# Architecture Security Advisory

Version: 1.0

# Collide+Power: A new software-based power side-channel

Non-Confidential

# Architecture Security Advisory

## Collide+Power: A new software-based power side-channel



## Arm Non-Confidential Document Licence ("Licence")

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-21585 version 4.0

## Web Address

http://www.arm.com

## Contact

arm-security@arm.com

# Contents

# 1 Introduction

Collide+Power is a new software-based power side-channel attack [2] capable of leaking data from the CPU memory hierarchy. The leakage is based on power consumption models of the adversary's own data loads/stores when co-located in the memory hierarchy with victim data.

The fundamental observation is that the power consumption of cache-related transactions can be correlated to the Hamming difference between their values. More specifically, the CPU power consumption varies in relation to the Hamming distance between previous and new values in the cache. An adversary can abuse this relationship by controlling the data being written to the cache while measuring CPU power consumption, this could allow the adversary to infer victim’s data without having access to it.

The sampling of power measurements can be done via available interfaces or, as in Hertzbleed [1], by translating the power consumption into the time domain. In their end-to-end exploit, the authors rely on timing measurements.

In contrast to traditional cache side-channels, Collide+Power doesn't leak metadata, e.g., memory accesses or control flow, but the secret data itself; similarly, in contrast to transient execution issues, it doesn't rely on speculative or out-of-order execution and it can target data at rest across security domains.

The authors present two different scenarios:

- MDS-Power: requires both victim and adversary to run in parallel in the same physical core, i.e., Simultaneous Multi-Threading (SMT); the victim continuously loads the target data, and the adversary repeats accesses with controlled guessed values while sampling the power consumption.
- Meltdown-Power: targets data at rest and does not required SMT; the victim executes and leaves the target data in the cache, then the adversary performs the required accesses while sampling the power consumption.

# 2 Are Arm cores affected by Collide+Power?

To our knowledge, Collide+Power techniques have not yet been successfully used against Arm based systems. However, software-based power side-channels have been recently demonstrated in Arm SoCs [9], and since they exploit the same fundamental principle, which is architecture agnostic, Collide+Power could also affect Arm based systems.

That said, software power measurement interfaces or power capping policies are implementation defined additions and not part of the Arm architecture, therefore it is difficult to make any general statements on which specific systems could be affected.

## 2.1 Variants

The MDS-Power variant requires SMT, thus Arm implementations without SMT, which are the vast majority, are not affected. On systems with SMT, core co-location is still needed between victim and adversary.

The Meltdown-Power variant requires loading the victim data into the cache in a controlled manner on every measurement. For that, the adversary schedules the victim and waits for (or forces[1]) it to do the desired load. This is an important limitation as it involves a large overhead due to context switch and victim execution that adds noise and hinders the practicality of the technique. Specifically, the authors are only able to leak values with repeated nibbles, i.e., with “amplification”, at a rate of 0.136 bit/h.

## 2.2 Cores not affected by Meltdown

Cores not affected by Meltdown, e.g., implementing FEAT_CSV3, and thus not enabling Meltdown mitigations, such as KPTI or FEAT_E0PDx, could be at higher risk against Collide+Power. An adversary could fetch any arbitrary privileged address, have a successful translation, and install the victim data into the cache without the overhead of invoking the victim. Note that the access will fault, but this can be suppressed by performing the access under speculation. All this without violating CSV3, since the data is only fetched but not forwarded and exfiltrated by younger instructions.

Under this hypothetical scenario, the adversary could use a power leakage model without evictions or multiple cache levels, which according to the paper provide the best noise-to-signal ratio, while also avoiding the SMT requirement or the expensive context switches. However, this resembles the prefetch with no-eviction scenario described in the paper and it could not create a strong enough signal for a Collide+Power analysis.

## 2.3 Other scenarios

Collide+Power targets the kernel (EL1) from an unprivileged process (EL0), but there are other scenarios that require consideration.

The risk of Non-Secure world targeting Secure or Realm world is similar. In practice, a privileged adversary has more control of the system’s resources and the victim, facilitating the analysis. It is unclear how many speculative prefetch gadgets would be available in Secure or Realm applications compared to a kernel.

[1] The exploit against a Linux kernel abuses speculative prefetch gadgets [3], which are more common than Spectre gadgets.

The same holds for a Virtual Machine (VM) running at EL1 targeting a hypervisor at EL2.

The risk of a VM to VM attack is very low, since normally the adversary VM would have no control over the victim VM execution, and controlling the co-location of target victim data would be difficult. This applies to both Non-Secure VMs and Realms.

If we consider the above scenarios in the context of the potential attack where no victim execution is required, due to the adversary being able to directly fetch the victim's data, the risk is almost negligible. In the EL1->EL1 and EL1->EL2 scenarios, Stage 2 translation tables ensure that the adversary will never be able to fetch or hit victim data. In the Non-Secure EL2 to Realm or Secure scenario, even if Non-Secure world could fetch an arbitrary PA under speculation, e.g., bypassing the Granule Protection Check somehow, the PAS would still be wrong and 1) if the victim's data was in the cache, the tag wouldn't match; 2) if the access reached main memory, on a system with Memory Protection Engine (MPE) an incorrect key would be used and the data in the installed cache line would contain garbage or ciphertext.

# 3 Recommendations

Since the problem is rooted in the way semiconductor chips are designed and physically built, it is very hard to entirely mitigate this type of threat.

However, despite the impact of a successful Collide+Power attacks being used is high, the level of complexity required, low scalability across devices, and low exfiltration rate, make the overall risk manageable.

The two main limitation factors are:

- The availability and precision of power consumption metrics.
- The overhead to achieve controlled co-location of victim's and adversary's data.

Regarding access to metrics: not exposing power consumption interfaces to untrusted software is a general design principle that noticeably hinders these threats. The residual risk is the indirect time-variations due to frequency scaling, which performs frequency adjustments based on the computing workloads, temperature, and/or remaining power buckets [8].

The second limiting factor implies that the risk would be increased if the adversaries were able to speculatively prefetch victim data in the target structure in a controlled manner without involving victim execution. Hence, making sure that such primitives do not exist is critical. The main recommendation is to enable E0PDx by default even on cores not affected by Meltdown.

E0PDx has the additional benefit of preventing KALSR side-channels.

In summary, on cores without SMT, the residual risk would be that of a Meltdown-Power scenario with an adversary inferring power consumption via time measurements. This threat is deemed unpractical due to the expected overhead, and thus can be accepted. On cores with SMT, the risk remains higher, although core co-location between untrusted parties can be restricted.

As discussed in the next section, this area of research is under active development and therefore future improvements might challenge some of the current assumptions and require a risk re-evaluation.

# 4 Conclusions

Power side-channel analysis techniques have been largely considered a physical type of threat (i.e., a threat involving an adversary with physical access to the device and measuring equipment, such as an oscilloscope), and as such they have been out-of-scope in most threat models. The main exception being cryptographic implementations (both software and hardware) on some critical systems, like smartcards.

Recent research shows a trend where the assumption of physical access to the target device to perform this class of attacks, e.g., power side-channels, is under dispute.

So far, the general recommendation against physical side-channels has been the implementation of software mitigations (like masking or hiding schemes) on cryptographic libraries, which were the main target. However, new techniques are targeting architectural and hardware primitives as well as components that are beyond the influence of software. When data at-rest from another security domain can be leaked, there is hardly anything that software can do to maintain confidentiality.

While the practicality and real impact of software-based power side-channel attacks is still unclear, there is a concern that the analysis methods and techniques will only keep improving. For example, recent works [9,10] have shown other software-based power side-channels going beyond cryptography; specifically, an adversary can use JavaScript in the web browser to perform timing measurements, which show differences due to GPU power-induced throttling, and infer the values of cross-origin pixels.

Given the mounting evidence, it is time to re-evaluate this threat model and start planning better ways to hinder these threats.

This situation also extends to fault injection attacks, a threat which (except for a few industries like credit cards or videogames, where theft and physical access are included in the threat model of most manufacturers) has been largely ignored. In this context, the physical access assumption has also been disputed several times with Rowhammer [4], CLKScrew [5], and other [6,7] research.

# 5 References

1. “Hertzbleed: Turning Power Side-Channel Attacks Into Remote Timing Attacks on x86” Wang et al. (2022)
2. “PLATYPUS: Software-based Power Side-Channel Attacks on x86” Lipp et al. (2021)
3. “Speculative Dereferencing: Reviving Foreshadow (Extended Version)” Schwarzl et al. (2021
4. “Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors” Kim et al. (2014)
5. “CLKSCREW: Exposing the Perils of Security Oblivious Energy Management” Tang et al. (2017)
6. “Plundervolt: Software-based Fault Injection Attacks against Intel SGX” Murdock et al. (2020)
7. “VoltJockey: Breaching TrustZone by Software-Controlled Voltage Manipulation over Multi-core Frequencies” Qiu et al. (2019)
8. “Frequency Throttling Side-Channel Attack” Liu et al. (2022)
9. “Hot Pixels: Frequency, Power, and Temperature Attacks on GPUs and Arm SoCs” Taneja et al. (2023)
10. “DVFS Frequently Leaks Secrets: Hertzbleed Attacks Beyond SIKE, Cryptography, and CPU-Only Data” Wang et al. (2023)
11. “System Control and Management Interface (SCMI) specification” https://developer.arm.com/Architectures/System%20Control%20and%20Management%20Interface