CHALLENGES OF USING PERFORMANCE COUNTERS IN SECURITY AGAINST SIDE-CHANNEL LEAKAGE

Maria Mushtaq, Pascal Benoit, Umer Farooq

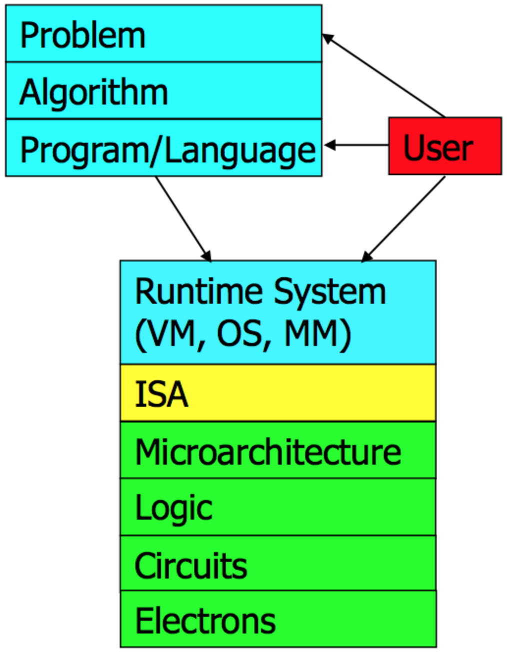
Outline

- Background
  - Side Channel Information Leakage –The Problem Statement
  - Performance-led Vulnerabilities in Intel's x86
  - Caches Leak Information
- Cache Side-Channel Attacks
  - Prime+Probe Attack
- Challenges of Using PCs
  - PCs Relevant to Cache-SCAs
  - Experiments and Results

Information Security

The Problem Statement

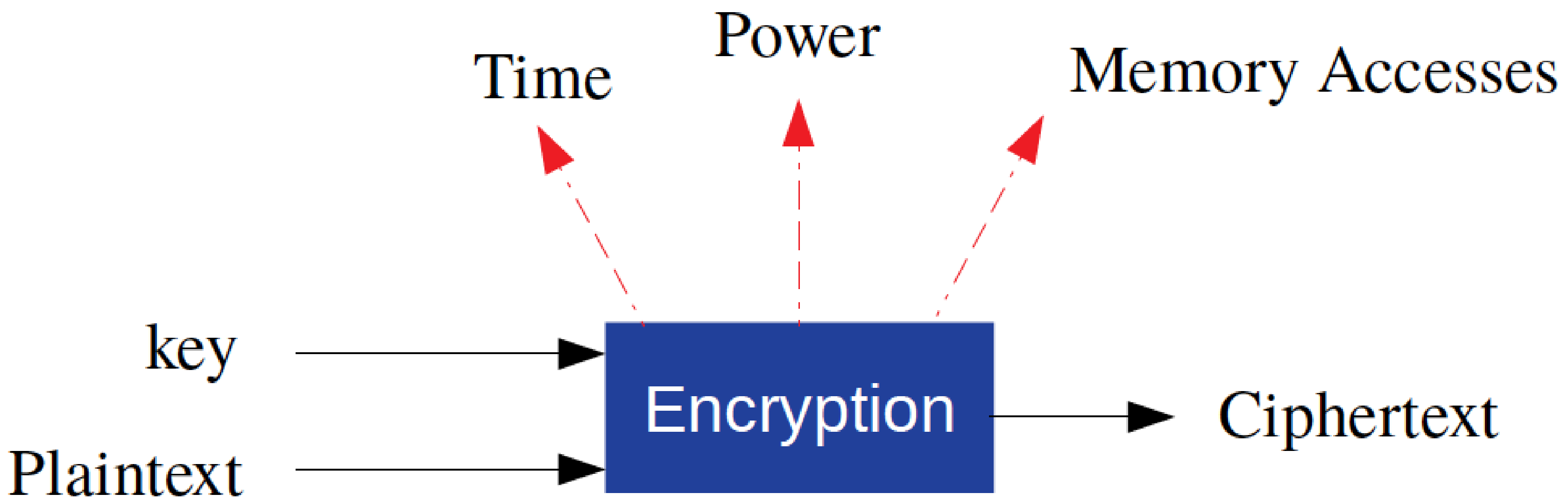
- Information leakage is possible *under safe* software!
  - Software is often encrypted by mathematically sound encryption techniques like RSA, AES, DES, ElGamal etc.



- Applications run over large untrusted computing base –underlying hardware is vulnerable
  - Micro-architectural features leak information on the state of program execution

Side Channel Leakage

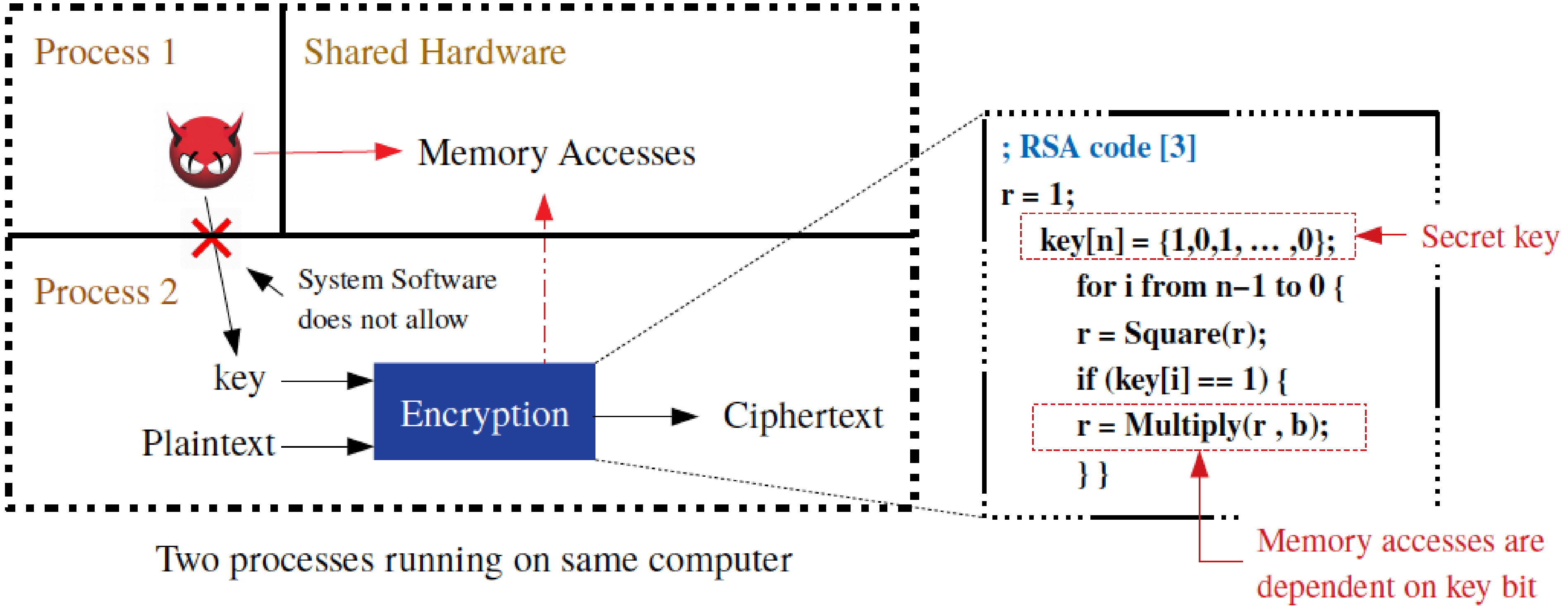
The Problem Statement



Mathematically Strong
Cryptographic Algorithms

Side Channel Leakage

The Problem Statement



```
; RSA code [3]
r = 1;
 key[n] = {1,0,1, … ,0};
    for i from n–1 to 0 {
    r = Square(r);
    if (key[i] == 1) {
    r = Multiply(r , b);
    } }
```

Two processes running on same computer

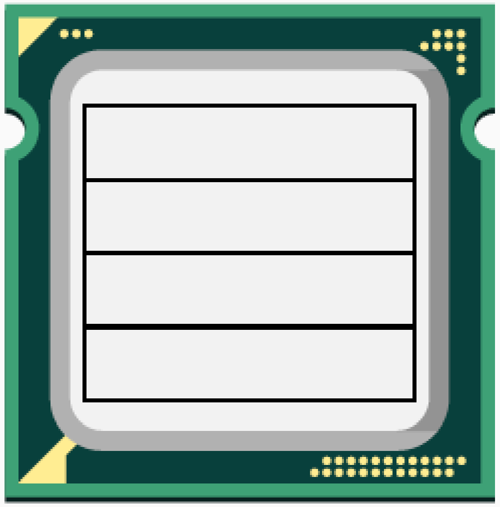
Vulnerabilities

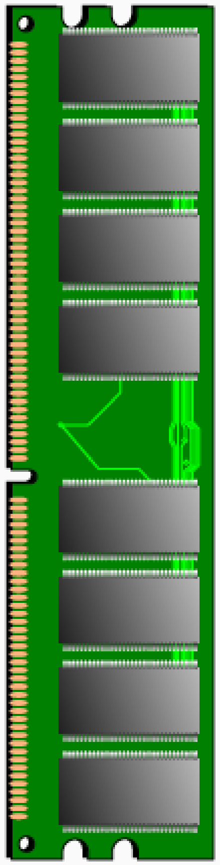
- Sharing –smaller footprint, better timing
  - Shared libraries, page sharing, de-duplication
- Inclusivity –coherency, performance
  - Intel's x86 performs complete reload of inconsistent data using *clflush* instruction (instruction privilege)
- Access time –caches help
  - Allows distinction b/w execution time & access patterns of processes
- Memory organization –better addressing modes
  - Exposes structured address space

Caches Leak Information

- Memory Access Time & Access Pattern

```
printf("%d", i);
printf("%d", i);
```

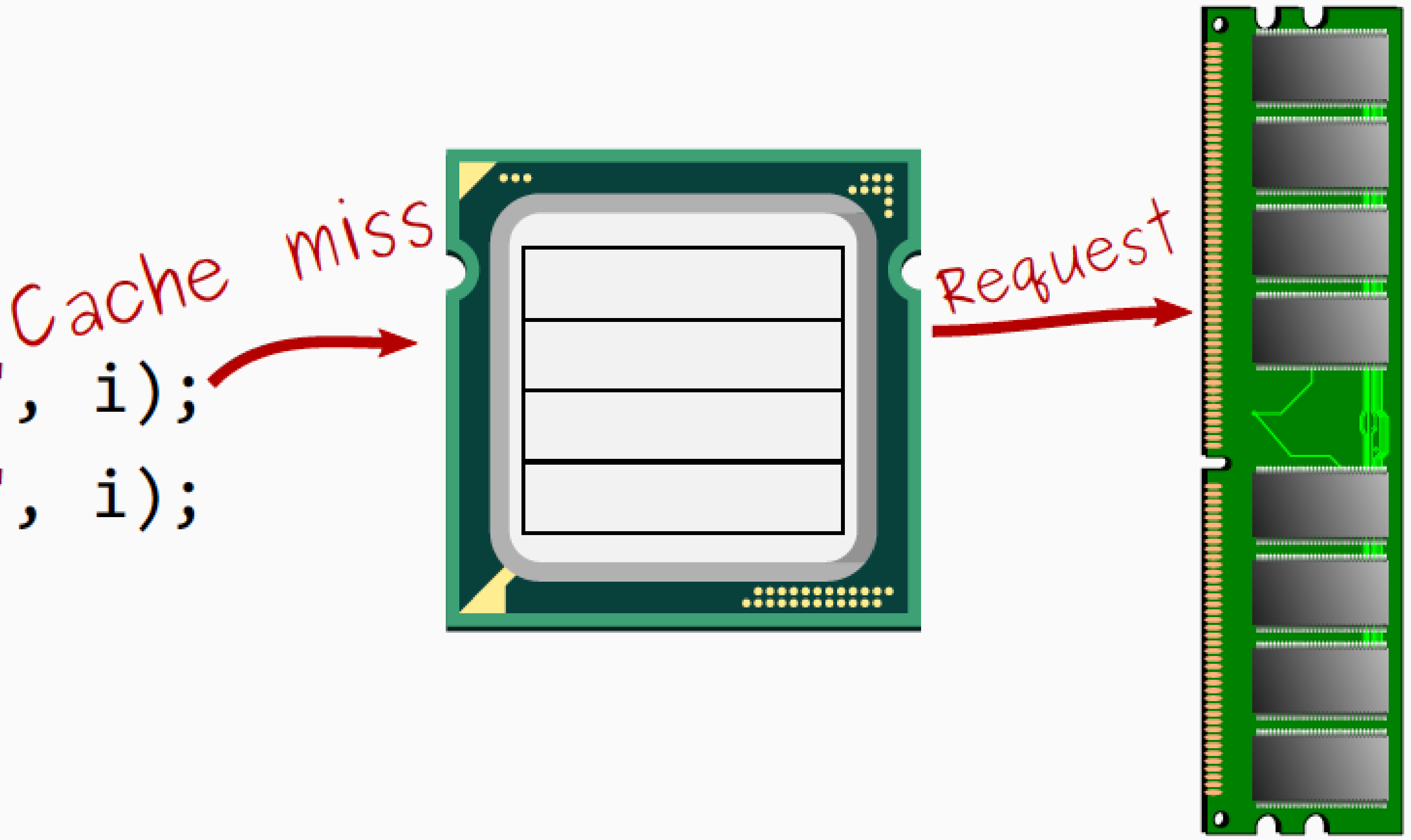




Caches Leak Information

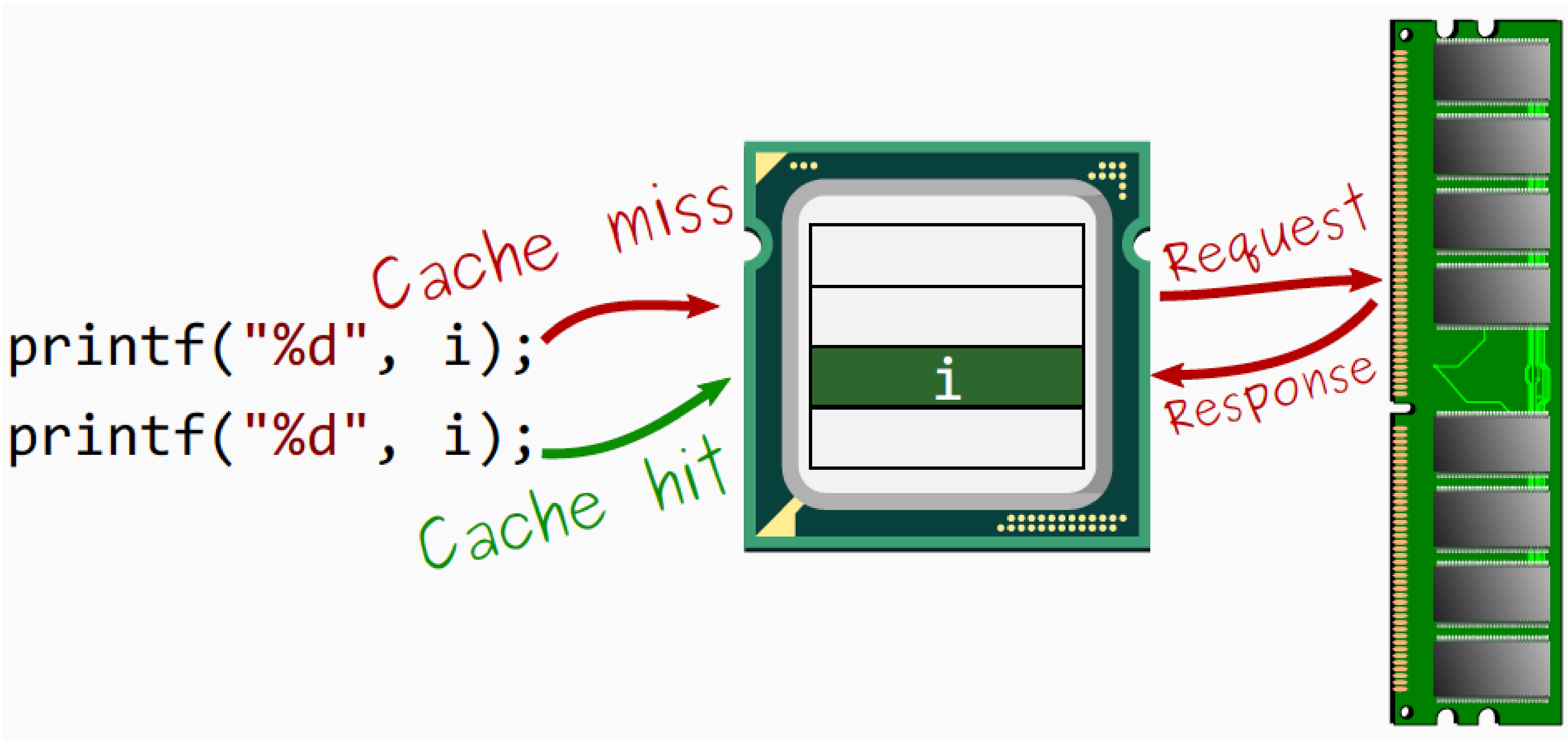
- Memory Access Time & Access Pattern

```
printf("%d", i);
printf("%d", i);
```



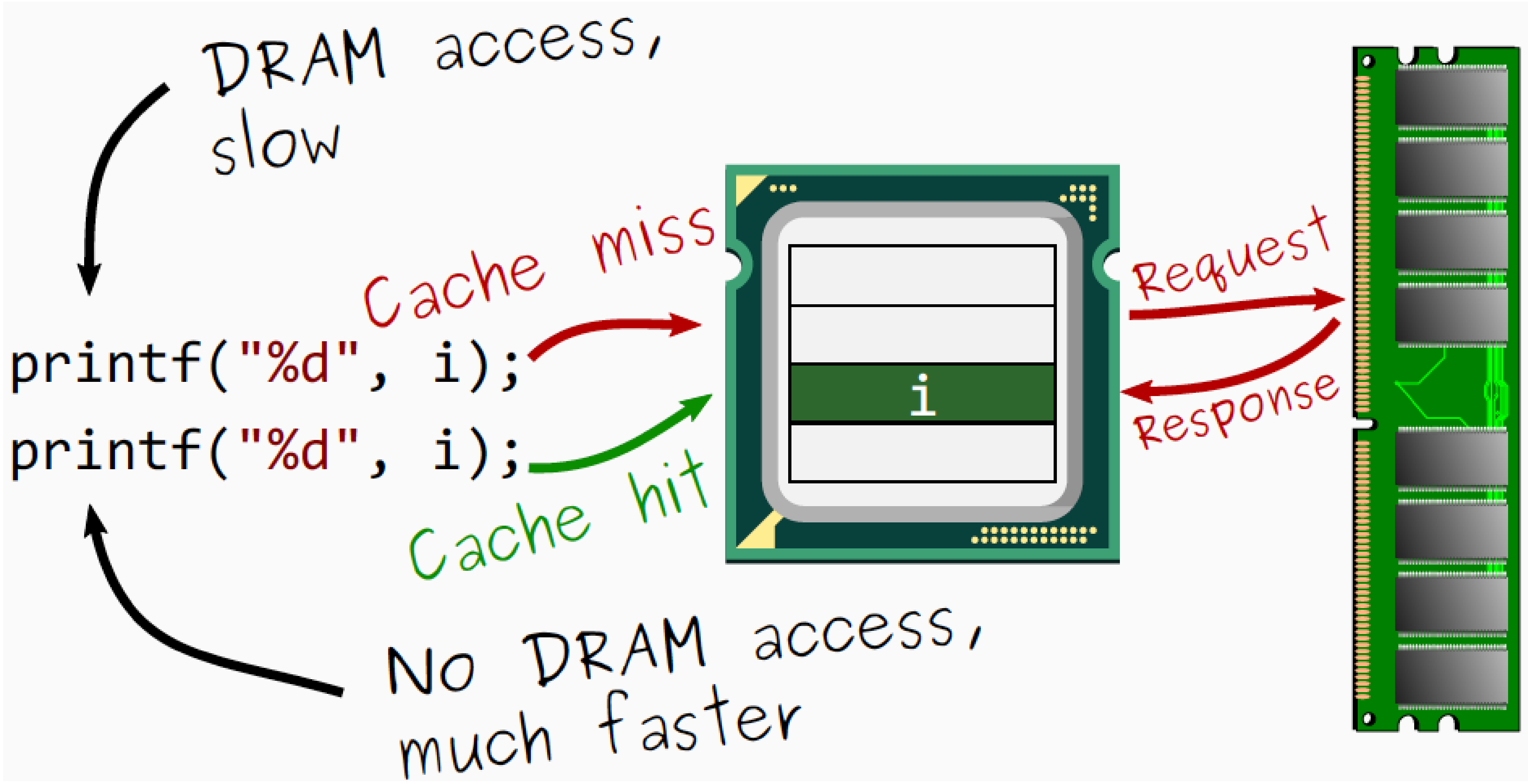
Caches Leak Information

- Memory Access Time & Access Pattern


```
printf("%d", i);
printf("%d", i);
```

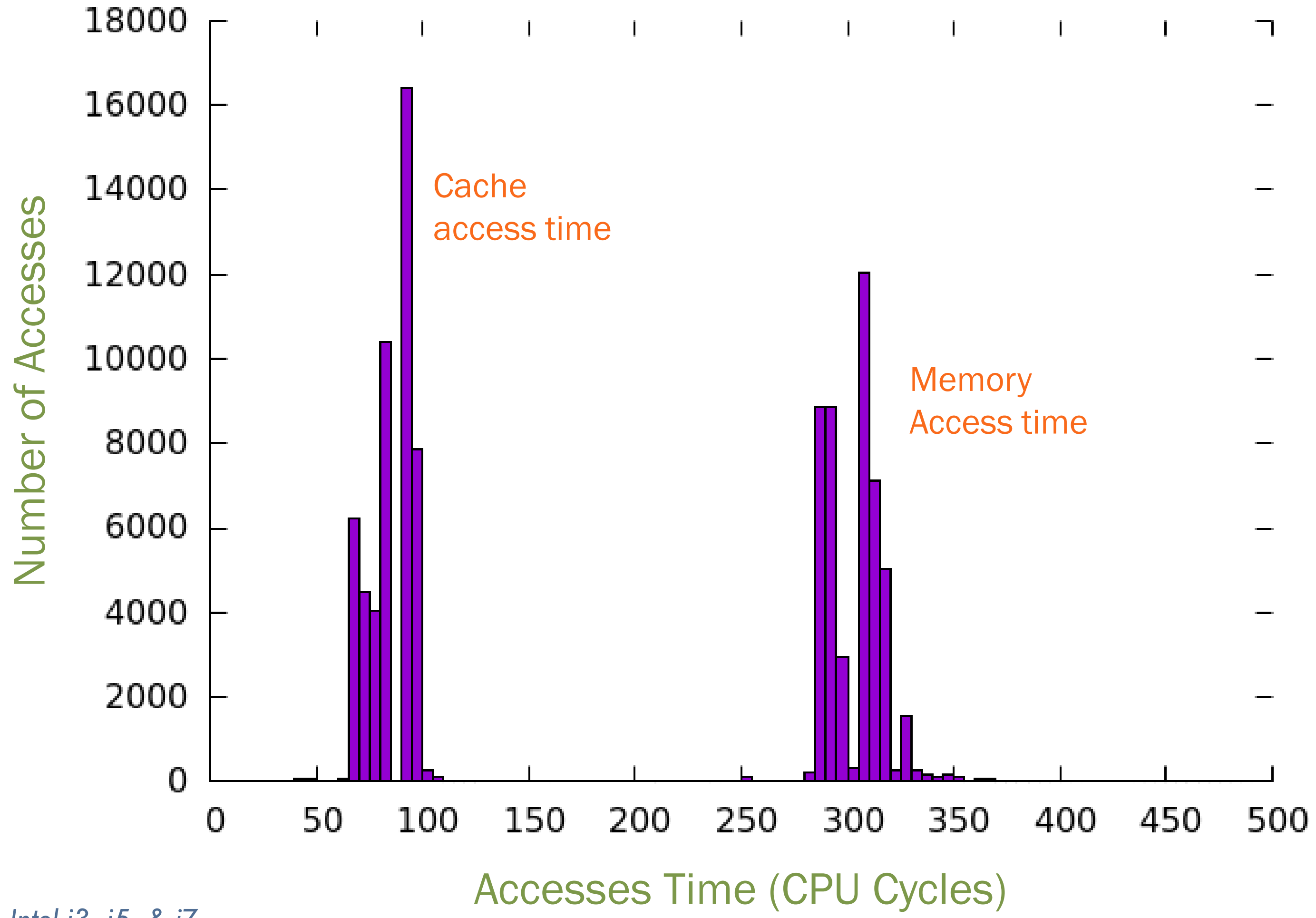
Caches Leak Information

- Memory Access Time & Access Pattern



Caches Leak Information

- Memory Access Time & Access Pattern



Results measured on Intel i3, i5, & i7

Outline

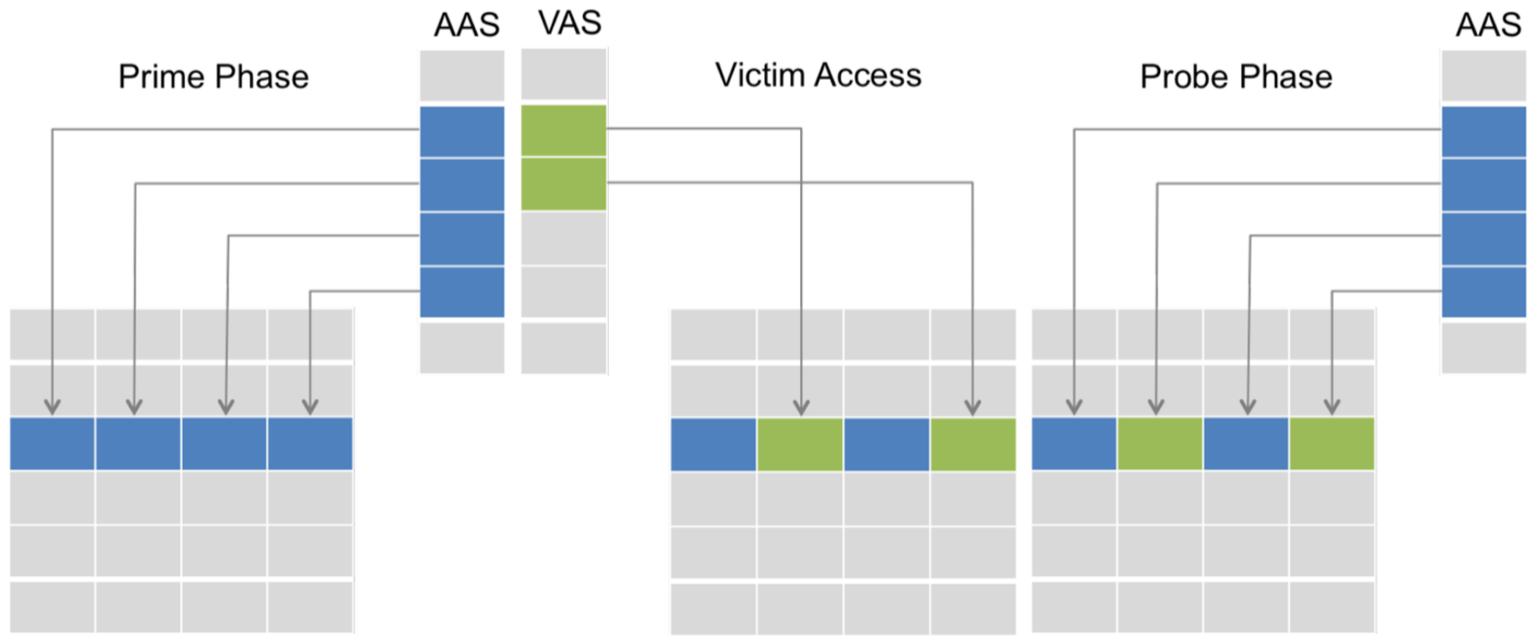
Cache Side Channel Attacks

- Recent Cache-based Side-Channel Attacks on Intel's x86 architecture

No.	Cache SCAs	Attacks Target
1.	Flush+Reload	AES & RSA Cryptosystem
2.	Flush+Flush	AES & RSA Cryptosystem
3.	Prime+Probe	AES & RSA Cryptosystem
4.	Spectre	Speculative Execution
5.	Meltdown	Out-of-Order Execution

Cache Side Channel Attacks

- Side-Channel Attacks on Intel's x86 architecture
  - Prime+Probe Attack



(a) Attacker's prime phase (b) Victim's memory accesses (c) Attacker's probe phase

AAS: Attacker's Address Space
VAS: Victim's Address Space

Outline

- Background
  - Side Channel Information Leakage –The Problem Statement
  - Performance-led Vulnerabilities in Intel's x86
  - Caches Leak Information
- Cache Side-Channel Attacks
  - Prime+Probe Attack
- Challenges of Using PCs
  - PCs Relevant to Cache-SCAs
  - Experiments and Results

Challenges of Using PCs

- PCs Relevant for Cache SCAs

#	Scope	Hardware Events
1	Cache Level 1	Data Cache Misses (L1-DCM)
2		Instruction Cache misses (L1-ICM)
3		Total cache misses (L1-TCM)
4	Cache Level 2	Instruction cache accesses (L2-ICA)
5		Instruction Cache misses (L2-ICM)
6		Total Cache accesses (L2-TCA)
7		Total cache misses (L2-TCM)
8	Cache Level 3	Instruction cache accesses (L3-ICA)
9		Total Cache accesses (L3-TCA)
10		Total cache misses (L3-TCM)
11	System-wide	Branch Miss Prediction (BR_MSP)
12		Total CPU Cycles (TOT_CYC)

Challenges of Using HPCs

- Discernible Information

  -Similar PCs for different attacks do not provide distinguishable information

  -Standalone PCs under multiple attacks further escalate the problem

- Non-deterministic Behavior

  -Time and security-critical applications require determinism

  -Non-determinism appears due to context switch, hardware interrupts etc

- Multiplexing Issues

  -Allows Multiple Counters to be used simultaneously

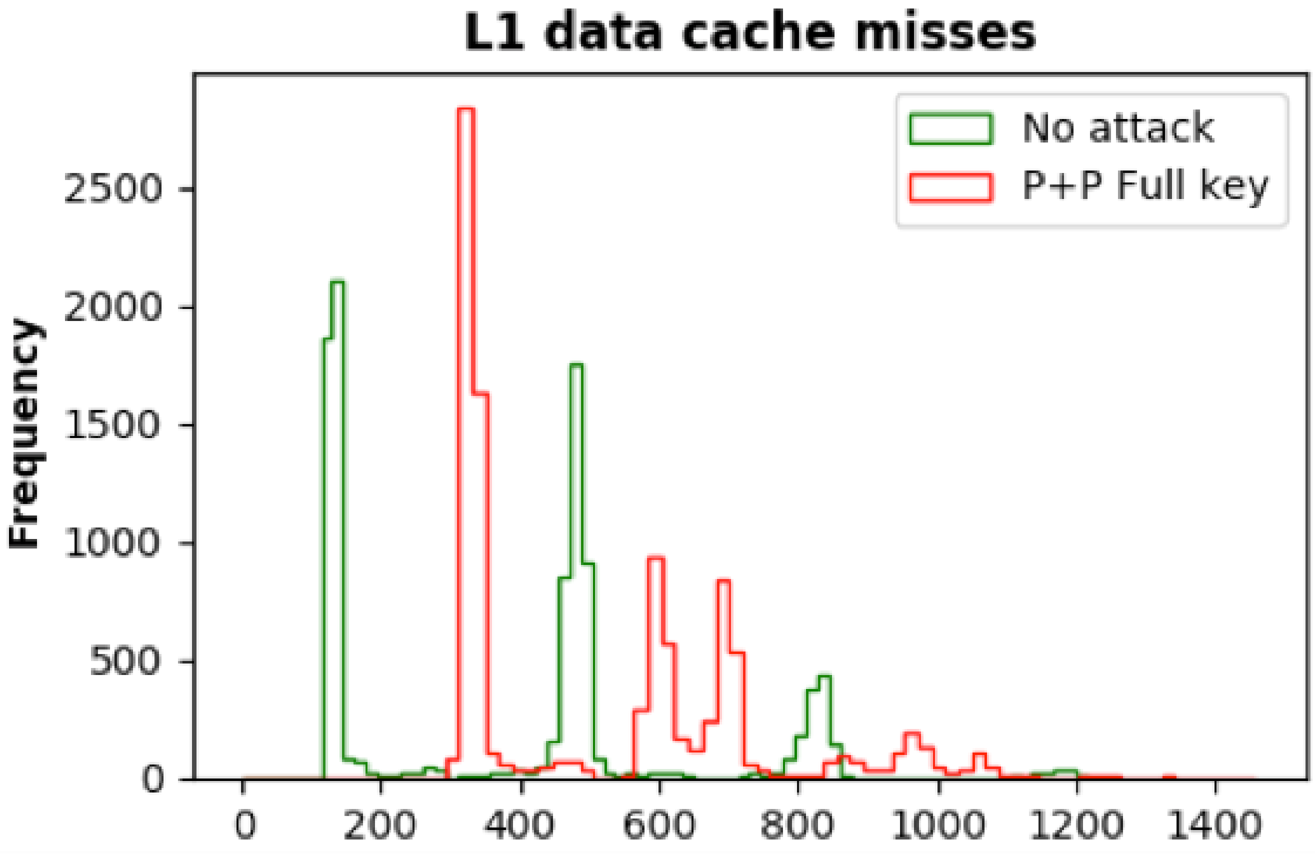
  -Time-sliced multiplexing leads to imprecise results

- Performance Overhead

  -Sampling frequency leads to increased overhead

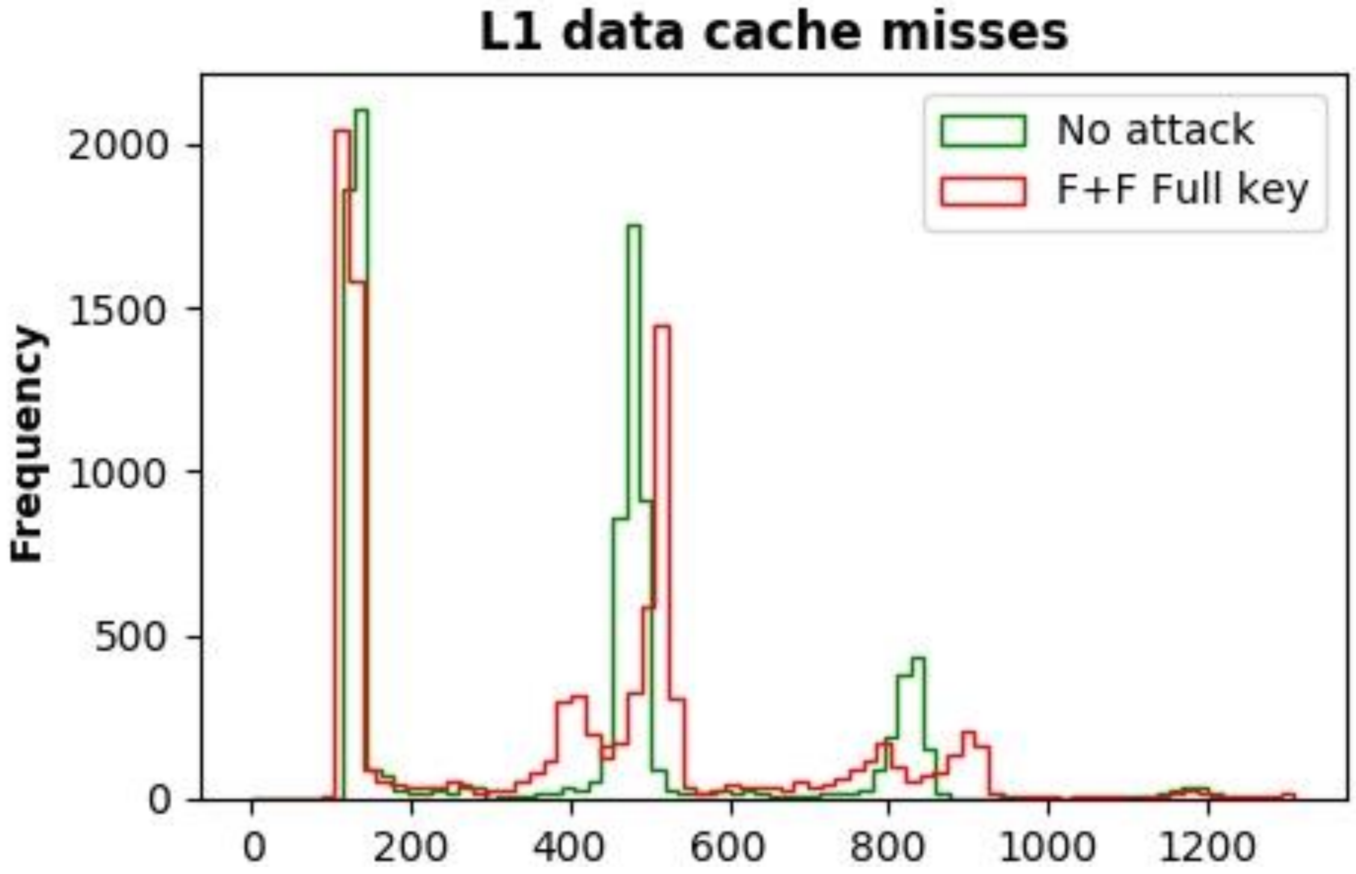
Challenges of Using HPCs

- Experiments and Results- Prime+Probe Attack



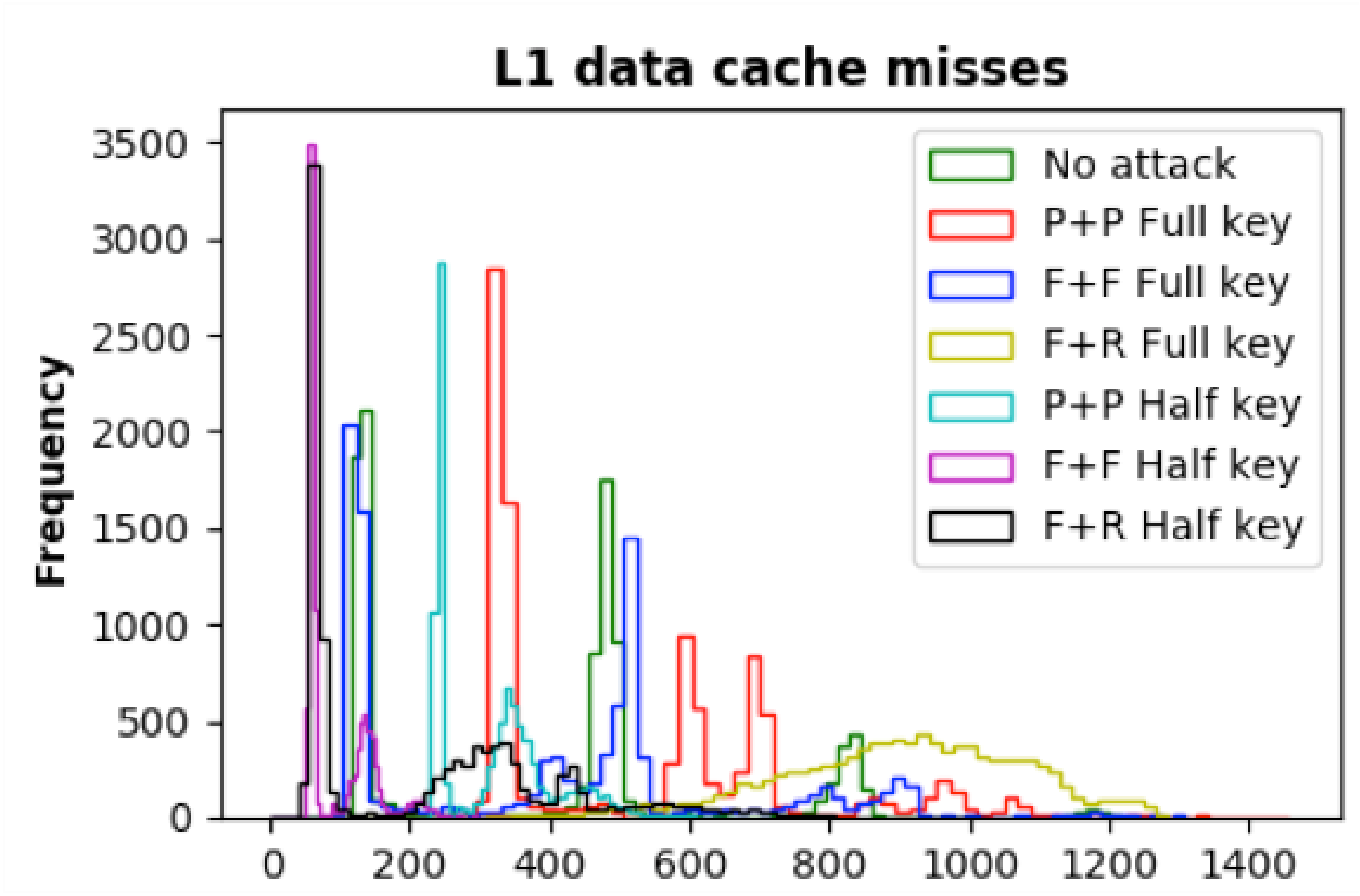
Challenges of Using HPCs

- Experiments and Results- Flush+Flush Attack



Challenges of Using HPCs

- Experiments and Results- Multiple Attacks



Conclusions

- SCAs are high resolution and stealthy attacks
- PCs can be effective in behavior analysis for security applications
- Standalone PCs, work good for similar attacks but do not discern between different behaviors
- An extensive analysis on different PCs is required to further detect and mitigate attack behaviors

Some Relevant Publications

1. M Mushtaq, et al., NIGHTs-WATCH: A Cache-Based Side-Channel Intrusion Detector using Hardware Performance Counters. ACM/IEEE International Symposium on Computer Architecture (ISCA) 2018, Hardware and Architectural Support for Security and Privacy (HASP), California, USA, 2018.

2. M Mushtaq, et al., Run-time Detection of Prime+Probe Side-Channel Attack on AES Encryption Algorithm. In the Proceedings of IEEE Global Information Infrastructure Symposium (GIIS), 2018, Thessaloniki, Greece.

3. M Mushtaq, et al., Machine Learning For Security: The Case of Side-Channel Attack Detection at Run-time. In the proceedings of 25th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2018, Bordeaux, France.

4. M Mushtaq, et al., Sherlock Holmes of Cache Side-Channel Attacks in Intel's x86 Architecture. 2019 IEEE Conference on Communications and Network Security (CNS), 2018, Washington, USA.

5. M Mushtaq, et al., WHISPER: A Tool for Run-Time Detection of Side-Channel Attacks. IEEE Access, 2020.

6. M Mushtaq, et al., Winter is here! A decade of cache-based side-channel attacks, detection & mitigation for RSA. Information Systems, 2020.

7. A Akram, et al., Meet the Sherlock Holmes' of Side Channel Leakage: A Survey of Cache SCA Detection Techniques. IEEE Access, 2020.

The Co-Authors

Maria

Pascal

Umer

Thank you!