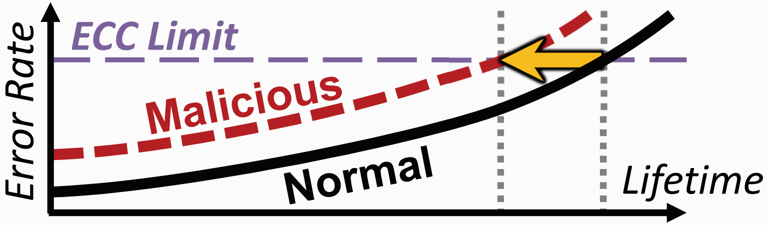
SEAGATE

ETHzürich SAFARI

## Using **real MLC NAND flash chips**, we show that two-step programming introduces **new reliability and security vulnerabilities**

We find that **cells with only one bit programmed** are **more vulnerable** to **interference during reads and writes** than fully-programmed cells





Vulnerabilities can be **exploited** to corrupt data and **reduce flash lifetime**

We propose **three solutions**
to minimize vulnerabilities at **negligible latency overhead**

## We propose three solutions to minimize vulnerabilities at negligible latency overhead

One solution completely eliminates vulnerabilities

*4.9% increase in flash programming latency*

# Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

**HPCA Session 3A – Monday, 3:15 PM, Salon F**

Carnegie Mellon University SEAGATE ETHzürich SAFARI

## We propose **three solutions** to minimize vulnerabilities at **negligible latency overhead**

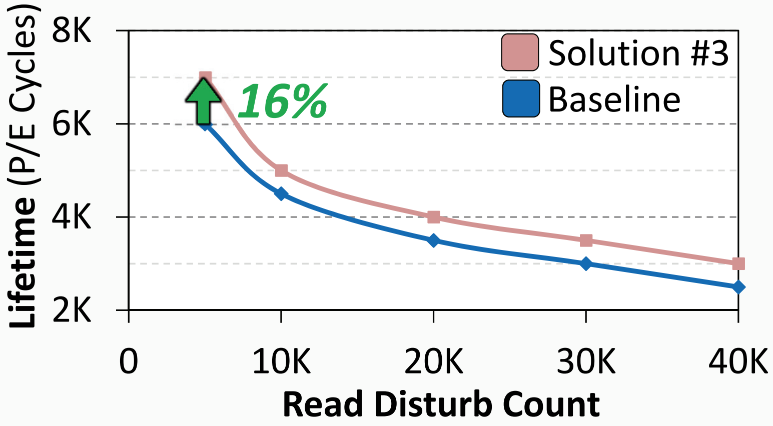
One solution **completely eliminates vulnerabilities**

*4.9% increase in flash programming latency*

Two other solutions **mitigate vulnerabilities**

*No increase in flash latency, errors not completely eliminated*

***Increases flash lifetime by 16%***



# Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

**HPCA Session 3A – Monday, 3:15 PM, Salon F**

Carnegie Mellon University

SEAGATE

ETHzürich SAFARI

## We propose three solutions to minimize vulnerabilities at negligible latency overhead

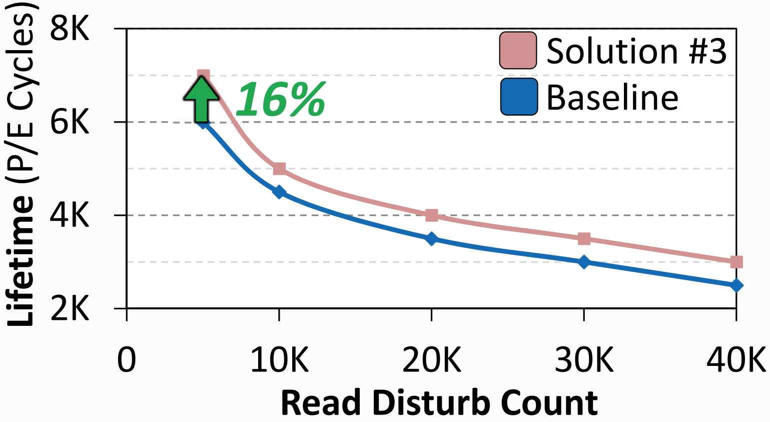
One solution **completely eliminates vulnerabilities**

*4.9% increase in flash programming latency*

Two other solutions **mitigate vulnerabilities**

*No increase in flash latency, errors not completely eliminated*

***Increases flash lifetime by 16%***



**Want more? Come to our talk! Read our paper!**

*Authors:* Yu Cai, **Saugata Ghose**, Yixin Luo, Ken Mai, Onur Mutlu, Erich F. Haratsch