LETTER *Special Section on Next-generation Security Applications and Practice*

Data Covert Channels between the Secure World and the Normal World in the ARM TrustZone Architecture

Haehyun CHO[†a)], ***Nonmember***

SUMMARY The ARM TrustZone architecture, which provides hardware-assisted isolation, is widely adopted in mobile and IoT devices. The security of ARM TrustZone relies on the idea of splitting system-on-chip hardware and software into two worlds, namely normal world and secure world. There are legitimate channels at the hardware level that the normal world and the secure world can use to communicate with each other. To protect these channels from being abused, research efforts were invested on restricting the access to these channels from normal world components. Therefore, only predefined and legitimate normal world components can use cross-world communication channels. In this work, we present a study on data covert channels that can bypass such protection mechanisms and smuggle sensitive information. We first analyze causes of the noise in the covert channel between two worlds. Then, we evaluate the accuracy and bandwidth of covert channels built by our Prime+Count method with one built by Prime+Probe method. Our results demonstrate that Prime+Count is an effective technique for enabling cross-world covert channels in the ARM TrustZone.

key words: *cache side-channel, covert channels, ARM TrustZone*

1. Introduction

Attackers and defenders have been fighting the "Eternal War in Memory" for a long time. While this cat-and-mouse game may seem discouraging to researchers, it *has* resulted in a gradual improvement of the security posture of modern applications. Lately, a new attack surface, the CPU caches, was highlighted by the research community. Especially, as mobile and "Internet of Things" devices experience unprecedented growth, cache attacks on the ARM architecture, the dominating computing platforms in both mobile and IoT markets [1], has drastically risen in importance.

Cache, a high-speed and small internal memory to buffer the data that is frequently used, is built for CPU to overcome the latency of system memory access. Therefore, there are noticeable time differences between an access to cache (cache hit) and an access to the main memory (cache miss), by which attackers infer sensitive data. Cache attacks are known to be effective in inferring sensitive information from cryptographic algorithms, in tracing execution flows of target programs, and so forth [2].

Unlike the most of previous research efforts on cache side-channel attacks, in our prior work [3], we proposed a new cache attack method to build a practical data covert channel in which two entities can communicate by using the cache side-channel. Especially, our work focused on cross-world covert channels in the TrustZone architecture. In the work, we demonstrated that unauthenticated normal world and secure world components can always smuggle sensitive information that is not supposed to leave the secure world to the normal world, such as private keys, user passwords, etc.

To this end, we designed the Prime+Count attack method that inspects how many cache sets or lines were used. For more background knowledge in particular, please refer to our previous work [3]. In case of the Prime+Probe method, attackers need to find which cache sets were occupied by a victim process (or a sender in the scenario to build a covert channel) [4]. In theory, it is possible to build a data covert channel between the normal world and the secure world in the TrustZone architecture by using Prime+Probe. However, because of ARM's pseudo-random replacement policy and the world switching introduced by TrustZone, we deemed Prime+Probe unreliable.

In this work, we first analyze and discuss the noise of the covert channel in the TrustZone architecture. We, then, evaluate the accuracy and bandwidth of covert channels built by Prime+Count with one built by Prime+Probe. Because the other cache attack methods such as Evict+Reload and Flush+Flush require the shared memory, we did not consider them. It is essential to use cache attack methods that do not required shared memory for building a data covert channel between the normal world and the secure world.

2. Background

Legitimate Communication Channels. There is a couple of ways to communicate between components in the normal world and the secure world: (1) When the world switching occurs through the SMC instruction, data stored in general registers of CPUs can be passed into the other world based on Secure Monitor Call Calling Convention (SMCCC) and (2) Share memory regions, mapped by components in the secure world, can be used to communicate between the two worlds. Also, these two channels can be used together for faster communication. Because adversaries can perform malicious actions by using those channels (e.g., sending crafted messages to trigger vulnerabilities in the secure world, or smuggling sensitive information from the secure world) researchers have proposed an extra monitoring layer where only verified components can used the channels to prevent sensitive data leakages through the communication channels and misuses of them [5].

Manuscript received February 6, 2022.
Manuscript revised May 31, 2022.
Manuscript publicized July 28, 2022.
[†]The author is with Soongsil University, Seoul, Korea.
a) E-mail: haehyun@ssu.ac.kr
DOI: 10.1587/transinf.2022NGL0002

Table 1 Experimental environment.

Device	SoC	CPU (cores)	L1 Cache	L2 Cache	Secure World OS	Normal World OS
Hikey board	HiSilicon Kirin 620	64-bit Cortex-A53 (8)	32KB, 4-way, 128 sets	512KB, 16-way, 512 sets	ARM Trusted Firmware, OP-TEE	Linux 4.1.0

Building Data Covert Channels. The legitimate communication channels, including parameter passed by registers and shared memories, between the normal world and the secure world can be monitored by a security solution such as SecRet [5]. SeCReT have not only tried to monitor such legitimate channels but also added extra layers of authentication and verification in these channels. Therefore, it is impossible for the attackers to transfer sensitive data and commands between the normal and secure world without being detected by the monitor mode code. However, unfortunately, data covert channels can be built by exploiting cache side-channels because there is a CPU cache shared between the two worlds to smuggle sensitive data such as private keys, user passwords, etc., out from the secure world to the normal world [3].

3. Noise of the Covert Channel

Unlike the other previous covert channels, we aimed at the cross-world covert channel which is along with the path of the world switching. Therefore, transferred data using the covert channel has noise as it moves through the path. And, to build a practical data covert channel in the cross-world scenario, it is critical to manage such noise.

We empirically analyzed the noise and found that there are two types of the noise in the channel, which can be classified according to the effects: the first type results in additional cache misses and the second type issues the cache hit at the address where the cache miss should be happened. The reason of the first type noise is the data used by the world switching and by functions that the channel have to be gone through, and thus, this noise is inevitable. On the other hand, inaccurate prime method is another source of the first type noise. Also, the uncertainty of the prime method can cause the second type noise. The second type noise can be occurred by the automated data prefetcher of the ARM architecture. In the rest of this section, we show that we can manage the noise caused by the inaccurate prime method and the data prefetcher with Prime+Count.

4. Experiments

Accuracy and Capacity. For the experiments, we implemented covert channels on the device as shown in Table 1. To this end, we used the *set-counting* mode of the Prime+Count as in [3]. On the other hand, to implement the Prime+ Probe, we used the same method as in [2]. To simulate the scenario where *Sender* smuggles sensitive information from the secure world to the normal world, we implemented *Receiver* in the normal world and *Sender* in the secure world. Prime+Count is a variant of Prime+Probe that counts *how many* cache *sets* or *lines* have been occupied instead of determine *which* cache *sets* have been occupied. Prime+Count, as a coarser-grained approach than Prime+Probe, significantly reduces the noise introduced by the random replacement policy and world switching. In addition, Prime+Count does not require shared memory space/objects.

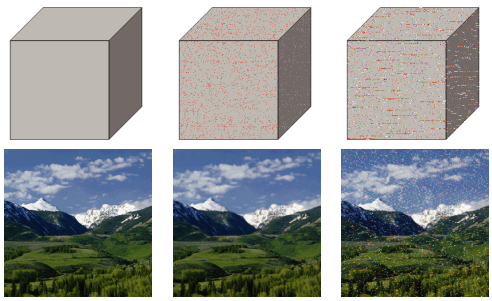

Fig. 1 Transferred images using covert channels to evaluate the accuracy. (a) Original images; (b) Prime +Count; (c) Prime+Probe.

Table 2 Capacities of two covert channels.

Attack Method	Bandwidth (Byte/Second)
Prime+Count	13,718.97
Prime+Probe	10,356.31

Firstly, we used the covert channels to transmit images from the secure world to the normal world by using the two methods. Figure 1 shows the results of experiments. The accuracy of the transferred images using Prime+Count is better (and thus is more effective to build a cross-world data covert channel) than the ones using Prime+Probe. This is because Prime+Count can handle noise in the covert channel by employing the instruction synchronization barrier (`ISB`) to deter the automatic data prefetcher and by using our Prime method with the `DC CISW` instruction by which we can clean and invalidate a specific cache line.

To evaluated the capacity of covert channels, we measured how many bytes can be transferred. Specifically, we measured the mean bandwidth between when the *Sender* occupies all cache sets and when the *Sender* does not load anything. Table 2 shows evaluation results: we can achieve higher capacity by using Prime+Count. This is mainly due to the fact that, albeit Prime+Probe can transfer more bits than Prime+Count at once, Prime+Probe costs thousands of CPU cycles even when it does *prime* the cache.

5. Related Work

Covert channel attacks for transferring data on the sly can

be implemented through various techniques using network packets [6], [7] and CPU caches [8]. Among them, recently, a lot of research effort has put onto building cache-based covert channels. Xu et al. proposed cache covert channels between virtual machine using L2 cache in cloud computing environment [9]. Also, Wu et al. designed a high-bandwidth and reliable data transmission cache-based covert channel in the cloud computing environment [10]. On the other hand, Lipp et al. demonstrated cache attacks are possible on the ARM architecture for the first time and showed that we can build data covert channels using the Prime+Probe method [2].

6. Conclusion

In this work, we analyzed the noise in the covert channel and found that, while there is noise that we have to embrace unavoidably, there is the noise caused by the inaccurate prime method and the data prefetcher, which is manageable. We, then, evaluated the performance of two cache attack methods in the cross-world covert channel scenario. As the evaluation results show, Prime+Count is effective to build data covert channels because it can manage the noise even though it is less fine-grained than any other cache attack methods. In our future work, we will focus on exploiting Prime+Count for other cases and extending this study to build covert channels in other systems.

Acknowledgments

This work was supported by the Soongsil University Research Fund (New Professor Support Research) of 2021.

References

[1] ARM, "ARM Q1 2017 Roadshow," http://www.arm.com/company/investors, 2017.

[2] M. Lipp, D. Gruss, R. Spreitzer, C. Maurice, and S. Mangard, "ARMageddon: Cache attacks on mobile devices," Proceedings of the 25th USENIX Security Symposium (Security), Austin, TX, pp.549–564, Aug. 2016.

[3] H. Cho, P. Zhang, D. Kim, J. Park, C.-H. Lee, Z. Zhao, A. Doupé, and G.-J. Ahn, "Prime+count: Novel cross-world covert channels on arm trustzone," Proceedings of the 34th Annual Computer Security Applications Conference, pp.441–452, 2018.

[4] D.A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: the case of AES," Proceedings of the Cryptographer's Track at the RSA Conference (CT-RSA), vol.3860, pp.1–20, 2006.

[5] J. Jang, S. Kong, M. Kim, D. Kim, and B.B. Kang, "SeCReT: Secure Channel between Rich Execution Environment and Trusted Execution Environment," Proceedings of the 2015 Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, Feb. 2015.

[6] T. Schmidbauer and S. Wendzel, "Sok: A survey of indirect network-level covert channels," Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security, ASIA CCS '22, New York, NY, USA, pp.546–560, Association for Computing Machinery, 2022.

[7] T. Schmidbauer and S. Wendzel, "Detection of computational intensive reversible covert channels based on packet runtime," Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol.13, no.1, pp.137–166, 2022.

[8] Z. Wang and R.B. Lee, "Covert and side channels due to processor architecture," Proceedings of the 22nd Computer Security Applications Conference (ACSAC), pp.473–482, 2006.

[9] Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting, "An exploration of L2 cache covert channels in virtualized environments," Proceedings of the 3rd ACM workshop on Cloud computing security workshop, pp.29–40, 2011.

[10] Z. Wu, Z. Xu, and H. Wang, "Whispers in the hyper-space: High-speed covert channel attacks in the cloud," Proceedings of the 21st USENIX Security Symposium (Security), Bellevue, WA, pp.159–173, Aug. 2012.