SILICON LABS

Why IoT Solutions Need Hardware-Based Security

Why IoT Solutions Need Hardware-Based Security

Sharon Hagi, Chief Security Officer

Security continues to be paramount for any electronics-based design, and this is especially the case for IoT deployments where complexity, resource constraints and a high degree of connectivity exist. While much talked about, the approach to IoT security needs to rely on established security principles as well as careful consideration to the ever-changing threat landscape. In this white paper we investigate some of the IoT security challenges design engineers face when bringing their products to market. In particular, the paper highlights why a software approach to IoT security implementation can no longer meet acceptable levels of assurance and protection as well as sets out the reasons why a hardware-based approach is now considered an essential element of any IoT solution design and implementation. This paper also emphasizes the benefits of taking a more holistic approach to IoT security, embracing a new mindset beyond past practices.

Contents

IoT Security – The Threat Landscape

The IoT is being integrated into the fabric of most industrial and commercial operations, including utilities, critical infrastructure, transportation, finance, retail and health. IoT devices are designed to sense and measure the physical world and gather data about every aspect of human activity. These devices facilitate highly distributed intelligence, automation, and autonomous command and control. The degree to which IoT enables this through smart, highly connected, pervasive and ubiquitous devices will allow companies to create truly revolutionary technologies that have the promise of improving every aspect of human life, social as well as economic for generations to come. That said, not a week goes by without a major media outlet highlighting a digital security breach, the result of which typically involves stolen consumer credit card details or a compromised operation. Unfortunately, such news only mentions one of many thousands of security attacks taking place over the internet every day. Such threats may serve to extract valuable data, create widespread disruption, or even more alarmingly, take control of critical systems. Distributed Denial of Service (DDoS) attacks are probably the most documented threat from a consumer's viewpoint. In 2016, the Mirai botnet (which caused internet-wide disruptions) was the first major wake-up call for organizations to acknowledge these types of threats. Since then, Mirai successors such as Aidra, Wifatch and Gafgyt as well as newcomers such as the BCMUPnP, Hunter52 and Torii53 botnets have amassed access to millions of IoT devices to spread their DDoS attack malware, cryptocoin-mining malware and spam relay proxies.

Security threats are omnipresent and increasing in magnitude as we deploy and connect more of our society and the workplace. Consider the impact on smart cities. Underpinned by ubiquitous wireless communication and machine/deep learning, the rationale behind a smart city includes demand-adaptive traffic control, automatic load-balancing management across the power grid and smart street lighting. Take one of these examples, such as smart traffic control, and imagine the potential attack surfaces presented to an adversary in terms of traffic flow sensors, traffic lights, automotive mesh networking for adaptive coordination between vehicles and the control equipment infrastructure across a large city. Taking control of the traffic lights or the communications between vehicles in a wireless mesh at a major intersection is no longer a scenario only played out in a Hollywood-blockbuster, but a somber reality.

Consider also the rise of internet-connected medical devices, smart-tags in stores to aid the retail shopping experience, and how our homes and appliances are getting connected. If you can turn on your furnace, unlock the front door and disarm the alarm system using your smartphone, can someone else?

We can all relate to the above examples, but what about use cases we, as consumers, don't see? Imagine an industrial Internet of Things (IIoT) deployment for an automated manufacturing environment -- what chaos could a security breach cause, and what might be the financial consequences of production downtime and damaged equipment?

With an exponential rise in the number of potential attack surfaces, IoT security must be pervasive, robust and resilient (Figure 1).

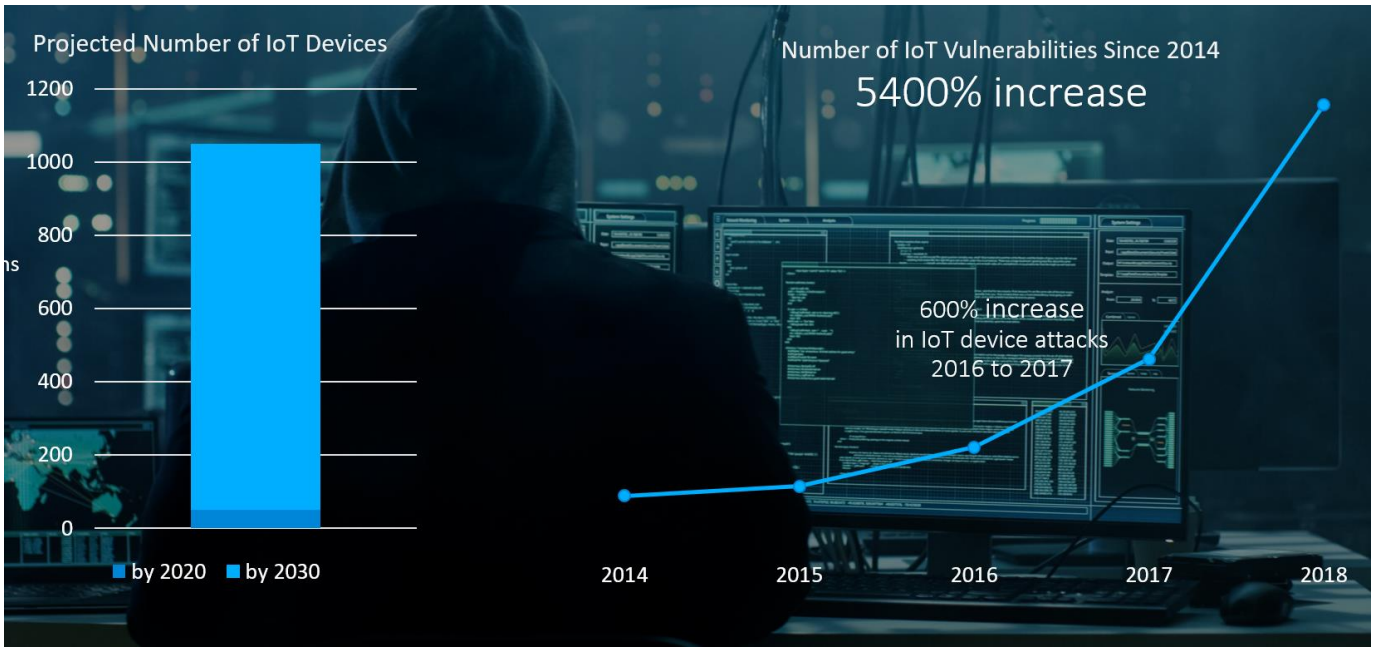

Figure 1 – Exponential growth in IoT devices and threats (Sources: Gartner, Softbank, IBM X-Force Threat Intelligence Index 2019, Symantec Internet Security Threat Report 2018)

Why a Software-Based Approach to IoT Security is Not Enough

Attempting to eavesdrop or illegally access information is not new. Some of the earliest recorded events include the efforts by the Dutch computer researcher Wim van Eck in 1985 at phreaking (reading) information from a visual display unit based on intercepting and decoding the display's electromagnetic fields. His groundbreaking work highlighted the fact that by using just a handful of inexpensive components, he could bypass the costly security methods in place.

Such non-invasive and passive electromagnetic side-channel attacks have become even more sophisticated today, and one of many tools in the adversary's armory. Other side-channel attack methods include differential power analysis (DPA) and are typically undertaken together with an electromagnetic side-channel attack. With this method of attack, sensitive information such as encryption keys, passwords and personally identifiable information can be 'leaked' as electromagnetic signals from the IoT device's microcontroller during execution of cryptographic processing instructions. A wide-bandwidth receiver, cheaply available today as a software-defined radio application, is used to detect and store electromagnetic signal patterns alongside the timeline of operation.

DPA is a slightly more sophisticated eavesdropping method. Simple power analysis works on measuring the device's processor power consumption during operation. Since the power consumed by processing devices varies depending on the function performed, it is possible to identify discrete functions by zooming into the power consumption timeline. AES, ECC and RSA-based cryptographic algorithm functions are compute-intensive and are identified from analysis of the power consumption measurements. Examining the power consumption at a micro-second interval can reveal individual numeric operations frequently used in cryptography, such as squaring and multiplication. DPA adds statistical and error-correcting techniques to simple power analysis to achieve high accuracy decoding of secret information.

Intercepting data transferred through wired or wireless communications methods can also reveal secret information. Covert channel and “man-in-the-middle attacks” are an effective way to listen in on the communications between an IoT device and the host system to gather data. Analysis of that data may give away equipment control protocols and the private keys necessary to take over the operation of a remote connected device.

Fault injection attack targeting unprotected microcontrollers and wireless system-on-chip (SoC) devices is another technique used by hackers. At its simplest level, this technique may involve reducing or glitching the supply voltage to a microcontroller so that it exhibits erratic error conditions. In turn, these errors may trigger other protected devices to open registers that hold secret information, which can then be accessed. Tampering with the system’s clock signal(s), either by changing the frequency, injecting false triggers or altering the signal level, can also result in unexpected device behavior that would then propagate around the IoT device, exposing secret information or creating the potential to take over the control functions. Both of these cases require physical access to the device printed circuit board (PCB) but are not invasive.

Gaining access to secure information is possible because many security techniques used to protect IoT devices are software based. Standard cryptographic encryption algorithms such as AES, ECC and RSA run as software-stacks on microcontrollers and embedded processors. Being able to observe the power consumption and use DPA techniques to capture secret keys and other sensitive information is very accessible today using equipment and software that costs less than $100 USD. Readily available DPA software tools automate the whole process, so you don’t even have to be an expert in the analysis methodology used.

These types of attack are no longer exclusively in the theoretical domain; they are being used widely today by hackers worldwide.

With a growing list of attack surfaces and vectors, developers of IoT devices and systems need to re-think their approach to implementing and incorporating security capabilities so that they can be more robust and resilient.

Changing the Mindset to a Hardware-Based Security Approach

If you are about to embark on the design of a new IoT device, we recommend that you conduct a thorough review of the likely attack surfaces the device would present and the threat model that it would have to be secure against. Often the design specification for embedded systems commences with the functional requirements of the product and how it should work. Reviewing and incorporating security needs into the product specifications from the beginning is a prudent first step. Most IoT devices are expected to be in service for many years, in which case over-the-air (OTA) firmware updates are a necessity, and that capability alone introduces further attack surfaces to be considered. To cover all attack vectors requires a silicon-to-cloud approach to implementing a hardware-based security design methodology.

Implementing a Hardware-Based Approach to IoT Security

In this section, we explore some of the hardware-based techniques used to provide a robust security regime for an IoT device. As the reader will learn, implementing security in hardware starts in the foundry and creates an immutable identification that cannot be changed. In so doing, the cost of attempting to compromise such an IC or device is significantly higher than the tools required to achieve the security breach through software. When selecting microcontrollers or wireless SoCs, embedded design engineers should consider that the review of a device’s hardware-based security features is as important as the other device criteria such as clock speed, power consumption, memory and peripherals.

Root of Trust

Establishing a root of trust (RoT) is the first step of a hardware-validated boot process for any processor-based device. Usually embedded as a root key or image in read-only memory (ROM) during IC wafer fabrication at the foundry, a RoT is immutable and forms the anchor to establish a chain of trust as the device initiates the bootstrap process. The RoT can also include the initial boot image, ensuring that from the execution of the first instruction, the device is running both authentic and authorized code. The RoT secures the device from being compromised by a rootkit attack.

Secure Boot Process

The next step in creating the chain of trust is booting the device using a secure boot process. After the first stage of boot has been completed using an authenticated and authorized RoT image, the second stage of boot commences. A secure loader then authenticates and executes the main application code. Figure 2 illustrates this approach using a dual-core device, although this process can also take place using a single-core device. The secure loader can instigate an update process before code execution if required. Wireless SoCs from Silicon Labs feature an enhanced Secure Boot implementation called Secure Boot with Root of Trust and Secure Loader (RTSL).

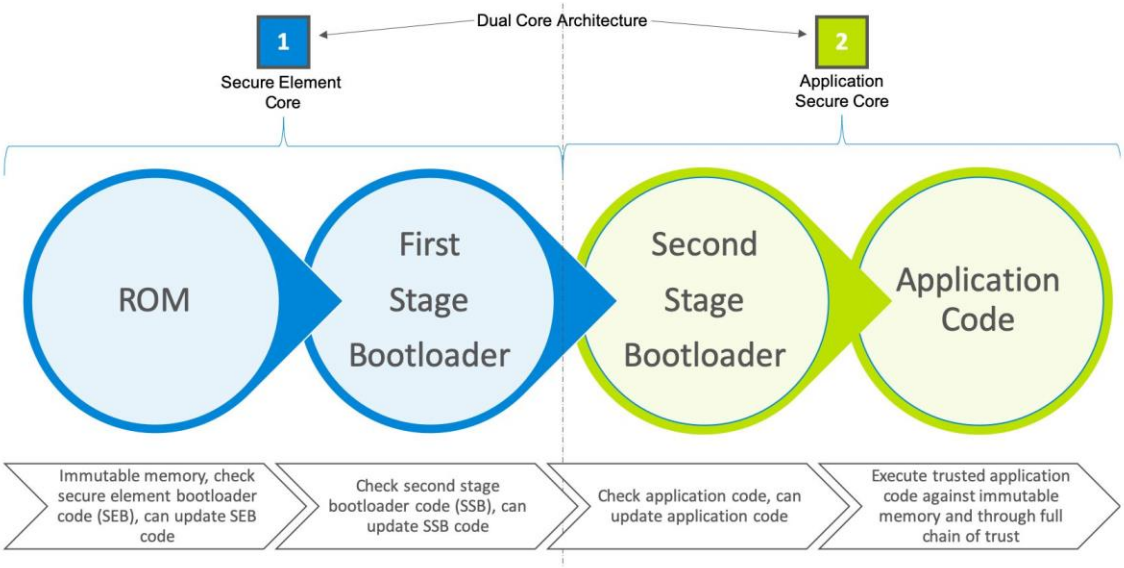

Figure 2 – The root of trust and secure boot process (Source: Silicon Labs)

Another hardware-based technique that greatly aids security is the use of physically unclonable functions (PUF). PUFs are physically created within the silicon die during wafer fabrication and, due to unpredictable atomic-level structure variations and their impact on intrinsic gate or memory cell electrical properties, provide a unique identity for the semiconductor device. Essentially, the unpredictable/chaotic variances create a unique “fingerprint” for each IC, in essence a digital birth certificate. They are unclonable since even if you attempted to recreate an identical IC using the same process and materials, the PUF generated would be different. A repeatable cryptographic key can be extracted from the PUF using techniques that include employing one-way transformation functions (exploiting spatial variability) or an iterative challenge-response mechanism (exploiting temporal variability). PUFs are extremely secure and tamper resistant. The PUF key encrypts all the keys in the secure key storage, is regenerated at startup and not stored in flash. A full attack must be launched against the single device to extract the key. PUF-wrapped keys can also be handled by the application while remaining confidential. The technology and sophistication required to practically access a silicon die at nanometer scale for the purpose of either reverse engineering or perfectly cloning molecular variations inherit in the PUF’s underlying implementation are out of reach for most, if not all adversaries.

Silicon Labs embeds hardware-based security at the heart of every secure wireless SoC and module. Security is integrated throughout the product lifecycle, from silicon to cloud, and from initial design to end-of-life (Figure 3).

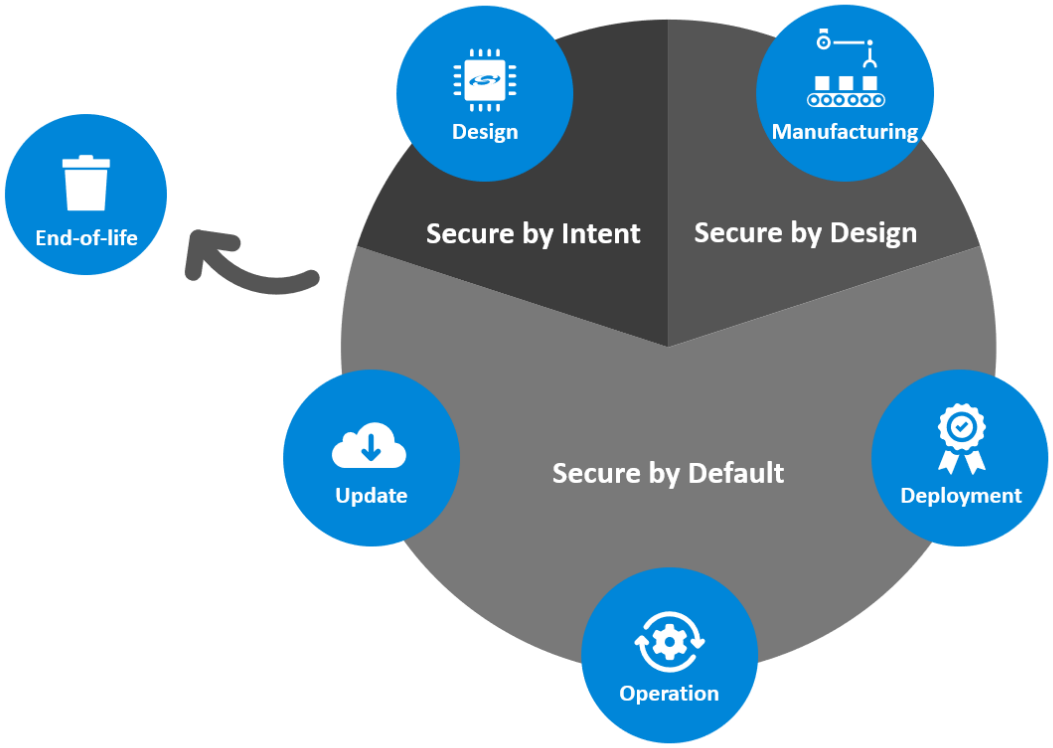

Figure 3 – Hardware security considerations through the device lifecycle (Source: Silicon Labs)

Secure Element

By provisioning the security features in hardware, adversaries face a difficult, costly and unproductive challenge attempting to access or intercept secret information. An example of a Silicon Labs wireless SoC with comprehensive hardware security functions is the Series 2 Wireless Gecko SoC.

The Series 2 SoCs feature a Secure Element, which isolates security from the host. Typically, attributes of a Secure Element are provided via separate chips, but Series 2 SoCs integrate everything onto one chip, resulting in stronger device security and lower bill of materials (BOM) costs for customers.

A Secure Element incorporates four key features to enhance device security: Secure Boot with RTSL, a dedicated secure core, a true random number generator (TRNG) and secure debug with lock/unlock (Figure 4). Secure Boot with RTSL provides trusted firmware execution and protection from remote attacks. A dedicated secure core incorporates DPA countermeasures, which include using random masks to protect the internal computations process and randomizing the timing of these computations performed in silicon. A TRNG helps create strong encryption keys by using non-deterministic high entropy random values and is compliant with NIST SP800-90 and AIS-31 standards. Secure debug locks debug interface to prevent access to chips in the field and allows debug interface authenticated unlock for enhanced failure analysis.

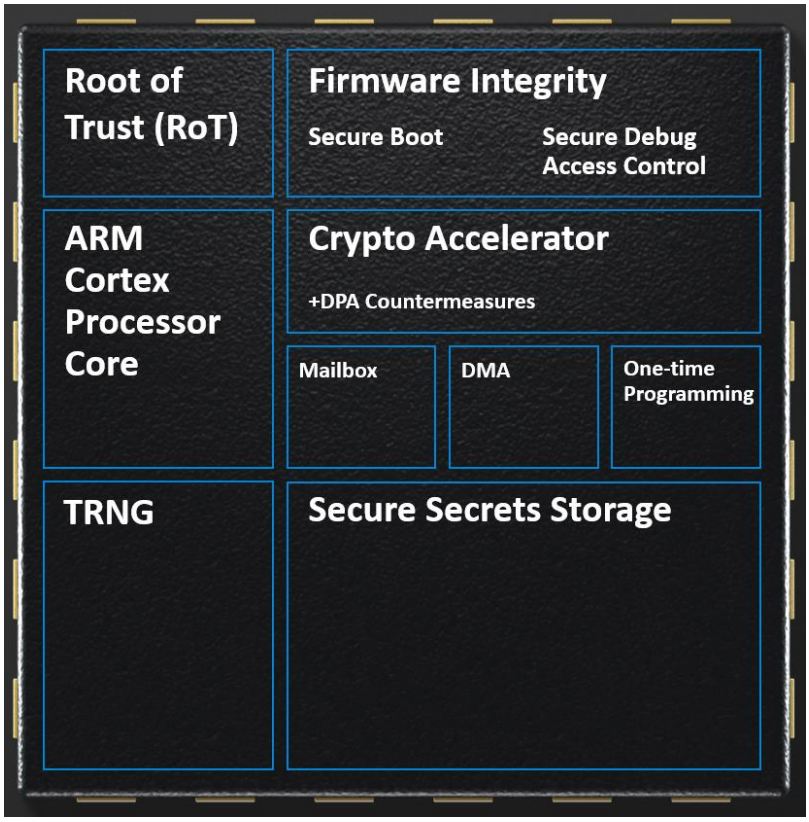

Figure 4: Series 2 Wireless Gecko SoC Architecture – Secure Element Core (Source: Silicon Labs)

Additional Considerations

Complementing the hardware security features described above, Silicon Labs provides Simplicity Studio, an integrated development environment (IDE) consisting of a range of software tools to streamline the development. Other features within Simplicity Studio include the ability to review a design's energy consumption profile and to analyze wireless network communication.

Silicon Labs is a participating member of the ioXt, the Internet of Secure Things Alliance. The ioXt has defined a certification process using internationally recognized security standards through which devices are assessed and rated for secure operation.

Conclusion

In addition to enabling robust security and reducing costs, using hardware-based IoT security provides an additional benefit: reduced power consumption. Running cryptographic algorithms in software can impose a significant compute load on a microcontroller, which increases power consumption and reduces battery life. Offloading crypto processing to a dedicated security core can result in a more energy-efficient and higher performance design.

The security threat to any connected device is omnipresent and ever-changing. Software-based security techniques have worked well in the past but have become an extension of the potential attack surface. Implementing security using hardware-based methods is now considered the only viable way a holistic and robust security regime can be implemented.