Always-trusted IoT — Making IoT Devices Trusted with Minimal Overhead

Zsolt István
zsolt.istvan@cs.tu-darmstadt.de
CS Department, TU Darmstadt, DE

Paul Rosero
paur@itu.dk
IT University of Copenhagen, DK

Philippe Bonnet
phbo@itu.dk
IT University of Copenhagen, DK

ABSTRACT

Internet-of-Things (Iot) devices are becoming increasingly prevalent, with many of them not only relaying data to the Cloud but also being capable of local computation. This capability could be used for many purposes: detecting sensor tampering, compression or anonymization of data before uploading to the cloud, or even participating in distributed Machine Learning.

IoT devices are not only at risk of malicious and misbehaving software, but due to their deployment in unprotected locations, they are also at risk of physical attackers and tampering. Even though there are many exciting local computation ideas, the authenticity of computations performed on most IoT devices cannot be guaranteed. In clouds, Trusted Execution Environments (TEEs) already offer trust in the computation carried out even in the presence of a physical attacker, without slowing applications down. In IoT devices, however, such TEEs introduce large performance overheads and increase energy consumption.

In this project we propose a radical way forward: to design IoT platforms with processors that do not rely on off-chip memory and instead keep application state on on-chip memory that is easier to protect. This design reduces the overhead of TEEs significantly: it eliminates the cost of securing off-chip memory from attackers. It is important to note that, in addition to fresh thinking on how to design processors with more on-chip memory, computation will also have to be re-imagined to fit in a reduced memory footprint.

ACM Reference Format:
Zsolt István, Paul Rosero, and Philippe Bonnet. 2022. Always-trusted IoT — Making IoT Devices Trusted with Minimal Overhead. In *Proceedings of Proceedings of the 5th Workshop on System Software for Trusted Execution (SysTEX '22 Workshop)*. ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

There are a growing number of proposals for local computation on IoT devices ranging from data anonymization, compression, outlier detection, to complex distributed algorithms, such as Federated Learning (FL) [3, 7, 9]. By performing computation locally and having to communicate less with the Cloud, IoT devices can protect data privacy, take decisions faster and operate more efficiently. Especially in the case of FL, pre-training locally and then merging results in the cloud allows devices to benefit from a centralized view on the data without sending all private information to the cloud – this increases privacy guarantees and reduces costly data movement. The latter aspect is important because IoT devices often have very reduced network bandwidths.

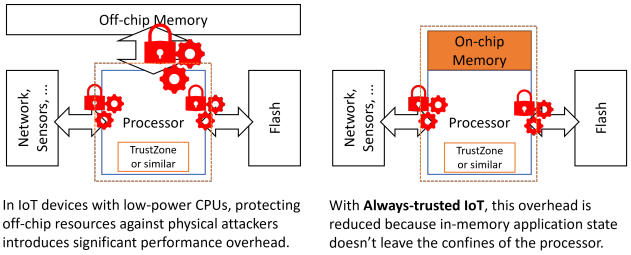

Figure 1: The most efficient way of performing main-memory encryption is to avoid it in the first place!

Unfortunately, for virtually all local computation ideas to realize their promise, there needs to be certainty that the IoT devices perform computations correctly and are not sending bogus results / sensor readings to the cloud. To protect the operating system and applications that share the same device from each other, practical solutions already exist, e.g., based on ARM TrustZone [5, 10, 11]. The same applies for detecting compromised devices in some scenarios, e.g., with MUD [12] that identifies them based on their network traffic patterns. Even remote attestation of the OS/application code could be added by using a small TPM [13]. However, when it comes to protection against attackers who gain physical access to the device and who could tamper with off-chip resources, the performance overhead of protection mechanisms is significant. Powerful server CPUs offer Trusted Execution Environments (TEEs), e.g., Intel SGX, with negligible energy or performance overheads. This makes it easy to deploy applications that benefit from trust guarantees. In contrast, in power-efficient IoT platforms (e.g. devices similar to a Raspberry Pi or NXP iMX board), it is relatively costly to perform encryption/decryption at high bandwidth.

Protecting IoT devices from attackers is a prerequisite for making them smarter. Securing communication between the IoT device and the "outside world" is achievable but networking and persistent storage have orders of magnitude lower bandwidth and higher latency than DRAM. Related work shows that securing off-chip memory through encryption can lead to slowdowns of several orders of magnitude [4, 8]. *This leads to an impasse: many of the local computation ideas require large memories but securing application state in memory outside the CPU is impractical in most IoT devices.*

2 ALWAYS-TRUSTED IOT

IoT devices are arguably in need of the guarantees that modern TEE technologies provide: they should be secured against physical attacks, attest that their firmware/software is unchanged and ensure confidentiality and integrity of sensitive data. Given that IoT devices are deployed in large numbers, providing these guarantees should be of high priority – even more so with the emergence of local computation in the IoT devices and the offloading of many cloud

SysTEX '22 Workshop, @ASPLOS, Lausanne, Switzerland
2022.

functions to the Edge. As outlined in the previous section, these goals conflict with the requirement that Iot devices remain energy efficient and cheap.

Our vision is to build future trusted IoT devices that provide TEE guarantees without significant performance or power overhead. We propose achieving this by not using off-chip memory and instead *relying on memory that is tightly integrated with the compute element of the IoT device*, making it impractical for attackers to tamper with its contents. In the absence of off-chip memory, there is no need for encryption or integrity checks [1] – hence, there is no need to add power-hungry dedicated silicon to perform these operations!

While our idea might sound as an "extreme design point", we are not necessarily saying that future IoT devices should have no off-chip DRAM. Instead, we advocate for treating on-chip memory as the default location for all local computation state and designing applications with this in mind. Off-chip memory could still be used to increase capacity through paging, but accessing it should be the exception rather than the rule.

There are existing proposals for using on-chip memory to secure secrets from attackers. e.g., the work from Colp et al. [2] and Zhang et al. [15]. The motivation of these works, however, is different from ours: they aim to either secure secrets that would otherwise reside as plain text in DRAM or to utilize the on-chip memory as a cache for encrypted off-chip memory. Our goal is to shift the way we think about the role of on-chip and off-chip resources of IoT processors and to design IoT platforms that prioritize the former.

3 OPEN QUESTIONS AND CHALLENGES

With this short paper, we would like to raise interest in the idea of Always-trusted IoT and initiate collaborations across the computer architecture, systems, and machine learning communities. These collaborations are needed because, in addition to adopting a new way at thinking about the role of on-chip memory in IoT devices, we need to achieve progress on several fronts:

1) First, we need to design SoCs/processors for IoT devices that have significantly more on-chip memory than today. This is necessary in order to accommodate a wide range of local computation ideas. Currently, it is not common to add large amounts of SRAM to IoT processors, or to include resources like HBM, these being reserved for HPC-like use-cases. It is also clear that it is not realistic to increase the memory size of IoT devices drastically – otherwise it might be cheaper to include crypto accelerator logic on the processor. Open questions in this area are, for instance: Among technologies for adding on-chip memory to small footprint SoCs, which ones are most energy efficient? What is the break-even point of energy consumption between performing more compute to secure off-chip memory versus adding on-chip memory to the device?

2) Second, we need to re-imagine the algorithms underlying local computation ideas, so that they can happen in a more modest memory footprint. Many "hot" ideas, such as Federated Learning (FL), trace their lineage to the cloud server world, where memory is abundant and wide SIMD units and GPUs can be used to perform high performance computation. In an IoT setting, however, it is an open questions to what extent can we reduce the memory footprint of FL algorithms without significantly impacting their final model quality. In this context there are already some existing work one could look at for inspiration, e.g., executing ML operations in power-constrained devices (e.g., the Raspberry Pi-targeting implementation of FedML [6]), as well as, exploiting quantization and reduced precision to fit working sets into small on-chip memories, for instance, on FPGAs [1, 14] – the setting, however, is new and off-the-shelf solutions will not work off the bat.

3) Third, it is worth considering what novel local computation ideas could be implemented in a future in which IoT devices have TEE guarantees without negative energy of performance impact thanks to the Always-trusted approach. Once computation on these devices can be trusted, we should be able to authenticate sensors and to detect tampering more easily, increasing this way the general trust in "smart infrastructure" powered by IoT devices.

[1] Of course, protecting against malicious local computation, securing DMA access, virtual memory protection, encryption of networking channels and persistent storage, such as SD cards, can and should be done according to the state of the art methods.

REFERENCES

[1] Michaela Blott, Thomas B Preußer, Nicholas J Fraser, et al. 2018. FINN-R: An end-to-end deep-learning framework for fast exploration of quantized neural networks. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 11, 3 (2018), 1–23.

[2] Patrick Colp, Jiawen Zhang, James Gleeson, Sahil Suneja, Eyal De Lara, Himanshu Raj, Stefan Saroiu, and Alec Wolman. 2015. Protecting data on smartphones and tablets from memory attacks. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*. 177–189.

[3] Angelo Feraudo, Poonam Yadav, Vadim Safronov, et al. 2020. CoLearn: Enabling federated learning in MUD-compliant IoT edge networks. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*. 25–30.

[4] Andrew Ferraiuolo, Andrew Baumann, Chris Hawblitzel, and Bryan Parno. 2017. Komodo: Using verification to disentangle secure-enclave hardware from software. In *Proceedings of the 26th Symposium on Operating Systems Principles*. 287–305.

[5] Le Guan, Peng Liu, Xinyu Xing, Xinyang Ge, et al. 2017. Trustshadow: Secure execution of unmodified applications with arm trustzone. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 488–501.

[6] Chaoyang He, Songze Li, Jinhyun So, et al. 2020. FedML: A Research Library and Benchmark for Federated Machine Learning. arXiv:2007.13518 [cs.LG]

[7] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, et al. 2021. A survey on federated learning for resource-constrained IoT devices. *IEEE Internet of Things Journal* (2021).

[8] Dayeol Lee, David Kohlbrenner, Shweta Shinde, Krste Asanović, and Dawn Song. 2020. Keystone: An open framework for architecting trusted execution environments. In *Proceedings of the Fifteenth European Conference on Computer Systems*. 1–16.

[9] Jed Mills, Jia Hu, and Geyong Min. 2019. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE Internet of Things Journal* 7, 7 (2019), 5986–5994.

[10] Sérgio Pereira, David Cerdeira, Cristiano Rodrigues, and Sandro Pinto. 2021. Towards a Trusted Execution Environment via Reconfigurable FPGA. *arXiv preprint arXiv:2107.03781* (2021).

[11] Sandro Pinto, Tiago Gomes, Jorge Pereira, et al. 2017. IIoTEED: An enhanced, trusted execution environment for industrial IoT edge devices. *IEEE Internet Computing* 21, 1 (2017), 40–47.

[12] José L Hernández Ramos, Sara N Matheu, Angelo Feraudo, et al. 2021. Defining the behavior of IoT devices through the MUD standard: review, challenges and research directions. *IEEE Access* (2021).

[13] Paul Georg Wagner, Pascal Birnstill, and Jürgen Beyerer. 2020. Establishing Secure Communication Channels Using Remote Attestation with TPM 2.0. In *International Workshop on Security and Trust Management*. Springer, 73–89.

[14] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. 2017. Zipml: Training linear models with end-to-end low precision, and a little bit of deep learning. In *International Conference on Machine Learning*. PMLR, 4035–4043.

[15] Meiyu Zhang, Qianying Zhang, Shijun Zhao, et al. 2019. Softme: A software-based memory protection approach for tee system to resist physical attacks. *Security and Communication Networks* 2019 (2019).