上海科技大学
ShanghaiTech University

Side-channel Information Leakage with CPU Frequency Scaling, but without CPU Frequency

Speaker: Yanpeng Hu
Li Zhu and Chundong Wang
ShanghaiTech University

立志成才报国裕民

Outline

- Introduction
- Motivation
- Experiments
  - Covert channel
  - Side-channel
- Mitigations
- Conclusion

Introduction

上海科技大学
ShanghaiTech University

- Modern operating systems dynamically adjust CPU frequency to balance performance and energy use.
- We observe that **I/O performance** (e.g., file access on fast storage) is influenced by runtime CPU frequency changes, reflecting the behavior of the running workload.
- This effect enables the creation of a **covert channel** across different physical cores.
- We present **IOLeak**, a novel side channel that uses **I/O latency** to infer workload activity. IOLeak enables stealthy attacks such as:
  - Cryptographic key extraction
  - Website fingerprinting

Outline

- Introduction
- Motivation
- Experiments
  - Covert channel
  - Side-channel
- Mitigations
- Conclusion

Dynamic voltage and frequency scaling

- Dynamic voltage and frequency scaling (DVFS) allows CPUs to adjust their **clock frequency** based on workload demands.
- The key idea is that during I/O operations, most of the time is spent waiting on device-level responses rather than utilizing the CPU.
- As a result, lowering the CPU frequency during these non-intensive tasks—such as storage I/O—can reduce energy usage without noticeably affecting performance.

However…

- As device I/Os continue to improve in speed, software overhead now constitutes a larger portion of total system latency.
- Consequently, I/O latency has become more sensitive to changes in CPU frequency.
- To validate this effect, we conducted an experiment.

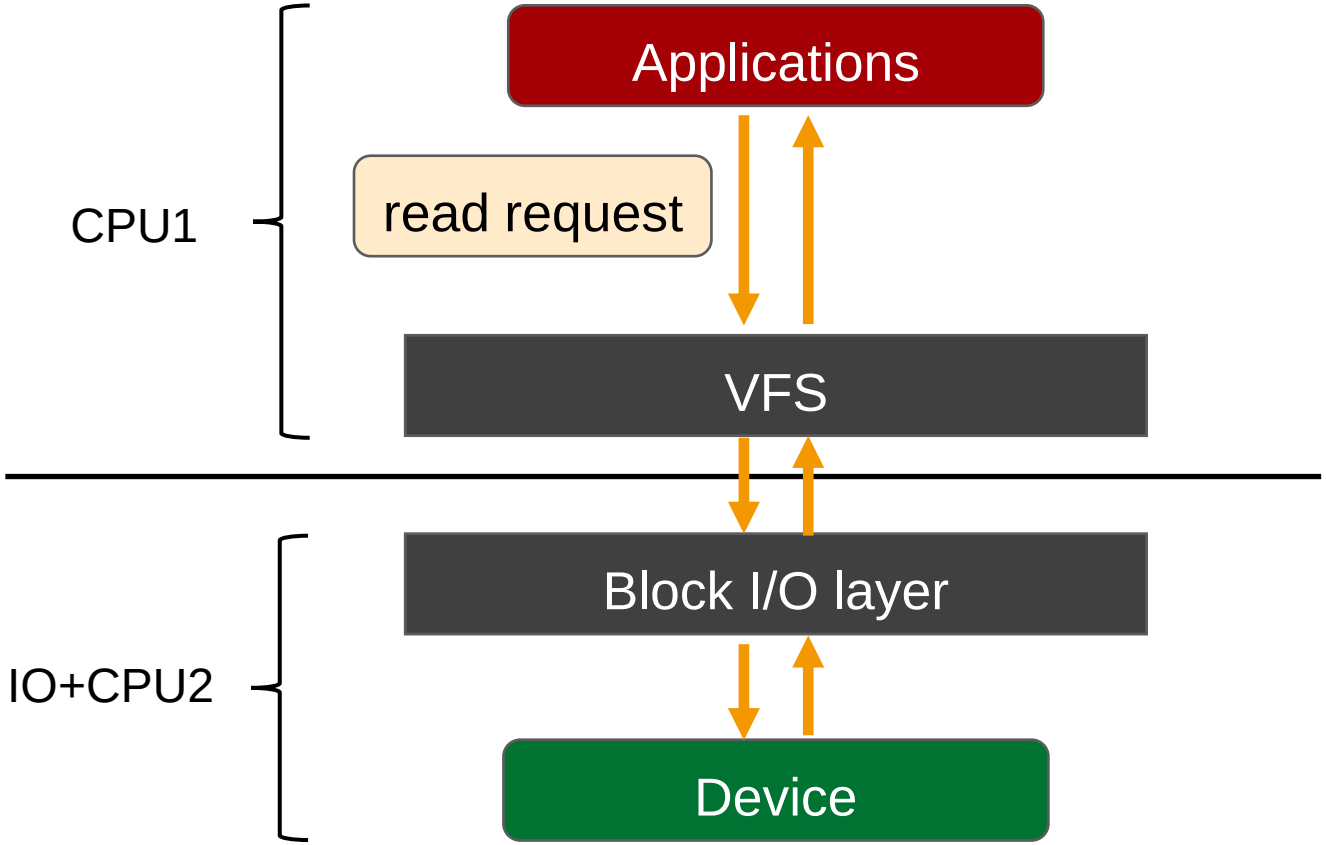
Testbed

上海科技大学
ShanghaiTech University

Machine	CPU	Storage Device
M1	12th Gen Intel® CoreTM i5-12500 (6 physical cores)	Intel 660p in NVMe SSD (512GB)
M2	Intel® Xeon® Gold 6348 (28 physical cores)	Samsung PM863a SATA SSD (960GB)

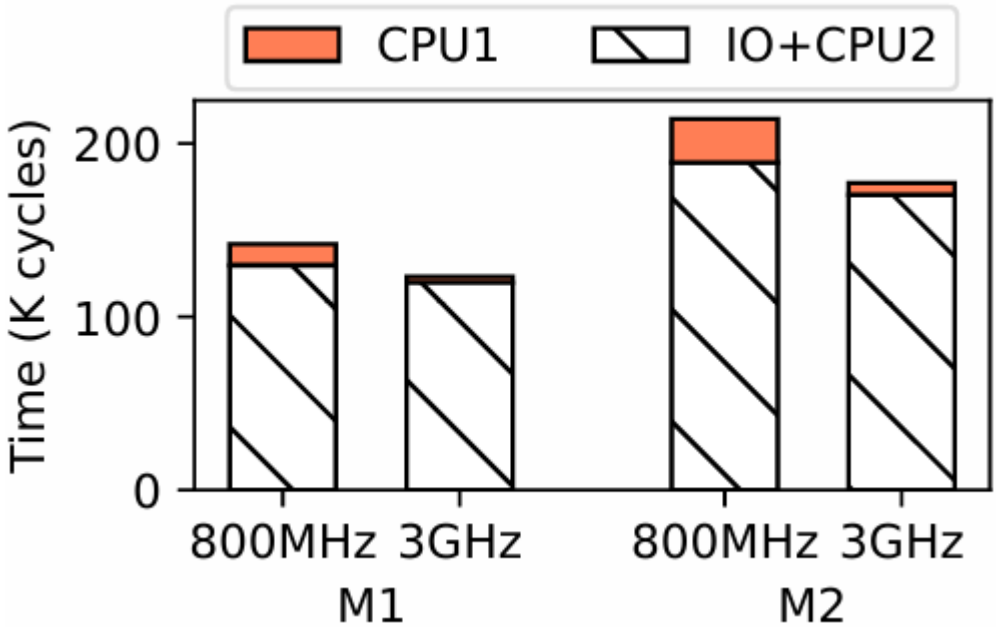
- Kernel: Linux kernel 6.8.0-51
- File system: ext4, default data=ordered mode

An abstraction of I/O stack



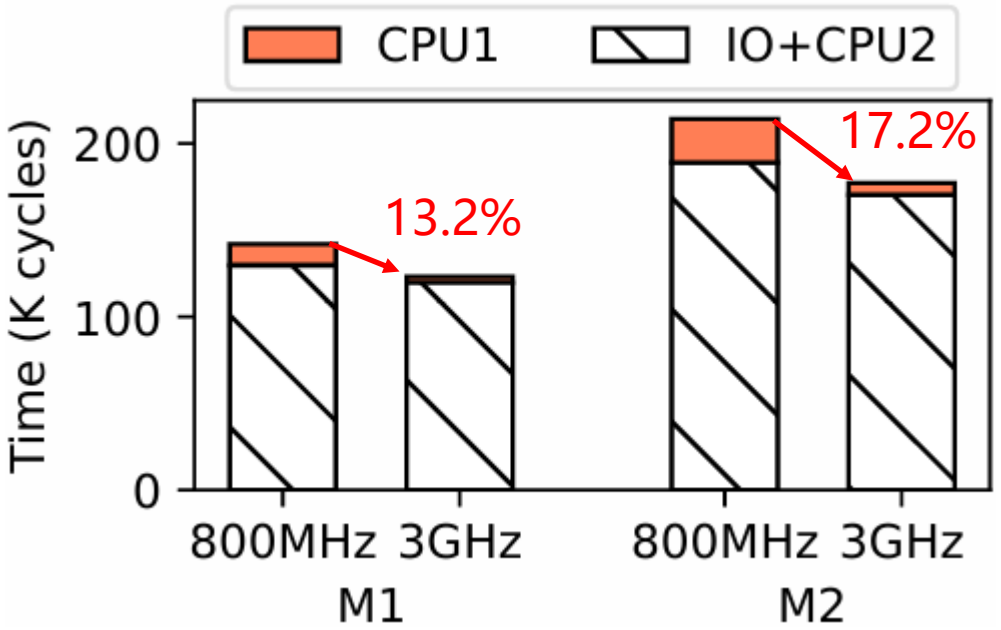
- We divide each read request into two stages: **CPU1** and **IO + CPU2**.
- We then perform a breakdown analysis to examine how CPU frequency affects I/O latency across these two stages.

Motivation



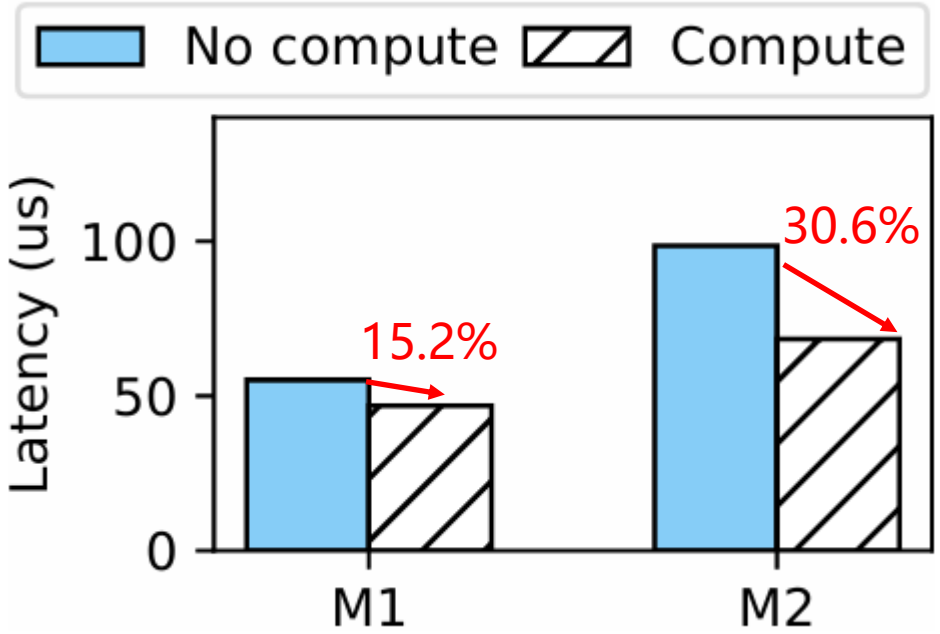
- We make a breakdown analysis on two machines.
- We fix CPU frequency on 800MHz and 3GHz to compare a file I/O latency.

Motivation



- On M1, The execution time for the CPU1 stage decreases by 73.3%, while the IO+CPU2 stage time reduces by 7.6%.
- The acceleration of computations helps explain why changes in CPU frequency can have such a noticeable impact on I/O latencies, particularly when using fast SSDs.

Motivation



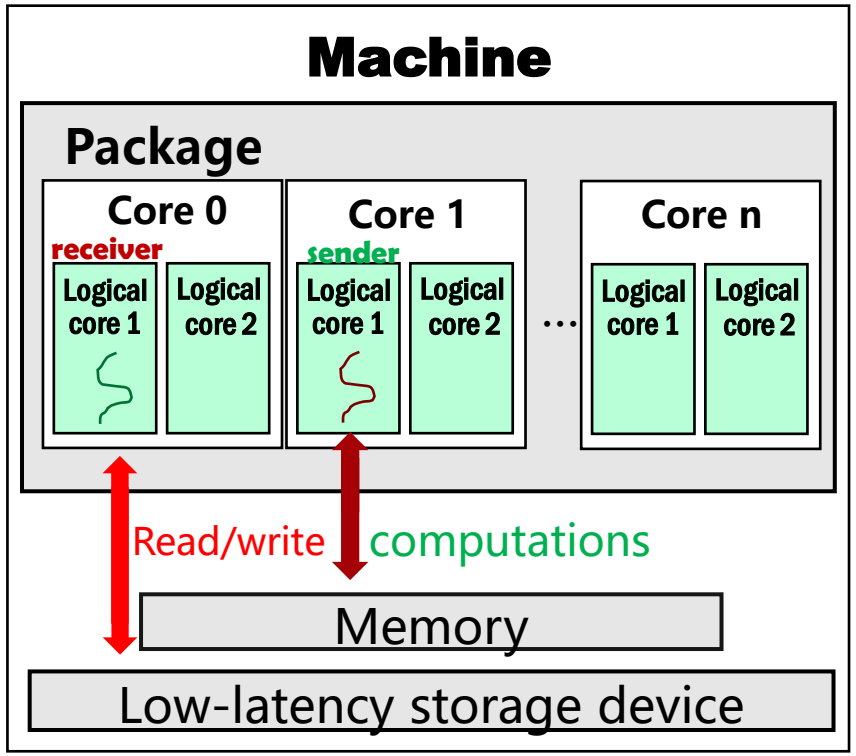
- We then test I/O latency under default powersave DVFS policy.
- When the CPU frequency scales up due to the computing task, the average I/O latency decreases by 15.2% on M1 and 30.6% on M2.

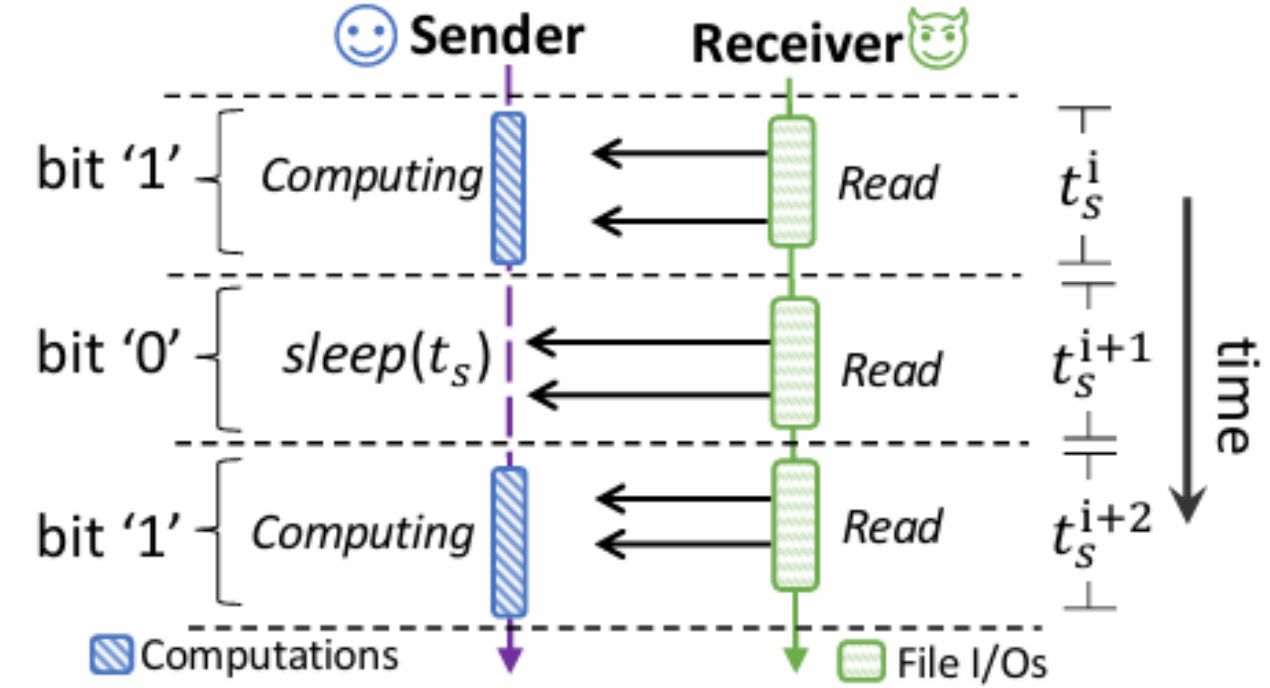
Outline

上海科技大学
ShanghaiTech University

- Introduction
- Motivation
- Experiments
  - Covert channel
  - Side-channel
- Mitigations
- Conclusion

Covert channel





Covert channel results

上海科技大学
ShanghaiTech University

BER: bit error rate

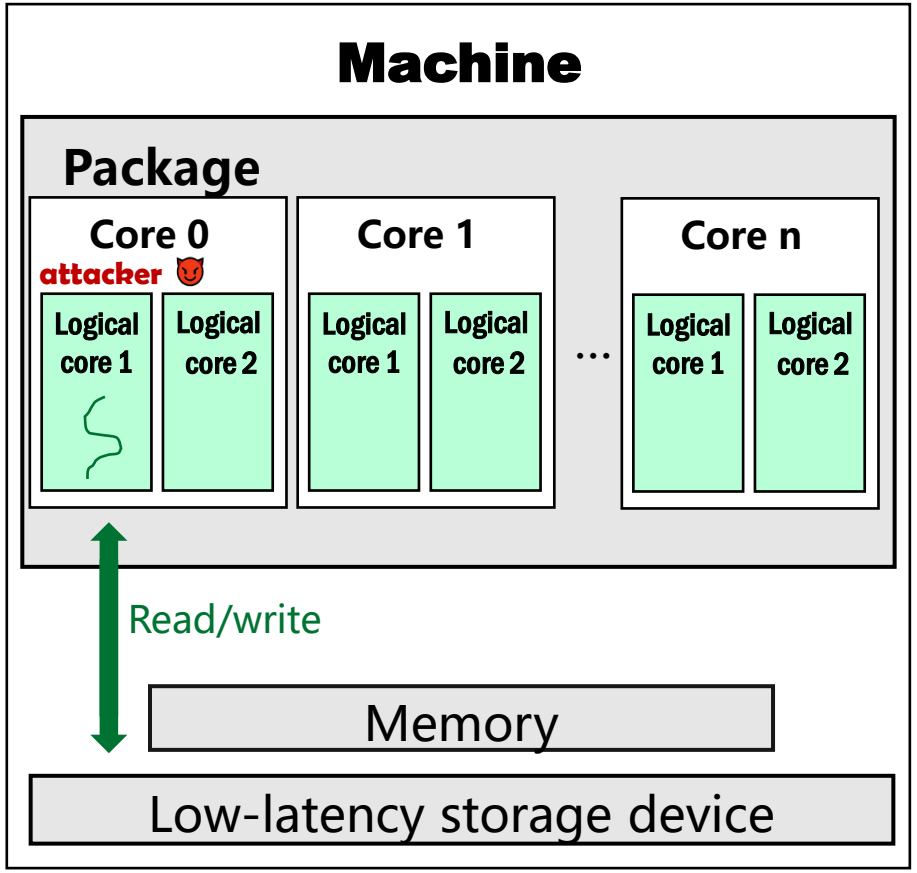
Time window size (ms)		30	60	90
Noiseless	Capacity (bps)	15.47	15.83	11.11
	BER	12.23%	0.57%	0%
Compute-intensive noise	Capacity (bps)	0.004	2.37	5.19
	BER	49.28 %	28.17%	12.12%
I/O-intensive noise	Capacity (bps)	0.03	14.70	7.55
	BER	48.13%	1.60%	5.83%

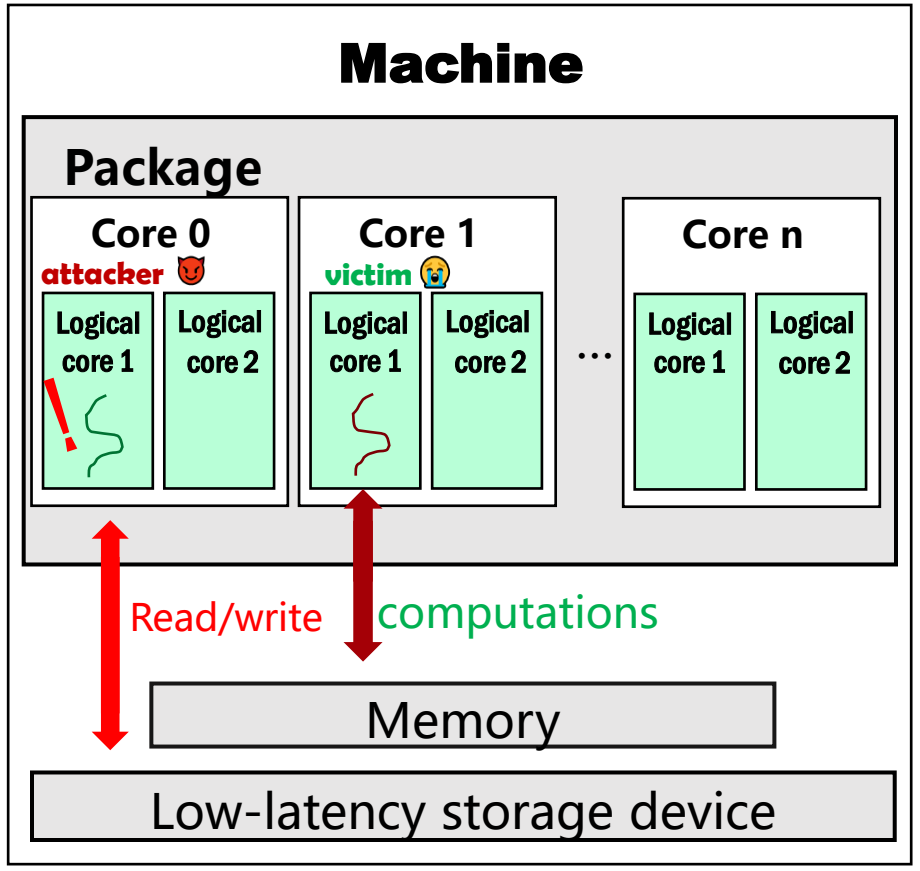
- Compute-intensive noise: stress-ng
- I/O-intensive noise: fio
- IOLeak achieves a capacity comparable to covert channels built by other researchers directly using power management (18.7 bps) and CPU frequency (46 bps).

Outline

- Introduction
- Motivation
- Experiments
  - Covert channel
  - Side-channel
- Mitigations
- Conclusion

Side-channel





Extracting Cryptographic Keys

上海科技大学
ShanghaiTech University

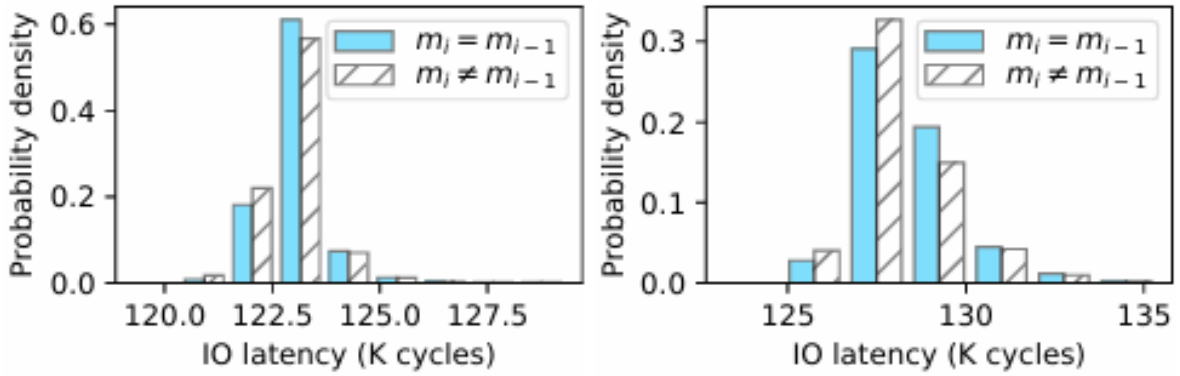
- Wang, et al.[1] demonstrated that SIKE is vulnerable under the chosen-ciphertext attack (CCA) model.
- The server's static secret key is an integer m with bit expansion $m = (m_{l-1}, \ldots, m_0)_2$, where l =378. An attacker who knows the i least significant bits of m can infer $i+1$ th bit. They managed to extract secret keys by monitoring variations in CPU frequency.
- We demonstrate IOLeak can also do secret key extractions.

[1] Wang, Yingchen, et al. "Hertzbleed: Turning power Side-Channel attacks into remote timing attacks on x86." *31st USENIX Security Symposium (USENIX Security 22)*. 2022.

Extracting Cryptographic Keys

上海科技大学
ShanghaiTech University

We choose two SIKE implementations as our victims: PQCrypto-SIDH and CIRCL.



(a) Attack to PQCrypto-SIDH (b) Attack to CIRCL

Figure 4: The distribution of I/O latencies when a challenge ciphertext introduces an anomalous zero value ($m_i \neq m_{i-1}$) or not ($m_i = m_{i-1}$)

- IOLeak can check whether $m_i = m_{i-1}$ by examine the IO latency distribution.
- The time complexity of recovering the secret key is reduced from $O(2^n)$ to $O(n)$.

TOAST LAB 吐 司 实 验 室

立志成才报国裕民

Fingerprinting Websites

上海科技大学
ShanghaiTech University

Browser	Google Chrome		Mozilla Firefox	
Metric	Top-1 Accuracy	Top-5 Accuracy	Top-1 Accuracy	Top-5 Accuracy
Machine 1	79.4%	97.5%	65.5%	83.0%
Machine 2	48.2%	76.7%	46.4%	77.0%

- Top-1 accuracy: The model answer (the one with highest probability) must be exactly the expected answer.
- Top-5 accuracy: Any of the model 5 highest probability answers must match the expected answer.
- Dataset: Alexa Top 100.

- IOLeak successfully **distinguishes between visits to different websites**

Outline

- Introduction
- Motivation
- Experiments
  - Covert channel
  - Side-channel
- Mitigations
- Conclusion

Mitigation against IOLeak

- Involve I/O-intensive tasks to defend against IOLeak.
- Fix the CPU frequency to prevent information from being leaked through CPU frequency scaling.
  - This, however, violates the intention of changing CPU frequency to achieve both power saving and high performance.
- Application-specific mitigation.

Outline

- Introduction
- Motivation
- Experiments
  - Covert channel
  - Side-channel
- Mitigations
- Conclusion

Conclusion

上海科技大学
ShanghaiTech University

- IOLeak is the first timing-based side channel that uses **storage I/O response latency** to reflect real-time CPU frequency of a victim workload.
- The IOLeak covert channel remains effective in various environments
  - Idle or low-noise systems.
  - Systems under some CPU-intensive workloads.
- IOLeak impacts real-world applications where CPU frequency varies at runtime.
  - Demonstrated capabilities:
    - Cryptographic key extraction from SIKE (Supersingular Isogeny Key Encapsulation)
    - Website finger-printing.

Q&A

上海科技大学
ShanghaiTech University

- Thank you for your time.
- If you have any questions, please contact:
  - Li Zhu: zhuli2023@shanghaitech.edu.cn
  - Chundong Wang: cd_wang@outlook.com