Chapter 20
Addressing Security Issues of the Internet of Things Using Physically Unclonable Functions

Ishfaq Sultan
University of Kashmir, India

Mohammad Tariq Banday
https://orcid.org/0000-0001-8504-5061
University of Kashmir, India

ABSTRACT

The spatial ubiquity and the huge number of employed nodes monitoring the surroundings, individuals, and devices makes security a key challenge in IoT. Serious security apprehensions are evolving in terms of data authenticity, integrity, and confidentiality. Consequently, IoT requires security to be assured down to the hardware level, as the authenticity and the integrity need to be guaranteed in terms of the hardware implementation of each IoT node. Physically unclonable functions recreate the keys only while the chip is being powered on, replacing the conventional key storage which requires storing information. Compared to extrinsic key storage, they are able to generate intrinsic keys and are far less susceptible against physical attacks. Physically unclonable functions have drawn considerable attention due to their ability to economically introduce hardware-level security into individual silicon dice. This chapter introduces the notion of physically unclonable functions, their scenarios for hardware security in IoT devices, and their interaction with traditional cryptography.

INTRODUCTION

The Internet of Things (IoT) exemplifies the interconnection of a vast number of ’Things’ (uniquely identifiable physical objects) through the Internet, with sensing, communication and actuation capabilities (Dragomir et al., 2016). Internet of Things (IoT) domain is an appealing target of numerous cyber-attacks because IoT devices generate, process, and exchange massive sums of privacy-sensitive

DOI: 10.4018/978-1-7998-5351-0.ch020

information, and security-critical data (Dorri et al., 2017). There are many constraints and restrictions in IoT devices in terms of power and computational resources, and the heterogeneous and ubiquitous nature of IoT initiate additional apprehensions concerning security establishment (Sain et al., 2017). IoT security needs to be part of the design at physical, network, and application levels. The IoT device itself needs to be designed using security principles. This covers the sensors that capture data, the data storage mechanism, and the micro-controller or actuator capable of controlling the device behavior, processing data and establishing a network connection (Wurm et al., 2016). Traditional security structures, such as public key cryptography, are not viable in IoT devices due to strict cost and power requirements. Physical and network attacks are common in the IoT domain due to backdoors created by a large number of IoT devices and the ensuing scale of IoT network. Software attacks, device cloning, eavesdropping, and data-stealing are also possible in IoT devices due to their always-connected feature (Mahalle & Railkar, 2015). The limited amount of energy accessibility of IoT devices can make them susceptible to resource enervation and denial of service attacks. Firmware and Software updates are inevitable due to the long life of IoT devices and hence, requires robust authentication procedures to evaluate the reliability and authenticity of any updates and patches, considering the tight power budget of IoT devices.

IoT needs security at the hardware level to ensure authenticity and integrity of hardware implementation of each node. Physically unclonable functions (PUFs) have been developed in the recent past as a potentially lightweight and secure solution for assuring security down to the hardware level. PUFs sometimes denoted as silicon biometrics (unique for each chip) are functions that map an *input digital challenge* with an *output digital response* repeatedly in an unpredictable manner, taking benefit from random process variations of the chip. In PUFs the key is naturally generated and embedded into the chip at the time of manufacturing, eliminating the need to store the key. PUFs are primarily utilized for device identification and authentication (Alvarez et al., 2015), lightweight encryption and secure key storage (Mathew et al., 2014), hardware entangled cryptography (Sadeghi & Naccache, 2010) and detection of malicious hardware (Maes, 2013). PUFs are very favorable primitives because of their randomness and unclonable feature, and hence, are extremely difficult to compute without the possession of PUF hardware. Although PUFs are established on measurements of the vast diversity of physical parameters, the ones obtained from measurements of integrated circuits are predominantly convenient because the output is easily integrated into computational operations. In traditional encryption schemes, data is usually encrypted using an externally stored key, or a key that is stored in an on-chip non-volatile memory for the transmission security. Storing the key in an on-chip non-volatile memory or off chip enables the retrieval of the key by intruders. Since non-volatile memory is easy to read and prone to attacks, PUFs can be used to replace the traditional key storage offering better robustness against intrusive attacks. This is possible because PUFs do not store data rather they restore the keys only when the chip is being powered on.

In literature, PUFs have been proposed for use in remote attestation (Schulz et al., 2011), protecting intellectual property (Alkabani et al., 2007), random number generators (Maiti et al., 2009) and authentication (Hamlet et al., 2014). The effectiveness of PUFs for the applications mentioned above is governed by the credibility of the randomness of the PUF output. Statistical tests can be employed to access random number generators. Although not all the tests are utilized to the comparatively short binary strings generated by PUFs, the applicable tests can be applied to boost conviction in the randomness of PUF responses. Determining the number of unique output variants per specific PUF design is a relevant query. Although it may not be achievable to determine the number of unique output variants of a PUF that is physically possible, we can approximate the quantity. We can compare the response from the