Computer Networks 183 (2020) 107593

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Survey paper

A survey on physical unclonable function (PUF)-based security solutions for Internet of Things

Alireza Shamsoshoara [a,*], Ashwija Korenda [a], Fatemeh Afghah [a], Sherali Zeadally [b]

[a] School of Informatics, Computing, and Cyber Systems, Northern Arizona University, Flagstaff, Arizona, United States of America
[b] College of Communication and Information, University of Kentucky, Lexington, Kentucky, United States of America

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Security
Hardware-based security
IoT
Physical unclonable functions (PUFs)
Memory-based PUFs
Key generation
Authentication

ABSTRACT

The vast areas of applications for IoTs in future smart cities, smart transportation systems, and so on represent a thriving surface for several security attacks with economic, environmental and societal impacts. This survey paper presents a review of the security challenges of emerging IoT networks and discusses some of the attacks and their countermeasures based on different domains in IoT networks. Most conventional solutions for IoT networks are adopted from communication networks while noting the particular characteristics of IoT networks such as the nodes quantity, heterogeneity, and the limited resources of the nodes, these conventional security methods are not adequate. One challenge towards utilizing common secret key-based cryptographic methods in large-scale IoTs is the problem of secret key generation, distribution, and storage and protecting these secret keys from physical attacks. Physically unclonable functions (PUFs) can be utilized as a possible hardware remedy for identification and authentication in IoTs. Since PUFs extract the unique hardware characteristics, they potentially offer an affordable and practical solution for secret key generation. However, several barriers limit the PUFs' applications for key generation purposes. We discuss the advantages of PUF-based key generation methods, and we present a survey of state-of-the-art techniques in this domain. We also present a proof-of-concept PUF-based solution for secret key generation using resistive random-access memories (ReRAM) embedded in IoTs.

1. Introduction

Wireless communication technologies have managed to imprint themselves into our daily lives through the Internet of Things (IoTs). IoT enables billions of "things" from tiny sensors to automobiles to interconnect with each other and share their data to create a wide range of value-added services. IoT systems currently impact different aspects of people's daily life. Various kinds of data including location information, medical information are constantly being collected by sensors and different electronic and tracking devices [1]. Some examples include using smart watches to set the credit cards with the cell phone's NFC interface to stimulate a transaction protocol and reduce the required time for shopping [2]; using IoT-based remote health monitoring systems to gather information from patients in a short time and inform the physicians to take timely actions [3]. Fig. 1 demonstrates some applications of IoT systems in our day-to-day life. Fig. 1 categorizes the IoT's applications into different branches such as smart home application, smart farming, security and privacy, and healthcare and wearable devices. While IoT networks have improved the quality of our life in many levels and opened up a path for several new services, due to the power and computing limitation, high mobility, and the dynamic nature of the network, the security threats and attacks can rapidly propagate throughout the entire network [4]. One key challenge in IoT networks is the lack of a unified security, identification and authentication standard, while new products and technologies come to market every day without paying enough attention to the potential security threats. Besides the security challenges [5], there are several other concerns regarding the large scale IoT networks [6] in terms of data fusion [7,8] and data management [9,10], complexity [11], spectrum scarcity [12–14], and so on. Moreover, several recent IoT technologies are still based on IPV4 for addressing which compromises the scalability of these networks.

According to Symantec [15], cyber-criminals progressively target IoT devices since they are developing and expanding rapidly. It is estimated that by the end of 2025, 75 billion IoT devices will be connected to the worldwide network [16]. In 2019, the attack traffic on IoT network increased by three-fold to 2.9 billion attacks which represents an increase of 300% from 2018. [17]

* Corresponding author.
E-mail address: Alireza_Shamsoshoara@nau.edu (A. Shamsoshoara).

https://doi.org/10.1016/j.comnet.2020.107593
Received 14 January 2020; Received in revised form 6 September 2020; Accepted 6 October 2020
Available online 21 October 2020

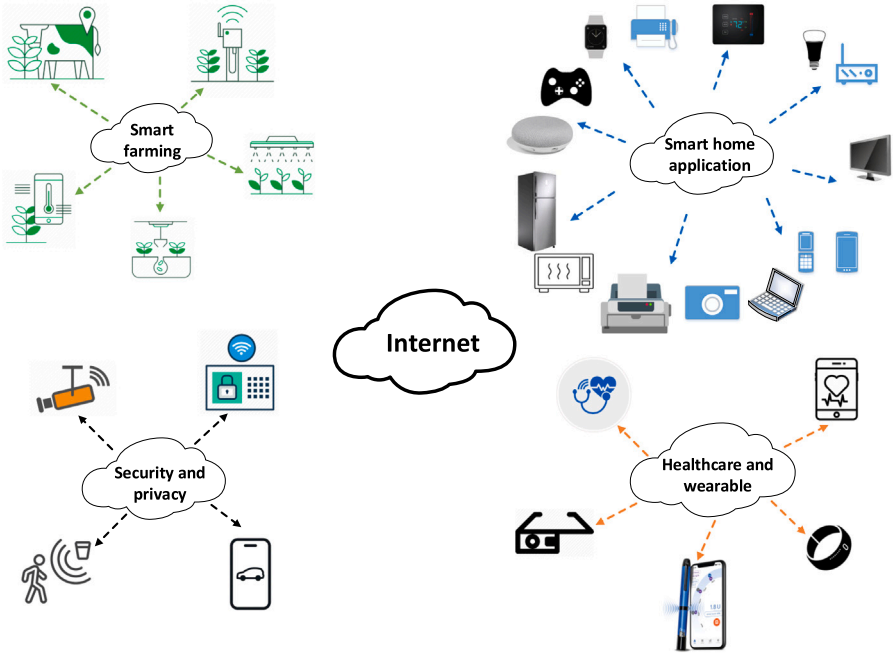


Fig. 1. Some examples of IoT applications.

Ransomware is one of the rapidly growing malwares types, in which the attacker uses Bitcoin or prepaid credit cards to demand money, where they do not need to decrypt the public or private keys for stealing cards information [18,19]. In Ransomware, the attacker prevent the victim from accessing his/her data unless ransom is paid by the victim [20]. For instance, in the earlier months of 2017, many individuals and businesses around the world were affected by two huge ransomware attacks, followed by a variant called "Wanacry" which affected 300,000 computers. Petya is another version of the Ransomware attack, which exploited the existence of a third party software to spread and grow itself in the networks [21–23]. The prevalence of this malware started from Ukraine, where 12,500 computers were affected. According to Microsoft, this malware has been outspread by using a third-party application [21]. The Dyn cyber attack used distributed denial of service (DDoS) which to target domain name systems (DNS) which are provided by the Dyn company [24]. In 2016, Dyn cyber attack used DDoS techniques on several Internet platforms and services in Europe and North America by exploiting security vulnerabilities in the IoT nodes [25]. These attacks have even penetrated into remote health monitoring systems. An article published in the Journal of the American College of Cardiology in February 2018, confirms that cardiac devices can be attacked by hackers leading to severe consequences or even death in some cases [26].

There are several factors contributing to the immense security vulnerability of IoTs including the limited energy available at IoT nodes, their low computational capability, the myriad of available "things" in a network as well as the heterogeneous nature of the network [27]. These characteristics, in particular, the number of connected devices, often result in inefficient performance of conventional security mechanisms. As stated above, it is anticipated that, about 75 billion IoT devices will be interconnected by the year 2025, which drastically increases the need for advanced security mechanisms for IoT. These "things" need to transmit the data they collected and intelligently respond and react to the received information. Therefore, it is crucial that information is received from and sent to an authenticated user.

The security challenges in IoTs can be broadly classified into identification, authentication, encryption, confidentially, jamming, cloning, hijacking, and privacy. Encryption has been widely used by several mechanisms in order to send their messages without the risk of being understood by the hackers. Cryptographic methods are a crucial element in securing IoT systems. In theory, encrypting messages does not allow the hackers to have access to the messages and eliminates the risk of data manipulation. However, encryption alone does not provide or guarantee integrity. For instance, an encrypted message can still be decrypted but the outcome is not fully clear. In addition, encryption by itself cannot avoid malicious third parties from transmitting encrypted packets in the network.

Several widely-used encryption mechanisms including public Key infrastructure (PKI), advanced encryption standard (AES), and elliptic curve cryptography (ECC) rely on secret keys [28,29]. In PKI-based systems, there are two sets of keys for each user, private and public keys. The private keys need to be kept secret while the public keys can be known by everyone. In these systems, one of the two keys is used for encrypting while the other one is used for decrypting. These secret (private) cryptographic keys are expected to not only be reliable, and robust but also perfectly reproducible. Hence, these keys are usually stored in the Non-volatile memory (NVM) of the devices such as ROM and one-time electronic fuse. However, due to the electrical nature of these memories they are highly susceptible to physical attacks. For instance, using a scanning electron microscope (SEM), attackers can implement many invasive threats on these chips. Moreover, using these kinds of memories requires additional fabrication steps during the production of the device. Antifuse or electronic fuse is another security technique which is being used for key storage using FinFET transistors [30]. The main benefit of this technology is that the information about the power consumption is not disclosed during the reading process. Furthermore, this technology enhances the reliability of the read procedure. The disadvantage is that it fails to remove the key from the device. Once the key is exposed, it cannot be eliminated from the chip.

The cryptographic keys are sensitive information and therefore, several mechanisms have been developed to protect these keys. White box cryptography (WBC) is a software based solution to protect these keys and allow secure distribution of valuable information [31]. WBC requires high processing power and memory and is only applicable to symmetric cryptographic methods; therefore, it will not be a competing candidate for the security of IoT networks. In addition, RSA encryption keys are stored in a specialized chip called trusted platform modules, introduced in [32], on an endpoint device (client device) to allow secured encryption. Physical computing devices called hardware security

modules (HSM) were designed to safeguard and manage digital keys. HSM require programming the equipment and interfaces to allow fast data transfer. Key management in an IoT network is even more difficult due to the increasing number of devices. The process of generation, distribution and storage of keys in large-scale IoT networks is still a major challenge of IoT security.

In general, there are two types of software-based, and hardware-based mechanisms to protect IoT devices from various attacks. Software-based security mechanisms rely solely on software to protect their messages. They are based on mathematical approaches (e.g., a discrete logarithmic problem) which may not be easily solvable using today's computers but when the existence of quantum computers becomes a reality, they can be solved in a shorter time compared to traditional methods in order to extract the keys [33]. Moreover, in software-based security mechanisms, the keys are stored in the NVM of the devices which are prone to attacks. Though software-based security systems were effective all these years, the advancement in hardware and computers may allow the hackers to break them using quantum computers [34–37]. As a lot of resources are put towards the creation of quantum computers, their existence will soon be a reality. Therefore, all the existing software security mechanisms are at high risk, which calls for additional security solutions. Hardware based security is one of the possible solutions to improve the current security mechanisms. Hardware-based security uses a dedicated hardware integrated circuit or processor to perform cryptographic functions and store the keys. They can prevent read-and-write access to data and offer a stronger protection against various attacks. The hardware-based mechanisms such as HSM have been used for crypto processing and strong authentication where it can encrypt, decrypt, store, and manage the digital keys. HSM have been used alongside with software mechanisms such as PKI, AES to encrypt their messages [38].

One of the main problems with the hardware-based security solutions is that they are prone to the Man-in-the-middle attacks. In these attacks, when the hardware security module is stolen, the attackers can clone the device. This can be compared to a simple physical lock and key, where the key is stolen and cloned to mimic the actual key. Physically unclonable functions (PUFs) can provide a solution to this mentioned problem. Physically unclonable functions were introduced by Gassend et al. in 2002 [39] as a security primitive based on hardware. PUF utilizes the intrinsic manufacturing variations in a device to generate a fingerprint of the hardware that offers the valuable advantage of unclonability. This property gives PUFs an edge over other hardware-based security schemes as the hacker cannot clone the intrinsic properties of the device even with physical access to it. Therefore, PUFs are unique to their device and can be used as a security primitive to enable device-based identification, and authentication. Furthermore, PUFs can provide a low cost alternative solution for on-demand generation of cryptographic keys from the device rather than the conventional methods, where the secret keys are produced and distributed by the server and stored in the IoT device memories [40].

The data derived from PUFs is often highly sensitive to environmental changes and the physical conditions where the device is being tested. In other words, the readings from the PUFs are not perfectly reproducible. Therefore, different types of PUFs have been used for the purpose of identification and authentication of devices, where a certain margin of error rate is tolerable. However, even a small amount of variation in the PUF's responses in different conditions can prevent them from being utilized in key generation because the key used for encryption needs to be perfectly reproducible to decrypt the messages. These PUF's responses act as unique fingerprints for the device which are not reproducible.

1.1. Review of recent relevant survey papers and the contributions of this paper

This survey focuses on the applicability of PUFs-based hardware security for generating keys and authenticating IoT devices. The paper offers a unique and timely survey compared to existing survey papers in the literature by investigating the role of PUFs in IoT security, in particular providing an additional level of security to common key-based cryptographic methods to generate the keys from the devices.

In 2019, the authors of [41] presented a review which discussed several characteristics of PUFs that contribute to their application in authentication followed by a comprehensive classification of different types of PUFs using three different classification systems based on their properties and applications, their parameters and also a chronological classification based on the time these PUF technologies were first introduced. The authors described relationships between PUF technologies that were not identified previously and investigated other novel forms of PUF which were not exploited. While the mentioned paper offers an inclusive classification of a wide range of different types of PUFs, it does not provide much insight on the challenges of PUFs when it comes to their applications in various security applications.

In [42], hardware based security techniques in a low-power system-on-chip (SoC) design was surveyed in order to investigate the hardware defenses suitable for it. The authors focus on mitigating the threats faced by the SoC-based embedded and mobile systems as they operate in uncontrolled low power environments.

In [43], the authors review the authentication protocols used in 19 different strong PUFs proposed between 2001 and 2014. The aforementioned survey reveals different security issues in these protocols and suggests more research is needed on the fundamental physics of a PUF in order to create a truly strong PUF, or else only conventional cryptographic Key generation methods are a promising alternative. The authors also recommend that some of the protocols might leverage the strong PUF to provide side channel attack resistance, but the same physical attack can still be launched on them by adding a machine learning block.

In [44], the authors state that the compatibility between IoT with limited resources and PUF is its main advantage over other cryptographic solutions proposed for IoT devices. The mentioned paper examines the challenges in utilizing PUF technology in IoT that must be addressed. The paper discusses different threats in using PUF with a focus on the man-in-the-middle and side-channel attacks (invasive, semi-invasive and non-invasive attacks) as well as the defense strategies against these attacks. The survey then describes the selection of PUF architectures for IoT based on their robustness to possible attacks, the uniqueness of Challenge Response Pairs (CRPs), and ease of implementation on FPGA. The paper also presents several ways to utilize PUFs to implement cryptographic schemes more efficiently by utilizing them for designing encryption keys, random numbers and electronic signatures.

In 2018, authors of [45] discuss the standards of wireless communication in cyber physical systems (CPS) and IoT and focus on security of these systems. The paper does not study the physical side channel attacks in implementation of security mechanisms for these systems. The paper explains in detail, the various wireless communication standards and protocols. Later, it briefly reviews the security threats in IoT and CPS domains and concludes the paper with recommendations on careful selection of devices from sensors to routers, and auditing the systems using dedicated third party surveillance technologies. The paper does not address the problems in identifying malicious nodes and authenticating known users in the network. In [46], several recent challenges are identified as a result of the introduction of IoT and CPS systems. The authors in [46] focus on different attacks and threats for these systems as well as the challenges related to implementation of CPS and IoT in the wireless network.

In [47], the security and privacy challenges of IoT in general and possible attacks in different layers of IoT devices were discussed. In particular, the mentioned paper discusses the security challenges of fog/edge computing-based IoT. In [48], the use of information-centric networking (ICN) as a possible protocol for addressing IoT devices in terms of in-network caching, content naming schemes, security schemes and mobility handling schemes was introduced. The authors of [47]

stress on the need for "larger and permanent naming scheme and addressing space for IoT contents and devices" [48]. In [49], the authors focused on securing communications between IoT devices using different protocols and mechanisms and the security weaknesses of IoT at different layers of communication were also discussed. In [50], the authors discussed the use of programmable hardware such as FPGAs in network infrastructure security. The author highlighted the role of hardware-based mechanisms to address some of the challenges in software-based methods, as well as the potential challenges due to the rising demands of intensive analysis and real time operation for sequential processing.

In another survey paper [51], the authors describe how remote attestation schemes can determine the level of integrity of a system and their application in IoT networks, cloud computing infrastructures, and content delivery networks. The authors surveyed hardware-based security devices and cryptographic primitives to achieve security integrity and efficiency and investigated PUFs as a possible solution for remote attestation.

Another recent survey paper [52] studies the application of quantum technology in cryptography and in particular physical security. The authors of the mentioned paper proposed a new type of PUFs called post-quantum PUFs using the fundamental of quantum phenomena.

The key contribution of this survey compared to the previously published surveys in IoT security is to study the role of memory-based PUFs in authentication and identification of the various IoT devices. More importantly, we discuss the potential advantages and challenges of using PUF-based secret key generation mechanisms to add another level of security to popular key-based cryptographic methods. Such mechanisms, if successful can enhance the security of a huge number of IoT devices against physical attacks. The current memory-based PUF technologies do not have the required robustness to generate fully reproducible responses for low-power IoT devices. This need calls for key generation schemes with error correction mechanisms to generate robust secret keys as required in cryptographic systems as discussed in this paper.

This survey paper is organized as follows. In Section 2, we discuss various security challenges in different domains of IoT networks, with a focus on TCP/IP Stack protocol. Moreover, we also briefly describe different attacks in an IoT network. Next, we explain the chain of integrated circuit manufacturing and point out hardware attacks based on the vulnerable points. In Section 3, the concept of PUF, their classification along with their application in different security applications are discussed. After explaining different IoT attacks, Section 4 investigates the role of PUFs in preventing hardware attacks. In Section 5, the role of fuzzy extractors in PUF to generate keys is described. In Section 6, we provide a survey on the state-of-the-art key generation mechanisms. Section 7 investigates different types of attacks on fuzzy extractors. In Section 8, the concluding remarks and future directions of research are discussed.

1.2. List of acronyms

All of the abbreviations used throughout this paper are summarized in Table 1.

2. Security challenges and attacks in IoTs

In this section, we classify the possible attacks on IoT networks from different perspectives and discuss the potential ways the PUF-based security solutions can contribute to mitigating such attacks. Section 2.1 studies the IoT security challenges in different domains such as data, communication, architecture, and application. Then, in Section 2.2, several traditional attacks including denial of service, Sybil are mentioned to familiarize the reader with different attacks and security challenges based on the TCP/IP stack layer. Next, considering the different layers of the IoT's architecture, Section 2.3 describes a taxonomy of attacks with respect to the IoT structure. Finally, Sections 2.4 and 2.5 focus on hardware-based attacks and hardware-based assisted security respectively. The goal of these last two sections is to familiarize the readers with a wider range of hardware-based attacks and possible hardware-based solutions in addition to the PUF-based security solutions which is the main focus of this survey.

2.1. Domain taxonomy to consider the security

In this paper, we address the security issues of IoTs and the potential impact of PUF technology to address some of these challenges. In IoT networks, the Internet connects enormous sensors and machines which form a huge network with mobility and heterogeneity characteristics that makes it difficult to protect the network against security attacks. Moreover, the limited energy and computation capability of IoT nodes restrict the utilization of some conventional security mechanisms in these networks [53,54]. One key challenge related to utilizing some traditional security protocols is the heterogeneous nature of IoT networks due to the wide variety of applications of these networks. As a result, different applications require different protection mechanisms. In most scenarios, these applications can be categorized by different characteristics such as user association, openness, and heterogeneity. The heterogeneity of IoTs can degrade the efficiency of cryptographic methods that rely on the key generation and key exchange [55]. Furthermore, noting the characteristics of IoT networks, the security threats such as DoS, routing attack, man-in-the-middle, side channel attack, replay attack, node capture, and mass node authentication are common attacks in these networks.

Fig. 2 shows the various security concepts for IoT based on four domains namely, application, architecture, communication, and data. Based on these stack layers, a security taxonomy can be defined in IoT. We developed Fig. 2 based on the discussions in [56–58] and the literature. Next, different concepts of security for each of the application, data, communication, and architecture domains are introduced.

2.1.1. Data

Data privacy and confidentiality are important aspects of security in different networks [57]. In general, *confidentiality* is a security concept which ensures that unauthorized users cannot access the data or try to hijack the information. Preserving the confidentiality of data is even more challenging in IoTs because of the large number of users and the diversity of network protocols and applications in these networks. Secure key management is one of the methods which can improve the confidentiality in IoT networks [47,59]. On the other hand, *data privacy* refers to the required regulations related to the collection, storage and sharing of data in such a way to protect the users' personal information (e.g., users' identity) from third parties. Most of the time, the main focus of confidentiality is on the encryption of the data; however, privacy defines the level of access to the received data for different users [47,60]. Finally, *trust* is a concept for the user to accept the security, privacy, and confidentiality in each network. Trust imposes privacy, confidentiality, and security among different layers of IoT, or between different users, devices and applications [47,61,62].

2.1.2. Communication

Communication in IoT networks is defined based on exchanging or sharing information between the users, devices or even exchanging information between different IoT layers. Noting the wide applications of IoT devices in different domains, several communication protocols have been used in IoT networks making these networks vulnerable to various communication attacks [58]. As a result, the communication medium is a bottleneck for different attacks such as eavesdropping [63] and Man-in-the-Middle (MitM) [64]. Many PUF solutions are available to handle security issues in the communication domain. For instance, in [65], the authors utilized PUF with an authentication key exchange and a broadcast authentication technique to develop a secure 2-way communication between smart grid meters and the utility infrastructure.

Table 1
Abbreviation table.

Acronyms	Paraphrase	Acronyms	Paraphrase
AES	Advanced Encryption Standard	CPS	Cyber Physical Systems
CRP	Challenge Response Pair	DoS	Denial of Service
DDoS	Distributed DoS	DES	Data Encryption System
DHT	Distributed Hash Tag	ECC	Elliptic Curve Cryptography
FAR	False Authentication Rate	FE	Fuzzy Extractor
FPGA	Field Programmable Gate Array	FRR	False Rejection Rate
HSM	Hardware Security Modules	IC	Integrated Circuit
ICN	Information Centric Network	IoT	Internet of Things
IP	Intellectual Property	LDPC	Low Density parity Check
MRAM	Magnetoresistive Random Access Memory	MQTT	Message Queuing Telemetry Transport
NVM	Non-Volatile Memory	OSI	Open System Interconnection
OSN	Online Social Network	PKI	Public Key Infrastructure
PMKG	Pattern Matching Key Generators	PUF	Physically Unclonable Function
Re-RAM	Resistive Random Access Memory	RF	Radio Frequency
RFID	Radio Frequency Identification	RO	Ring Oscillator
RSA	Rivest Shamir Aldeman	SDA	Software Defined Network
SEA	Secure and Efficient Architecture	SEM	Scanning Electron Microscope
SMB	Server Message Block	SoC	System on Chip
SOA	Secure-Oriented Architecture	SRAM	Static Random Access Memory
SS	Secure Sketch	SSL	Secure Socket Layer
TCP	Transmission Control Protocol	TSL	Transport Layer Security
ULP	Ultra Low Power	UUID	Universally Unique IDentifier
WBC	White Box Cryptography		

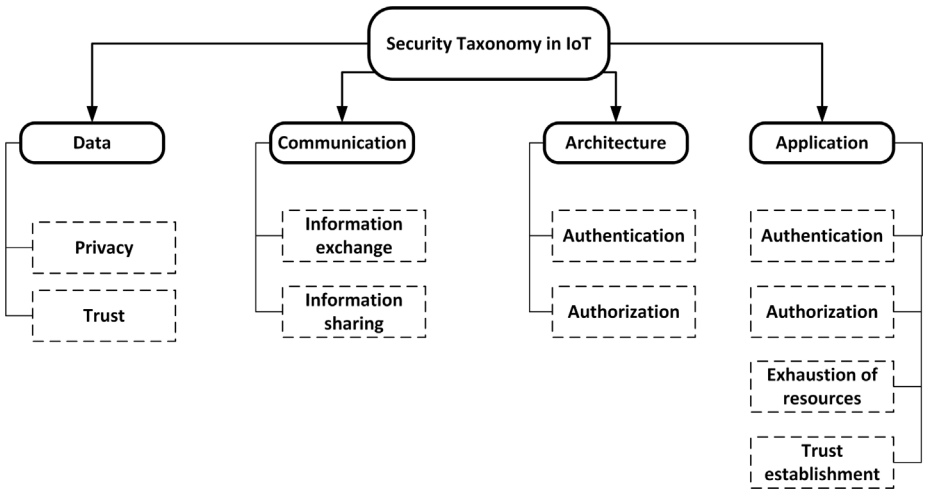


Fig. 2. Domains and security concepts in IoT.

2.1.3. Architecture

There are no global and specific architectures for IoT networks to validate the security concepts for authorization and authentication. However, various architectures such as software-defined network (SDN) [66], secure and efficient architecture (SEA) [67], smart city [68], service-oriented architecture (SOA), object security architecture (OSCAR) [69], and black SDN [70] are proposed to examine both authentication and authorization.

2.1.4. Application

Scope, scale, heterogeneity, accessibility, and repeatability are among the application features that can be used to evaluate different security techniques. Trust establishment, exhaustion, authorization, and authentication are considered as different security metrics [71,72]. Since there is no definite architecture for IoT devices, various techniques have been developed for authentication and authorization in this domain [73]. Noting the wide range of applications of IoT, attacks on these systems can impact several critical domains.

2.2. Attacks on IoT devices

The specific characteristics of IoT devices such as low price, low power, and low computational capability as well as the heterogeneity and large-scale of the network limit the applications of common security mechanisms. Therefore, IoTs are prone to several advanced attacks and security issues [74–76] that call for novel security mechanisms in different domains, including identification/authentication, reliability, confidentiality, and non-renunciation. In this section, we review some common attacks on IoTs such as spoofing, altering, replay routing attack, DoS, node capture attack, and Sybil attack.

2.2.1. Denial of service attack (DoS)

In this attack, the attacker attempts to exploit all the reserves and resources in the network which can seriously degrade the network performance. The DoS attack is also called a computational resource attack. These attacks are categorized into two groups: Distributed DoS (DDoS) and individual (single) DoS [77,78]. In a single DoS attack, the intruder as a single entity tries to exhaust the resources of the target entity. However, in a DDoS attack, multiple attackers exploit the single entity or a single attacker compromises multiple users to flood the target machine with lots of requests.

2.2.2. Sybil attack

Networks with a large number of users are more susceptible to Sybil attacks. In this attack, a single node is identified with different IDs.

This means that the unification of entities will be eliminated from the network [79]. Based on [80], in 2012, 20 million users on Twitter and 76 million users on Facebook were fake. Online social networks (OSNs) such as Facebook, Instagram, or Twitter are prone to this kind of attack as they have lots of users. One of the purposes of the Sybil attack is to hijack the information from the OSNs and websites. Since the quantity of IoT sensors and applications is increasing rapidly, they are also vulnerable to Sybil attacks. A Sybil attack can cause users to produce fake and false reports. Users might also receive spam messages from fake profiles and fail to keep their privacy. Different mechanisms including feature-based mobile Sybil detection, cryptography-based mobile Sybil detection, and friend relationship-based sybil detection (FRSD) are being used to defend IoT networks against Sybil attacks [80].

2.2.3. Spoofed, alter, or replay routing information

In these types of attacks, an attacker changes the routing information or tried to manipulate the routing packets by listening to the legitimate transmitter and impersonating the identity of the real transmitter. Then, it sends fake data to the receiver and introduces loops into the network [79,81].

2.2.4. Attacks based on access-level

Based on the level of access to the network, these types of attacks are categorized into two different branches namely, passive and active attacks.

Passive Attacks: In most passive attacks, the attacker just eavesdrops the communication between the legitimate transmitter and its receiver to exploit their data. [78,82,83].

Active Attacks: In active attacks, the intruder attempts to disturb the connection between the legitimate entities, perform impersonation itself, or even disrupt the connection by manipulating the routing information [78,82,84,85].

2.2.5. Attacks in communication protocols

The communication functions of IoT networks are commonly described by the TCP/IP model. Table 2 presents the taxonomy of attacks that are possible on the TCP/IP protocol stack.

2.2.6. Attacks based on device property

IoT devices are categorized into two groups: high-end and low-end device classes. According to these types, attacks might have different effects on the devices. They might just result in abnormal behavior or they might stop the devices from working [82].

High-end device class attacks: In this class of attacks, powerful devices such as laptops and computers are used to launch attacks on the IoT network. Most of the time, the Internet protocol is used between the attacker and the IoT network. In these types of attacks, the intruders can use the computing power of CPUs and even GPUs to launch attacks on the IoT network [125,126].

Low-end device class attacks: In contrast to the previous class of attacks, in this class, the devices which have low power and energy are engaged in attacks on IoT devices. The attacker uses the radio connection between itself and the IoT device to perform the attack. As an example, smart watches or smart home gadgets are very common in every home. These tiny devices connect to your smart home network which includes TV, refrigerator, cooling system, home security and they can control the configurations of these features. However, these smart home utilities could also be attacked by these little IoT devices [127,128].

Table 2
Taxonomy of attacks based on different layers of the TCP/IP reference model.

Layer	Attacks	Attackers' strategies
Physical	Jamming [86–88]	With radio interference
	Tampering [89,90]	Making fake nodes
Data link	Collision [91–93]	Transmit data simultaneously in the same frequency channel
	Exhaustion [82,94,95]	Multiple collisions and continuous re-transmission until the node runs out of resource
	Unfairness [96,97]	Repeatedly ask for the channel to limit others' request
Network	Spoofed, or Replayed routing information [82,98,99]	Routing loops, changing the source of the route, Repelling network from selected nodes
	Selective forwarding [100–102]	Send selected information to the legitimate receiver
	SinkHole [103–105]	Become the target of all nodes in order to gather all information
	Sybil [63,80,106]	Create lots of pseudonymous identities to undermine the authorized system
	Acknowledgment spoofing [107]	Spoof the link layer acknowledgment
	Hello flood [108–110]	Exploit Hello messages to flood the network with these tiny messages
	WormHoles [111,112]	Re-transmit data to the IoT nodes
Transport	SYN flooding [113–115]	Resend request multiple times to fill the capacity of the transport layer
	De-synchronization, [116–118]	Reinitialize the connection in order to disrupt it
Application	Reliability attacks: Data aggregation distortion, Selective message Forwarding, Clock skewing [102,119–124]	Impersonate itself as a reliable node in the IoT network and sends corrupted data

2.2.7. Attacks based on transmitting data

Sensing and collecting information from the surrounding environment are the main goals for most IoT networks. For this reason, thousands of sensors are being used to gather information. These sensors are also prone to different sorts of attacks which can be used to launch network attacks which can be categorized into six groups such as (1) man-in-the-middle attack, (2) message replay attack, (3) fabrication attack, (4) alteration attack, (5) eavesdropping attack, and (6) interruption attack. PUFs can provide lots of solutions for these kind of attacks. For instance, a controlled PUF(C-PUF) can be introduced to handle man-in-the-middle attack. C-PUF is a specific type of PUF that can only be accessed using a specific algorithm which is physically linked to PUF [129]. The algorithm uses a collision resistant hash function. Thus the only way for the man-in-the-middle attack to be successful is to use the user's program. Therefore, if the user's program has a security leak then the attack can still be successful but not through the PUF itself. Section 4 describes the algorithm further. Utilizing this controlling algorithm prevents the man-in-the-middle attack [129,130].

2.2.8. Host-based attacks

In this attack, the intruder targets the host resources such as the operating system (OS) or the hardware. The assumption in this attack is that the intruder has managed to access to the host. Host-based attacks are categorized into three groups: hardware-, software- and user-based attacks. The IoT nodes are usually tiny devices with some applications or software embedded in the OS. The attackers target

these three resources of IoT devices and compromise each group with different impact on the overall network [131]. In Section 2.4, we focus more on hardware-based attacks and possible hardware-based assisted security solutions.

2.3. Classification of IoT security attacks on different layers of IoT networks

This section categorizes common attacks based on the IoT ecosystem. Although there is no well-defined layered model for IoT, Fig. 3 illustrates a three-layered model for IoT devices [132–134]. These layers include perception, network, and application. First, we describe each layer and then we present a classification of attacks with respect to the different layers in Table 3.

2.3.1. Perception layer

The perception layer is the lower layer of the IoT networks which handles the interconnection of the nodes in the network. For instance, Arduino boards can use the Ethernet to get access to the Internet, Raspberry Pi can use the Ethernet, WiFi module, or the Bluetooth module to connect to the Internet or other nodes. Each of the communicating devices should have a unique identification called the Universally Unique IDentification (UUID) [138]. Most of the time, these IDs are interchangeable. Hence, these UUIDs as System-on-Chip(SoC)s are embedded in the hardware or provided by a secondary chip [139].

2.3.2. Network layer

Addressing, network administration, communication channels, and interfaces are the main parts of the network layer. This layer is also responsible for all communications and connectivity for all devices in the network using multiple communication protocols [133]. Unlike the Internet, no well-established or standard protocol exists for the network layer in IoT devices. However, Constrained Application Protocol (CoAP) [140] and Message Queuing Telemetry Transport (MQTT) 3.1 [141] are two common protocols for the IoT networks. This layer transmits information within the network (other nodes) or outside of the network (e.g., the Internet or a sensor network). Since devices in an IoT network have a limited amount of energy and computation, the role of addressing, forwarding, and routing is pivotal in such networks.

2.3.3. Application layer

This layer makes sure that different entities in the network communicate using the same type of service. This layer is also known as the service-oriented layer [142] which handles data for different applications based on user requirements and demands. For instance, for applications such as smart transportation, smart home, and eHealth, it can store data into an appropriate database [142].

In Table 3, we provide an overview of some of the main attacks on different layers of the IoT networks. Since the majority of IoT devices have limited on-board power and computation capabilities, existing encryption methods cannot be performed at the device level.

Memory-based PUFs have gained high importance in recent years because they are available as embedded memories in every IoT Device as cache or storage, and unlike other PUF technologies, they require minimal or no additional hardware [143–146]. Moreover, several memory-based technologies such as memristors can offer a short process time and a low power supply to generate the PUF responses which makes them a good security primitive for IoT. The PUFs require a density of 128–256 bits for key generation, which is very small compared to the memory needed in IoT. This will allow us to increase the devices' security as only a small percentage of the entire memory cells will be used for Key generation, therefore identifying those cells will be a challenging task for the hacker. Moreover, by utilizing advanced protocols, which will allow us to change the memory cells we use to extract the PUF response, we can extract many different keys which will increase the security because every time a different key is utilized to authenticate a PUF. Therefore, memory-based PUFs can offer a unique solution for identification, authentication and even extracting the private cryptographic keys from the embedded memory in these devices without introducing additional fabrication costs to the device.

2.4. Hardware-based attacks

In this section, we focus on hardware-based attacks in IoT networks to lay the groundwork to discuss the role of PUFs in securing the IoT devices. Based on the production line for integrated circuits (ICs), there are several vulnerabilities based on hardware designs. Fig. 4 shows this semiconductor manufacturing process chain which consists of different tasks.

As a result of the fast and growing tendency in IC industry and production, the global supply line can be targeted and attacked at different vulnerable points. Some common threats faced by the manufacturing process include fake copy, side-channel attack, reverse engineering, intellectual property (IP) hijacking, and hardware Trojans. Next, we review hardware-based attacks and threats discussed in [147,148].

2.4.1. Fake replica

In this attack, the intruder counterfeits the original IP illegally. Fake replica and piracy are totally different. Piracy means overbuilding the entire IC. This might happen because the attacker gets access to the design information at different points such as the design or the fabrication. However, a fake replica might happen at different stages such as recycling, packaging, or the new vendor [149]. Fake replica or counterfeiting can be very harmful to the industry. Since the attacker uses the reputation of the original designer, instead he/she uses expired or old designs to rebuild the ICs or IPs. In most cases, the attacker's intention is to make profit by selling fake products. However, he/she can also put malicious circuits such as Trojans into those ICs and compromise different critical products and applications such as airplanes, vehicles, drones and UAVs, elevators, and so on.

2.4.2. Side-channel attack

In some cases, physical states' parameters such as power consumption, timing values, or electromagnetic reflection from hardware can reveal important information to the intruder. In most cases, such information sets can be extracted when the application is being executed where the attacker can perform different tests. Such attacks which involve extracting the behavior of devices [150], are very common in public-crypto systems such as Rivest–Shamir–Adleman (RSA). RSA uses public and private keys which encrypts and decrypts messages based on modular operations and large exponential values. Two common approaches include calculating the multiplication chain: the first one uses the naive multiplication operation and the second one uses square-and-multiply method [151,152]. In both scenarios, the attacker can use delay analysis to perform a timing side-channel attack. Delay analysis measures the execution time for a number of multiplications which the system uses to calculate the exponential results [153]. Using the execution time and by exploiting the information regarding the implementation method, the attacker can extract secret information such as the secret private key. For instance, in RSA, the modular exponentiation utilizes the square-and-multiply algorithm to perform multiplication; however, using the statistical analysis and timing analysis with this attack, it is possible to recover the secret key. In addition to timing side-channel attacks, other attacks such as measuring the photonic emissions, systemic acoustic noise, power consumption, and electromagnetic emissions are common in crypto systems [154–156]. One possible solution to mitigate this kind of attack is to use a key-based PUF to extract the key from the device. In [157], the authors proposed a public key exchange method using a PUF which is hard to break by physical and side-channel attacks. Using simulation, they showed that if an attacker uses all available computational resources, then it takes 500 years to break this protocol.

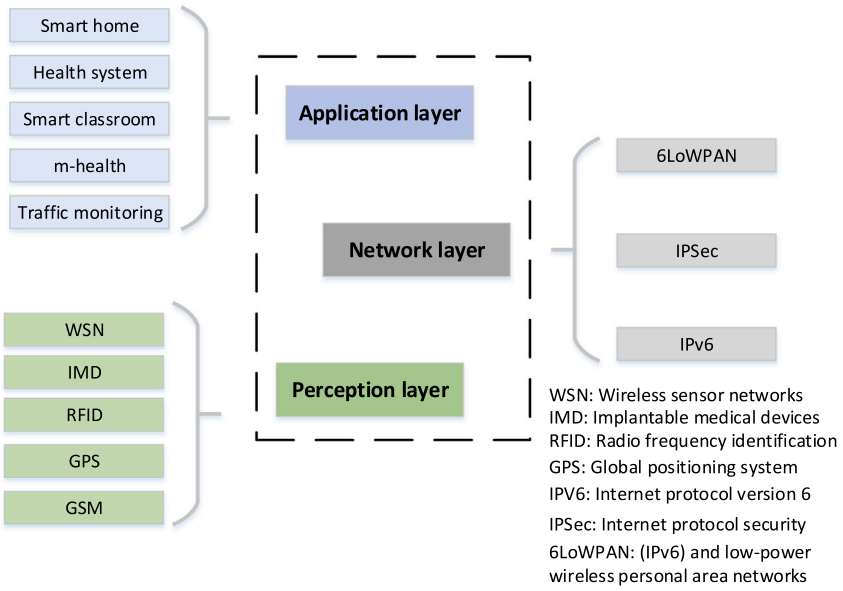


Fig. 3. IoT structure.

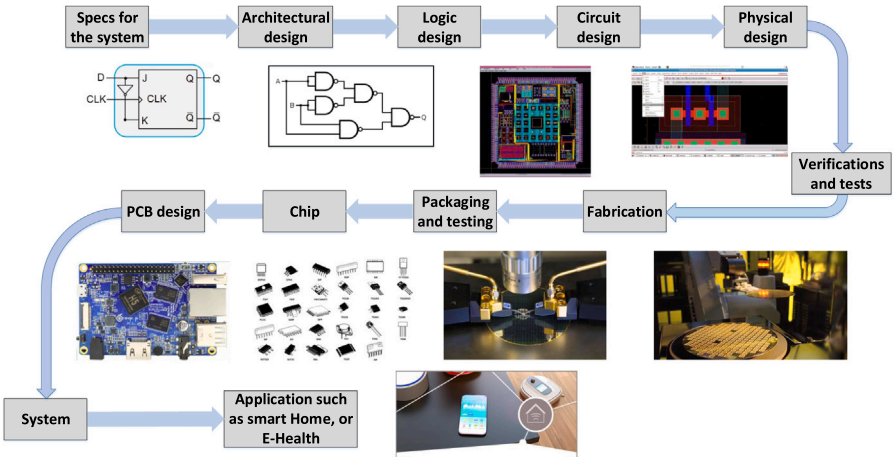


Fig. 4. Semiconductor manufacturing process chain for ICs from design home to the application.

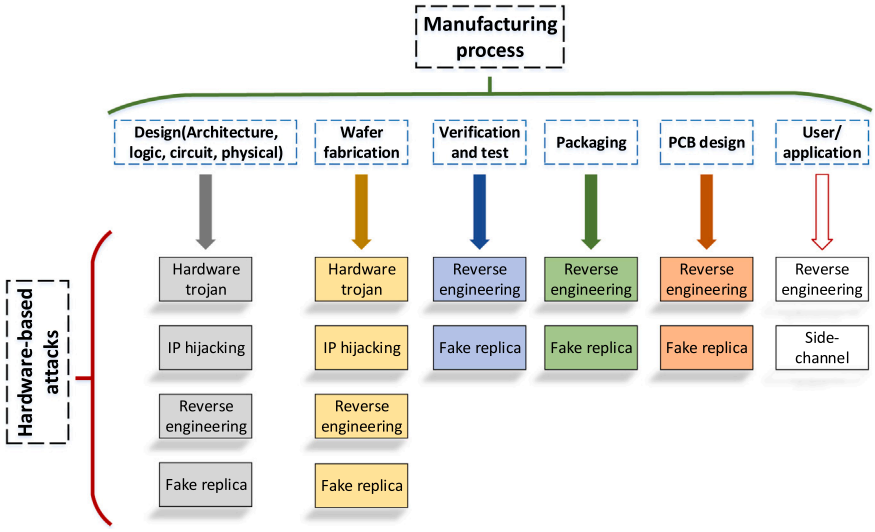


Fig. 5. Hardware-based attacks based on different entities in the semiconductor manufacturing process.

Table 3
Taxonomy of attacks with respect to the different layers in the IoT structure.

Encryption attack [61]	Perception attacks [135]	Network attacks [136]	Application attacks [137]
Side channel attack	Node tampering RF interference	Sybil attack Route information attack	Virus and worms Spyware and adware
Man-in-the-middle attack	Node jamming	Sinkhole attack	Trojan horse
	Malicious node injection Physical damage	RFID spoofing RFID cloning	Denial of service
Crypto attacks	Social engineering Sleep deprivation attack	Man in the middle attack Denial of service	Malevolent script

2.4.3. Reverse Engineering (RE)

Reverse engineering is the process wherein the intruder follows a reverse path from the application to the design point for the IC or the IP to reconstruct it, modify it, or implant malicious circuit into it. RE may involve different steps such as (i) detecting the technology model which is being used in the design and fabrication steps [158], (ii) taking out different parts of design such as gate, logic, circuit, and physical [159], and (iii) discovering and observing the functionality of the IP or the IC [147,160]. RE might have different objectives such as hijacking the design, illegally replicating the IC and announcing the technology used in the design. The intruder might use a table of information based on a defined pair of inputs and outputs to evaluate the behavior of the circuit. In this way, the attacker can verify the gate level design from the IP/IC. The attacker's incentive might be hijacking at the gate-level, circuit-level, or physical design by performing reverse engineering in order to extract an abstract level of the IP. The attacker can use the abstract level to reproduce the product and to sell it illegally or implant a malicious circuit into the product. One solution for mitigating this attack is to use PUF. In [161], the authors proposed new approaches using PUFs to obfuscate the hardware. The authors hide the circuit functionality using two methods: (i) Hiding the signal path and (ii) Replacing a logic using PUFs. They showed that these techniques are resilient to reverse engineering attacks.

2.4.4. Intellectual Property (IP) hijacking

When the IC is designed, the designers of the IP company or people involved in the fabrication process might hijack the design information without respecting the copyright terms. Moreover, an attacker at the fabrication stage may reproduce additional chips to sell them on the black market. In these cases, unreliable people can steal the design information and assert a right to possess the proprietary of the IP or the IC [162]. One possible solution to protect against the IP hijacking is to use PUFs. For instance, in [163], the authors used the variation of delays in specific arrays of gates in an FPGA to employ a unique signature for IP protection and anti-hijacking.

2.4.5. Trojans in hardware

Malicious modifications to an IC can be defined as a hardware Trojan. This Trojan can mislead the communication or cause a failure in control and processing units. In this kind of attack, the intruder can modify and alter the circuit or add a malicious circuit to it. Since the testing procedures are usually slow and expensive, it is difficult to identify the hardware Trojans after wafer fabrications. Moreover, the technology is merging with Nano- and Pico-meter fabrication design and because of the large space inside of ICs, there are many locations for implanting Trojans. Such locations include different design points such as logic, circuit, and physical and the fabrication process [164–166]. In [167], the authors proposed novel hardware protection techniques using PUF to prevent the use of hardware Trojan and unauthorized overproduction. The authors minimized the *rare values* in the IC to make it difficult for an attacker to use these values with hardware Trojans.

After introducing hardware-based attacks, Fig. 5 summarizes these attacks and indicates which entities are vulnerable to specific attacks in the semiconductor manufacturing chain.

These attacks mentioned above are not unique only to IoT devices and they can affect the whole process of IC fabrication. However, considering the fast growth production rate of IoT devices during recent years, the lack of standards in this domain, and the high demand for low cost devices, these hardware attacks are more likely to be implemented on IoT devices compared to old-fashioned traditional devices. Another reason is that IoT has the largest application for digital device marketing. Hence compared to other applications, it is more likely to have devastating outcomes on IoT devices because of their mass production. In [168], the authors present the pyramid of attacks depicted in Fig. 6 which is based on [169,170]. The peak of the pyramid is the most vulnerable element of the IoT stack with the least impact which are the sensors in IoT networks. The bottom of the pyramid is the hardware platform consisting of the systems on chips, microcontrollers, and Field-programmable gate arrays, which has the most impact on the system in case of any attacks on the IoT systems. In this section, we focus on the manufacturing chain process because it has the most impact in case of attacks.

2.5. Hardware-based assisted security

In previous sections, an overview of possible attacks on IoT networks was provided. Such attacks can be generally categorized into two classes. (i) The attacker does not have physical access to the IoT device, hence, the attacker exploits software or network connections to gain access to the IoT device remotely. In this case, the attacker can draw out the cryptographic keys and disturb the authentication mechanism. (ii) In the second case, the attacker has physical access to the IoT device or the chip. For instance, the intruder can perform fake replica, reverse engineering or the IP hijacking [171]. Hence, the existence of an environment is necessary to avoid these kind of adversaries. In the following, two types of hardware-based security methods are proposed. These methods work based on environment splitting which means dividing the hardware and environment into two sections: (i) the secured area, and (ii) the unsecured area. In the first approach, which is "ARM TrustZone", a new state is defined in the processor to bring a meaningful separation. In the latter one, a specific hardware "Security Controller" such as a microcontroller takes the responsibility to define the reliable environment [172].

2.5.1. ARM TrustZone

This approach is a system-wide method to utilize the security option at the low level for the microcontrollers with the cortex-based cores. Cortex-based cores are a specific family of ARM microcontrollers. This technology initiates at the hardware level on a single core which divides the processor into two secured and unsecured areas. Since attackers can target the boot up procedure for microcontrollers, this method also secures the boot up process. Core families such as ARM Cortex-A and Cortex-M series support the TrustZone feature [173,174]. The new secure state in the processor splits all partitions in the CPU [172]. Using this method, all signals and interrupts of the secured area are isolated from the unsecured one. Fig. 7 shows the schematic for this technique.

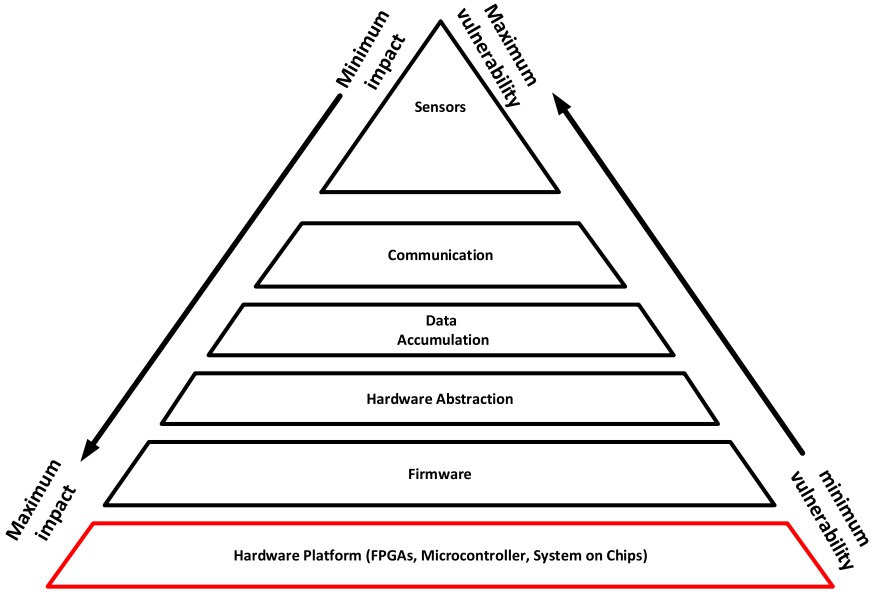


Fig. 6. The attack pyramid for IoT devices based on vulnerability and impacts. *Source:* Adopted from [168].

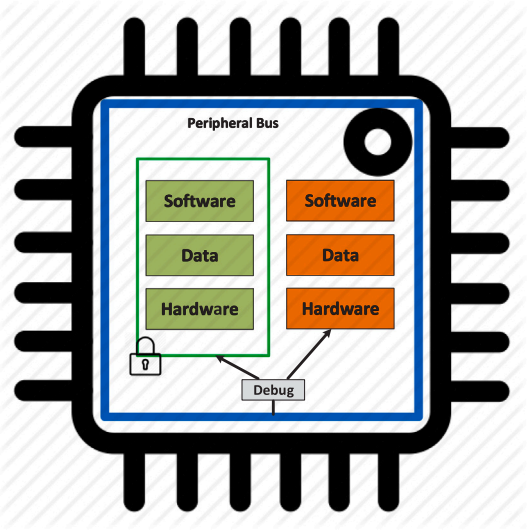


Fig. 7. TrustZone concept in ARM based microcontroller. The concept is depicted based on the explanation in [175].

2.5.2. Security controller

In the past, several organizations have utilized individual cryptoprocessors which are also known as hardware security modules. Moreover, ATM technologies also utilize those cryptoprocessors as a concept of security in smart cards. In this approach, the security controller or the secure microcontroller is an individual IC in the IoT device which brings a group of predefined cryptographic tasks. The security controller safeguards the confidentiality and the authenticity of the cryptosystems [171].

PUF is a principle which is being used for authentication and authorization that does not call for any non-volatile memory [39,176]. They can be also used for cryptographic key generation, where the digital key is not saved in the device, rather, it is extracted from the physical features of the device. PUFs exploit the random disorder of the physical system in the environment or from the manufacturer. These disorders cannot be re-fabricated again. Hence, they are called intrinsic behavior of the device. Although these features are disadvantageous from the perspective of integrated circuits, they can be advantageous from a security perspective. In recent years, PUFs face many critical challenges such as reproducibility, wireless transmission of data, and exposure to the predictive models for attacks.

In general, cryptographic methods are currently the most reliable way to secure IoT devices in a vulnerable environment. However, power utilization and key storage are amongst the main concerns when implementing these cryptographic methods in IoT networks.

In the following, Section 3 introduces the concept of PUFs and their roles to prevent the aforementioned attacks in this section.

3. Physically Unclonable Functions (PUFs)

3.1. Introduction to PUFs

PUFs use the unique variations introduced in the fabrication of the device, to extract a fingerprint unique to the device. One or more specific parameters of the device such as threshold voltage, critical dimensions etc., are measured when an external stimulus is applied. When the devices' parameter is being measured for the first time, the measurement is called an "original response" for a specific input stimulus or a specific address in the memory-called as a "challenge" used to obtain this measurement and they are both stored in the server. When the same parameter is measured again, and the same external stimulus is applied it is called a response. These challenges and responses form a pair, called the *Challenge Response Pair (CRP)* and are generally compared with each other to validate the identity of the device. The error between the challenge and response of a PUF during the registration and authentication phases is referred to as the *Challenge Response Pair error (CRP error)*.

Subject to the number of possible CRPs a PUF has, they can be broadly classified into: "strong PUFs" and "weak PUFs". Weak PUFs leverage the manufacturing variability and allow digitization of some "fingerprint" of the hardware device. The number of responses in a weak PUF is a function of the number of components in the device used for generation of CRPs [177]. This fact results in a small number of CRPs with stable responses which are usually robust to environmental conditions. Due to high stability and reproducibility of weak PUF responses, they are generally used for secret key generation. "Strong PUFs" have a large number of CRPs in a device. Ideally, if the number of unique CRPs is high, even though an attacker gets temporary accesses

to the system, he/she will not be able to apply all the responses (brute force attack) and get access to the system. Hence, strong PUFs are generally used for authentication [177]. However, a large set of PUF responses may offer stronger cryptographic strength as it leads to longer cryptographic keys [178]. Independent CRP refers to the fact that if one CRP is known, one cannot predict the other CRPs in the PUF, hence there is no shared information between two CRPs. PUFs can also be classified based on how their unique-randomness was obtained. If the PUFs had their variation obtained by externally applying additional steps as in the case of coating PUFs they are called *explicit PUFs*. If the randomness was natural through variations in the manufacturing process they are called *implicit PUFs* [41].

Usability of PUF can be determined by two statistical parameters of *intra-distance* and *inter-distance* which are defined as follows in [179]:

- *"Intra-distance: the Hamming or the fractional Hamming distance between two different responses to the same PUF challenge"* [179]
- *"Inter-distance: the Hamming or the fractional hamming distance between two responses of two different PUFs to a given challenge"* [179].

These measurements indicate the PUFs reproducibility and uniqueness, respectively.

PUF-based security mechanisms depend on the unique CRP's produced from a device [180]. Every PUF device initially needs to be registered with the server in order to use it with any cryptographic method. During the registration phase, the server uses a stimulus to challenge the client's PUF and as a result a corresponding original response will be produced. This challenge and response pair is stored in the server's memory. During the authentication process, the server uses the same challenge for the client's PUF to extract the corresponding response. These responses depend on the manufacturing behavior and variations in PUF. The user is authenticated if the number of bits in error between a CRP at the registration phase and authentication phase is statistically low enough [181,182].

Another application of PUFs is to utilize the high randomness introduced during its manufacturing to create a secure key from the device. Such key generation requires ideal PUFs that are robust, tamper evident, and unpredictable. In order to correct the noise in a PUF response and generate cryptographic keys, the concepts of a secure sketch in a fuzzy extractor which are described in Sections 5 and 6, are utilized. Secure sketches use the concept of error correction coding to ensure that we recover the original PUF data from the noisy PUF.

3.2. Types of PUFs

Different components can be utilized to extract fingerprints from a device. Initially randomness was physically introduced into a device to extract a fingerprint, whose examples are optical and coating PUFs. An optical PUF uses the physical property of a transparent material, in which the light particles scatter in an uncontrolled manner. When a laser beam falls on it, a unique and random pattern is produced [183]. A coating PUF can be built by filling the space between a network of metal wires on top of an IC with a randomly doped opaque with dielectric particles [177]. Due to the random placement of doping, each couple of wires will have a random capacitance value. This PUF is generally used on the top layer of the ICs which is generally used to protect the underlying circuits from attackers' inspection. The capacitance between the wires will change when a portion of the coating is removed. These PUFs have been used as RFID tags [184].

Current technologies prefer to utilize the PUFs designed based on intrinsic variations, because they are already embedded in the device. *Silicon PUFs* exploit the intrinsic variations in the IC manufacturing process. *Leaked current-based PUFs* were dependent on the concept that the combination of different intrinsic variations in a circuit will result in a different leakage current. Another example of silicon PUFs are *Delay-based PUFs*, where distinct delays are caused in a circuit due to the manufacturing variations in its components even in an identical layout. The most popular examples of delay-based PUFs are *Arbiter PUFs* and *Ring Oscillator PUFs* [185]. These PUFs need huge groups of device components to make them secure. These PUFs tend to take a substantial amount of chip space and are vulnerable to side channel attacks because they give off information due to heat and therefore, they may not be suitable for IoT nodes. Therefore, PUFs which can be easily deployed, occupy less space, and require less power are required for security purposes. These characteristics can be found in PUFs made from memory devices.

3.2.1. Memory based PUF's

Memory-based PUFs utilize the memory chips readily available in different devices and hence can be easily deployed in any device, to allow PUF based security in a network. PUFs can be made from different types of memory including SRAM, Flash, MRAM, memristor, and ReRAM.

SRAM cells are made of cross-coupled inverters which are connected by access transistors and because of the intrinsic manufacturing variations, SRAM cells typically settle into a "0" or "1" state consistently [186,187]. SRAM PUFs were introduced in [188] where the initial values of the cells, on powering on the SRAM were used to generate a unique fingerprint. SRAMs tend to emit energy when they switch states which can be detected by checking the wavelength of the laser by using a signal analyzer. When this side channel information is leaked, it can provide enough information to the attacker about the device in order to clone it [189,190].

In our experiments, we utilize addressable PUF generator protocol to extract the fingerprint of the Memory device [191]. In this protocol, a random number and a password which is known to both the client and server is exclusively-or'ed and sent to a hash function. This random number acts as our challenge for the protocol. The message digest obtained from the hash gives us the information in terms of address of the memory cell, from which the response of the PUF is extracted. This protocol can also be further extended to use multiple addresses to extract a key from different places of the device. A new key can also be easily developed by changing the Random number which will give rise to new addresses for fingerprint extraction.

The manufacturing of a Re-RAM is very similar to memories which use CMOS technology and hence can be easily integrated into the IC. Voltage is used to program Re-RAM and erase cells while current is used to read the resistance values of the cells [192]. Only specific parameters such as low and high state resistance, which are used in making the CRPs in the device can be measured consistently.

Re-RAM-based PUFs use the value of V_{set} after programming or the resistance of each cell, to differentiate between "0" and "1" states. The flipping probability of cells during response generation is higher when their resistance values are closer to the threshold, when they are subjected to voltage changes, temperature, aging or electromagnetic interference. This could result in 5%–20% CRP matching error rate if the number of cells whose resistance is close threshold is high.

PUFs designed using Re-RAMs are immune to side channel attacks without direct access to the chip as they work at or below noise level. Re-RAM and magnetoresistive random-access memory (MRAM) offer low power options when compared to current Flash technologies because they rely on resistance. Table 4 compares the operation requirements for Flash, Re-RAM and MRAM.

3.3. Comparison and applications of different types of PUFs

Table 5 demonstrates different PUFs based on their types and their specific features. Since the goal of PUFs is to enhance the security of entities and nodes in a network, they can utilize the physical and application to enhance the security. We can also develop a cross-layer design which considers PUF along with these two layers. However, there is no need for any additional implementation, it is possible to

Table 4
Operation requirements for current and novel memory technologies [193–195].

Operation	Flash	ReRAM	MRAM
Program parameter	NOR Vds = 5 V; NAND Vgb = 15 V	Vset = +100 mV	Current: 500 μA
Program power required	1 mJ/bit	10 pJ/bit	100 pJ/bit
Program speed (ns)	5000 ns/block	2–20 ns	2–20 ns
Read parameter	Voltage: 10 mV	Current: 1–20 μA	Current: 1–20 μA
Read power required	10 pJ	1 pJ	1 pJ
Read speed (ns)	50 ns	2–20 ns	2–20 ns

use the built-in structures and element to enhance the security for the aforementioned layers (physical and application). Based on the security challenges mentioned in Sections 2.2 and 2.3, PUFs are introduced to address several security issues of IoT networks. In most scenarios, PUFs are used to achieve authentication and authorization.

In [196], PUF was utilized to create chaotic Pseudo random number generators (random number generators which are extremely sensitive to initial seed), and the PUF was utilized to create the initial seed. PUFs were also used in [197], to enhance the strength of the password generated by improving the entropy by using PUF as an entropy pump which showed a 48% improvement in entropy from human passwords.

In [198], PUFs were part of the PKI architecture proposed which did not require storage of reference patterns and passwords in order to mitigate attacks related to stolen databases. PUFs were also a part of digital signature schemes proposed in [199]. In order to verify the digital signature, these schemes require error correction mechanisms to obtain a stable response from the PUF. PUFs are also deemed one of the candidates for a true random number generator in [200], where the random number was generated using a plurality of cells from the PUF response.

3.4. Robustness of PUF responses

Analog physical parameters which are used to extract fingerprint of a device, are prone to noise and may change due to temperature, supply voltage and other parameters. The digital fingerprint of the PUF may vary due to changes in any of the parameters mentioned before. Differential design techniques are applied in order to mitigate some of the environmental dependencies in a PUF to make it more stable [177].

Although differential design techniques may improve reliability, the change in environmental conditions will introduce noise in the PUF output. Noise may cause one or more PUF output bits to be flipped resulting in server not authenticating a valid client resulting in a false negative authentication. Mechanisms to reduce the *intra hamming distance* of the PUF responses are required to address the various challenges that hinder their perfect reproducibility. Different error correction coding techniques are being employed to improve the reproducibility of the PUF using "*Fuzzy Extractor*" schemes elaborated in Section 5. These mechanisms should be designed to reduce the noise in a client PUF causing false negative authentication while ensuring that another client device is not mistaken for the authenticated client, i.e., false positive authentication should not increase while reducing false negative authentications. These mechanisms should reduce any false authentication rates while improving the quality of PUF. Various error correcting codes have been utilized to reduce the intra PUF variation factor and improve the similarity of PUF responses for the same query [177]. Adding redundant information (parity bit or helper data) will increase the probability of error detection and correction during the challenge response authentication. Linear block codes and 2-D Hamming codes were used earlier in several PUFs.

4. PUF-based security solutions

In Section 2, we reviewed different possible attacks in IoT networks from various aspects such as applications, architecture, communication, and data planes. Moreover, we also discussed hardware-based attacks and possible solutions in addition to PUFs. In this section, we study the role of PUFs in mitigating different types of attacks.

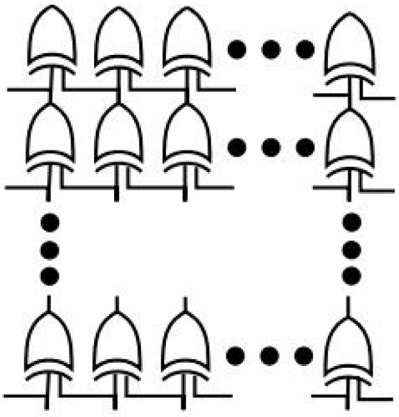


Fig. 8. Rectangular PPUF circuit using XOR gates [157].

We mentioned Controlled PUFS (C-PUFs) in Section 2 as one possible solution to prevent man-in-the-middle attacks. The authors in [129] used C-PUFs to increase the robustness and reliability of normal PUFs. C-PUFs can be only accessed by an algorithm that is physically linked to the PUF. In this scheme, the C-PUF is achieved using multiple layers. For instance, they placed one random hash function before the PUF to prevent the man-in-the-middle attack by avoiding a *chosen challenge attack* on PUFs. This technique does not allow the attacker to utilize the model-based adversary attack to extract the PUF parameters. Also, an error correcting code is placed after the PUF output to reduce the noisy output measurements which result in more robust responses. Also, the authors located another random hash function after the error correcting code to make the relationship between the response and the physical measurements more complex. The authors proposed the idea of multiple personalities for this controlled-PUF and they showed this C-PUF is robust to prevent man-in-the-middle attacks.

The authors in [157] proposed a new class of PUF called *Public-PUF (PPUF)* and investigated their role in preventing side channel attacks. They showed that the attacker can reverse engineer this PPUF; however, getting a response or output takes a long time. The authors proposed a rectangular circuit built from XOR gates with a width of w and a height of h. The output results from this circuit depends on the gates' delays and even using the simulation it needs a long time to simulate the output. The PPUF needs to reach the steady state point before starting the simulation because of the intermediate gates' delays. As a result, in addition to the simulation time to predict the output, an additional time is required for the circuit to reach a steady state. Fig. 8 shows one simple structure of this PPUF. The authors used these delays to generate the public key in cryptographic methods.

To evaluate the performance of this PUF, the authors showed that if an attacker utilized a 10 GHz processor and assuming that simulating one number in the PPUF requires 3×10^{12} cycles, then simulating a PPUF with $w \approx 10^4$ needs 1.7×10^{29} cycles of simulation to extract the secret key. Based on [219], if the attacker uses two billion computers with the power of 10 GHz per computer which is very optimistic, then

Table 5
Comparison of different types of PUFs.

	Type	Name	Weak/strong	Ref	Comment
Special fabrication	Coating		Weak	[201–203]	Smaller number of CRP
	Optical		Strong	[202,204,205]	Difficult to evaluate the uniqueness
Silicon PUF	Delay based	Arbiter	Strong	[206–208]	Vulnerable to attacks
		Ring oscillator	Weak	[185,209,210]	Needs large power and space
	Memory based	Re-RAM	Strong	[192,211,212]	Very sensitive to environmental and voltage fluctuations
		Butterfly	Weak	[213,214]	Unstable adjoining will effect PUF response
		SRAM	Weak	[215–218]	Vulnerable to side-channel attacks

the total required time is 8.5×10^9 or 264 years. This approach of PUF is resilient against different physical attacks such as side channels.

Hardware obfuscation using PUFs is one of the methods that researchers use to avoid the reverse engineering attacks. Using the hardware obfuscation, the designers can hide some part of the design or chips during the fabrication process [220]. The general idea is to keep some modules of the design such that the chip does not work properly during the design. However, it is possible to use an activation function to actuate those modules in a post-fabrication process [221,222]. Without the activation function or the key, the attacker cannot understand the whole design of the IC, which makes the reverse engineering process less likely. Many authors used PUF to generate a unique key or activation function for each chip in such a way that the attacker cannot use the leaked key for other chips to reverse engineer them [161,223]. The authors of [224] proposed a new method for the activation key using strong PUFs. The authors proposed that characterizing the strong PUF is almost impossible. The authors showed a scheme where they used a small portion of CRP's space from the strong PUF in the activation phase. As a result, the size of the look-up-table is not costly for the designer; however, the attacker has to deal with the entire space of the strong PUF to extract the key.

Hardware Trojans and IP hijacking are common types of security attacks. Many illegal companies are investing on IP hijacking and reverse engineering such as mask theft and illegal overproduction. PUFs can offer effective countermeasures against these kind of attacks. The authors of [167] proposed logic encryption for ICs. Encryption techniques can be categorized in two types, (first) Some logic gates will be added to the main circuit to hide the true logic and behavior of the circuit, (second) A set of specific sequences of inputs are required to reach a valid state of functionality. In this case, the state transition will be modified such that the attacker cannot predict the behavior. The authors used a combinational method to minimize the probability of *rare values*. *Rare values* are those true low controllability signals and conditions that can trigger the inserted Trojan. To design the encryption algorithm, the authors used PUF with additional logic gates to generate a random unique key for each IC. The authors claimed that using this PUF and encryption method, the attacker cannot exploit those *rare values* to initiate the Trojan.

5. Fuzzy extractors

Fuzzy extractors (FE) are mechanisms which help to extract original registered responses from noisy responses. In [28], the authors proposed the FE in order to correct noisy biometric responses. To correct the noisy response, the initial response of the client needs to be registered. During the registration process, from an initial input w, a uniformly random string R and non-secret string P (helper data) are extracted using a fuzzy extractor. This mechanism allows the string R to be used as a key and reproduced exactly with the help of P, even though the input changes to some w' but remains close to w. These mechanisms are said to be *information-theoretic secure*, i.e., a crypto-system whose security is derived only from information theory, where the adversary cannot break the encryption due to insufficient information, thereby allowing them to be used in cryptography.

FEs are constructed using *Secure Sketch* (SS), which is a combination of two procedures "sketch" and "recover" that allow precise reconstruction of the initial input from noisy input by making use of some helper data P.

Helper Data P is extracted from initial input w, which can be made publicly available in the "sketch" phase. This output P will be used in the "recover" phase along with noisy input w' to recover w. This method is secure because the publicly available *Helper Data* reveals little to no information about w. Fig. 9 describes a secure sketch (see Fig. 10).

FE is defined with a combination of a pair of procedures called "generate" and "reproduce". In the "generate" phase, the fuzzy extractor uses the "sketch" phase of the SS where *Helper data*, P and *Key*, R are extracted from the given input w. The "reproduce" phase uses the "recover" phase of the secure sketch which makes use of the *Helper data* to reproduce the initial input w from a noisy input w' along with the random extractor used in the "sketch" phase, to extract the randomness from the recovered w. The ability to recover w from w' is highly dependent on the technique used to correct the data. In this case, based on the initial FE proposed by Dodis et al. in [28], the authors utilized the concepts of correcting the received communication message by using error correction codes. All the other FEs proposed in the literature also followed the same trend, utilizing different types of error correction codes to correct noisy input data. These error correcting codes are used in the "sketch" phase of the FE. If the hamming distance between the noisy input w' and input w is too large, it may not be possible to recover w from w'. Fig. 11 shows the construction of a FE using a secure sketch. As Secure Sketches have an error tolerance capability, error correcting codes are utilized in their construction. The error correcting code C is used to correct errors in w' by shifting the codeword, although w' may not be in C. Two different constructions are used for secure sketch are presented [28]:

- *"Code-Offset Construction: For input w, a uniformly random codeword c is selected from C, and SS(w) to be the shift needed to get from c to w: $SS(w) = w - c$. To compute $Rec(w', s)$, the shift s is subtracted from w' to get $c' = w' - s$: c' is decoded to retrieve c and w is computed by shifting back to get $w = c + s$. When code C is linear, the information in s is essentially the syndrome of w"* [28].
- *"Syndrome Construction: The sketch SS(w) computes $s = syn(w)$, where syn is the syndrome. To recover the key, a unique vector e is chosen such that $syn(e) = syn(w') - s$ and output $w = w' - e$"*.

Bose–Chaudhuri–Hocquenghem (BCH) codes which are one of the cyclic error correcting codes, have been implemented in fuzzy extractors, where 192 syndrome bits out of 255 bits were exposed as helper data offered a stability of 88% to a PUF correcting 30 errors in a noisy PUF [225]. High performance error correction coding schemes such as Convolution Coding and low density parity check (LDPC) are applicable to PUF error correction but are highly complex to implement. It is practical to use good error correction codes which consider maximum likelihood estimation, implementation complexity and secret key leakage to improve the performance of the PUF.

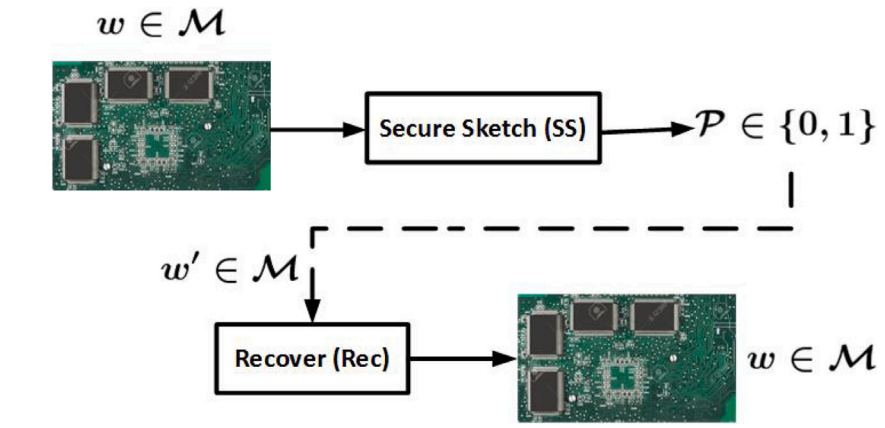


Fig. 9. Secure sketch.

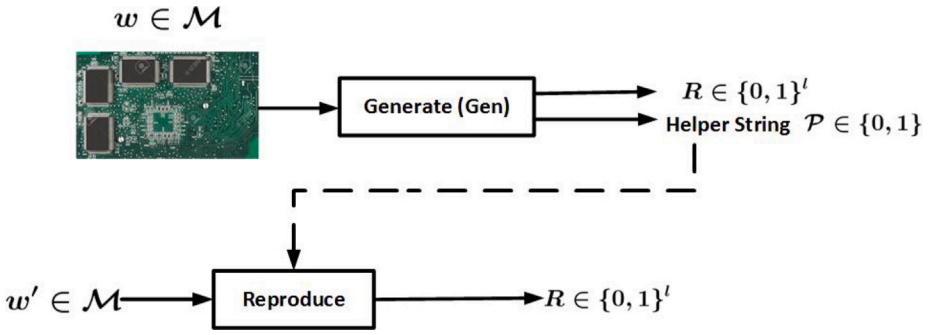


Fig. 10. Block diagram of a fuzzy extractor.

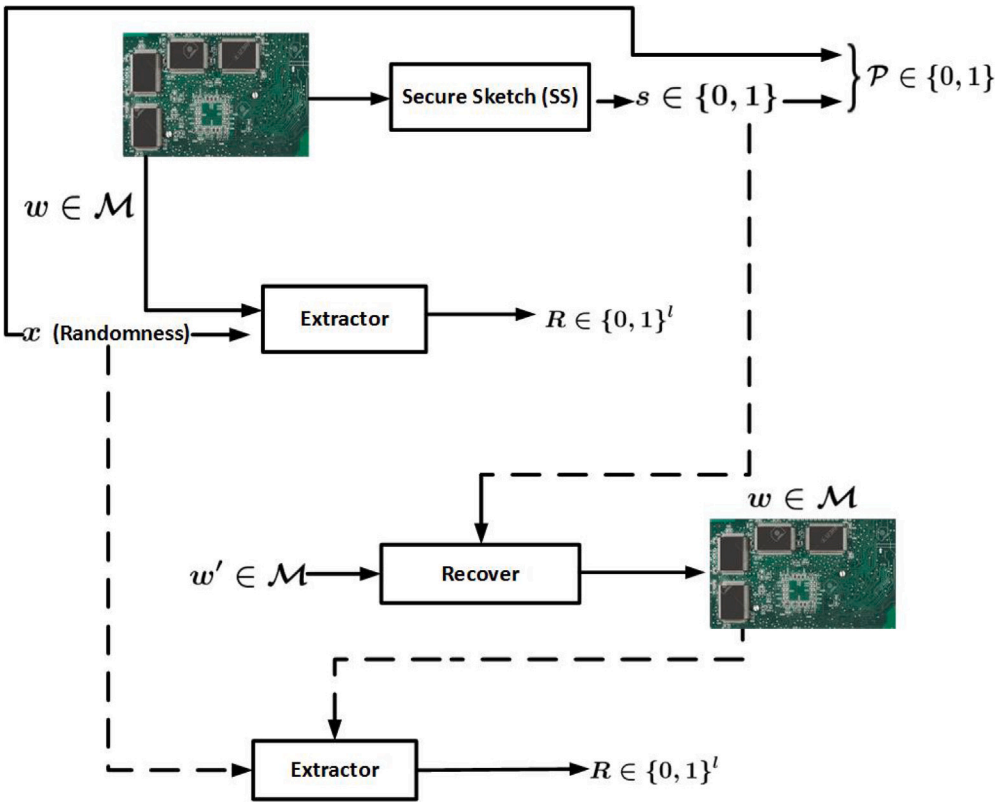


Fig. 11. Construction of a fuzzy extractor using secure sketch extractor.

6. Generation of cryptographic keys from PUFs using fuzzy extractors

Secret key generation using PUFs will allow users to produce a key from their own devices which need not be stored in the device. Using PUFs to produce keys will make the device unclonable and hence less susceptible to hacking. Additionally, the use of PUFs eliminates the security issues related to key storage and distribution. Different PUFs have been used in the past to generate reliable and reproducible cryptographic keys using FE.

Different methods to generate reproducible keys using PUFs have been proposed in the literature [204,206,226–229]. The key generation scheme proposed in [226] uses BCH codes and random number generators for the construction of fuzzy extractors. The main goal of BCH codes is to help reconstruct the PUF estimate from noisy PUF data in order to be used in the "secure sketch" phase of the FE.

The error correction capability of the fuzzy extractor is dependent on the strength of the decoder to recover the message. This strength generally enhances with increasing the complexity of the system. If a system with high complexity is being utilized in a fuzzy extractor, there is a good chance that the system is not only correcting all the errors, but might over correct the message thereby introducing more errors into the message. This scheme may be suitable for authentication where a certain error margin can be tolerated. In key generation schemes, the key which is used for encrypting needs to be reproduced with a zero error margin in order to decrypt the encrypted data.

Many papers in the literature have proposed different FE architectures. In [230], a practical and modular design for key generation using a PUF was proposed which was evaluated on a set of FPGA devices on a ring oscillator PUF. This fuzzy extractor was able to produce a secure 128-bit key with a failure rate $< 10^{-9}$ in 5.62 ms. In this research, the author used a repetition based BCH code, where the syndrome generation and error decoding are fully combinatorial.

In [231], an implementation of a fuzzy extractor on a FPGA was done in order to measure the efficiency in terms of the required hardware resources. The authors proposed decoder architectures for Reed Muller and Golay codes. In [232], a helper data algorithm using soft decision information was proposed. The authors argued that this technique could lower the entropy loss of a helper data algorithm. The authors chose to use soft decision maximum likelihood decoding and generalized multiple concatenated codes as other soft decision decoders such as belief propagation for LDPC and Viterbi algorithm for convolutional codes that require very long data streams.

In [233], a helper data algorithm based on generalized concatenated, Reed Muller and Reed Solomon codes was proposed, where a simplex inner code (16,5,8) was selected with a RM outer code of 128 bits to obtain a codeword of length $128 \times 16 = 2048$.

In a recent research work proposed in [234], a FE structure based on Polar codes for SRAM PUFs was proposed. This work utilized complex Hash-Aided SC decoder to ensure that the key was reproducible. Using this methods the authors in [234] were able to reproduce their key using 896 helper bits, with a failure probability of 10^{-9} for a key length of 128 bits. In [235], we used a custom-built Arduino shield to read the fingerprint of SRAM PUF devices. We later compared the efficiency of different fuzzy extractor schemes to generate reliable and reproducible keys while estimating the FAR and FRR. The FE techniques proposed in the literature utilize long Helper data bits and complex decoder structures to extract the low failure probability rates. There is a high probability in increasing the FAR and FRR of the PUF based Keys when using high complex-sophisticated decoders. These decoders were designed to extract the message from very noisy channels, which may not be the best solution to PUF based keys. Using decoders which were designed to extract messages from very noisy channels may cause the fuzzy extractor to estimate the PUF response in an incorrect way leading to an increase in the false authentication rate. Further research to understand the average error rate of each PUF and using mechanisms appropriate for the PUF will help mitigate such situations. Table 6 presents a comparison of the results obtained by these different architectures.

Fuzzy extractors may be used to extract a PUF response using a noisy PUF response, but many research efforts have focused on attacking a PUF using a fuzzy extractor as they produce the helper data. This is due to the fact, that while the publicly available *helper data* is used to recover the noisy responses in the client side, it can also be a primary target for attackers to extract the response from the PUF. These extractions can further lead to characterizing the entire device if enough responses can be extracted from a single PUF. Some of the attacks on Fuzzy extractors using "helper data manipulation" are discussed in Section 7.

In general, all the Key generation protocols proposed strive to achieve a probability of error of 10^{-6} to 10^{-10}. The tolerable error rate for a PUF-based key generation scheme is highly dependent on the application the device utilizes. When used for a sensitive application, a very low probability of error in the range of 10^{-10} is desired which is possible at a cost of increased complexity. Therefore, there is a trade-off between the desired complexity and the probability of error.

The concept of on-demand key generation has also gained huge popularity in recent years. Theoretically, the number of attacks possible on a PUF can also be reduced when a different key is utilized each time a message is transmitted. There are many factors that depend on the generation of a new key. Some of these factors include the entropy space of the PUF, time delay to read the PUF response, time delay caused in the generation of a key using a FE, and so on. A strong PUF has a high entropy and will be able to generate multiple keys for each message while a weak PUF has to repeat the same keys over a period of time. In a SRAM PUF, there is an additional delay cost because one has to wait for the PUF to turn ON/OFF in order to read the PUF response. This cost is also a key factor in generating the on-demand-keys. The fuzzy extractor utilized will also have to extract a new key each time a new PUF response is chosen and store it in the server for decryption which also adds to the time delay. If the application can tolerate the added delay, and the entropy of the PUF allows it to generate on-demand-keys, then this idea will be popular in the cryptographic community. This is an open challenge in this area, where researchers are constantly trying to increase the entropy of the PUF while reducing the delay caused by on-demand key generation. Depending on the application, a trade-off should be established between the delay caused due to current hardware technology and the need for a new key each time it is required.

7. Attacks on fuzzy extractors via helper data manipulation

In [236], the authors proposed a method to extract the cryptographic keys from PUF using helper data manipulation attacks by tracking the power or electromagnetic measurements of the power consumed when manipulated versions of helper are used to generate keys. The value of power is extracted from every trace and the bit pattern corresponding to minimum power is considered to be the correct codeword.

As ring oscillator (RO) PUFs are easily deployable on FPGAs. ROs have been gaining a lot of attention in many research works for some time. In [237], the authors had demonstrated manipulation attacks on temperature aware cooperative RO PUFs [210], sequential pairing algorithm [238], group-based RO PUFs [239], all entropy distiller constructions [240]. The attacks proposed in this paper statistically estimate the failure rate when the Helper data is manipulated to extract the key. In temperature aware cooperative RO PUFs, errors were injected into the Helper data which comprises cooperating pairs, while calculating the failure rate. In a similar way, steep polynomials were injected into the entropy distiller, Helper data in Group RO PUF, to overshadow the random frequency variations. In Sequentially paired RO PUFs, a sequential pairing algorithm is used to select disjunct pairs.

Table 6
Comparison of failure probability and flipping probability of different fuzzy extractor schemes proposed in the literature [180].

Fuzzy extractor construction	Key length	Helper data bits	Failure probability	Flipping probability
BCH repetition code [230]	128	2052	10^{-9}	13%
Reed Muller generalized multiple concatenated coding [232]	128	13 952	10^{-6}	15%
Generalized concatenated (GC) Reed Muller [233]	2048	2048	$5.37.10^{-10}$	14%
GC Reed Solomon [233]	1024	1024	$3.47.10^{-10}$	14%
Polar codes with SC [234]	128	896	10^{-6}	15%
Polar codes with Hash-Aided SCL decoder [234]	128	896	10^{-9}	15%
Serially concatenated BCH-Polar codes with SC decoder [180]	250	**262**	$\mathbf{10^{-8}}$	15%
Serially concatenated BCH-Polar codes with belief propagation decoder [180]	250	262	10^{-10}	15%

In order to attack this RO PUF, response bits are matched with other responses and the failure rate is calculated to understand the secret key.

Different constructions of error correction codes used in FE are discussed in [241]. Some of their findings include new threats using Helper data algorithms in terms of leakage and manipulation. The authors also derive a way to accurately calculate the leakage of repetition codes in case of bias. The leakage caused due to soft decision coding is underestimated according to the authors. The authors also propose a "divide and conquer manipulation attack" for parallel, concatenated and soft decision codes.

In [242], the authors discuss using the Helper data manipulation attack on Pattern Matching Key Generators (PMKG) which was proposed in [243]. PMKGs utilize sub-strings of a long PUF response stream called *patterns*. In these mechanisms, the indices of the sub string are stored as the *Key* while the patterns are stored as public *helper data*. The key is reconstructed using a matching procedure when a regenerated response is inserted across the patterns. In [242], the full bitstreams of a PUF and their secret indices were extracted by manipulating the Helper data and statistically observing the failure rate of a PMKG. The attack was further demonstrated on a 4-XOR arbiter PUF.

8. Conclusion and future discussion

This survey paper reviews different security challenges in IoT networks and devices. Different domains such as data, communication, architecture, and application are considered for the security taxonomy in IoTs. Software attacks are discussed based on these aforementioned domains. Then, we reviewed the semiconductor manufacturing process chain which consists of different tasks to accomplish the hardware design for ICs. This chain brings various vulnerabilities in different points which can be considered as hardware attacks. PUFs are considered as one of the potential solutions to counter the hardware based attacks. Noting the challenges of using PUFs due to the variations between different responses. We surveyed various methods to extract keys from noisy PUF responses using fuzzy extractors schemes. The importance and need for using the fuzzy extractors schemes for generating cryptographic keys from PUFs is discussed. A brief discussion on the probability of FE schemes to over correct the Noisy PUF responses and leading to false authentication rate or false rejection rate is discussed.

The future directions of using PUFs need to focus not only on extracting reproducible schemes from Noisy PUFs but also to understand the behavior of each PUF for different environmental and physical conditions. This challenge required designing schemes that will tailor themselves to the expected error in each device, thereby not allowing under correction or over correction of the PUF responses; thereby reducing the False Rejection Rate and False Authentication Rate of the PUF.

CRediT authorship contribution statement

Alireza Shamsoshoara: Conceptualization, Methodology, Writing - original draft. **Ashwija Korenda:** Conceptualization, Methodology, Writing - original draft. **Fatemeh Afghah:** Supervision, Conceptualization, Methodology, Writing - original draft. **Sherali Zeadally:** Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments which helped us improve the organization, content, and quality of this paper. This material is based upon the work supported by the National Science Foundation under Grant No. 1827753.

References

[1] M. Bhayani, M. Patel, C. Bhatt, Internet of things (IoT): In a way of smart world, in: Proceedings of the International Congress on Information and Communication Technology, Springer, 2016, pp. 343–350.
[2] M. Galleso, Samsung Gear S3 Classic and Frontier: An Easy Guide to Best Features, Lulu Press, Inc, 2016.
[3] D. Lu, T. Liu, The application of IoT in medical systems, in: IT in Medicine and Education (ITME), 2011 International Symposium on, Vol. 1, IEEE, 2011, pp. 272–275.
[4] R. Kloti, V. Kotronis, P. Smith, Openflow: A security analysis, in: Network Protocols (ICNP), 2013 21st IEEE International Conference on, IEEE, 2013, pp. 1–6.
[5] Z.-K. Zhang, M.C.Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, S. Shieh, IoT security: Ongoing challenges and research opportunities, in: Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on, IEEE, 2014, pp. 230–234.
[6] L. Fuhong, L. Qian, Z. Xianwei, C. Yueyun, H. Daochao, Cooperative differential game for model energy-bandwidth efficiency tradeoff in the internet of things, China Commun. 11 (1) (2014) 92–102.
[7] A. Valehi, A. Razi, B. Cambou, W. Yu, M. Kozicki, A graph matching algorithm for user authentication in data networks using image-based physical unclonable functions, in: 2017 Computing Conference, IEEE, 2017, pp. 863–870.
[8] A. Valehi, A. Razi, Maximizing energy efficiency of cognitive wireless sensor networks with constrained age of information, IEEE Trans. Cogn. Commun. Netw. 3 (4) (2017) 643–654.
[9] K. Främling, S. Kubler, A. Buda, Universal messaging standards for the IoT from a lifecycle management perspective., IEEE Internet Things J. 1 (4) (2014) 319–327.
[10] A. Shamsoshoara, Y. Darmani, Enhanced multi-route ad hoc on-demand distance vector routing, in: Electrical Engineering (ICEE), 2015 23rd Iranian Conference on, IEEE, 2015, pp. 578–583.
[11] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, K. Leung, A survey on the IETF protocol suite for the internet of things: Standards, challenges, and opportunities, IEEE Wirel. Commun. 20 (6) (2013) 91–98.
[12] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, J. Ashdown, Distributed cooperative spectrum sharing in UAV networks using multi-agent reinforcement learning, in: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2019, pp. 1–6.
[13] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, J. Ashdown, K. Turck, A solution for dynamic spectrum management in mission-critical uav networks, 2019, arXiv preprint arXiv:1904.07380.
[14] F. Afghah, A. Shamsoshoara, L. Njilla, C. Kamhoua, A reputation-based stackelberg game model to enhance secrecy rate in spectrum leasing to selfish IoT devices, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2018, pp. 312–317, http://dx.doi.org/10.1109/INFCOMW.2018.8406970.

[15] A.D. Rayome, As IoT attacks increase 600% in one year, businesses need to up their security - tech Republic, 2018, https://www.techrepublic.com/article/as-iot-attacks-increase-600-in-one-year-businesses-need-to-up-their-security/, (Accessed on 09/26/2018).

[16] S. 2020, IoT: number of connected devices worldwide 2012-2025 | statista, 2020, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, (Accessed on 04/27/2020).

[17] Z. Doffman, Cyberattacks on IOT devices surge 300% in 2019, 'measured in billions', report claims, 2019, https://www.forbes.com/sites/zakdoffman/2019/09/14/dangerous-cyberattacks-on-iot-devices-up-300-in-2019-now-rampant-report-claims/#3e0d17195892, (Accessed on 04/28/2020).

[18] B. Vigliarolo, Wannacry: A cheat sheet for professionals - techRepublic, 2017, https://www.techrepublic.com/article/wannacry-the-smart-persons-guide/, (Accessed on 09/26/2018).

[19] G. O'Gorman, G. McDonald, Ransomware: A Growing Menace, Symantec Corporation, 2012.

[20] R. Brewer, Ransomware attacks: Detection, prevention and cure, Netw. Secur. 2016 (9) (2016) 5–9.

[21] N. Heath, Petya ransomware: Where it comes from and how to protect yourself, 2017, https://www.techrepublic.com/article/petya-ransomware-where-it-comes-from-and-how-to-protect-yourself/, (Accessed on 09/26/2018).

[22] A. Sapienza, A. Bessi, S. Damodaran, P. Shakarian, K. Lerman, E. Ferrara, Early warnings of cyber threats in online discussions, in: Data Mining Workshops (ICDMW), 2017 IEEE International Conference on, IEEE, 2017, pp. 667–674.

[23] I. Yaqoob, E. Ahmed, M.H. ur Rehman, A.I.A. Ahmed, M.A. Al-garadi, M. Imran, M. Guizani, The rise of ransomware and emerging security challenges in the internet of things, Comput. Netw. 129 (2017) 444–458.

[24] S. Hilton, Dyn analysis summary of friday october 21 attack | dyn blog, 2016, https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/, (Accessed on 04/01/2020).

[25] Dyn cyberattack, 2018, URL https://en.wikipedia.org/wiki/2016_Dyn_cyberattack.

[26] Can your cardiac device be hacked? 2018, URL https://www.acc.org/about-acc/press-releases/2018/02/20/13/57/can-your-cardiac-device-be-hacked.

[27] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris, J. Henkel, Computation offloading and resource allocation for low-power IoT edge devices, in: Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, IEEE, 2016, pp. 7–12.

[28] Y. Dodis, L. Reyzin, A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2004, pp. 523–540.

[29] A. Shamsoshoara, Overview of blakley's secret sharing scheme, 2019, arXiv preprint arXiv:1901.02802.

[30] S.-Y. Chou, Y.-S. Chen, J.-H. Chang, Y.-D. Chih, T.-Y.J. Chang, 11.3 a 10nm 32kb low-voltage logic-compatible anti-fuse one-time-programmable memory with anti-tampering sensing scheme, in: Solid-State Circuits Conference (ISSCC), 2017 IEEE International, IEEE, 2017, pp. 200–201.

[31] M. Joye, On white-box cryptography, Secur. Inform. Netw. (2008) 7–12.

[32] S.L. Kinney, Trusted Platform Module Basics: using TPM in Embedded Systems, Elsevier, 2006.

[33] E. Gerjuoy, Shor's factoring algorithm and modern cryptography: An illustration of the capabilities inherent in quantum computers, Amer. J. Phys. 73 (6) (2005) 521–540.

[34] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, D. Smith-Tone, Report on Post-Quantum Cryptography, US Department of Commerce, National Institute of Standards and Technology, 2016.

[35] M. Amy, O. Di Matteo, V. Gheorghiu, M. Mosca, A. Parent, J. Schanck, Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3, in: International Conference on Selected Areas in Cryptography, Springer, 2016, pp. 317–337.

[36] C.J. Mitchell, The impact of quantum computing on real-world security: A 5g case study, 2019, arXiv preprint arXiv:1911.07583.

[37] V. Mavroeidis, K. Vishi, M.D. Zych, A. Jøsang, The impact of quantum computing on present cryptography, 2018, arXiv preprint arXiv:1804.00200.

[38] E. Barker, A. Roginsky, Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths, NIST Spec. Publ. 800 (2011) 131A.

[39] B. Gassend, D. Clarke, M. Van Dijk, S. Devadas, Silicon physical random functions, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM, 2002, pp. 148–160.

[40] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R.S. Chakraborty, D. Mahata, M.M. Prabhu, Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database, IEEE Trans. Dependable Secure Comput. (2018).

[41] T. McGrath, I.E. Bagci, Z.M. Wang, U. Roedig, R.J. Young, A puf taxonomy, Appl. Phys. Rev. 6 (1) (2019) 011303.

[42] A. Ehret, K. Gettings, B.R. Jordan, M.A. Kinsy, A survey on hardware security techniques targeting low-power soc designs, in: 2019 IEEE High Performance Extreme Computing Conference (HPEC), 2019, pp. 1–8.

[43] J. Delvaux, R. Peeters, D. Gu, I. Verbauwhede, A survey on lightweight entity authentication with strong PUFs, ACM Comput. Surv. 48 (2) (2015) 1–42.

[44] A. Babaei, G. Schiele, Physical unclonable functions in the internet of things: State of the art and open challenges, Sensors 19 (14) (2019) 3208.

[45] A. Burg, A. Chattopadhyay, K.-Y. Lam, Wireless communication and security issues for cyber–physical systems and the internet-of-things, Proc. IEEE 106 (1) (2018) 38–60.

[46] M. Wolf, D. Serpanos, Safety and security in cyber-physical systems and internet-of-things systems, Proc. IEEE 106 (1) (2018) 9–20.

[47] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications, IEEE Internet Things J. 4 (5) (2017) 1125–1142.

[48] S. Arshad, M.A. Azam, M.H. Rehmani, J. Loo, Recent advances in information-centric networking based internet of things (ICN-IoT), IEEE Internet Things J. (2018).

[49] J. Granjal, E. Monteiro, J.S. Silva, Security for the internet of things: a survey of existing protocols and open research issues, IEEE Commun. Surv. Tutor. 17 (3) (2015) 1294–1312.

[50] H. Chen, Y. Chen, D.H. Summerville, A survey on the application of FPGAs for network infrastructure security, IEEE Commun. Surv. Tutor. 13 (4) (2011) 541–561.

[51] I. Sfyrakis, T. Gross, A survey on hardware approaches for remote attestation in network infrastructures, 2020, arXiv preprint arXiv:2005.12453.

[52] S. Chowdhury, A. Covic, R.Y. Acharya, S. Dupee, F. Ganji, D. Forte, Physical security in the post-quantum era: A survey on side-channel analysis, random number generators, and physically unclonable functions, 2020, arXiv preprint arXiv:2005.04344.

[53] D.M. Mendez, I. Papapanagiotou, B. Yang, Internet of things: Survey on security and privacy, 2017, arXiv preprint arXiv:1707.01879.

[54] F.A. Alaba, M. Othman, I.A.T. Hashem, F. Alotaibi, Internet of things security: A survey, J. Netw. Comput. Appl. 88 (2017) 10–28.

[55] K. Zhao, L. Ge, A survey on the internet of things security, in: Computational Intelligence and Security (CIS), 2013 9th International Conference on, IEEE, 2013, pp. 663–667.

[56] O. El Mouaatamid, M. Lahmer, M. Belkasmi, Internet of things security: Layered classification of attacks and possible countermeasures, Electron. J. Inform. Technol. 9 (9) (2016).

[57] A. Botta, W. De Donato, V. Persico, A. Pescapé, On the integration of cloud computing and internet of things, in: 2014 International Conference on Future Internet of Things and Cloud, IEEE, 2014, pp. 23–30.

[58] I.A.T. Hashem, V. Chang, N.B. Anuar, K. Adewole, I. Yaqoob, A. Gani, E. Ahmed, H. Chiroma, The role of big data in smart city, Int. J. Inf. Manage. 36 (5) (2016) 748–758.

[59] S. Capkun, L. Buttyán, J.-P. Hubaux, Self-organized public-key management for mobile ad hoc networks, IEEE Trans. Mob. Comput. 2 (1) (2003) 52–64.

[60] L. Zhang, Z. Cai, X. Wang, Fakemask: A novel privacy preserving approach for smartphones, IEEE Trans. Netw. Serv. Manag. 13 (2) (2016) 335–348.

[61] I. Andrea, C. Chrysostomou, G. Hadjichristofi, Internet of things: Security vulnerabilities and challenges, in: Computers and Communication (ISCC), 2015 IEEE Symposium on, IEEE, 2015, pp. 180–187.

[62] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM, 2002, pp. 41–47.

[63] P. Pongle, G. Chavan, A survey: Attacks on RPL and 6lowpan in IoT, in: Pervasive Computing (ICPC), 2015 International Conference on, IEEE, 2015, pp. 1–6.

[64] J. Han, M. Ha, D. Kim, Practical security analysis for the constrained node networks: Focusing on the DTLS protocol, in: 2015 5th International Conference on the Internet of Things (IoT), IEEE, 2015, pp. 22–29.

[65] M. Delavar, S. Mirzakuchaki, M.H. Ameri, J. Mohajeri, PUF-based solutions for secure communications in advanced metering infrastructure (AMI), Int. J. Commun. Syst. 30 (9) (2017) e3195.

[66] A.L. Valdivieso Caraguay, A. Benito Peral, L.I. Barona Lopez, L.J. Garcia Villalba, SDN: Evolution and opportunities in the development IoT applications, Int. J. Distrib. Sens. Netw. 10 (5) (2014) 735142.

[67] S.R. Moosavi, T.N. Gia, A.-M. Rahmani, E. Nigussie, S. Virtanen, J. Isoaho, H. Tenhunen, SEA: a secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways, Procedia Comput. Sci. 52 (2015) 452–459.

[68] A. Gaur, B. Scotney, G. Parr, S. McClean, Smart city architecture and its applications based on IoT, Proc. Comput. Sci. 52 (2015) 1089–1094.

[69] M. Vučinić, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, R. Guizzetti, OSCAR: Object security architecture for the internet of things, Ad Hoc Netw. 32 (2015) 3–16.

[70] S. Chakrabarty, D.W. Engels, A secure IoT architecture for smart cities, in: 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2016, pp. 812–813.

[71] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660.

[72] H. Rahimi, A. Zibaeenejad, P. Rajabzadeh, A.A. Safavi, On the security of the 5g-IoT architecture, in: Proceedings of the International Conference on Smart Cities and Internet of Things, ACM, 2018, p. 10.

[73] D. Chen, G. Chang, L. Jin, X. Ren, J. Li, F. Li, A novel secure architecture for the internet of things, in: Genetic and Evolutionary Computing (ICGEC), 2011 Fifth International Conference on, IEEE, 2011, pp. 311–314.

[74] M. Nawir, A. Amir, N. Yaakob, O.B. Lynn, Internet of things (IoT): Taxonomy of security attacks, in: Electronic Design (ICED), 2016 3rd International Conference on, IEEE, 2016, pp. 321–326.

[75] S. Li, L. Da Xu, S. Zhao, The internet of things: a survey, Inform. Syst. Front. 17 (2) (2015) 243–259.

[76] W. Trappe, R. Howard, R.S. Moore, Low-energy security: Limits and opportunities in the internet of things, IEEE Secur. Priv. 13 (1) (2015) 14–21.

[77] E. Alsaadi, A. Tubaishat, Internet of things: Features, challenges, and vulnerabilities, Int. J. Adv. Comput. Sci. Inform. Technol. 4 (1) (2015) 1–13.

[78] A. Belapurkar, A. Chakrabarti, H. Ponnapalli, N. Varadarajan, S. Padmanabhuni, S. Sundarrajan, Distributed Systems Security: Issues, Processes and Solutions, John Wiley & Sons, 2009.

[79] D.N. Sushma, V. Nandal, Security threats in wireless sensor networks, IJCSMS Int. J. Comput. Sci. Manag. Stud. 11 (01) (2011) 59–63.

[80] K. Zhang, X. Liang, R. Lu, X. Shen, Sybil attacks and their defenses in the internet of things, IEEE Internet Things J. 1 (5) (2014) 372–383.

[81] M.H. Yılmaz, H. Arslan, A survey: Spoofing attacks in physical layer security, in: Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th, IEEE, 2015, pp. 812–817.

[82] M.M. Hossain, M. Fotouhi, R. Hasan, Towards an analysis of security issues, challenges, and open problems in the internet of things, in: Services (SERVICES), 2015 IEEE World Congress on, IEEE, 2015, pp. 21–28.

[83] S. Alam, D. De, Analysis of security threats in wireless sensor networks, 2014, arXiv preprint arXiv:1406.0298.

[84] A. Mayzaud, R. Badonnel, I. Chrisment, A taxonomy of attacks in RPL-based internet of things, Int. J. Netw. Secur. 18 (3) (2016) 459–473.

[85] U. Sabeel, S. Maqbool, Categorized security threats in the wireless sensor networks: Countermeasures and security management schemes, Int. J. Comput. Appl. 64 (16) (2013).

[86] A. Mosenia, N.K. Jha, A comprehensive study of security of internet-of-things, IEEE Trans. Emerg. Top. Comput. 5 (4) (2017) 586–602.

[87] M. Li, I. Koutsopoulos, R. Poovendran, Optimal jamming attacks and network defense policies in wireless sensor networks, in: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, IEEE, 2007, pp. 1307–1315.

[88] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, G. Pantziou, A survey on jamming attacks and countermeasures in WSNs, IEEE Commun. Surv. Tutor. 11 (4) (2009).

[89] A. Becher, Z. Benenson, M. Dornseif, Tampering with motes: Real-world physical attacks on wireless sensor networks, in: International Conference on Security in Pervasive Computing, Springer, 2006, pp. 104–118.

[90] K. Lemke, Embedded security: Physical protection against tampering attacks, in: Embedded Security in Cars, Springer, 2006, pp. 207–217.

[91] K. Schramm, T. Wollinger, C. Paar, A new class of collision attacks and its application to DES, in: International Workshop on Fast Software Encryption, Springer, 2003, pp. 206–222.

[92] A. Bogdanov, Multiple-differential side-channel collision attacks on AES, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2008, pp. 30–44.

[93] S.K. Sanadhya, P. Sarkar, New collision attacks against up to 24-step SHA-2, in: International Conference on Cryptology in India, Springer, 2008, pp. 91–103.

[94] L.M.R. Tarouco, L.M. Bertholdo, L.Z. Granville, L.M.R. Arbiza, F. Carbone, M. Marotta, J.J.C. de Santanna, Internet of things in healthcare: Interoperatibility and security issues, in: Communications (ICC), 2012 IEEE International Conference on, IEEE, 2012, pp. 6121–6125.

[95] T. Heer, O. Garcia-Morchon, R. Hummen, S.L. Keoh, S.S. Kumar, K. Wehrle, Security challenges in the IP-based internet of things, Wirel. Pers. Commun. 61 (3) (2011) 527–542.

[96] P. Varga, S. Plosz, G. Soos, C. Hegedus, Security threats and issues in automation IoT, in: Factory Communication Systems (WFCS), 2017 IEEE 13th International Workshop on, IEEE, 2017, pp. 1–6.

[97] M. Burhanuddin, A.A.-J. Mohammed, R. Ismail, M.E. Hameed, A.N. Kareem, H. Basiron, A review on security challenges and features in wireless sensor networks: Iot perspective, J. Telecommun., Electron. Comput. Eng. (JTEC) 10 (1–7) (2018) 17–21.

[98] J. Liu, Y. Xiao, C.P. Chen, Authentication and access control in the internet of things, in: Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on, IEEE, 2012, pp. 588–592.

[99] M. Brachmann, S.L. Keoh, O.G. Morchon, S.S. Kumar, End-to-end transport security in the IP-based internet of things, in: Computer Communications and Networks (ICCCN), 2012 21st International Conference on, IEEE, 2012, pp. 1–5.

[100] L.K. Bysani, A.K. Turuk, A survey on selective forwarding attack in wireless sensor networks, in: Devices and Communications (ICDeCom), 2011 International Conference on, IEEE, 2011, pp. 1–5.

[101] W.Z. Khan, X. Yang, M.Y. Aalsalem, Q. Arshad, Comprehensive study of selective forwarding attack in wireless sensor networks, Int. J. Comput. Netw. Inf. Secur. 3 (1) (2011) 1.

[102] C. Karlof, D. Wagner, Secure Routing in Sensor Networks: Attacks and Countermeasures, 2003.

[103] I. Krontiris, T. Giannetsos, T. Dimitriou, Launching a Sinkhole attack in wireless sensor Networks; the intruder side, in: IEEE International Conference on Wireless & Mobile Computing, Networking & Communication, IEEE, 2008, pp. 526–531.

[104] B.G. Choi, E.J. Cho, J.H. Kim, C.S. Hong, J.H. Kim, A Sinkhole Attack detection mechanism for LQI based mesh Routing in WSN, in: Information Networking, 2009. ICOIN 2009. International Conference on, IEEE, 2009, pp. 1–5.

[105] N. Gandhewar, R. Patel, Detection and prevention of Sinkhole Attack on AODV protocol in mobile Adhoc Network, in: Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on, IEEE, 2012, pp. 714–718.

[106] T. Borgohain, U. Kumar, S. Sanyal, Survey of security and privacy issues of Internet of Things, 2015, arXiv preprint arXiv:1501.02211.

[107] D. Senie, P. Ferguson, Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing, Network (1998).

[108] M.A. Hamid, M. Rashid, C.S. Hong, Routing security in sensor network: Hello flood attack and defense, in: IEEE ICNEWS, 2006, pp. 2–4.

[109] V.P. Singh, S. Jain, J. Singhai, Hello flood attack and its countermeasures in wireless sensor networks, Int. J. Comput. Sci. Issues (IJCSI) 7 (3) (2010) 23.

[110] K. Sharma, M. Ghose, Wireless sensor networks: An overview on its security threats, IJCA (2010) 42–45, Special Issue on "Mobile Ad-hoc Networks" MANETs.

[111] K.S. Win, Analysis of detecting Wormhole Attack in wireless networks, in: World Academy of Science, Engineering and Technology, Citeseer, 2008, pp. 422–428.

[112] R.H. Jhaveri, A.D. Patel, J.D. Parmar, B.I. Shah, et al., MANET routing protocols and Wormhole attack against AODV, Int. J. Comput. Sci. Netw. Secur. 10 (4) (2010) 12–18.

[113] A.D. Wood, J.A. Stankovic, Denial of service in sensor networks, computer 35 (10) (2002) 54–62.

[114] P. Yi, Z. Dai, Y. Zhong, S. Zhang, Resisting flooding attacks in Ad hoc Networks, in: Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on, Vol. 2, IEEE, 2005, pp. 657–662.

[115] W. Eddy, TCP SYN Flooding Attacks and Common Mitigations, Tech. rep., RFC, 2007.

[116] L. Joncheray, A simple active attack against TCP, in: USENIX Security Symposium, 1995, pp. 2–15.

[117] T. Roosta, S. Shieh, S. Sastry, Taxonomy of security attacks in sensor networks and countermeasures, in: the First IEEE International Conference on System Integration and Reliability Improvements, Vol. 25, 2006, p. 94.

[118] A.-S.K. Pathan, H.-W. Lee, C.S. Hong, Security in wireless sensor networks: Issues and challenges, in: Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, Vol. 2, IEEE, 2006, pp. 6–pp.

[119] S.J. Murdoch, Hot or not: Revealing hidden services by their clock skew, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, ACM, 2006, pp. 27–36.

[120] M. Manzo, T. Roosta, S. Sastry, Time synchronization attacks in sensor networks, in: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, 2005, pp. 107–116.

[121] C. Arackaparambil, S. Bratus, A. Shubina, D. Kotz, On the reliability of wireless fingerprinting using clock skews, in: Proceedings of the Third ACM Conference on Wireless Network Security, ACM, 2010, pp. 169–174.

[122] B. Yu, B. Xiao, Detecting selective forwarding attacks in wireless sensor networks, in: Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, IEEE, 2006, pp. 8–pp.

[123] S. Roy, S. Setia, S. Jajodia, Attack-resilient hierarchical data aggregation in sensor networks, in: Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, 2006, pp. 71–82.

[124] M. Rezvani, A. Ignjatovic, E. Bertino, S. Jha, Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks, IEEE Trans. Dependable Secur. Comput. 12 (1) (2015) 98–110.

[125] A.W. Atamli, A. Martin, Threat-based security analysis for the Internet of Things, in: 2014 International Workshop on Secure Internet of Things, IEEE, 2014, pp. 35–43.

[126] Y. Lu, L. Da Xu, Internet of Things (IoT) cybersecurity research: a review of current research topics, IEEE Internet Things J. 6 (2) (2018) 2103–2115.

[127] H. Lin, N. Bergmann, IoT privacy and security challenges for Smart Home Environments, Information 7 (3) (2016) 44.

[128] E. Baccelli, C. Gündoğan, O. Hahm, P. Kietzmann, M.S. Lenders, H. Petersen, K. Schleiser, T.C. Schmidt, M. Wählisch, RIOT: An open source operating system for low-end embedded devices in the IoT, IEEE Internet Things J. 5 (6) (2018) 4428–4440.

[129] B. Gassend, D. Clarke, M. Van Dijk, S. Devadas, Controlled physical random functions, in: 18th Annual Computer Security Applications Conference, 2002. Proceedings, IEEE, 2002, pp. 149–160.

[130] M. Tehranipoor, C. Wang, Introduction to hardware security and trust, Springer Science & Business Media, 2011.

[131] S.T. Eckmann, G. Vigna, R.A. Kemmerer, STATL: An attack language for state-based intrusion detection, J. Comput. Secur. 10 (1–2) (2002) 71–103.

[132] H. Ning, S. Hu, Technology classification, industry, and education for future Internet of Things, Int. J. Commun. Syst. 25 (9) (2012) 1230–1241.

[133] X. Yang, Z. Li, Z. Geng, H. Zhang, A multi-layer security model for Internet of Things, in: Internet of Things, Springer, 2012, pp. 388–393.

[134] Y. Song, Security in Internet of Things, 2013.

[135] K. Sonar, H. Upadhyay, A survey: DDOS attack on Internet of Things, Int. J. Eng. Res. Dev. 10 (11) (2014) 58–63.

[136] S.A. Kumar, T. Vealey, H. Srivastava, Security in Internet of Things: Challenges, solutions and future directions, in: 2016 49th Hawaii International Conference on System Sciences (HICSS), IEEE, 2016, pp. 5772–5781.

[137] L. Nastase, Security in the Internet of Things: A survey on application layer protocols, in: 2017 21st International Conference on Control Systems and Computer Science (CSCS), IEEE, 2017, pp. 659–666.

[138] P. Leach, M. Mealling, R. Salz, A Universally Unique Identifier (UUID) URN Namespace, Tech. rep., Network Working Group, 2005.

[139] Z. Song, A.A. Cárdenas, R. Masuoka, Semantic middleware for the Internet of Things, in: Internet of Things (IOT), 2010, IEEE, 2010, pp. 1–8.

[140] Z. Shelby, K. Hartke, C. Bormann, The Constrained Application Protocol (CoAP), Tech. rep., IETF, 2014.

[141] U. Hunkeler, H.L. Truong, A. Stanford-Clark, MQTT-s—A publish/subscribe protocol for wireless sensor networks, in: Communication Systems Software and Middleware and Workshops, 2008. Comsware 2008. 3rd International Conference on, IEEE, 2008, pp. 791–798.

[142] R. Khan, S.U. Khan, R. Zaheer, S. Khan, Future internet: the Internet of Things architecture, possible applications and key challenges, in: Frontiers of Information Technology (FIT), 2012 10th International Conference on, IEEE, 2012, pp. 257–260.

[143] P. Koeberl, Ü. Kocabaş, A.-R. Sadeghi, Memristor PUFs: a new generation of memory-based physically unclonable functions, in: 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2013, pp. 428–431.

[144] D.E. Holcomb, W.P. Burleson, K. Fu, Power-up SRAM state as an identifying fingerprint and source of true random numbers, IEEE Trans. Comput. 58 (9) (2009) 1198–1210, http://dx.doi.org/10.1109/TC.2008.212.

[145] S. Sutar, A. Raha, V. Raghunathan, Memory-based combination PUFs for device authentication in embedded systems, IEEE Trans. Multi-Scale Comput. Syst. 4 (4) (2018) 793–810.

[146] C. Keller, F. Gürkaynak, H. Kaeslin, N. Felber, Dynamic memory-based physically unclonable function for the generation of unique identifiers and true random numbers, in: 2014 IEEE International Symposium on Circuits and Systems (ISCAS), 2014, pp. 2740–2743.

[147] M. Rostami, F. Koushanfar, R. Karri, A primer on hardware security: Models, methods, and metrics, Proc. IEEE 102 (8) (2014) 1283–1295.

[148] M. Rostami, F. Koushanfar, J. Rajendran, R. Karri, Hardware security: Threat models and metrics, in: Proceedings of the International Conference on Computer-Aided Design, IEEE Press, 2013, pp. 819–823.

[149] F. Koushanfar, S. Fazzari, C. McCants, W. Bryson, P. Song, M. Sale, M. Potkonjak, Can EDA combat the rise of electronic counterfeiting? in: DAC Design Automation Conference 2012, IEEE, 2012, pp. 133–138.

[150] P. Rohatgi, Improved techniques for side-channel analysis, in: Cryptographic Engineering, Springer, 2009, pp. 381–406.

[151] J.P. ChristofPaar, B. Preneel, Understanding Cryptography: A Textbook for Students and ractitioners, Springer, 2010.

[152] P. Mahajan, A. Sachdeva, A study of Encryption Algorithms AES, DES and RSA for security, Glob. J. Comput. Sci. Technol. (2013).

[153] Wikipedia, RSA (cryptosystem) - Wikipedia, 2019, https://en.wikipedia.org/wiki/RSA_(cryptosystem), (Accessed on 04/26/2019).

[154] P. Rohatgi, Electromagnetic attacks and countermeasures, in: Cryptographic Engineering, Springer, 2009, pp. 407–430.

[155] A. Schlösser, D. Nedospasov, J. Krämer, S. Orlic, J.-P. Seifert, Simple photonic emission analysis of AES, J. Cryptogr. Eng. 3 (1) (2013) 3–15.

[156] D. Genkin, A. Shamir, E. Tromer, RSA key extraction via low-bandwidth acoustic cryptanalysis, in: Annual Cryptology Conference, Springer, 2014, pp. 444–461.

[157] N. Beckmann, M. Potkonjak, Hardware-based public-key cryptography with public physically unclonable functions, in: International Workshop on Information Hiding, Springer, 2009, pp. 206–220.

[158] Y. Bi, Enhanced Hardware Security Using Charge-Based Emerging Device Technology (Thesis in Ph.D.), University of Central Florida, 2016.

[159] R. Torrance, D. James, The state-of-the-art in semiconductor reverse rngineering, in: 2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2011, pp. 333–338.

[160] S.M. Saeed, X. Cui, R. Wille, A. Zulehner, K. Wu, R. Drechsler, R. Karri, Towards reverse engineering reversible logic, 2017, arXiv preprint arXiv:1704.08397.

[161] J.B. Wendt, M. Potkonjak, Hardware obfuscation using PUF-based logic, in: 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2014, pp. 270–271.

[162] J.A. Roy, F. Koushanfar, I.L. Markov, Ending piracy of Integrated Circuits, Computer 43 (10) (2010) 30–38.

[163] J.H. Anderson, A PUF design for secure FPGA-based embedded systems, in: 2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, 2010, pp. 1–6.

[164] R. Karri, J. Rajendran, K. Rosenfeld, M. Tehranipoor, Trustworthy hardware: Identifying and classifying hardware Trojans, Computer 43 (10) (2010) 39–46.

[165] M. Tehranipoor, F. Koushanfar, A survey of hardware Trojan Taxonomy and detection, IEEE Des. Test Comput. 27 (1) (2010) 10–25.

[166] A. Waksman, S. Sethumadhavan, Silencing hardware Backdoors, in: 2011 IEEE Symposium on Security and Privacy, IEEE, 2011, pp. 49–63.

[167] S. Dupuis, P.-S. Ba, G. Di Natale, M.-L. Flottes, B. Rouzeyre, A novel hardware logic encryption technique for thwarting illegal overproduction and hardware Trojans, in: 2014 IEEE 20th International on-Line Testing Symposium (IOLTS), IEEE, 2014, pp. 49–54.

[168] V. Venugopalan, C.D. Patterson, Surveying the hardware Trojan threat landscape for the internet-of-things, J. Hardw. Syst. Secur. 2 (2) (2018) 131–141.

[169] CISCO, Jim_green_cisco_connect.pdf, 2014, https://www.cisco.com/c/dam/global/en_ph/assets/ciscoconnect/pdf/bigdata/jim_green_cisco_connect.pdf, Accessed on 03/26/2020).

[170] CISCO, Microsoft word - IoT Reference Model White Paper June 4, 2014.doc, 2014, http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf, (Accessed on 03/26/2020).

[171] C. Lesjak, D. Hein, J. Winter, Hardware-security technologies for industrial IoT: TrustZone and security controller, in: IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2015, pp. 002589–002595.

[172] Arm TrustZone Technology, 2019, https://developer.arm.com/ip-products/security-ip/trustzone, accessed: 2019-12-04.

[173] Technologies trustzone for cortex-a, 2019, https://www.arm.com/why-arm/technologies/trustzone-for-cortex-a, accessed: 2019-12-04.

[174] Technologies trustzone for cortex-m, 2019, https://www.arm.com/why-arm/technologies/trustzone-for-cortex-m, accessed: 2019-12-04.

[175] Trustzone Technology for Armv8-M, 2020, https://developer.arm.com/ip-products/security-ip/trustzone/trustzone-for-cortex-m, accessed: 2020-01-13.

[176] U. Rührmair, S. Devadas, F. Koushanfar, Security based on physical unclonability and disorder, in: Introduction to Hardware Security and Trust, Springer, 2012, pp. 65–102.

[177] C. Herder, M.-D. Yu, F. Koushanfar, S. Devadas, Physical unclonable functions and applications: A tutorial, Proc. IEEE 102 (8) (2014) 1126–1141.

[178] A. Maiti, I. Kim, P. Schaumont, A robust physical unclonable function with enhanced challenge-response set, IEEE Trans. Inf. Forensics Secur. 7 (1) (2012) 333–345.

[179] I.A. Bautista Adames, J. Das, S. Bhanja, Survey of emerging technology based physical unclonable funtions, in: Proceedings of the 26th Edition on Great Lakes Symposium on VLSI, ACM, 2016, pp. 317–322.

[180] A.R. Korenda, F. Afghah, B. Cambou, A secret key generation scheme for internet of things using ternary-states ReRAM-based physical unclonable functions, in: 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2018, pp. 1261–1266.

[181] B. Cambou, F. Afghah, Physically unclonable functions with multi-states and machine learning, in: 14th International Workshop on Cryptographic Architectures Embedded in Logic Devices (CryptArchi), 2016, p. 1.

[182] U. Rührmair, H. Busch, S. Katzenbeisser, Strong PUFs: models, constructions, and security proofs, in: Towards Hardware-Intrinsic Security, Springer, 2010, pp. 79–96.

[183] R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, Physical one-way functions, Science 297 (5589) (2002) 2026–2030.

[184] P. Tuyls, L. Batina, RFID-tags for anti-counterfeiting, in: Cryptographers' Track at the RSA Conference, Springer, 2006, pp. 115–131.

[185] A. Shamsoshoara, Ring oscillator and its application as physical unclonable function (PUF) for password management, 2019, arXiv preprint arXiv:1901.06733.

[186] A. Vijayakumar, V. Patil, S. Kundu, On improving reliability of SRAM-based physically unclonable functions, J. Low Power Electron. Appl. 7 (1) (2017) 2.

[187] C. Böhm, M. Hofer, W. Pribyl, A microcontroller SRAM-PUF, in: Network and System Security (NSS), 2011 5th International Conference on, IEEE, 2011, pp. 269–273.

[188] D.E. Holcomb, W.P. Burleson, K. Fu, et al., Initial SRAM state as a fingerprint and source of true random numbers for RFID tags, in: Proceedings of the Conference on RFID Security, Vol. 7, 2007, p. 2.

[189] B. Cambou, M. Orlowski, PUF designed with resistive RAM and ternary states, in: Proceedings of the 11th Annual Cyber and Information Security Research Conference, ACM, 2016, p. 1.

[190] C. Helfmeier, C. Boit, D. Nedospasov, S. Tajik, J.-P. Seifert, Physical vulnerabilities of physically unclonable functions, in: Proceedings of the Conference on Design, Automation & Test in Europe, European Design and Automation Association, 2014, p. 350.

[191] B. Cambou, Encoding data for cells in a PUF that corresponds to a response in a challenge response pair, 2019, US Patent 10, 439, 828.

[192] P. Pavan, R. Bez, P. Olivo, E. Zanoni, Flash memory cells-an overview, Proc. IEEE 85 (8) (1997) 1248–1271.

[193] Y.-H. Chang, J.-W. Hsieh, T.-W. Kuo, Endurance enhancement of flash-memory storage systems: an efficient static wear leveling design, in: Proceedings of the 44th Annual Design Automation Conference, ACM, 2007, pp. 212–217.

[194] H. Akinaga, H. Shima, Resistive random access memory (reram) based on metal oxides, Proc. IEEE 98 (12) (2010) 2237–2251.

[195] S. Tehrani, Status and outlook of MRAM memory technology, in: Electron Devices Meeting, 2006. IEDM'06. International, IEEE, 2006, pp. 1–4.

[196] S. Kalanadhabhatta, D. Kumar, K.K. Anumandla, S.A. Reddy, A. Acharyya, PUF-based secure chaotic random number generator design methodology, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. (2020) 1–5.

[197] Q. Wang, M. Gao, G. Qu, PUF-PassSE: A PUF based password strength enhancer for IoT applications, in: 20th International Symposium on Quality Electronic Design (ISQED), IEEE, 2019, pp. 198–203.

[198] B. Cambou, Addressable PUF Generators for Database-Free Password Management System, CryptArchi, 2018.

[199] B.F. Cambou, Secure digital signatures using physical unclonable function devices with reduced error rates, 2020, US Patent App. 16/560, 502.

[200] B. Cambou, Data compiler for true random number generation and related methods, 2019, US Patent 10, 175, 949.

[201] P. Tuyls, G.-J. Schrijen, B. Škorić, J. Van Geloven, N. Verhaegh, R. Wolters, Read-proof hardware from protective coatings, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2006, pp. 369–383.

[202] B. Skoric, G.-J. Schrijen, W. Ophey, R. Wolters, N. Verhaegh, J. van Geloven, Experimental hardware for coating PUFs and optical PUFs, in: Security with Noisy Data, Springer, 2007, pp. 255–268.

[203] B. Skoric, S. Maubach, T.A. Kevenaar, P. Tuyls, Information-theoretic analysis of coating PUFs, IACR Cryptol. ePrint Arch. 2006 (2006) 101.

[204] K. Kursawe, A.-R. Sadeghi, D. Schellekens, B. Skoric, P. Tuyls, Reconfigurable physical unclonable functions-enabling technology for tamper-resistant storage, in: Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on, IEEE, 2009, pp. 22–29.

[205] U. Rührmair, C. Hilgers, S. Urban, A. Weiershäuser, E. Dinter, B. Forster, C. Jirauschek, Optical PUFs Reloaded, Eprint. Iacr. Org, 2013.

[206] G.E. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in: Proceedings of the 44th Annual Design Automation Conference, ACM, 2007, pp. 9–14.

[207] S. Morozov, A. Maiti, P. Schaumont, An analysis of delay based PUF implementations on FPGA, in: International Symposium on Applied Reconfigurable Computing, Springer, 2010, pp. 382–387.

[208] K. Fruhashi, M. Shiozaki, A. Fukushima, T. Murayama, T. Fujino, The arbiter-PUF with high uniqueness utilizing novel arbiter circuit with delay-time measurement, in: Circuits and Systems (ISCAS), 2011 IEEE International Symposium on, IEEE, 2011, pp. 2325–2328.

[209] A. Maiti, P. Schaumont, Improving the quality of a physical unclonable function using configurable ring oscillators, in: Field Programmable Logic and Applications, 2009. FPL 2009. International Conference on, IEEE, 2009, pp. 703–707.

[210] C.-E. Yin, G. Qu, Temperature-aware cooperative ring oscillator PUF, in: Hardware-Oriented Security and Trust, 2009. HOST'09. IEEE International Workshop on, IEEE, 2009, pp. 36–42.

[211] B. Cambou, M. Orlowski, PUF designed with resistive RAM and ternary states, in: Proceedings of the 11th Annual Cyber and Information Security Research Conference, ACM, 2016, p. 1.

[212] F. Afghah, B. Cambou, M. Abedini, S. Zeadally, A reram physically unclonable function (ReRAM PUF)-based approach to enhance authentication security in software defined wireless networks, Int. J. Wirel. Inf. Netw. (2018) 1–13.

[213] S.S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, P. Tuyls, The butterfly PUF protecting IP on every FPGA, in: Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on, IEEE, 2008, pp. 67–70.

[214] U. Rührmair, D.E. Holcomb, PUFs at a glance, in: Proceedings of the Conference on Design, Automation & Test in Europe, European Design and Automation Association, 2014, p. 347.

[215] J. Guajardo, S.S. Kumar, G.-J. Schrijen, P. Tuyls, FPGA intrinsic PUFs and their use for IP protection, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2007, pp. 63–80.

[216] J. Guajardo, S.S. Kumar, G.-J. Schrijen, P. Tuyls, Physical unclonable functions and public-key crypto for FPGA IP protection, in: Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on, IEEE, 2007, pp. 189–195.

[217] M. Claes, V. van der Leest, A. Braeken, Comparison of SRAM and FF PUF in 65 nm technology, in: Nordic Conference on Secure IT Systems, Springer, 2011, pp. 47–64.

[218] S. Assiri, B. Cambou, D.D. Booher, D.G. Miandoab, M. Mohammadinodoushan, Key exchange using ternary system to enhance security, in: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2019, pp. 0488–0492.

[219] scmo, How many computers are there in the world? — SCMO, 2019, https://www.scmo.net/faq/2019/8/9/how-many-compaters-is-there-in-the-world#:~:text=In%202019%2C%20there%20were%20over,servers%2C%20desktops%2C%20and%20laptops. (Accessed on 06/30/2020).

[220] F. Koushanfar, Hardware metering: A survey, in: Introduction to Hardware Security and Trust, Springer, 2012, pp. 103–122.

[221] S. Zamanzadeh, A. Jahanian, Higher security of ASIC fabrication process against reverse engineering attack using automatic netlist encryption methodology, Microprocess. Microsyst. 42 (2016) 1–9.

[222] J. Rajendran, Y. Pino, O. Sinanoglu, R. Karri, Security analysis of logic obfuscation, in: Proceedings of the 49th Annual Design Automation Conference, 2012, pp. 83–89.

[223] Y. Alkabani, F. Koushanfar, Active hardware metering for intellectual property protection and security, in: USENIX Security Symposium, 2007, pp. 291–306.

[224] S. Khaleghi, W. Rao, Hardware obfuscation using strong PUFs, in: 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, 2018, pp. 321–326.

[225] G.E. Suh, C.W. O'Donnell, S. Devadas, AEGIS: A single-chip secure processor, Inf. Secur. Tech. Rep. 10 (2) (2005) 63–73.

[226] H. Kang, Y. Hori, T. Katashita, M. Hagiwara, K. Iwamura, Performance analysis for PUF data using fuzzy extractor, in: Ubiquitous Information Technologies and Applications, Springer, 2014, pp. 277–284.

[227] T. Ziola, Z. Paral, S. Devadas, G.E. Suh, V. Khandelwal, Authentication with physical unclonable functions, 2014, US Patent 8, 782, 396.

[228] B. Škorić, P. Tuyls, W. Ophey, Robust key extraction from physical uncloneable functions, in: International Conference on Applied Cryptography and Network Security, Springer, 2005, pp. 407–422.

[229] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. Van Dijk, S. Devadas, Extracting secret keys from integrated circuits, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 13 (10) (2005) 1200–1205.

[230] R. Maes, A. Van Herrewege, I. Verbauwhede, PUFKY: A fully functional PUF-based cryptographic key generator, in: Cryptographic Hardware and Embedded Systems–CHES 2012, Springer, 2012, pp. 302–319.

[231] C. Bösch, J. Guajardo, A.-R. Sadeghi, J. Shokrollahi, P. Tuyls, Efficient helper data key extractor on FPGAs, in: International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2008, pp. 181–197.

[232] R. Maes, P. Tuyls, I. Verbauwhede, Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs, in: CHES, Vol. 9, Springer, 2009, pp. 332–347.

[233] S. Puchinger, S. Müelich, M. Bossert, M. Hiller, G. Sigl, On error correction for physical unclonable functions, in: SCC 2015; 10th International ITG Conference on Systems, Communications and Coding; Proceedings of, VDE, 2015, pp. 1–6.

[234] B. Chen, T. Ignatenko, F.M. Willems, R. Maes, E. van der Sluis, G. Selimis, High-rate error correction schemes for SRAM-PUFs based on polar codes, 2017, arXiv preprint arXiv:1701.07320.

[235] A.R. Korenda, F. Afghah, B. Cambou, C. Philabaum, A proof of concept SRAM-based physically unclonable function (PUF) key generation mechanism for IoT devices, in: 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), IEEE, 2019, pp. 1–8.

[236] D. Merli, D. Schuster, F. Stumpf, G. Sigl, Side-channel analysis of PUFs and fuzzy extractors, in: International Conference on Trust and Trustworthy Computing, Springer, 2011, pp. 33–47.

[237] J. Delvaux, I. Verbauwhede, Key-recovery attacks on various RO PUF constructions via helper data manipulation, in: Proceedings of the Conference on Design, Automation & Test in Europe, European Design and Automation Association, 2014, p. 72.

[238] C.-E.D. Yin, G. Qu, LISA: Maximizing RO puf's secret extraction, in: 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), IEEE, 2010, pp. 100–105.

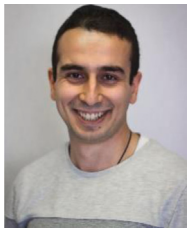
[239] C.-E. Yin, G. Qu, Q. Zhou, Design and implementation of a group-based RO PUF, in: 2013 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2013, pp. 416–421.

[240] C.-E. Yin, G. Qu, Improving PUF security with regression-based distiller, in: Proceedings of the 50th Annual Design Automation Conference, 2013, pp. 1–6.

[241] J. Delvaux, D. Gu, D. Schellekens, I. Verbauwhede, Helper data algorithms for PUF-based key generation: Overview and analysis, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 34 (6) (2014) 889–902.

[242] J. Delvaux, I. Verbauwhede, Attacking PUF-based pattern matching key generators via helper data manipulation, in: Cryptographers' Track at the RSA Conference, Springer, 2014, pp. 106–131.

[243] Z. Paral, S. Devadas, Reliable and efficient PUF-based key generation using pattern matching, in: 2011 IEEE International Symposium on Hardware-Oriented Security and Trust, IEEE, 2011, pp. 128–133.



Alireza Shamsoshoara received the B.Sc. degree in Electrical Engineering from Shahid Beheshti University (SBU), Tehran, Iran, and the M.Sc. degree in Electrical and Communication Engineering from Khaje Nasir Toosi University of Technology (KNTU), Tehran, Iran in 2012 and 2014 respectively. Also, he received the M.Sc. degree in Informatics from Northern Arizona University in 2018. Currently, he is a Ph.D. student in the School of Informatics, Computing & Cyber Systems at Northern Arizona University. His main research interests are security, wireless networks, UAV networks, spectrum sharing, and machine learning.

Ashwija Korenda received the B.Sc degree in Electronic and Communication Engineering from Jawaharlal Technological University (JNTU), Hyderabad, India, and the M.Sc. degree in Electrical Engineering from Northern Arizona University (NAU), Flagstaff, USA in 2015 and 2018 respectively. Currently, she is a Ph.D. student in School of Informatics, Computing and Cyber Systems at Northern Arizona University. Her main interests are security, physically unclonable functions, error correction coding and cognitive radios.

Fatemeh Afghah received the B.Sc. and M.Sc. degrees (with Hons.) in Electrical Engineering from Khajeh Nassir Toosi University of Technology, Tehran, Iran, and the Ph.D. degree in Electrical & Computer Engineering from the University of Maine, ME, USA, in 2005, 2008, and 2013, respectively. She was a Visiting Student with the Department of Electrical and Computer Engineering, University of Maryland, College Park from 2011 to 2012. Currently, she is an Associate Professor with the School of Informatics, Computing and Cyber Systems, Northern Arizona University (NAU), Flagstaff, AZ, USA, where she is the Director of Wireless Networking and Information Processing Laboratory.

Sherali Zeadally received the Bachelor's degree from the University of Cambridge, Cambridge, U.K., in 1991 and the Doctorate degree from the University of Buckingham, Buckingham, U.K., in 1996, both in computer science. He is an Associate Professor with the College of Communication and Information, University of Kentucky, Lexington, KY, USA. Dr. Zeadally is a Fellow of the British Computer Society and the Institution of Engineering Technology, U.K.