Using Rowhammer for Fault Analysis of Block Ciphers and a Mitigation Technique thereof

Anirban Chakraborty¹, Sarani Bhattacharya², Sayandeep Saha¹, Manaar Alam¹ and Debdeep Mukhopadhyay¹

1. Indian Institute of Technology Kharagpur, India
2. Katholieke Universiteit Leuven, Belgium
Exploiting Page Frame Cache

**Adversary**

Running on a standard system, requests some amount of memory. The buddy allocator scans the list of available page frame blocks and finds out a block that can satisfy the request.

De-allocates a pre-chosen memory page (by using `unmap` function). The ‘unmapped’ page resides in the page frame cache of the CPU.

The Adversary successfully coerced the operating system to allocate selected memory pages to the Victim process.

**Victim**

Sends a request for additional memory pages. Due the presence of PFC, the buddy system will serve the page that was ‘unmapped’ by the adversary.
The bin-partitioning

- Access the first page and put it in $bin_0$
- Next, access the next page and the first page simultaneously and check their access times

- If access time for next page is more than a pre-defined threshold, it signifies a row conflict
- The pair of pages must be located in the same bank but different rows.
- In that case, put the new page in $bin_0$
- If the access time is less than threshold, put it in the next bin, i.e., $bin_1$
ExplFrame on OpenSSL AES T-tables

• OpenSSL v1.1.1 AES T-tables T0 – T3
• The adversary waits for the victim to load the T-tables into memory
• Due to the presence of PFC, the T-tables will be allocated in the same vulnerable page
• Once one of the T-tables is placed in a freed page, the adversary again starts Rowhammer-ing on the same bin.

The attacker allocates some memory, partitions it into bins and conducts rowhammer on a bin to find vulnerable pages.
The attacker unmaps the vulnerable pages. Due to the property of page frame cache, atleast one of the AES T-tables gets allocated in a vulnerable page.
The attacker once again conducts rowhammer in the same bin. After some time, a fault is induced in one of the T-tables located in a vulnerable page.

Victim

A flip!

1 June 2021

Anirban Chakraborty/ IIT Kharagpur
Central Idea – Guess a part of the key and partially decrypt up to the round where bias is observed

Identify the bias using Squared Euclidean Imbalance (SEI) test

- We target the 9th round of AES computation
- Fault is induced in table T0 and we encrypt 20,000 plaintexts with the faulty T-tables
- The blue region in the convergence plot represents SEI values for wrong key
- The red line refer to SEI for correct key
Overview of our Diagnostic tool

Five Step Approach

1. Access
2. Address Map
3. Access Map
4. Data Preparation
5. Authorization
Profiler: Generating Access Patterns

- Identification of a probable Adversary
- Generation of Access Patterns
Analyzer: Determining Rowhammer Process

Offline Phase

• Obtains data for both benign and malicious patterns
• Trains a CNN to differentiate patterns that induce bit flips

Online Phase

• Obtains access pattern for unknown process
• Uses the already trained CNN to classify the process
Future Works

• Explore the ExplFrame attack on ECC protected memory
• Can we replace supervised learning used in the diagnostic tool with an unsupervised one?
Thank You!