IP Protection and Hardware Assisted Security for IoT Devices

Parvesh Kumar Chaudhary, Ravi Kumar

Abstract— The Internet of Things, which is also known as the IoT, is a collection of smart devices that are connected to the internet and in turn to each other, which send and receive user data. Examples include wearable devices, vehicles, home appliances and industrial appliances which contain electronics, software, actuators and connectivity which allow these things to connect, interact and exchange data.

Integration of IoT devices into the standard Internet exposes it to several security challenges as the majority of Internet technologies and communication protocols were not designed to support IoT. The large scale use of IoT devices in Industrial automation, public utility systems like urban transportation systems, home automation has led to security concerns. Recently unprotected IoT devices were used in launching large scale DDOS attacks against chosen victim like Mirai, Bashlite. Iot devices are often deployed in locations that can be accessed easily for extended periods of time and are vulnerable to physical damage, tampering with switches and making connections to debugging and test ports. Side-channel attacks may allow the attacker to get encryption keys and other data by observing the power consumption, temperature fluctuations or electromagnetic emissions of a hardware device such as CPU or cryptographic circuit.

In this paper we provide an overview of the state-of-art methods and recommend ways for IP core protection and hardware assisted security of our IoT devices and reducing IoT attack surface. We utilise the most relevant hardware design practises providing a discussion of the benefits and limitations with reference to currently available hardware.

Index Terms— Crypto processor in IoT, Embedded Devices, Embedded device hardening, Hardware Assisted Security, Internet-of-Things (IoT), IoT Security, IP protection, Physical attacks on IoT.

1 INTRODUCTION

The Internet of Things(IoT) is collection of devices such as smart home devices, wearable devices, vehicles, manufacturing equipments that contains sensors, actuators, electronics, software and connectivity which allows these things to connect, interact and exchange data. The definition of the Internet of Things has evolved due to convergence of multiple technologies, cloud computing, Embedded Systems, Real-time analytics and machine learning. Embedded System, wireless sensor networks, control system, automation and others all contribute to enabling the Internet of things. The extensive set of applications for IoT devices is usually divided into clients from consumer, commercial, industrial and infrastructure spaces. In recent years there has been explosive growth of devices connected and controlled by the internet. The specifications of the IoT devices can be very different from one device to another but there are basic characteristics shared by most. IoT creates opportunities for a lot of direct integration of the physical world into computer –based systems, leading to potency enhancements, economic advantages and reduced human exertions. It has been estimated that there will be 30 Billion IoT devices by 2020.[1] The global market value of IoT is projected to reach $7.1 trillion by 2020.[2]

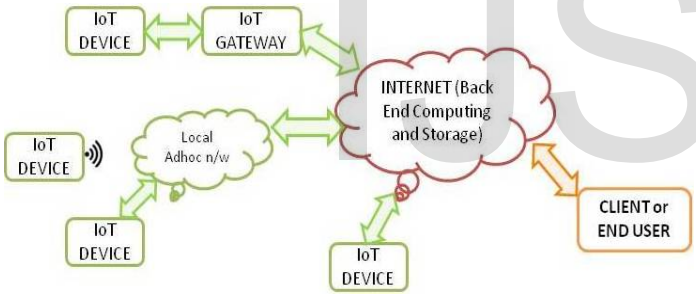
The key factors behind this huge number of IoT devices is the reducing cost of small, energy efficient, powerful processors that are capable of being embedded in almost every single electronic device of embedded in almost every single electronic device ever manufactured. Adding more and more Embedded devices to an IoT ecosystem has security implications especially as the devices are remotely deployed and these devices are difficult to secure physically and standards and guidelines from organizations such as the International Telecommunication Union (ITU), Department of Telecommunication (DoT) and the Trusted Computing Group (TCG) are still topics of discussion.

The devices in the Internet of Things can be of wide variety of data producing applications, devices, sensors or custom objects ranging from new to already existing devices which may or may not be complimented by a cloud. Many existing embedded devices produce useful data, for example cars and elevators, which can help with maintenance intervals and already exist in their millions. These mainly just lack Internet connectivity and back end cloud services to process the data. Now a day's many new IoT devices have Internet connectivity built into them and are produced in large quantities in an IoT enabled ecosystem. Many embedded electronics devices in IoT ecosystem are custom built.

In this paper we look at IoT architecture for custom devices and provide recommendations to improve security of the IoT devices against the embedded hardware attacks. Embedded electronics hardware is the root of trust for all secure computations and communications. Any cryptographic algorithm is implemented either directly in hardware or in software which eventually run on hardware. So in this paper we focus on the

recommendations for embedded electronics hardware design point of view right from the electronics hardware design for ensuring secure data collection, storage and transmission to an online server. The security of the application running on this hardware, security of online cloud service providers and social engineering attacks are considered out of the scope for this paper.

Figure 1 below shows a general high-level IoT ecosystem architecture. The IoT devices fall into three main categories (a) Sensors, which gather data (b) Actuators, which effect actions (c) Gateways, which act as communication hubs and may also implement some automation logic. At the lowest level we have IoT devices (Sensors or Actuators) which collect or produce data and can have limited storage and processing. These devices may be directly connected to the internet or often due to power, size, connectivity or RF range restrictions, they might connect via a local wired or wireless networks to an IoT Gateway. These gateways act as communication hubs, local data buffers, signal boosters and usually connected to Internet. Finally the data is sent to a server or data warehouse or cloud with enough storage and processing power to manage the data streams from millions of devices.



The user of these services access the data held in the data centre or cloud. For IoT computing to become a reality we need to establish the validity and authenticity of the data from the producer/sensors to the back end data centre/cloud. For this, there are two areas of concern (a) transmitting the data to the cloud (b) building an IoT device that is secure. For a secure IoT device, our focus in this paper would be on the design practices of embedded electronics hardware design.

2 Why Hardware Security Required

Attackers generally try to take advantage of poor design, but unintentional leakage of data or information due to ineffective security control measures can also bring dire consequences to consumers and suppliers.

IoT devices, services and software, and the communication network that connect them, are at risk of attack by a variety of malicious parties, bedroom hackers, professional criminals or even state sponsored hackers. Possible reasons of attacking hardware of IoT device could be

- Theft of services
  - Getting a service for free (Games, phone cards, pay-TV…)
- Theft of Intellectual Property(IP)
  - Reverse engineering/cloning/counterfeiting for market place advantage
- Theft of sensitive data/personal information
  - Bypass security to get access/control (steal encryption keys, PINs, …)

The consequences to consumers or suppliers of such attacks could include:

- Inconvenience and irritation
- Infringement of privacy
- Loss of life, money, goods, time, property, health, relationship, etc.
- Loss of trust
- Damage to reputation
- Compromised intellectual property
- Financial loss
- Possible prosecution

3 Building a Secure IoT Device Hardware

In this section we shall discuss IoT hardware based attacks and possible ways that would help to build a secure IoT device hardware, because hardware is the root of trust for all computations and communications. To build a secure IoT device requires an understanding of what we are trying to protect in our product, why we are protecting it and what types of attackers will likely target our product. The followings are the suggestions to avoid specific IoT hardware based attacks.

3.1 Reverse engineering

In Reverse Engineering type of attack, the attacker determines the part numbers of the major Integrated Circuits ICs) present on the target board and reconstructs netlist of the target design. Datasheets of the major components are available on the internet, so it becomes easy for the attacker to understand what the components do may provide details for particular signal lines that may be useful for active probing during operations. Many of the weakness, security vulnerabilities, and design flaws of a product are identified when analyzing the circuit board. A number of ways can be implemented at the circuit board design level to help prevent some attacks.

At the time of deciding placement of components on the PCB, important components that are most likely be targeted for an attack (Programmable devices, Memory devices, etc) should be made difficult to access. Use IC packages (like BGA or QFN) in which it is difficult to do casual probing,

manipulations and attack on the signals. One more solution is to make use of Chip-on-Board (COB) packaging, in which the silicon die of the IC is mounted to the PCB directly and protected by epoxy encapsulation. However, the protective layer can be removed by scraping, heating, cooling and by use of chemicals. X-ray can be used to get an image.

For PCB layout design, proper engineering practices should be followed always. Prefer to use multilayer PCB design (at least 4 layer or 6 layer PCB) and if possible use buried vias, which connect two or more inner layers but no outer layer, to reduce potential probing points for the attacker. Traces should be as short as possible. Differential signal lines should be aligned parallel and should be of equal length. Noisy power supply section should be kept away from sensitive analog and digital components. Properly designed ground plane should be employed to reduce EMI emissions. If test points are required, use copper-filled pad as compared to through-hole pad.

IoT device operation and information can be probed by simply removing the solder mask on the circuit board and tapping the address, data and control bus lines with a logic analyzer, oscilloscope or custom hardware [3]. Routing of critical bus lines onto the internal layers of the board would thwart such an attack. If a multilayer board is not used, protective encapsulation could be applied to the target traces.

Debug or external interfaces like JTAG, USB, RS232 or Ethernet are used for connecting to external peripherals, field programming or testing during development. Products often include development or programming interfaces that are not meant for end user use. The final production boards should not include any testing ports which are not required by the user, because such ports can benefit an attacker.

When an attacker gets access to an interface, the attacker will first try to probe the connections to determine the functionality of that interface if not known by obvious. To probe any connection, the attacker makes use of multimeter, Oscilloscope or Logic analyzer to know the type of signals by monitoring the test points for any device-generated signals and then manually toggling the state of the pins to induce a device response. By knowing the state of the pins can help an attacker make an educated guess on the type interface the product is using. Once the attacker comes to know about the type of interface, it is easy for an attacker to monitor the communication by using a dedicated protocol analyzer or software based tool. [4]

3.2 Memory Devices, PLDs and FPGAs

In embedded systems, most of the memory devices are insecure. Many EEPROM and SFLASH ICs do not provide any kind security to the data written into them. A hacker can get information stored in these non-volatile memory ICs just by tapping the physical connections to other device or by taking out the complete memory IC from the PCB board and mounting it on some other board. So to store the secret data one should consider of using secure memory devices. Through dynamic, symmetric mutual authentication, data encryption, and the use of encrypted checksums, secure memory devices provides a secure place for storage of sensitive information within a system. Such memory devices uses security features to stop regular device programmers or attacker from accessing stored data or data at rest, such as boot-block protection in Flash memory. Reading RAM or other volatile storage areas while the device is in operation may yield temporary stored data or plaintext components. The CryptoMemory family of EEPROMs and FLASH from various vendors includes features like encryption using password and authentication to allow access to data. The Atmel AT88SC0104C is an example of a secure memory device that has high security features like Encrypted checksum, stream encryption, four key sets for Authentication and encryption, eight sets of two 24-bit passwords, anti-tearing function etc.[5]

Protecting our intellectual property inside programmable logic devices (PLDs) and field programmable gate arrays (FPGAs) is as important as protecting firmware and data in memory of the device. Essentially, SRAM-based devices are the most vulnerable to attack due to their requirement to have configuration memory external to the device (configuration or program firmware in separate non-volatile memory), which is then loaded into the micro-controller or FPGA on power-up. The bit stream between the configuration memory and micro-controller or FPGA simply needs to be monitored to retrieve the entire configuration. If possible use such devices which eliminate the need for external configuration memories required by SRAM-based FPGAs.

The easiest attack against low-density PLDs with dedicated inputs and outputs and other circuitry is simple I/O scan attack. In simple I/O scan attack hacker attempts to reverse-engineer a programmable logic design by cycling through all possible combinations of inputs and then monitoring the outputs to determine the internal logic functions. To avoid such attacks, one should use unused input pins on the device to detect probing or tampering. Unused pins on these devices can be set as input pins, and if the detect a level change, the device can assume it is being probed and perform a countermeasure or response or connect the unused pins to a fixed state. An attacker may put the FPGA into an indeterminate state through fault-generation attacks, so while designing state machines in FPGAs and PLDs, ensure that all conditions are covered and that defaults are in place for unused conditions. Take the advantage of any on-chip security features available on that device, like enabling the simple fuse/software protection bits available in the FPGAs, adds a level of protection compared to no enabling it at all.

3.3 Anti-tamper mechanism

The purpose of Anti-tamper techniques is to prevent the attacker to perform any unauthorized physical or electronic action against the device. Anti-tamper techniques are divided into four groups

- Tamper Resistance
- Tamper Evidence
- Tamper Detection
- Tamper Response

Anti-tamper techniques are most effectively used to prevent access to any critical components on the device. These anti-tamper techniques are very much necessary for physical security of IoT embedded systems and must be properly implemented to be successful. Generally, the existing anti-tamper mechanisms on the device can be discovered by attempted or complete disassembly of the target product. This may require an attacker to obtain more than one device in order to sacrifice one for the purpose of discovering such anti-tamper mechanisms. Once the anti-tamper mechanisms are know, an attacker can form hypotheses about how to attack and bypass them.

A comprehensive guide[6] describes physical tamper mechanisms attacks to many (if not all) known attack types and provides lists of solutions to implement to protect against such attacks ranging from cheap and easy to extremely costly and complex.

Physical attacks are mainly classified as Invasive (micro-probing, fault injection) and Non-invasive (measuring side-channel signals like power consumption, timing etc, fault generation by changing supply voltage and clock signal). Physical attacks also include different machining methods like manual material removal, mechanical, water, laser, chemical usage for material removal.

To counter these physical attacks anti-tamper techniques are used which include physical barriers (secure enclosures, chip coatings, insulator based substrate); Tamper evidence solutions include doing things which cannot be revert back to original condition like use of brittle packages, crazed aluminium, polished packages, bleeding paint and holographic tape; Detect tampering includes making use of Anti-tamper sensors (voltage, probe, wire, printed circuit board, stressed glass, piezo-electric, motion, ultrasonic, microwave, infrared, acceleration, radiation, or temperature) and different methods to identify hardware Trojans; and respond to tampering by erasing memory containing secret data, RAM power drop, RAM overwrite or physical destruction.

3.4 Cryptographic Processors and Secure Access Modules

The term Cryptographic processors and secure access modules originated from the notion of a protected system than can execute sensitive function in a trusted manner. By making use of separate cryptographic processor and secure access modules in the devices we can make sure that application software or firmware will execute in trusted manner only. A secure boot and the root of trust are cornerstones of an electronic device's trustworthiness. A secure boot and the root of trust can be done if secure memory, cryptographic processors and secure access modules are there.

The topic of Cryptographic processors and secure access modules is extremely broad and cannot be covered in sufficient depth in this article. Programmable secure coprocessor performs several tasks together with response hardware tamper events, authentication, self-initialisation, randomness of keys, coprocessor, persistent storage, and third party programming interface APIs. Generally, for a secure application, cryptographic functions in our design should be moved out of application firmware and into a dedicated cryptographic device. The strength of cryptography relies on the secrecy of a key, so the secret key must be stored in hardware from which attacker cannot get rid of it. One must have complete understanding of the requirements and functionality of an encryption solution before its implementation into a system. Improper implementation of encryption system could make the product easy to break.

The [7] thesis provides in-depth knowledge of secure coprocessor design and implementation. Cryptographic devices available from the vendors include IBM 4758, Philips Semiconductors' VMS747 security processor and many more. Physical security is a central assumption on which secure devices are built, without a secure design, even the best cryptography system or the most secure kernel/service will fail.

4 Conclusion

IoT devices are going to become part of our life and unsecure IoT devices can help attackers to create huge disturbance to our day to day activities and public services. To protect the IP of an IoT device or to properly design embedded security systems we must have in-depth understanding of what we are trying to protect, why we are protecting it, and what type of attackers will likely target out product. Based on these factors, we can choose or implement required secure hardware designs features from a wide variety of secure hardware design features available to make a successful attack more difficult.

5 References

[1] P. Middleton, P. Kjeldsen, and J. Tully, (2013), "Forecast: The internet of things, worldwide,"

[2] Hsu, Chin-Lung; Lin, Judy Chuan-Chuan (2016). "An empirical

examination of consumer adoption of Internet of Things services: Network externalities and concern for information privacy perspectives"

[3] A. Huang, *Hacking the Xbox: An Introduc_on to Reverse Engineering*, No Starch Press, 2003

[4] Logic Analyser, http://www.tek.com/learning/logic-analyser-tutorial

[5] Atmel CryptoMemory Specification, https://store.comet.bg/download-file.php?id=9908

[6] Weingart's "Physical Security Devices for Computer Subsystems: A Survey of Aacks and Defenses"

[7] B.S. Yee, "Using Secure Coprocessors," Carnegie Mellon University, 1994