F4kEOUT: Leveraging Cache Contention for Cross-Thread Data Leakage

Skylar Gagnon
cng1022@usnh.edu
Advised by Dr. Dean Sullivan
Department of Electrical and Computer Engineering, University of New Hampshire, Durham NH

University of New Hampshire

Introduction

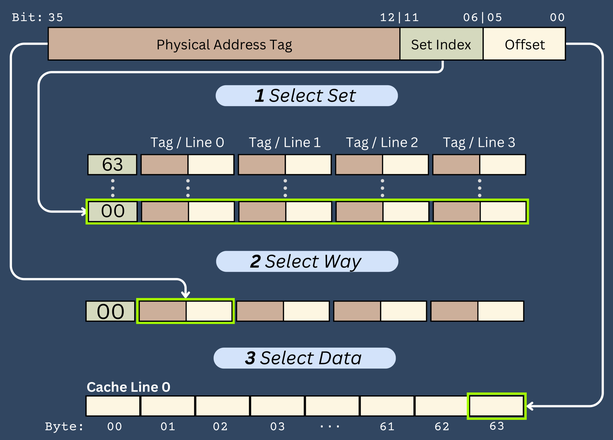
Motivation
With the rise of cloud computing and resource sharing, securing hardware means protecting billions from data theft.

Goals
- Develop a timing channel using 4k aliasing
- Expand to a 4k aliasing side channel attack
- Create a transient variant of the side channel

Why
- Prior works only leak crypto keys
- No transient variants of a 4k side channel
- Break transient attack mitigations
- Expose new transient attack surfaces

Background

The Cache: Where a computer keeps data for rapid retrieval. Two actions can be performed: storing data (write) or accessing data (read).



4k Aliasing: When multiple requests are made to the same cache set at the same time, some may be denied and reissued.

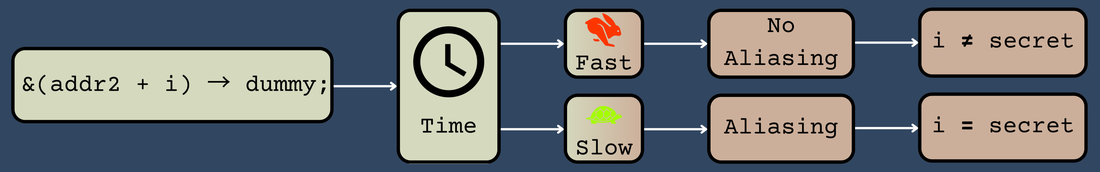
Attack Overview

Victim Thread

```
while(true) {
    &(addr1 + secret) ← dummy;
}
```

Attacker Thread

```
for(i = 0; i < 0x1000; i++) {
    start = time();
    &(addr2 + i) → dummy;
    end = time();
}
```



Experiments

Data Encoding

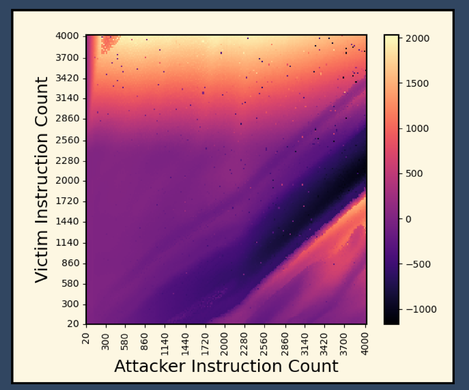
Only bits 11:6 cause aliasing, meaning only **6 bits** can be used for encoding.

Optimal Parameters

Parameter 1: Action Pairs
- Victim (V) or Attacker (A)
- Perform read or write action
- Addresses can alias (4k) or not (No-4k)

	A-Read		A-Write	
	V-Read	*V-Write*	*V-Read*	*V-Write*
No-4k	67 ± 2	71 ± 2	75 ± 2	84 ± 4
4k	64 ± 2	265 ± 10	77 ± 2	84 ± 4

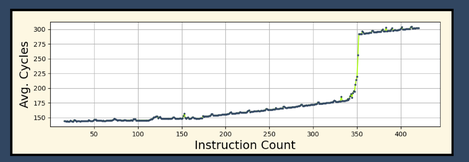
Parameter 2: Instruction Count
- The victim and attacker can both perform any number of actions (instructions), which in turn can affect the cycle penalty
- A **bigger penalty** means **better transmission**



Time Constraint

How many writes?
- Transient variant only
- Victim instruction count has a cap
- Maximum* is about **350 instructions**

* on an 11th Gen Intel Core i9-11900K



Results

F4kEOUT Attack
- Victim performs writes based on secret data
- Attacker monitors the cache using reads
- Based on timing, victim data is leaked

Accomplishments
- Successful across address spaces
- Successful across threads
- Able to transmit **1.023 KB/s** accurately

Recall	Precision	Accuracy	F1-Score
96.8%	99.0%	99.4%	97.8%

Evaluated on 11th Gen Intel Core i9-11900K running Ubuntu 22.04 with kernel version 6.5.0-17

Transient Variant
- Should be possible based on experiments
- Currently, no working proof of concept

Discussion

Action Pairs
- Higher No-4k average in (A-Read, V-Read) test
- Difference in averages in (A-Write, V-Read) test

Instruction Count
- Aliased instructions faster in some cases

Conclusions

With the current state of hardware, multitenant cloud services are **not safe**.

Software Mitigation Options
- Disabling Simultaneous Multi-Threading (SMT)
- One Party per Core Policy

Hardware Mitigation Options
- No aliasing across threads
- Cache Partitioning

Future Work
- Implement end-to-end attacks
- Explore unexpected behavior from experiments
- Resolve issues with the transient variant