Memory Protection for IoT Devices

Authors
Ionuț Mihalcea, MSc (Royal Holloway, 2022)
Konstantinos Markantonakis, ISG, Royal Holloway

Abstract

Internet-connected computing devices are quickly becoming a pervasive part of our environment. Their growing adoption in fields such as healthcare and infrastructure is predicated on increased efficiency and novel functionality. However, embedding the Internet of Things into systems critical to our society also brings more significant risks that we must defend against. Many of these devices are susceptible to attacks over the internet and must operate in physically exposed locations. This article makes a case for the security mechanisms necessary to protect them from their physical environment. While safeguarding data found in transit between devices is ubiquitous, similar protections for the data in use on a device are far less common. We highlight the complexity of thwarting an attacker who can access a device's memory and discuss the performance cost of such defences.[a]

[a]This article is based on an MSc dissertation written as part of the MSc in Information Security at the ISG, Royal Holloway, University of London. The full thesis is published on the ISG's website at `https://www.royalholloway.ac.uk/research-and-teaching/departments-and-schools/information-security/research/explore-our-research/isg-technical-reports/`.

1 Introduction

As highlighted by recent surveys (e.g., [1]), multiple industry sectors have been adopting the Internet of Things (IoT) with increasing momentum and interest. Computing devices are embedded in household items, industrial machinery, and critical infrastructure. An essential driver of this growth is a desire to reduce costs and improve efficiency. Given the current economic outlook, the focus on tightening budgets appears unavoidable in the near future. However, this increased deployment base should come with increased scrutiny of its challenges.

Many business leaders are concerned with the security of IoT devices, and for a good reason. When these platforms become critical components in electricity control or oil refinement, the threats they can pose grow to a societal level. A security failure at this level can lead to substantial economic loss and an imminent threat to human life. The IoT field must contain these risks and provide safety guarantees to mature, and policymakers are slowly stepping in to ensure that[2].

Security is by no means a niche or new endeavour in computing. Specialised protocols such as the Transport Layer Security (TLS) protocol protect data travelling between machines. Complex software and hardware features oversee and prevent malicious code from taking over our computers. But not all of these were designed with the limitations and context of IoT devices in mind. For example, the performance of these devices is bounded by their cost, size, and available power. Their constrained performance, in turn, limits the processing capacity that can be expended on their security features.

While a personal computer in an office might be compromised if a post-it with the password is stuck on the monitor, defensive measures are commonplace. On the other hand, the very nature of some IoT deployments leaves them completely exposed. Anyone

A more atypical security problem raised by IoT is that of physical security.

[1] `https://www.computerweekly.com/news/252525441/IoT-success-fuelling-further-expansion`
[2] `https://www.atlanticcouncil.org/in-depth-research-reports/report/security-in-the-billions/`

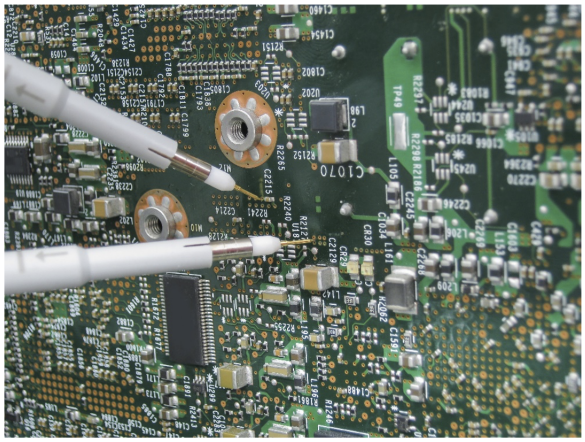

Figure 1: Physical access to a device yields attack avenues far beyond those available to remote attackers. Source: `nano-di.com/resources/blog/2019-implementing-physical-layer-security-in-iot-devices-using-additive-manufacturing`

could roll up in the field with a laboratory worth of tools
and plenty of time to spare. The devices must be assumed under an attacker's physical control - what measures must we put in place to secure them?

2 IoT security Landscape

IoT devices run familiar software stacks and present standard network interfaces despite their limitations. The whole security context, therefore, provides a mix-and-match between generic and context-specific issues to tackle[3]. For example, OWASP IoT Top 10 cites insecure network services among the things to avoid in IoT systems – a common worry for any internet-connected device. The same list also mentions the lack of physical hardening, a markedly IoT-specific issue given the lax physical controls in their environment. Other methodologies and industry frameworks, such as the Platform Security Architecture[4], address similar concerns.

This asymmetry is also reflected in the ecosystem of software and hardware solutions tackling these threats. The ease with which software vulnerabilities can be exploited over a network makes them ideal for infiltrating otherwise physically secure machines. Entire shadow industries have spawned to capitalise on these threats at the expense of businesses. Research and development of solutions against these vulnerabilities have amassed significant attention. Engineers are now spoiled for choice in terms of security mitigations, from using more secure programming languages[5] to hardware features[6] that help detect memory safety violations. Defending IoT devices from remote threats is thus bound to become more accessible.

However, IoT systems can only sometimes rely on a physically secure environment. Their exposure can be used to gain direct access and control over the device's data. Telecommunications can be intercepted or changed, data stored on hard disks can be extracted or corrupted, and even signals sent between discrete components of the device (such as the processor and main memory) can be exfiltrated or modified. Without physical hardening, general-purpose IoT devices represent easy targets for a well-equipped and knowledgeable adversary. While secure protocols designed for constrained devices exist to protect data in flight, and tooling exists to enforce confidentiality and integrity for data at rest, protecting data in use is still a nascent field.

[3] `https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf`
[4] `https://www.psacertified.org/`
[5] `https://www.rust-lang.org/`
[6] `https://www.arm.com/blogs/blueprint/securing-software`

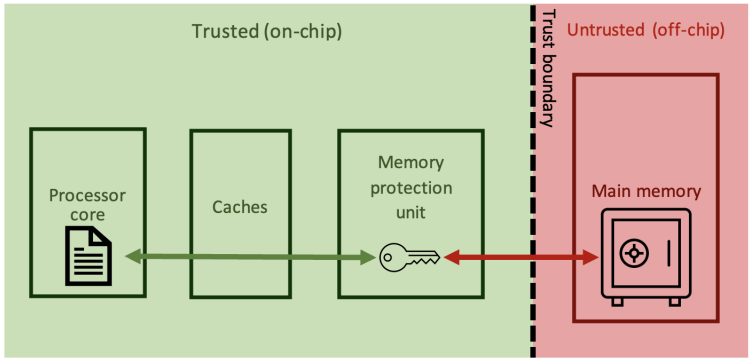

Figure 2: A processor can safeguard itself by relying on a memory protection engine to cryptographically protect all data leaving the chip

3 A Silicon Bastion

The processor and the primary memory that serves its instructions and data sit at the core of every computer. If we seek the protection of data in use, we must require that only the authorised components of the device can view or change data in primary memory. In most larger devices – including IoT devices – the processor and memory are separate components connected by a data bus. An unfortunate consequence of such designs is the ease with which this bus can be probed. This allows an attacker to snoop on or alter all data passing between processor and memory, voiding our previous requirement. A few avenues are open that take us to our desired security stance.

The most apparent solution is to combine the processor and memory as one component or to make their physical connection inaccessible. The chips inside millions of bank cards use this approach, protecting the valuable secrets stored inside them. Unfortunately, scaling up such a design to the size of an IoT device is too costly.

Another option is establishing a secure communication channel between processor and memory, transforming the data-in-use problem to data-in-flight instead. Data would be cryptographically protected when travelling between the two components with a minimal performance cost. This fulfils our requirement, with one major caveat: in many cases, the memory component can be easily replaced with one controlled by the attacker, who could then remove the protection.

A more straightforward and more robust solution is to transform the processor into a bastion. All data leaving it is cryptographically protected with keys that only the processor knows. Retrieved data is verified and released, ensuring no one has interfered. Can such an approach be efficient enough for IoT devices?

4 Cryptographic Memory Protection

Modern processors and cryptographic algorithms have been co-designed to ensure superior performance even in constrained situations. Software applications can benefit from this to protect their data. Confidentiality and integrity can be efficiently provided for messages sent over the network or even to the data kept in memory by the application. However, for the application to use these algorithms, it must also keep hold of the correct cryptographic keys, which end up in memory. This is the cryptographic equivalent of a post-it with the password stuck to the monitor – if both the key and the encrypted data are kept in memory, an attacker who can read the memory can decrypt the data.

Thus, the processor is responsible for protecting applications from an untrustworthy environment. It must secure all data leaving its confines, verifying it when retrieved from memory. Since the crypto-

graphic keys used by the processor are central to the security of this bastion, they must remain inside, accessible only to the processor.

Cryptographic algorithms that grant confidentiality and integrity at a low cost are readily available. However, deploying these algorithms in the context of data in memory comes with a few subtle problems. Take confidentiality: if all memory blocks are encrypted in the same way with the same key, different memory blocks that contain an identical value would also be identical when encrypted. If an attacker controls what gets stored somewhere in memory, they could try guessing the value of some desired data until their encrypted values match. Hence, the processor must perform a variation of the encryption mechanism for each location. Since memory is organised as numbered data blocks, the solution is simple: use this address to enable differentiation.

The primary role of integrity protection is to guarantee that the data was not modified outside the processor. Algorithms that facilitate it produce an integrity tag, an identity document for the protected data, which can be verified with the correct key. The tag is stored alongside the data it represents. Despite these guarantees, there are still means to subvert the defences via replay attacks. An attacker can record the data along with its identity tag. When new data is written at that exact location later, they could erase it and write the old values back, replaying them. The processor needs a more trustworthy view of what data it owns to prevent this threat. The solutions bring more complicated trade-offs.

5 The Cost of Complete Protection

Sensible trade-offs are predicated on clear and accurate metrics. A significant consequence of primary memory protection is the impaired performance of the running applications. Thus, we are primarily concerned with their speed of execution, which we can measure and compare for a wide range of applications which stress the system in diverse ways. Since memory protection adds extra friction to data traffic between processor and memory, how the system behaves when the data bus is under increased load is also worth understanding.

We can enforce confidentiality and integrity protection – without considering replay attacks – for less than 5% performance degradation on average, even on IoT devices. This performance is achievable even in high memory traffic circumstances. 3-7% of the physical memory available on the device must also be reserved by the processor to store integrity tags.

However, solving the issue of replay attacks comes at a steeper cost. One solution would be to construct something akin to an integrity tag for the entire memory and have that stored securely within the processor. Tree-based data structures (such as Merkle Trees) represent the core mechanism for efficient verification. The tree's root is stored and updated regularly within the processor and used to anchor the verification of any data coming from memory. A considerable drawback of these structures lies in the extra memory traffic needed to probe and maintain the tree. Depending on the data structure, applications can see a slowdown between 5 and 25%. When the data bus is under significant load, the overhead of the integrity checks shoots up to over 100%.

Another option would be to store the integrity tags within the same package as the processor. This prevents a replay of old data since the integrity tag can no longer be changed to an older version. It also reduces the performance penalty to 5% even when the data bus is stressed. The drawback is a considerable cost in silicon space that needs to be dedicated to the integrity tags, enough to store up to 7% of the main memory.

An alternative way to prohibit replay attacks is to count how often each memory location has had data written to it. This counter is then included in the cryptographic computations, much like the address of the memory block. The problem is that the counter must also be saved in memory and thus needs to be protected, leading to another tree-like structure. An advantage of counters, however, is that they can be compressed. By allocating less than 1% of the primary memory size as storage space for counters within the processor chip, the performance overhead in a stressed system can be reduced to less than 50%.

Such comprehensive defensive solutions first saw use in server processors, insulating applications from their environment and any other software running on the platform. Their availability and user base have since grown to high-end consumer devices. However, the production and performance costs still represent an oversized burden for smaller, less powerful IoT devices. As is often the case, the features that first appear in specialised, high-performance systems trickle down to cheap, general-purpose devices.

With the increased interest and mounting regulatory pressure, the IoT ecosystem is thus ripe for investments towards enabling comprehensive security.

6 Conclusion

The widespread adoption of IoT devices must be met with a comparable push towards securing them. While much progress has been made on some fronts, security against physical attacks is still lagging. Our existing mechanisms, whilst demonstrating an ability to secure the confidentiality and integrity of data in main memory, come at a prohibitive performance cost. This trade-off remains an issue that needs to be addressed.

As the field continues to evolve and the threat of physical attacks becomes increasingly prominent, further research and development efforts are required. A commitment to innovation in this field is needed to pre-empt the increasing complexity of security threats.

Biographies

Ionuț Mihalcea is a senior software engineer in Arm's Architecture and Technology group. He works towards enabling hardware-backed security mechanisms by developing and maintaining open-source projects and contributing to standards and initiatives in the confidential computing space. He has a keen interest in applied cryptography.

Konstantinos Markantonakis is a Professor at the Information Security Group at Royal Holloway University of London and the Director of the Information Security Group Smart Card and IoT Security Centre (SCC) with research, teaching, and managerial responsibilities. He is also the Director of the Transformative Digital Technologies, Security and Society Catalyst responsible for coordinating multidisciplinary and impactful research. His main research interests involve smart card security and applications, IoTs/CPS, embedded system security and trusted execution environments, secure payment systems, cloud computing and transparent and explainable AI. He has published more than 210 papers, articles and book chapters in international conferences/journals. He continues to act as a consultant on a variety of information security topics.

Series editor: Dr Maryam Mehrnezhad, ISG, Royal Holloway