A Comprehensive Study of Security of Internet-of-Things

Arsalan Mohsen Nia, *Student Member, IEEE* and Niraj K. Jha, *Fellow, IEEE*

This paper can be cited as: A. Mohsen Nia and N. K. Jha, "A comprehensive study of security of Internet-of-Things," in IEEE Trans. Emerging Topics in Computing, DOI: 10.1109/TETC.2016.2606384
The latest version of this manuscript is available on http://ieeexplore.ieee.org/document/7562568/

***Abstract*—Internet of Things (IoT), also referred to as the Internet of Objects, is envisioned as a transformative approach for providing numerous services. Compact smart devices constitute an essential part of IoT. They range widely in use, size, energy capacity, and computation power. However, the integration of these smart things into the standard Internet introduces several security challenges because the majority of Internet technologies and communication protocols were not designed to support IoT. Moreover, commercialization of IoT has led to public security concerns, including personal privacy issues, threat of cyber attacks, and organized crime. In order to provide a guideline for those who want to investigate IoT security and contribute to its improvement, this survey attempts to provide a comprehensive list of vulnerabilities and countermeasures against them on the edge-side layer of IoT, which consists of three levels: (i) edge nodes, (ii) communication, and (iii) edge computing. To achieve this goal, we first briefly describe three widely-known IoT reference models and define security in the context of IoT. Second, we discuss the possible applications of IoT and potential motivations of the attackers who target this new paradigm. Third, we discuss different attacks and threats. Fourth, we describe possible countermeasures against these attacks. Finally, we introduce two emerging security challenges not yet explained in detail in previous literature.**

***Index Terms*—Availability, confidentiality, countermeasures, integrity, Internet of Things, privacy, security, vulnerabilities.**

I. Introduction

Internet of Things (IoT) does not have a unique definition. However, a broad interpretation of IoT is that it provides any service over the traditional Internet by enabling human-to-thing, thing-to-thing, or thing-to-things communications [1]. IoT represents the interconnection of heterogeneous entities, where the term entity refers to a human, sensor, or potentially anything that may request/provide a service [2].

The emergence of the IoT paradigm is one of the most spectacular phenomena of the last decade. The development of various communication protocols, along with the miniaturization of transceivers, provides the opportunity to transform an isolated device into a communicating thing. Moreover, computing power, energy capacity, and storage capabilities of small computing or sensing devices have significantly improved while their sizes have decreased drastically. These technological advances in electronics and computer science have led to an exponential increase in the number of Internet-connected sensing and computing devices (also known as smart devices) that can provide services only limited by human imagination.

As a side effect, the number of potential threats and possible attacks against security or privacy of a thing or an individual has grown drastically. Unfortunately, these security needs are not yet well-recognized. Thus, security threats and common privacy concerns need to be studied and addressed in depth. This would greatly simplify the development of secure smart devices that enable a plethora of services for human beings, ranging from building automation to health monitoring, in which very different things, e.g., temperature sensor, light sensor, and medical sensors, might interact with each other or with a human carrying a smart computing device, e.g., a smartphone, tablet, or laptop.

IoT security is an ongoing research topic that is attracting increasing attention in academic, industrial as well as governmental research. Many organizations worldwide and multinational corporations are involved in the design and development of IoT-based systems [3]. In order to provide a large number of reliable services, designers encounter several challenges, in particular, in security-related research areas. Several research efforts are currently attempting to discover potential threats and provide countermeasures against them. This survey summarizes these IoT security threats and countermeasures in a level-by-level fashion. More specifically, it:

- describes a comprehensive reference model of IoT,
- provides the readers with a definition of information, assurance, and security (IAS) octave requirements in the scope of IoT,
- summarizes the threats in the edge-side layer of the reference model,
- reviews the proposed countermeasures to address possible threats, and
- introduces two emerging security challenges not yet explained in detail in previous literature.

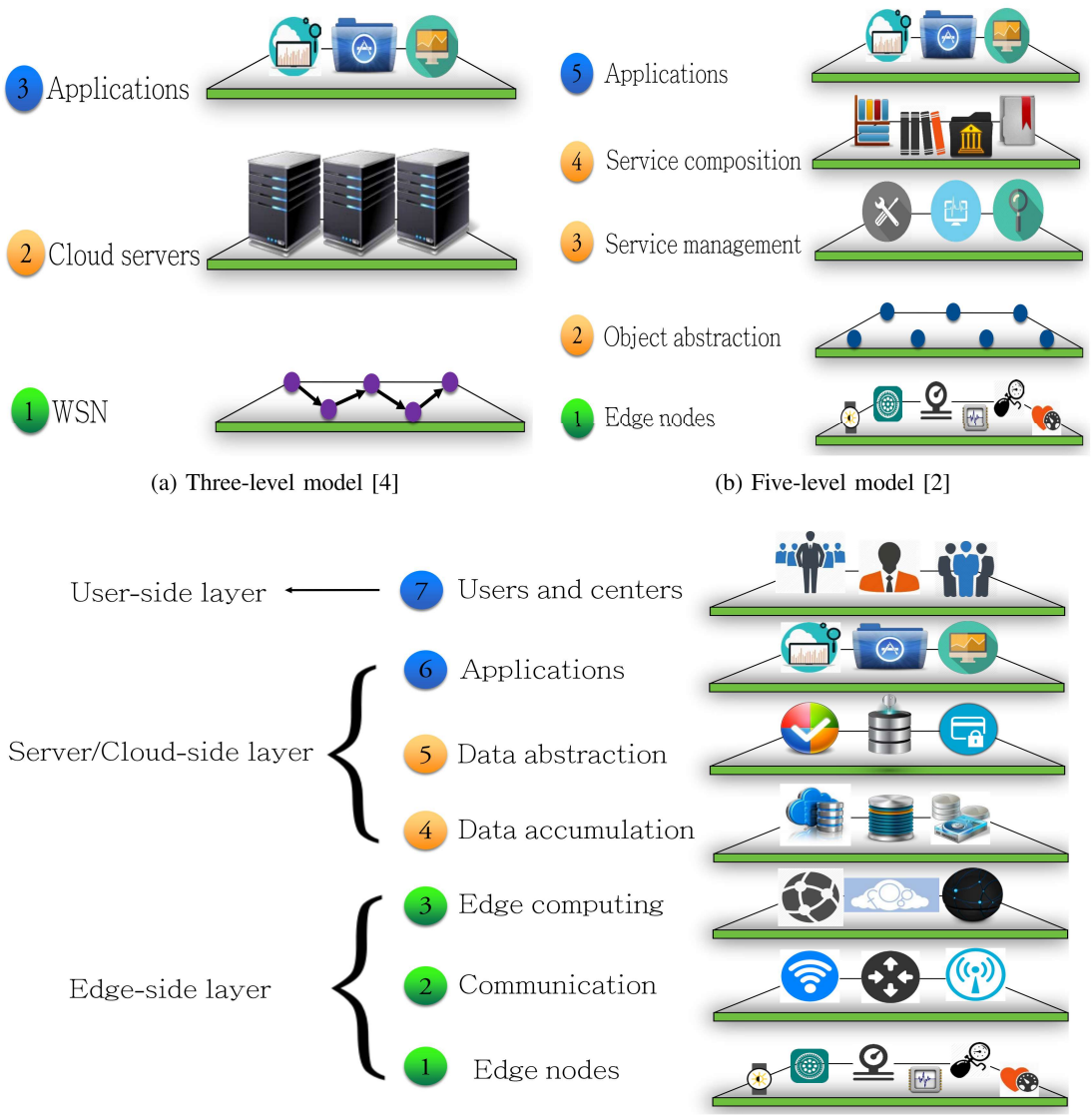
The main objective of the paper is to give the reader the opportunity to explore which attacks have been launched, how they have been addressed, and which threats still lurk.

The remainder of the paper is organized as follows. In Section II, we describe our reference model. In Section III, we discuss the scope of IoT applications and attackers. Moreover, we define the security requirements in the context of IoT. Then, in Section IV, we describe possible attacks against IoT. We summarize countermeasures against these attacks in Section V. In Section VI, we introduce two emerging security challenges. Finally, we provide conclusions and hints for future research in Section VII.

Acknowledgments: This work was supported by NSF under Grant no. CNS-1219570.

Arsalan Mohsen Nia and Niraj K. Jha are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA (e-mail: {$arsalan \| jha$}@princeton.edu).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing



(a) Three-level model [4]

(b) Five-level model [2]

(c) CISCO's seven-level model [5]

Fig. 1: Three IoT reference models.

II. IoT Reference Models

Three IoT reference models have been widely discussed in academic and industrial publications. Fig. 1 shows these models and their different levels. The three-level model [4] is among the first reference models proposed for IoT. It depicts IoT as an extended version of wireless sensor networks (WSNs). In fact, it models IoT as a combination of WSNs and cloud servers, which offer different services to the user. The five-level model [2] is an alternative that has been proposed to facilitate interactions among different sections of an enterprise by decomposing complex systems into simplified applications consisting of an ecosystem of simpler and well-defined components [2]. In 2014, CISCO suggested a comprehensive extension to the traditional three-level and five-level models. CISCO's seven-level model has the potential to be standardized and thus create a widely-accepted reference model for IoT [5]. In this model, data flow is usually bidirectional. However, the dominant direction of data flow depends on the application. For example, in a control system, data and commands travel from the top of the model (applications level) to the bottom (edge node level), whereas, in a monitoring scenario, the flow is from bottom to top.

Providing detailed descriptions of all three IoT reference models is beyond the scope of this paper. In order to summarize IoT security attacks and their countermeasures in a level-by-level fashion, we use the CISCO reference model in this paper. Next, we briefly describe each level of this model.

- ***Level 1-Edge devices***: The first level of this reference model typically consists of computing nodes, e.g., smart controllers, sensors, RFID readers, etc., and different versions of RFID tags. Data confidentiality and integrity must be taken into account from this level upwards.
- ***Level 2-Communication***: The communication level consists of all the components that enable transmission of information or commands: (i) communication between devices in the first level, (ii) communication between the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

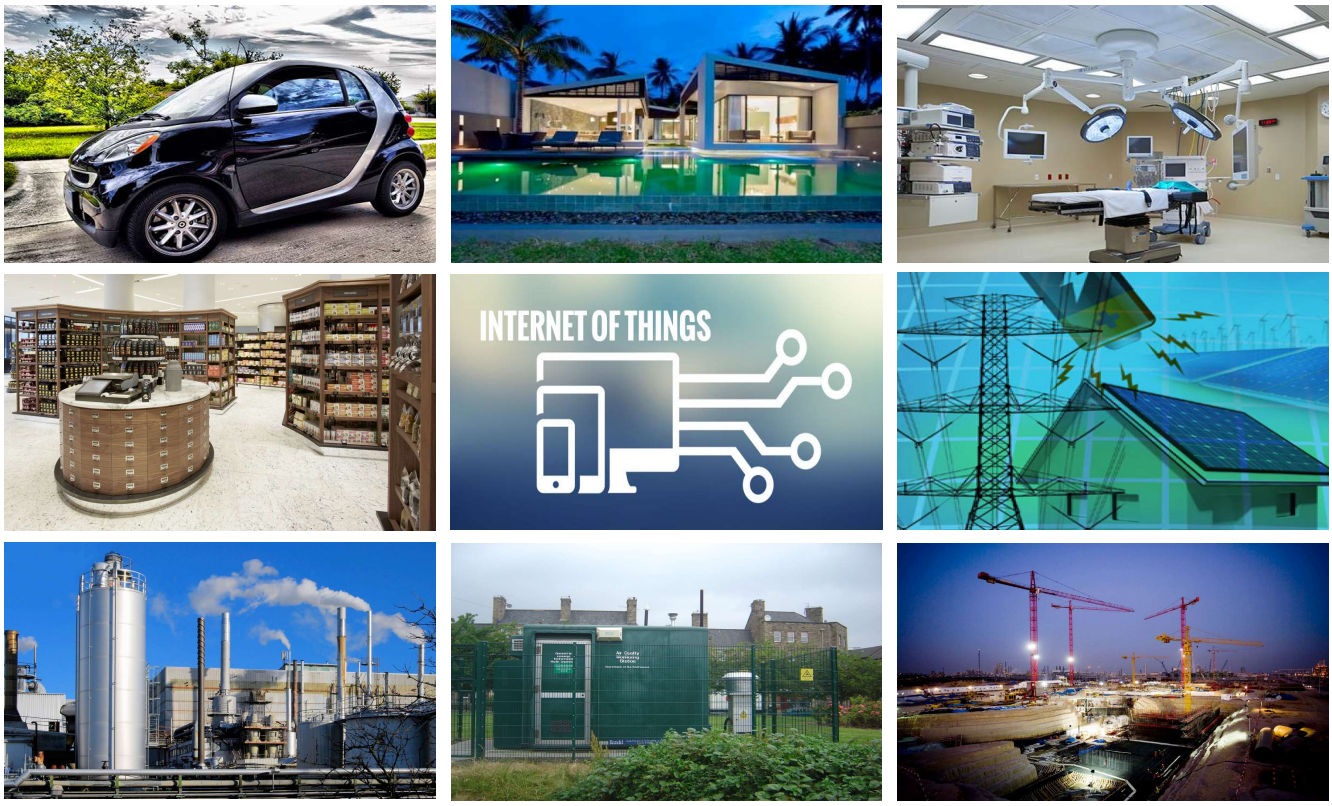


Fig. 2: Different applications of IoT. Starting from the top-left corner and moving clockwise, the applications are: (a) smart vehicles, (b) smart buildings, (c) health monitoring, (d) energy management, (e) construction management, (f) environmental monitoring, (g) production and assembly line management, and (h) food supply chain.

components in the second level, and (iii) transmission of information between the first and third levels (edge computing level).

- ***Level 3-Edge computing***: Edge computing, also called fog computing, is the third level of the model in which simple data processing is initiated. This is essential for reducing the computation load in the higher level as well as providing a fast response. Most real-time applications need to perform computations as close to the edge of the network as possible. The amount of processing in this level depends on the computing power of the service providers, servers, and computing nodes. Typically, simple signal processing and learning algorithms are utilized here.
- ***Level 4-Data accumulation***: Most of the applications may not need instant data processing. This level enables conversion of data in motion to data at rest, i.e., it allows us to store the data for future analysis or to share with high-level computing servers. The main tasks of this level are converting the format from network packets to database tables, reducing data through filtering and selective storing, and determining whether the data are of interest to higher levels.
- ***Level 5-Data abstraction***: This level provides the opportunity to render and store data such that further processing becomes simpler or more efficient. The common tasks of entities at this level include normalization, denormalization, indexing and consolidating data into one place, and providing access to multiple data stores.
- ***Level 6-Applications***: The application level provides information interpretation, where software cooperates with data accumulation and data abstraction levels. The applications of IoT are numerous and may vary significantly across markets and industrial needs.
- ***Level 7-Users and centers***: The highest level of the IoT is where the users are. Users make use of the applications and their analytical data.

III. Background

In this section, we first discuss the scope of IoT applications and, then, describe the potential attackers and their motivation. Finally, we define the security requirements in the context of IoT.

A. Scope of Applications

Smart homes and buildings, electronic health aids, and smarter vehicles are just some of the IoT instances. Each smart device may provide several services to enable a more intuitive environment. However, we are not even close to exhausting the possible uses of IoT. The IoT provides an opportunity to combine sensing, communication, networking, authentication, identification, and computing, and enables numerous services upon request such that access to the information of any smart thing is possible at any time. Fig. 2 demonstrates various

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

applications of the IoT, which we describe next:

1. Smart vehicles: Smart vehicles have started to revolutionize traditional transportation. Small IoT-based systems can enable remote locking/unlocking of cars, download of roadmaps, and access to traffic information. Moreover, Internet-connected cars provide significant protection against theft.

2. Smart buildings: Smart homes and buildings enable effective energy management. For example, smart thermostats, which have embedded sensors and data analysis algorithms, can control air conditioners based on user preferences and habits. Moreover, smart controllers can adjust lighting based on user's usage. Several household items, e.g., refrigerators, televisions, and security systems, could have their own processing units, and provide over-the-Internet services. These smart devices greatly enhance users' convenience. Remotely-controllable devices receive commands from users to perform actions that have an effect on the surrounding environment. Thus, attacks on these devices may lead to physical consequences [6].

3. Health monitoring: Recent advances in biomedical sensing and signal processing, low-power devices, and wireless communication have revolutionized healthcare. IoT-based long-term personal health monitoring and drug delivery systems, in which various physiological signals are captured, analyzed, and stored for future use, provide a fundamentally new approach to healthcare. Smart medical devices are already in use in fitness, diet, and health monitoring systems. The future of IoT-based healthcare systems lies in designing personal health monitors that enable early detection of illnesses.

4. Energy management: Use of smart IoT-based systems, which integrate embedded sensors and actuation components, enables a proactive approach to optimizing energy consumption. In particular, power outlets, lamps, fridges, and smart televisions, which can be controlled remotely, are expected to share information with energy supply companies to optimize the energy consumption in smart homes. Moreover, such things allow the users to remotely control or manage them, and enable scheduling that can lead to a significant reduction in energy consumption.

5. Construction management: Monitoring and management of modern infrastructure, e.g., bridges, traffic lights, railway tracks, and buildings, are one of the key IoT applications [7]. IoT can be used for monitoring any sudden changes in structural conditions that can lead to safety and security risks. It can also enable construction and maintenance companies to share information about their plans. For example, a construction company can let GPS companies know its maintenance plans for the roads and, based on that, the smart GPS devices can choose an alternative route, which avoids the road under construction.

6. Environmental monitoring: The use of smart things with embedded sensors enables environmental monitoring as well as detection of emergency situations, e.g., a flood, which require a fast response. In addition, the quality of air and water can be examined by IoT-based devices. Moreover, humidity and temperature can be easily monitored [8].

7. Production and assembly line management: IoT-based smart systems allow rapid manufacturing of new products and an interactive response to demands by enabling communication between sensors and controlling/monitoring systems [9]. Moreover, intelligent management approaches that use real-time measurements can also enable energy optimization and safety management.

8. Food supply chain: The food supply chain model is fundamentally distributed and sophisticated. IoT can provide valuable information for managers of this chain. Although IoT is already in use within the supply management systems, its current benefits are limited. One of the most obvious and significant advantages of IoT in supply management is that it ensures security and safety of the products by utilizing IoT-based tracking [10]. These devices can raise a warning in case of a security breach at any unauthorized level of the supply management system.

B. Potential attackers and their motivations

Next, we briefly discuss who the attackers that target the IoT might be, and what motivations they may have.

IoT-based systems may manage a huge amount of information and be used for services ranging from industrial management to health monitoring. This has made the IoT paradigm an interesting target for a multitude of attackers and adversaries, such as occasional hackers, cybercriminals, hacktivists, government, etc.

Potential attackers might be interested in stealing sensitive information, e.g., credit card numbers, location data, financial accounts' passwords, and health-related information, by hacking IoT devices. Moreover, they might try to compromise IoT components, e.g., edge nodes, to launch attacks against a third-party entity. Consider an intelligence agency that infects millions of IoT-based systems, e.g., remote monitoring systems, and smart devices, e.g., smart televisions. It can exploit the infected systems and devices to spy on a person of interest or to conduct an attack on a large scale. Also, hacktivists or those in opposition might be interested in attacking smart devices to launch protests against an organization.

C. Definition of security in the scope of IoT

Next, we define two of the most commonly-used terms in the scope of IoT: a secure thing and a security attack. When defining what a secure thing is, it is important to understand the characteristics that define security. Traditionally, security requirements are broken down into three main categories: (i) confidentiality, (ii) integrity, and (iii) availability, referred to as the CIA-triad. Confidentiality entails applying a set of rules to limit unauthorized access to certain information. It is crucial for IoT devices because they might handle critical personal information, e.g., medical records and prescription. For instance, an unauthorized access to personal health devices may reveal personal health information or even lead to life-threatening situations [11]. Integrity is also necessary for providing a reliable service. The device must ensure that the received commands and collected information are legitimate. An integrity compromise may lead to serious adverse consequences. For example, integrity attacks against medical devices, e.g., an insulin pump [12] or a pacemaker

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

TABLE I: Security requirements

Requirement	Definition	Abbreviations
Confidentiality	Ensuring that only authorized users access the information	C
Integrity	Ensuring completeness, accuracy, and absence of unauthorized data manipulation	I
Availability	Ensuring that all system services are available, when requested by an authorized user	A
Accountability	An ability of a system to hold users responsible for their actions	AC
Auditability	An ability of a system to conduct persistent monitoring of all actions	AU
Trustworthiness	An ability of a system to verify identity and establish trust in a third party	TW
Non-repudiation	An ability of a system to confirm occurrence/non-occurrence of an action	NR
Privacy	Ensuring that the system obeys privacy policies and enabling individuals to control their personal information	P

[13], may have life-threatening outcomes. IoT availability is essential for providing a fully-functioning Internet-connected environment. It ensures that devices are available for collecting data and prevents service interruptions. The insufficiency of the CIA-triad in the context of security has been addressed before [14]–[16]. Cherdantseva et al. show that the CIA-triad does not address new threats that emerge in a collaborative security environment [14]. They provide a comprehensive list of security requirements by analyzing and examining a variety of information, assurance, and security literature. This list is called the IAS-octave and is proposed as an extension to CIA-triad. Table I summarizes the security requirements in the IAS-octave, and provides their definitions and abbreviations. In the rest of this paper, we will also target IAS-octave requirements. We define:

- *Secure thing*: A thing that meets all of the above-mentioned security requirements.
- *Security attack*: An attack that threatens at least one of the above-mentioned security requirements.

IV. Vulnerabilities of IoT

This section provides an in-depth analysis of possible attacks and vulnerabilities at each level of the edge-side layer (edge nodes, communication, and edge computing). Fig. 3 summarizes several attacks and their countermeasures that are discussed in this work. We describe the left side of this figure (attacks) in this section. The security requirements and their abbreviations that are used in Fig. 3 are given in Table I.

A. Edge nodes

In this section, we discuss various attacks against the first level of the reference model that includes computing nodes and RFID tags.

1) Edge computing nodes: We begin with attacks against the edge computing nodes, e.g., RFID readers, sensor nodes, and compact controlling nodes.

Hardware Trojan: Hardware Trojans have emerged as a major security concern for integrated circuits [17]–[21]. Hardware Trojan is a malicious modification of an integrated circuit, which enables the attacker to use the circuit or to exploit its functionality to obtain access to data or software running on the integrated circuits (ICs) [22]. In order to insert a hardware Trojan in the original circuitry, the attacker maliciously alters the design before/during fabrication and specifies a triggering mechanism that activates the malicious behavior of the Trojan [17]. Trojans are generally divided into two categories based on their triggering mechanisms [21], [22]: (i) externally-activated Trojans, which can be triggered by an antenna or a sensor that can interact with the outside world, and (ii) internally-activated Trojans that are activated after a certain condition is met inside the integrated circuit, e.g., a Trojan that wakes up after a specific timespan when it receives a triggering signal from a countdown circuitry added by the attacker.

Non-network side-channel attacks: Each node may reveal critical information under normal operation, even when not using any wireless communication to transmit data. For example, the electromagnetic (EM) signature, i.e., the EM waves emitted by the node, can provide valuable information about the status of the device. The declassification of TEMPEST documents [23] in 2007, and the recent publications of some EM-based attacks [24]–[26] have started to develop the idea of non-network side-channel threats. For example, in a recent work, researchers were able to demonstrate how the acoustic/EM signals leaked from a medical device can provide valuable information about the patient or the device [26]. As mentioned in that work, detection of the existence of known signals or protocols may endanger the safety of the user, e.g., if the user has a device that is very expensive. Moreover, this type of attack may lead to a serious privacy issue in medical systems. For example, consider a subject who wears a medical device indicating a certain medical condition that carries a social stigma. Detecting the presence of this device can embarrass the patient. In addition, specific side-channel information from the devices may provide significant information about the individual's health condition, e.g., glucose level, blood pressure, etc.

Denial of Service (DoS) attacks: There are three well-known types of DoS attacks against edge computing nodes: battery draining, sleep deprivation, and outage attacks. Next, we describe each one.

1. Battery draining: Due to size constraints, nodes usually have to carry small batteries with very limited energy capacity. This has made battery-draining attacks a very powerful attack that may indirectly lead to serious consequences, such as a node outage or a failure to report an emergency. For example, if an attacker can find a way to deplete the battery of a smoke detector, he will be able to disable the fire detection system [27]. Such attacks could destroy a network if recharging the nodes is difficult [28]. An example of a battery-draining attack

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

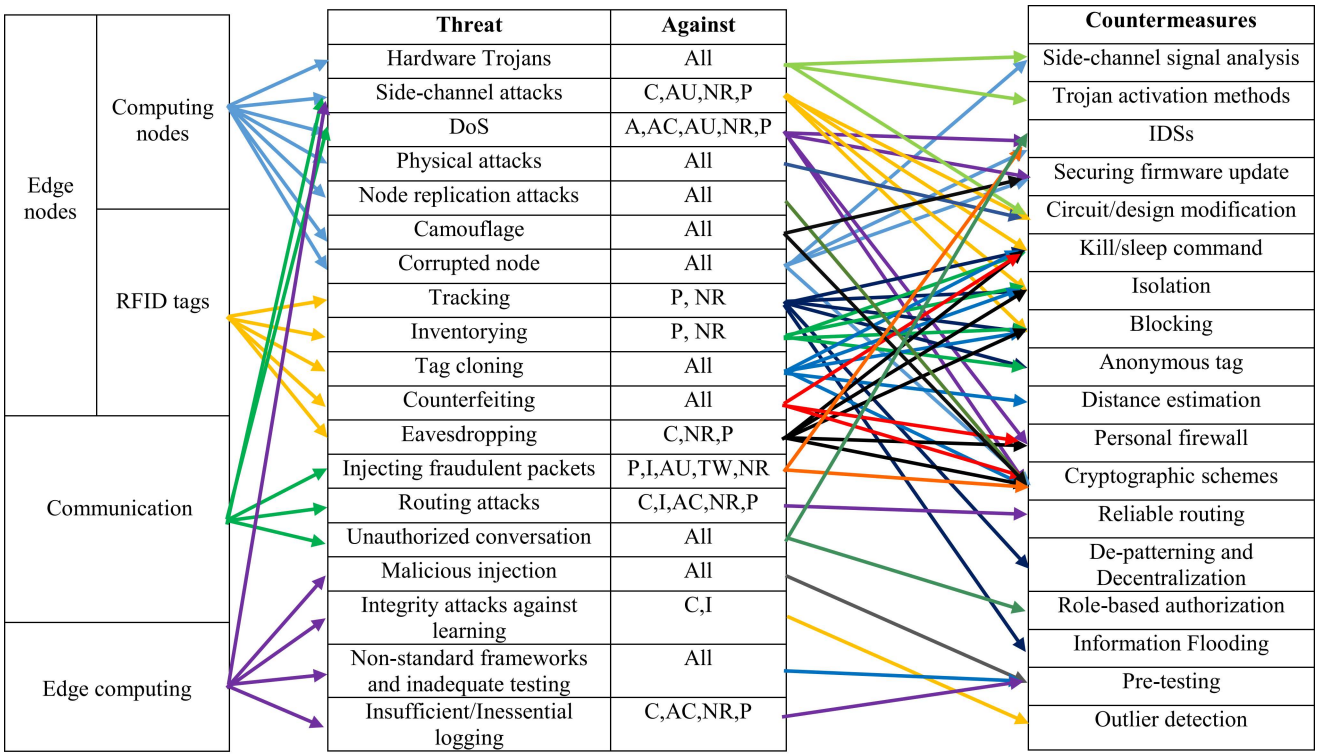


Fig. 3: Summary of attacks and countermeasures

is when an attacker sends tons of random packets to a node and forces the node to run its checking mechanisms, e.g., authentication mechanism. Several battery-draining attacks have been discussed in the literature [29]–[32].

2. *Sleep deprivation*: Sleep deprivation is a specific type of DoS attack in that the victim is a battery-powered node with a limited energy capacity. In this type of attack, the attacker attempts to send an undesired set of requests that seem to be legitimate. Therefore, detection of this type of attack is much harder than that of a simple battery-draining attack. The idea of sleep deprivation was first described by Stajano [33]. The research effort by Martin et al. is one of first publications to closely examine the impact of sleep deprivation attacks on energy-constrained devices [29].

3. *Outage attacks*: Edge node outage occurs when an edge device stops performing its normal operation. In some cases, a set of devices or an administrator device may stop functioning. Outage may be a result of an unintended error in the manufacturing process, battery draining, sleep deprivation, code injection or unauthorized physical access to the node. One of the most famous examples of outage attacks is injecting Stuxnet [34] into Iran's nuclear process control program. Stuxnet manipulates the industrial process control sensor signals such that the infected system loses its ability to detect abnormal behavior. Therefore, the system does not shut down even in an emergency situation [34], [35].

Physical attacks/tampering: Edge devices operate in hostile environments in which physical access to the devices may be possible, thus making them highly vulnerable to hardware/software attacks. The attacker, with a physical access to the device, may extract valuable cryptographic information, tamper with the circuit, modify programming, or change the operating system [36]–[40]. Physical attacks against the edge nodes may cause permanent destruction. Therefore, their main purpose is to extract information for future use, e.g., find the fixed shared key. Such a well-known recent attack was on the Nest thermostat [40], in which the attacker tries to replace the default firmware with a malicious one. This attack enables the attacker to control the thermostat, even when he no longer has physical access to the device.

Node replication attacks: In such an attack, the attacker adds a new node, e.g., a malicious one, to an existing set of nodes by replicating one node's identification number. This attack can lead to a significant reduction in network performance. Moreover, the attacker can easily corrupt or misdirect packets that arrive at the replica [41]. This attack usually causes severe damage to the system by enabling the attacker to obtain required access to extract cryptographic shared keys [42]. Moreover, node replicas may revoke authorized nodes by executing node-revocation protocols [41], [43].

Camouflage: In this type of attack, the attacker inserts a counterfeit edge node or attacks an authorized node in order to hide at the edge level. Afterwards, the modified/counterfeit node can operate as a normal node to obtain, process, send, or redirect packets [42], [44]. Moreover, such a node can function in a passive mode in which it only conducts traffic analysis.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

Corrupted/malicious node: The main goal of corrupting nodes is to gain unauthorized access to the network they belong to. Malicious nodes injected into a network can obtain access to other nodes, possibly controlling the network on behalf of the attacker [42]. A malicious node can also be used by the attacker to inject false data into the system or prevent delivery of true messages [44].

2) RFID tags: Next, we discuss the attacks against RFID tags.

Tracking: Covert reading of RFID tags is a significant threat. Unfortunately, almost all such tags provide a unique identifier. As a result, a nearby unauthorized reader can easily and effectively read a tag that is attached to a product or an individual. Such a reading provides very strong tracking information [45]. In the simplest form of the attack, an attacker uses a large number of RFID readers to read these fixed identifiers. The threat grows and becomes more important when a tag identifier is combined with personal information, e.g., credit/loyalty card number and personal profile [46].

Inventorying: There are certain types of tags that carry valuable information about the products they are attached to. In particular, electronic product code (EPC) tags have two custom fields: a manufacturer code and a product code. As a result, an individual who has an EPC tag is subject to inventorying [47], i.e., a tag reader can examine what products the individual has. This threat leads to serious privacy concerns. For example, the attacker might recognize what types of medical device, e.g., an insulin pump, a patient is wearing and, therefore, what illnesses, e.g., diabetes, he suffers from.

Physical attacks/tampering: This type of attack can be launched when the attacker has full physical access to a tag. In this attack, the tags can be physically manipulated and modified in a laboratory setup [48]. There are several known physical attacks against RFIDs. Among them are probe attacks, material removal, circuit manipulation, and clock glitching [49]. These attacks are used for extracting information from the tag, or modifying the tag for counterfeiting.

Tag cloning: Tag cloning (also referred to as spoofing) and impersonation of RFID tags could be very profitable to hackers, and extremely dangerous for the company's reputation. Potential damage can be amplified through a high level of automation [50]. An attacker may use tag cloning to access restricted areas, bank accounts, or sensitive information.

Counterfeiting: In counterfeiting, the attacker modifies the identity of an item, typically by means of tag manipulation. Generally, the attacker needs less information to launch counterfeiting attacks relative to spoofing attacks. In these attacks, a tag is partially manipulated. Westhues describes how an RF tape-recorder can be constructed to read commercial proximity cards and partially simulate their signals to bypass building security systems [51].

DoS attacks: In DoS attacks, the RF channels are jammed such that the tags cannot be read by the tag readers and, as a result, the intended services based on the RFID tags become unavailable. For example, an attacker can lock down a whole building by jamming all the RFID-based doors. Additional vulnerabilities of RFID authentication protocols to DoS attack have been discussed in [52].

Eavesdropping: In this attack, the main goal of the attacker is to intercept, read, and save messages for future analysis. The intercepted data can be used as an input to other attacks, such as tag cloning. The concept of eavesdropping attacks against RFIDs is not new and is frequently mentioned in the literature. Recent reports by the National Institute of Standards and Technology [53] and the Department of Homeland Security [54], in addition to several published surveys, e.g. [55]–[57], all mention risks of eavesdropping in the RFID environment. In particular, several practical attack scenarios and their experimental setups have been discussed in [57].

Side-channel attacks: Such attacks use state-of-the-art tools to intercept and process communications in order to extract information from various patterns, even when the messages are encrypted. For example, if an attacker reads the tags at the entrance of a building, he can guess the number of individuals in the building at any moment by counting the number of communications. Over-the-air timing attacks against RFID tags and their efficacy are open research problems [55]. Carluccio et al. have described the use of EM emanations to launch a power-analysis attack against RFIDs [58].

B. Communication

Next, attacks against the communication level of the IoT reference model are discussed.

Eavesdropping: At the communication level, eavesdropping (also called sniffing) refers to intentionally listening to private conversations over the communication links [59]. It can provide invaluable information to the attacker when the data are unencrypted. In this situation, usernames and/or passwords are often easy to extract. When packets also carry access control information, such as node configuration, shared network password, and node identifiers, eavesdropping can provide critical information. The attacker can use and process this captured information to design other tailored attacks. For example, if an attacker can successfully extract the information that is required to add a new node to the set of authorized nodes, he will easily be able to add a malicious node to the system.

Side-channel attacks: Although side-channel attacks are not easy-to-implement, they are powerful attacks against encryption. They pose a serious threat to the security and reliability of cryptographic implementations. As mentioned earlier, side-channel attacks can also be launched at the edge node level. In contrast to the attack at the edge node level, the side-channel attacks at the communication level are usually non-invasive. They only extract information that is often unintentionally leaked. Typical examples of unintentional information leakage are time between two consecutive packets, frequency band of communications, and communication modulation. An important characteristic of non-invasive attacks is that they are undetectable, and as a result, there is no easy defense against them except to minimize leakage or else add noise to the leaked information.

DoS attacks: The most common and well-known DoS attack at the communication level is a standard attack that jams the transmission of radio signals. Two types of active jamming attacks have been defined in the literature [60],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

[61]: (i) continuous jamming that involves complete jamming of all transmissions, and (ii) intermittent (also called non-continuous) jamming in which jamming is periodic and, as a result, the nodes can send/receive packets periodically. While the goal of constant jamming is to block all transmissions, with intermittent jamming, the attacker intends to lower the performance of time-sensitive systems. Consider a fire detection system that can detect unusual changes in the level of gases in the environment and calls the fire department in case of an emergency. An attacker can easily make the system unreliable by intermittently jamming node-to-node and node-to-base transmissions. In this scenario, the system will become out-of-service if the attacker uses constant jamming. Several research efforts have examined the possibility and effectiveness of launching DoS attacks against various transmission protocols, including Bluetooth [62], ZigBee [63], and 6LowPan [64]. In addition to active jamming attacks, the attacker can launch DoS against communication using malicious nodes or routers. The attacker may insert a node/router that intentionally violates the communication protocol in order to generate collisions or jam the communications [42]. A malicious router/node may also refuse to route messages or attempt to misdirect them. This could be done intermittently or constantly. Constant DoS attacks are usually easy to detect, whereas detection of intermittent ones requires accurate and efficient monitors.

Injecting fraudulent packets: An attacker can inject fraudulent packets into communication links using three different attack methods: (i) insertion, (ii) manipulation, and (iii) replication (also called replay) [42]. In insertion scenarios, the attacker inserts new packets in network communication. In other words, an insertion attack has the ability to generate and send malicious packets that seem legitimate. Manipulation attacks involve capturing the packet, and then modifying, e.g., updating header information, checksum, and data, and sending the manipulated packet. In replication attacks, the attacker captures the packets that have been previously exchanged between two things in order to replay the same packets. Generally, a stateless system, which does not keep track of previous packets or previous state of the system, is quite vulnerable to replay attacks.

Routing attacks: Attacks that affect how messages are routed are called routing attacks. An attacker may use such attacks to spoof, redirect, misdirect, or drop the packets at the communication level. The simplest type of routing attack is an altering attack in which the attacker changes the routing information, e.g., by generating routing loops or false error messages. In addition to altering attacks, several other serious attacks have been proposed, e.g., Black Hole [65], [66], Gray Hole [66], Worm Hole [67], Hello Flood [68], [69], and Sybil [70]. We briefly describe them next.

1) Black Hole: A Black Hole attack is launched by using a malicious node, which attracts all the traffic in the network by advertising that it has the shortest path to the destination in the network. As a result, all packets are sent to the malicious node, and the attacker can process the packets or simply drop them.
2) Gray Hole: A Gray Hole attack is a variation of Black Hole attack in which the nodes selectively drop some packets.
3) Worm Hole: A Worm Hole attack is a severe attack that can be launched even when authenticity and confidentiality are guaranteed in all communications. In this attack, an attacker first records packets at one location in the network and then tunnels them to a different location.
4) Hello Flood: A Hello Flood attack is based on the fact that a node must broadcast "HELLO PACKETS" to show its presence to neighbors. The receiving nodes may assume that they are within the communication range of the sender. In this attack, an attacker uses a malicious node with high transmission power to send "HELLO PACKETS" to every other node in the network and claim to be their neighbor.
5) Sybil: In a Sybil attack, the attacker adds/uses Sybil nodes, which are nodes with fake identities. Sybil nodes can out-vote honest nodes in the system.

Unauthorized conversation: Every edge node needs to communicate with other nodes in order to share data or access their data. However, each node should only talk to a subset of nodes that need its data. This is an essential requirement for every IoT system, in particular, ones consisting of both insecure and secure nodes. For example, in a smart home scenario, the thermostat requires the smoke detector's data in order to shut down the heating system in an emergency situation. However, if the insecure smoke detector can share (get) information with (from) every other node, an attacker might be able to control the whole home automation system by hacking the smoker detector.

C. Edge computing level

Edge (fog) computing is an emerging technology. Thus, its vulnerabilities have not yet been adequately explored. The few research efforts that address attacks on edge computing mainly focus on possible threats to sensor networks [71], [72]. Next, we discuss and suggest some attack scenarios against an edge computing based scheme. Although some of these attacks were designed to target conventional systems and networks, they are also applicable to the edge computing based systems.

Malicious injection: Insufficient validation of the input may enable malicious input injection. An attacker could inject a malicious input that causes the service providers to perform operations on behalf of the attacker. For example, an attacker may add an unauthorized component to one of the levels below (communication or edge node levels) that is capable of injecting malicious inputs into the servers. Afterwards, the attacker might be able to steal data, compromise database integrity, or bypass authentication. Standard database error messages returned by a database may also assist the attacker. In situations where the attacker has no knowledge of the database's tables, forcing an exception may reveal more details about each table and the names of its fields [73].

Integrity attacks against machine learning: Two types of attacks can be launched against machine learning methods that are used in IoT systems: causative and exploratory [74]. In causative attacks, the attacker changes the training process

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

by manipulating the training dataset, whereas in exploratory attacks, he exploits vulnerabilities without altering the training process. Recent research has introduced a new type of causative attack, called the poisoning attack [75]–[77]. In a poisoning attack, the attacker adds precisely-selected invalid data points to the training dataset. In an edge computing based system, an attacker might be able to launch this attack against the learning algorithm by directly accessing the server or computing nodes, or he might be able to add malicious data to the dataset by adding a sufficient number of malicious nodes to lower levels of the IoT model. The main motivation is to cause the classification algorithm to deviate from learning a valid model by manipulating the dataset.

Side-channel attacks: Earlier, we mentioned several types of side-channel attacks against the components at the edge node and communication levels. In addition, an attacker might use the information leaked from additional components, e.g., service providers and servers, to launch side-channel attacks. For example, a service, which generates verbose fault warnings, provides a useful tool for designers and developers. However, the same warnings can provide extravagant information in operational environments.

Non-standard frameworks and inadequate testing: Non-standard coding flaws can give rise to serious privacy and security concerns. Moreover, since the nodes typically need to connect to intermediate servers, the consequences of a compromise might be amplified. The development of an edge computing based system is a sophisticated process because it requires combining heterogeneous resources and devices that are often made by different manufacturers [78]. In addition, there is neither a generally-accepted framework for the implementation of edge computing based systems nor a standard set of policies. As a result, several privacy and security flaws of these systems may remain undetected.

Insufficient/inessential logging: Logging is a nice approach for detecting an intrusion or a hacking attempt. Developers should log events such as successful/unsuccessful authentication attempts, successful/unsuccessful authorization attempts, and application errors. The edge computing based systems may be damaged as a result of insufficient logging [79]. It is also recommended that the log files be encrypted.

V. Countermeasures

In this section, the right side of Fig. 3 that consists of several countermeasures is discussed. Next, we describe each defense in a level-by-level fashion.

A. Solutions for security issues in edge nodes

Next, we describe countermeasures for addressing attacks against the edge nodes.

1) Computing nodes: We start with solutions for attacks against computing nodes.

Side-channel analysis: Side-channel signal analysis provides an effective approach for the detection of both hardware Trojans and malicious firmware/software installed on a device.

1. Trojan detection: Side-channel signals, including timing [22], [80], [81], power [82]–[84], and spatial temperature [82], [85] can be used for Trojan detection. The presence of a Trojan in a circuit commonly affects power and/or delay characteristics of wires and gates in the circuit, and alters heat distribution on the silicon IC. In order to detect a hardware Trojan, side-channel signal-based Trojan detection mechanisms compare physical characteristics and/or the heat distribution map of a suspicious IC to the ones of a Trojan-free reference IC. Power-based analyses offer an activity monitoring method that can be utilized to detect suspicious activities within the IC, enabling detection of Trojans. Timing-based methods enable the detection of Trojans by testing the IC using efficient delay tests, which are sensitive to small changes in the circuit delay along the affected paths and can differentiate Trojans from process variations. Spatial temperature-based mechanisms rely on infrared imaging techniques, which provide thermal maps of ICs. Silicon is transparent in the infrared spectral region and this transparency enables us to obtain maps of thermal infrared emissions using infrared imaging techniques [85].

2. Malicious firmware/software detection: The effectiveness of side-channel signal analysis in detecting malicious firmware/software installed on a device has been shown by several previous research efforts [86]–[88]. As mentioned earlier in Section IV, side-channel signals can reveal valuable information about the device's operation. Similar to the Trojan detection mechanism, malware detection methods can process side-channel signals to detect abnormal behaviors of the device, e.g., a significant increase in its power consumption, which are the results of a malware installed on the device.

Trojan activation: Trojan activation strategies aim to partially/fully activate the Trojan circuitry to facilitate Trojan detection. Several Trojan activation approaches have been proposed in the last decade [22], [89], [90]. The common goal of such strategies is to magnify and detect the disparity between the behavior, outputs, or side-channel leakages of a Trojan-free circuit and the ones of a Trojan-inserted circuit. For example, Chakraborty et al. proposed MERO [91], an efficient methodology to derive a compact set of test patterns (minimizing test time and cost), while maximizing the Trojan detection coverage. MERO can increase the detection sensitivity of many side-channel Trojan detection. The basic concept is to detect low probability conditions at the internal nodes, select candidate Trojans triggerable by a subset of these rare conditions, and then derive an optimal set of vectors that can trigger each of the selected low probability nodes.

Policy-based mechanisms and intrusion detection systems (IDSs): Policy-based approaches are promising mechanisms for solving security and privacy problems at this IoT level. Violation of essential policies can be detected continuously by introducing an IDS [42]. An IDS ensures that general rules are not broken. It provides a reliable approach to defend against battery-draining and sleep deprivation attacks by detecting unusual requests to the node. Several recent and ongoing research efforts provide efficient IDS designs for monitoring the edge nodes and detecting potential threats [92]–[96].

Circuit modification: Changing the circuit is one of the most effective defenses against physical, side-channel, and Trojan attacks. In the following, for each of these attacks, we briefly discuss how specific circuit changes and modifications may

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

address/prevent the attack.

1. Tamper proofing and self-destruction: Nodes may be integrated with physical hardware that enhances protection against physical attacks. For example, to protect against tampering of sensors, several mechanical/electrical tamper-proofing methods for designing the physical packages of the nodes have been proposed and have traditionally been used in home automation sensors, e.g., smoke detectors. Moreover, using self-destruction mechanisms provides an alternative approach to defend against physical attacks [97].

2. Minimizing information leakage: There are also some well-known approaches for addressing side-channel attacks including, but not limited to, adding randomized delay [58] or intentionally-generated noise [98], balancing Hamming weights [99], using constant execution path code [99], improving the cache architecture [100], and shielding [26].

3. Integrating Physically Unclonable Function (PUF) into the circuity: A PUF is a noisy function embedded into an integrated circuit [101]. When queried with a challenge x, a PUF generates a response y that depends on both x and the unique intrinsic physical properties of the device [102], [103]. PUFs are assumed to be physically unclonable, unpredictable, and tamper-evident. PUFs enable unique device identification and authentication [102], [104], and offer Trojan detection mechanisms [22]. Any unintended modification of the circuit physical layout changes the circuit parasitic parameters that can be detected by Trojan detection methods.

Securing firmware update: Each firmware update can be launched remotely or directly. In the case of remote firmware update, the base or server broadcasts a command (CMD) to announce that there is a new version of firmware available. Then, a node with the new firmware broadcasts an advertisement (ADV) to neighboring nodes. The nodes that are willing to update their firmware and have also received ADV compare the new version with their existing version, and send requests (REQ) if they need an update. Eventually, the advertiser starts sending data to the requesters. Providing a secure method for remotely updating the firmware requires authentication of CMD, ADV, REQ, and data packets. Moreover, risks posed by DoS attacks during each step of the protocol should be considered [105]. In addition to remote firmware updates, some nodes support direct updates of the firmware, e.g., using a USB cable. In this case, the integrity of the firmware should be checked, and the user, who tries to update the firmware, should be authenticated, because a lack of sufficient integrity check mechanisms may enable an attacker to replace legitimate device firmware with a malicious one [40].

2) RFID tags: Next, we describe solutions and suggestions for addressing attacks against RFID tags.

Kill/sleep command: A kill scenario is built into the manufacturing process of RFID tags. An RFID tag has a unique PIN, e.g., a 32-bit password. Upon receiving the correct PIN from the RFID reader, the tag can be killed, i.e., the tag will not be able to transmit any further information after receiving this command [56]. There is an alternative approach called a sleep command that puts the tags to sleep, i.e., makes them inactive for a period of time [55]. Although these ideas seem simple at first glance, designing and implementing secure and effective PIN management schemes need sophisticated techniques.

Isolation: A very effective way of protecting the privacy of tags is to isolate them from all EM waves. One way is to build and use isolation rooms. However, building such rooms is usually very expensive. An alternative approach is to use an isolation container that is usually made of a metal mesh [55]. This container, which can block EM waves of certain frequencies, is called a Faraday cage [106]. Another approach to is to jam all nearby radio channels using an active RF jammer which continuously interrupts specific RF channels.

Blocking: Juels et al. proposed a protection scheme called blocking [45]. It adds a modifiable bit to the tag that is called a privacy bit. A '0' privacy bit indicates that public scanning is allowed for the tag, whereas a '1' bit marks the tag as private. This scheme requires a certain type of tag (called blocker tag), which is a special RFID tag that prevents unintended scanning. However, the idea of using blocker tags has two main limitations: (i) it requires the use of a modified version of RFID tags, and (ii) unreliable transmission of the tags may easily lead to privacy failure even when the blocking scheme is implemented. Another blocking approach, called soft blocking, has been proposed in [107]. It relies on auditing of reader configurations to enforce a set of policies that is defined in software. This set guarantees that readers can only read public tags. Then, a monitoring device can passively examine if a reader is violating tag policies.

Anonymous tag: A novel idea based on look-up table mapping has been proposed by Kinoshita [108]. The key contribution of his work is a scheme to store a mapping between an anonymous ID and a real ID of each tag in such a way that a attacker cannot find the mapping algorithm to recover the real ID from the anonymous one. The mapping may represent a key encryption algorithm or a random value mapped to the real ID. Note that although the anonymous ID emitted by an RFID tag has no intrinsic valuable information, it can still enable tracking as long as the ID is fixed over time [56]. In order to address the tracking problem, the anonymous ID should be re-issued frequently.

Distance estimation: Use of signal-to-noise ratio as a metric to determine the distance between a reader and a tag is proposed in [109]. For the first time, Fishkin et al. claim that it is possible to derive a metric to estimate the distance of a reader that tries to read the tag information. This enables the tag to only provide distance-based information. For example, the tag might release general information, e.g., the product type, when scanned at 10 meters distance, but release its unique identifier at less than 1 meter distance.

Personal firewall: A personal RFID firewall [110] examines all readers' requests to tags. The firewall can be assumed to be implemented in a device that supports high computation needs and provides enough storage capacity, e.g., a cellphone. The firewall enables the setting of sophisticated policies. For example, "my tag should not release my personal information when I am not within 50 meters of my work place".

Crypthographic schemes: Three types of cryptographic schemes are widely discussed in the previous literature to address the security attacks against RFID tags:

1. Encryption: Full encryption usually requires significant

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

hardware. Therefore, its implementation in RFID tags has not been feasible due to the need for the tags to be low-cost (a few cents). Feldhofer [111] proposed an authentication mechanism based on the Advanced Encryption Standard (AES). However, for a standard implementation of AES, 20-30K gates are typically needed [112], whereas RFID tags can only store hundreds of bits and support 5-10K logic gates. The limitations arising from gate count and cost suggest that the tag can only devote 250-3500 gates to the security mechanism. The traditional implementation of AES was not appropriate until Jung et al. proposed a novel implementation of AES that requires only 3595 logical gates [113]. However, no fully-developed version of AES has been implemented in any RFID tag.

2. Hash-based schemes: Such schemes are widely used for addressing security issues in the RFID technology. Recent research on hash functions can be found in [114]–[118]. A simple security mechanism based on hash functions is proposed in [46]. In this work, two states are defined for each tag: (i) locked state in which a tag responds to all queries with its hashed key, and (ii) unlocked state in which the tag carries out its normal operation. To unlock a tag, the reader sends a request, including the hashed key, to a back-end database and waits to get the key. After getting the key, the reader sends the key to the locked tag. Then, the tag changes its state to unlocked. Although this significantly improves RFID security, the problem of tracking still remains. To address this issue, Weis et al. [46] propose a more sophisticated scheme, in which the hashed key is changed in a manner that is unpredictable.

3. Lightweight cryptographic protocols: In order to address the security and privacy issues of RFID tags by taking into account their cost requirements, several lightweight cryptographic protocols have been suggested. For example, Peris et al. propose a minimalist lightweight mutual authentication protocol for low-cost RFID tags [112]. They claim that their method provides an adequate security level for certain applications, and can be implemented with only slightly more than 300 gates, which is quite acceptable even for the most limited RFID tags. Moreover, a simple scheme for mutual authentication between tags and readers is proposed by Molnar et al. [119]. Their protocol uses a shared secret and a pseudorandom function to protect the messages exchanged between the tag and the reader. Another example is extremely-lightweight challenge-response authentication protocols described in [120]. These protocols can be used in authenticating tags, but can be broken by a powerful adversary [56].

Circuit modification: In addition to the previously-mentioned applications of PUF (device identification/authentication and hardware Trojan detection), several research efforts have proposed different anti-counterfeiting mechanisms to prevent RFID tag cloning by integrating PUFs into RFID tags [121]–[124]. For example, consider an authentication mechanism that aims to identify the user based on his RFID tag. It can generate a set of challenge-response pairs for each tag during the enrollment phase and store it in a database. At a later point in time, during verification, it can compare the response provided by the user's PUF-based RFID tag for a chosen challenge from the database with the corresponding response in the database [123], [124].

B. Solutions for security issues in communication

In this section, we discuss solutions for addressing the security issues that exist at the communication level of the reference model.

Reliable routing: An essential characteristic of IoT networks that complicates implementation of secure routing protocols is that intermediate nodes or servers might require direct access to message content before forwarding it. As mentioned earlier, several valid attacks against routing have been proposed in the literature. Karlof et al. have addressed most major attack scenarios [125]. They provide the first detailed security analysis of major routing protocols and practical attacks against them, along with countermeasures. Various other research efforts have also tried to address security and privacy concerns in routing [126]–[129].

IDS: IDS is essentially needed at the communication level as a second line of defense to monitor network operations and communication links, and raise an alert in case of any anomaly, e.g., when a pre-defined policy is ignored. Traditional IDS approaches [130]–[132] are usually customized for WSNs or for the traditional Internet. However, few recent IDS proposals address the security and privacy concerns of IoT directly. SVELTE [133] is one of the first IDSs designed to meet the requirements of the IPv6-connected nodes of IoT. It is capable of detecting routing attacks, such as spoofed or altered information, and Black Hole attack. Another intrusion detection method for the IoT has been proposed in [134].

Cryptographic schemes: Using cryptographic schemes, e.g., strong encryption, to secure communication protocols is one of the most effective defenses against a variety of attacks, including eavesdropping and simple routing attacks, at the communication level. Several encryption methods have been proposed to address security issues in communication [135], [136]. The encryption-decryption techniques, developed for traditional wired networks, are not directly applicable to most IoT components, in particular, to small battery-powered edge nodes. Edge nodes are usually tiny sensors that have limited battery capacity, processing power, and memory. Using encryption increases memory usage, energy consumption, delay, and packet loss [137]. Variants of AES have yielded promising results for providing secure communication in IoT. Moreover, different lightweight encryption methods have been proposed, e.g., CLEFIA [138] and PRESENT [139]. Unfortunately, at this time, there are no promising public key encryption methods that provide enough security while meeting lightweight requirements [137].

De-patterning and decentralization: De-patterning and decentralization are two of the major methods proposed to provide anonymity and defense against side-channel attacks. There is always a trade-off between anonymity and the need to share information. De-patterning data transmissions can protect the system against side-channel attacks, e.g., traffic analysis, by inserting fake packets that can significantly alter the traffic pattern, when required. An alternative method for ensuring anonymity is distribution of sensitive data through a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2016.2606384, IEEE Transactions on Emerging Topics in Computing

spanning tree such that no node has a complete view of the original data. This method is called decentralization [140].

Role-based authorization: In order to prevent a respone to requests by intruders or malicious nodes in the system, a role-based authorization system verifies if a component, e.g., edge node, service provider, or router, can access, share, or modify the information. Moreover, for every communication, the authorization system should check whether the two parties involved in the action have been validated and have required authority [141].

Information flooding: Ozturk et al. propose flooding based anti-traffic analysis mechanisms to prevent an external attacker from tracking the location of a data source, since that information may release the location of things [142]. They have proposed three different approaches to flooding: (i) baseline, (ii) probabilistic, and (iii) phantom. In baseline flooding, every node in the network forwards a packet once and only once. In probabilistic flooding, only a subset of nodes within the entire network contributes to data forwarding and the others discard the messages they receive. In phantom flooding, when the source sends a message, the message unicasts in a random fashion (referred to as a random walk phase). Then, the message is flooded using the baseline flooding technique (referred to as the flooding phase).

C. Solutions for security issues at the edge computing level

In this section, we describe countermeasures and solutions for addressing the security attacks and issues at the edge computing level.

Pre-testing: Testing of updates and design implementations is important before they can be used in a critical system [143]. The behavior of the whole system and its components, e.g., routers, edge nodes, servers, etc., should be closely examined by feeding different inputs to the system and monitoring the outputs. In particular, pre-testing attempts to identify the set of possible attack scenarios and simulate these scenarios to see how the system responds [144]. It also specifies what information should be logged and what information is too sensitive to be stored. In addition, the input files should be closely examined to prevent the danger of malicious injection. For example, the attacker should not be able to execute any command by injecting it into the input files.

Outlier detection: The common goal of almost all defenses against integrity attacks on machine learning methods is to reduce the influence of adding invalid data points to the result. These invalid data points are deemed outliers in the training set. Rubinstein et al. have designed a defense framework against poisoning attacks based on robust statistics to alleviate the effect of poisoning [145]. In addition, a bagging defense against such integrity attacks has been proposed by Biggio et al. [146]. They examine the effectiveness of using bagging, i.e., a machine learning method that generates multiple versions of a predictor and utilizes them to get an aggregated predictor by getting averages over the versions or using a plurality vote [147], in reducing the influence of outlying observations on training data. Mozaffari-Kermani et al. have presented several countermeasures against poisoning attacks in the area of healthcare [148]. They have evaluated the effectiveness of their schemes and identified the machine learning algorithms that are easiest to defend.

IDS: IDSs can detect the existence of a malicious node that tries to inject invalid information into the system or violate the policies. Several recent research efforts have proposed IDS based methods to address the injection issue [149]–[152]. For example, Son et al. describe the design and implementation of DIGLOSSIA [149], a new tool that precisely and efficiently detects code injection attacks on servers.

VI. Emerging Challenges

So far, we have summarized several attacks against security of things/individuals along with countermeasures against the attacks. Next, we discuss two emerging security challenges not yet explained in detail in previous literature.

A. Exponential increase in the number of weak links

The majority of IoT-based services rely on compact battery-powered devices with limited storage and computation resources. Due to the special characteristics of these devices and cost factors considered important by manufacturers, several *already-in-market devices* do not support highly-secure cryptographic protocols. This has led to the emergence of an enormous number of weak links in the network/system that can be exploited by an attacker to target other presumably-secure entities in the network. A few research efforts [153], [154] have recently demonstrated the possibility of targeting weak edge nodes to extract the home user's WiFi password. Chapman [153] has demonstrated how Internet-connected light bulbs can reveal the user's WiFi password to the attacker. In [154], a similar attack is discussed, which extracts the WiFi password by targeting the user's smart lock. The endless variety of IoT applications magnifies the impact of these weak edge nodes.

B. Unexpected uses of data

The widespread use of ubiquitous computing enabled by IoT technologies has led to the pervasive deployment of Internet-connected sensors in modern day living. In recent years, a few research efforts have attempted to shed light on unexpected uses of different types of environment-/user-related data collected by Internet-connected sensors [155]–[158]. For example, McKenna et al. have provided a list of privacy-sensitive information, e.g., number of residents, personal habits, and daily routines, that can be inferred from smart homes' electricity load data collected by smart meters [155]. Despite the existence of previous research efforts, the extent of private information that can be inferred from presumably non-critical data is neither well-known nor well-understood.

VII. Conclusion

The emergence of the IoT paradigm in the last decade has led (and will continue to lead) to several threats and possible attacks against security or privacy of things or individuals. Unfortunately, the security threats are not well-recognized in

the domain of IoT. This survey attempted to summarize several IoT security attacks or concerns and countermeasures against them in a level-by-level fashion. The main objective of this paper was to give the reader an opportunity to explore which threats have been launched, how they have been addressed, and which threats still remain. Given the wide applicability of IoT, these threats should be addressed proactively and aggressively by industrial/academic research communities as well as manufacturers.

REFERENCES

[1] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in *Proc. IEEE World Forum on Internet of Things*, 2014, pp. 287–292.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things architecture, possible applications and key challenges," in *Proc. IEEE 10th Int. Conf. Frontiers of Information Technology*, 2012, pp. 257–260.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[5] "The Internet of Things reference model." CISCO, 2014. [Online]. Available: http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf

[6] M. Mozaffari-Kermani, M. Zhang, A. Raghunathan, and N. K. Jha, "Emerging frontiers in embedded security," in *Proc. IEEE Int. Conf. VLSI Design*, 2013, pp. 203–208.

[7] K. Su, J. Li, and H. Fu, "Smart city and the applications," in *Proc. IEEE Int. Conf. Electronics, Communications and Control*, 2011, pp. 1028–1031.

[8] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE J. Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.

[9] E. Fleisch, "What is the Internet of Things? An economic perspective," *Economics, Management, and Financial Markets*, no. 2, pp. 125–157, 2010.

[10] M. Tajima, "Strategic value of RFID in supply chain management," *J. Purchasing and Supply Management*, vol. 13, no. 4, pp. 261–273, 2007.

[11] M. Zhang, A. Raghunathan, and N. K. Jha, "Trustworthiness of medical devices and body area networks," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1174–1188, 2014.

[12] C. Li, A. Raghunathan, and N. K. Jha, "Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system," in *Proc. IEEE 13th Int. Conf. e-Health Networking Applications and Services*, 2011, pp. 150–156.

[13] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Proc. IEEE Symp. Security and Privacy*, 2008, pp. 129–142.

[14] Y. Cherdantseva and J. Hilton, "A reference model of information assurance & security," in *Proc. IEEE 8th Int. Conf. Availability, Reliability and Security*, 2013, pp. 546–555.

[15] D. B. Parker, *Fighting Computer Crime*. Scribner New York, NY, 1983.

[16] M. Whitman and H. Mattord, *Principles of Information Security*. Cengage Learning, 2011.

[17] S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan, "Hardware Trojan attacks: Threat analysis and countermeasures," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1229–1247, 2014.

[18] H. Salmani and M. M. Tehranipoor, "Vulnerability analysis of a circuit layout to hardware Trojan insertion," *IEEE Trans. Information Forensics and Security*, vol. 11, no. 6, pp. 1214–1225, 2016.

[19] T. Wehbe, V. J. Mooney, D. C. Keezer, and N. B. Parham, "A novel approach to detect hardware Trojan attacks on primary data inputs," in *Proc. ACM Wkshp. Embedded Systems Security*, 2015, p. 2.

[20] S. Bhasin and F. Regazzoni, "A survey on hardware Trojan detection techniques," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2015, pp. 2021–2024.

[21] D. M. Shila and V. Venugopal, "Design, implementation and security analysis of hardware Trojan threats in FPGA," in *Proc. IEEE Int. Conf. Communications*, 2014, pp. 719–724.

[22] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.

[23] "NSA TEMPEST Series." [Online]. Available: http://cryptome.org/#NSA-TS

[24] H. Tanaka, "Information leakage via electromagnetic emanations and evaluation of TEMPEST countermeasures," in *Information Systems Security*. Springer, 2007, pp. 167–179.

[25] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," in *Proc. USENIX Security Symposium*, 2009, pp. 1–16.

[26] A. M. Nia, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Physiological information leakage: A new frontier in health information security," *accepted for publication in IEEE Trans. Emerging Topics in Computing*.

[27] A. Brandt and J. Buron, "Home automation routing requirements in low-power and lossy networks." [Online]. Available: https://tools.ietf.org/html/rfc5826

[28] S. Seys and B. Preneel, "Authenticated and efficient key management for wireless ad-hoc networks," in *Proc. 24th Symp. Information Theory in the Benelux*, 2003, pp. 195–202.

[29] T. Martin, M. Hsiao, D. Ha, and J. Krishnaswami, "Denial-of-service attacks on battery-powered mobile computers," in *Proc. IEEE 2nd Conf. Pervasive Computing and Communications*, 2004, pp. 309–318.

[30] M. Khouzani and S. Sarkar, "Maximum damage battery depletion attack in mobile sensor networks," *IEEE Trans. Automatic Control*, vol. 56, no. 10, pp. 2358–2368, 2011.

[31] A. Agah and S. K. Das, "Preventing DoS attacks in wireless sensor networks: A repeated game theory approach," *Int. J. Network Security*, vol. 5, no. 2, pp. 145–153, 2007.

[32] E. Y. Vasserman and N. Hopper, "Vampire attacks: Draining life from wireless ad-hoc sensor networks," *IEEE Trans. Mobile Computing*, vol. 12, no. 2, pp. 318–332, 2013.

[33] F. Stajano, "The resurrecting duckling," in *Proc. Security Protocols*. Springer, 2000, pp. 183–194.

[34] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, "Stuxnet under the microscope," ESET LLC, Tech. Rep., 2011.

[35] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," in *Proc. ACM 6th Symp. Information, Computer and Communications Security*, 2011, pp. 355–366.

[36] X. Wang, S. Chellappan, W. Gu, W. Yu, and D. Xuan, "Search-based physical attacks in sensor networks," in *Proc. IEEE 14th Int. Conf. Computer Communications and Networks*, 2005, pp. 489–496.

[37] A. Becher, Z. Benenson, and M. Dornseif, *Tampering with Motes: Real-world Physical Attacks on Wireless Sensor Networks*. Springer, 2006.

[38] R. Anderson and M. Kuhn, "Tamper resistance-a cautionary note," in *Proc. 2nd USENIX Wkshp. Electronic Commerce*, vol. 2, 1996, pp. 1–11.

[39] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's Intranet of Things to a future Internet of Things: A wireless-and mobility-related view," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 44–51, 2010.

[40] G. Hernandez, O. Arias, D. Buentello, and Y. Jin, "Smart Nest thermostat: A smart spy in your home," in *Proc. Black Hat USA*, 2014.

[41] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proc. IEEE Symp. Security and Privacy*, 2005, pp. 49–63.

[42] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: A survey," *Security in Distributed, Grid, Mobile, and Pervasive Computing*, vol. 1, p. 367, 2007.

[43] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. Security and Privacy*, 2003, pp. 197–213.

[44] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *arXiv preprint arXiv:0909.0576*, 2009.

[45] A. Juels, R. L. Rivest, and M. Szydlo, "The blocker tag: Selective blocking of RFID tags for consumer privacy," in *Proc. ACM 10th Conf. Computer and Communications Security*, 2003, pp. 103–111.

[46] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and privacy aspects of low-cost radio frequency identification systems," in *Security in Pervasive Computing*. Springer, 2004, pp. 201–212.

[47] A. Juels, "RFID security and privacy: A research survey," *IEEE J. Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[48] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "RFID systems: A survey on security threats and proposed solutions," in *Proc. Personal Wireless Communications*. Springer, 2006, pp. 159–170.

[49] S. H. Weingart, "Physical security devices for computer subsystems: A survey of attacks and defenses," in *Proc. Cryptographic Hardware and Embedded Systems*. Springer, 2000, pp. 302–317.

[50] M. Lehtonen, D. Ostojic, A. Ilic, and F. Michahelles, "Securing RFID systems by detecting tag cloning," in *Pervasive Computing*. Springer, 2009, pp. 291–308.

[51] J. Westhues, "Hacking the prox card," *RFID: Applications, Security, and Privacy*, pp. 291–300, 2005.

[52] D. N. Duc and K. Kim, "Defending RFID authentication protocols against DoS attacks," *Computer Communications*, vol. 34, no. 3, pp. 384–390, 2011.

[53] T. Karygiannis, B. Eydt, G. Barber, L. Bunn, and T. Phillips, "Guidelines for securing radio frequency identification (RFID) systems," *NIST Special Publication*, vol. 80, pp. 1–154, 2007.

[54] T. Karygiannis, B. Eydt, G. Barber, and L. Bunn, "DHS emerging applications and technology subcommittee." [Online]. Available: http://www.dhs.gov/xlibrary/assets/privacy/privacy_advcom_rpt_rfid_draft.pdf

[55] I. Syamsuddin, T. Dillon, E. Chang, and S. Han, "A survey of RFID authentication protocols based on hash-chain method," in *Proc. IEEE 3rd Int. Conf. Convergence and Hybrid Information Technology*, vol. 2, 2008, pp. 559–564.

[56] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "RFID systems: A survey on security threats and proposed solutions," in *Proc. Personal Wireless Communications*. Springer, 2006, pp. 159–170.

[57] G. Hancke, "Eavesdropping attacks on high-frequency RFID tokens," in *Proc. 4th Wkshp. RFID Security*, 2008, pp. 100–113.

[58] D. Carluccio, K. Lemke, and C. Paar, "Electromagnetic side channel analysis of a contactless smart card: First results." [Online]. Available: http://www.iaik.tu-graz.ac.at/research/krypto/events/index.php

[59] A. Mukherjee, "Physical-layer security in the Internet of Things: Sensing and communication confidentiality under resource constraints," *Proceedings of the IEEE*, vol. 103, no. 10, pp. 1747–1761, 2015.

[60] G. Noubir and G. Lin, "Low-power DoS attacks in data wireless LANs and countermeasures," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 29–30, 2003.

[61] Q. I. Sarhana, "Security attacks and countermeasures for wireless sensor networks: Survey," *Int. J. Current Engineering and Technology*, pp. 628–635, 2013.

[62] A. D. Wood, J. Stankovic, and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15. 4-based wireless networks," in *Proc. IEEE 4th Communications Society Conf. Sensor, Mesh and Ad-Hoc Communications and Networks*, 2007, pp. 60–69.

[63] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, "Short paper: Reactive jamming in wireless networks: How realistic is the threat?" in *Proc. ACM 4th Conf. Wireless Network Security*, 2011.

[64] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56.

[65] Z. Karakehayov, "Using REWARD to detect team black-hole attacks in wireless sensor networks," in *Proc. Wkshp. Real-World Wireless Sensor Networks*, 2005, pp. 20–21.

[66] B. Revathi and D. Geetha, "A survey of cooperative black and gray hole attack in MANET," *Int. J. Computer Science and Management Research*, vol. 1, no. 2, pp. 205–208, 2012.

[67] O. Garcia-Morchon, S. Kumar, R. Struik, S. Keoh, and R. Hummen, "Security considerations in the IP-based Internet of Things." [Online]. Available: https://tools.ietf.org/html/draft-garcia-core-security-04

[68] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based Internet of Things," *Int. J. Distributed Sensor Networks*, vol. 2013, 2013.

[69] V. P. Singh, S. Jain, and J. Singhai, "Hello flood attack and its countermeasures in wireless sensor networks," *Int. J. Computer Science*, vol. 7, no. 3, p. 23, 2010.

[70] J. R. Douceur, "The Sybil attack," in *Peer-to-peer Systems*. Springer, 2002, pp. 251–260.

[71] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. IEEE Federated Conf. Computer Science and Information Systems*, 2014, pp. 1–8.

[72] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *J. Concurrency and Computation: Practice and Experience*, 2015. [Online]. Available: http://dx.doi.org/10.1002/cpe.3485

[73] S. W. Boyd and A. D. Keromytis, "SQLrand: Preventing SQL injection attacks," in *Applied Cryptography and Network Security*. Springer, 2004, pp. 292–302.

[74] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Information, Computer and Communications Security*, 2006, pp. 16–25.

[75] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[76] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. Tygar, "Adversarial machine learning," in *Proc. ACM 4th Wkshp. Security and Artificial Intelligence*, 2011, pp. 43–58.

[77] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-H. Lau, S. Rao, N. Taft, and J. Tygar, "Stealthy poisoning attacks on PCA-based anomaly detectors," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 2, pp. 73–74, 2009.

[78] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the Internet of Things," in *Proc. ACM 2nd SIGCOMM Wkshp. Mobile Cloud Computing*, 2013, pp. 15–20.

[79] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Security Privacy*, vol. 9, no. 2, pp. 50–57, Mar. 2011.

[80] A. Nejat, S. M. H. Shekarian, and M. S. Zamani, "A study on the efficiency of hardware Trojan detection based on path-delay fingerprinting," *Microprocessors and Microsystems*, vol. 38, no. 3, pp. 246–252, 2014.

[81] N. Yoshimizu, "Hardware Trojan detection by symmetry breaking in path delays," in *Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust*, 2014, pp. 107–111.

[82] A. N. Nowroz, K. Hu, F. Koushanfar, and S. Reda, "Novel techniques for high-sensitivity hardware Trojan detection using thermal and power maps," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1792–1805, 2014.

[83] T. Iwase, Y. Nozaki, M. Yoshikawa, and T. Kumaki, "Detection technique for hardware Trojans using machine learning in frequency domain," in *Proc. IEEE 4th Global Conf. Consumer Electronics*. IEEE, 2015, pp. 185–186.

[84] M. Tehranipoor, H. Salmani, and X. Zhang, "Hardware Trojan detection: Untrusted manufactured integrated circuits," in *Integrated Circuit Authentication*. Springer, 2014, pp. 31–38.

[85] K. Hu, A. N. Nowroz, S. Reda, and F. Koushanfar, "High-sensitivity hardware Trojan detection using multimodal characterization," in *Proc. IEEE Design, Automation & Test in Europe Conference & Exhibition*, 2013, pp. 1271–1276.

[86] S. S. Clark, B. Ransford, A. Rahmati, S. Guineau, J. Sorber, W. Xu, and K. Fu, "WattsUpDoc: Power side channels to nonintrusively discover untargeted malware on embedded medical devices," in *Proc. USENIX Wkshp. Health Information Technologies*, 2013.

[87] M. Msgna, K. Markantonakis, D. Naccache, and K. Mayes, "Verifying software integrity in embedded systems: A side channel approach," in *Proc. Int. Wkshp. Constructive Side-Channel Analysis and Secure Design*, 2014, pp. 261–280.

[88] M. Msgna, K. Markantonakis, and K. Mayes, "The B-side of side channel leakage: Control flow security in embedded systems," in *Proc. Int. Conf. Security and Privacy in Communication Systems*, 2013, pp. 288–304.

[89] N. Lesperance, S. Kulkarni, and K.-T. Cheng, "Hardware Trojan detection using exhaustive testing of K-bit subspaces," in *Proc. IEEE Asia and South Pacific Design Automation Conference*, 2015, pp. 755–760.

[90] X. Ye, J. Feng, H. Gong, C. He, and W. Feng, "An anti-Trojans design approach based on activation probability analysis," in *Proc. IEEE Int. Conf. Electron Devices and Solid-State Circuits*, 2015, pp. 443–446.

[91] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "MERO: a statistical approach for hardware Trojan detection," in *Proc. Cryptographic Hardware and Embedded Systems*. Springer, 2009, pp. 396–410.

[92] S. S. Doumit and D. P. Agrawal, "Self-organized criticality and stochastic learning based intrusion detection system for wireless sensor networks," in *Proc. IEEE Conf. Military Communications*, vol. 1, 2003, pp. 609–614.

[93] C.-C. Su, K.-M. Chang, Y.-H. Kuo, and M.-F. Horng, "The new intrusion prevention and detection approaches for clustering-based sensor networks [wireless sensor networks]," in *Proc. IEEE Conf.*

Wireless Communications and Networking, vol. 4, 2005, pp. 1927–1932.

[94] A. Agah, S. K. Das, K. Basu, and M. Asadi, "Intrusion detection in sensor networks: A non-cooperative game approach," in *Proc. IEEE 3rd Int. Symp. Network Computing and Applications*, pp. 343–346.

[95] A. P. R. da Silva, M. H. Martins, B. P. Rocha, A. A. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proc. ACM 1st Int. Wkshp. Quality of Service & Security in Wireless and Mobile Networks*, 2005, pp. 16–23.

[96] M. S. I. Mamun, A. Kabir, M. Hossen, M. Khan, and R. Hayat, "Policy based intrusion detection and response system in hierarchical WSN architecture," *arXiv preprint arXiv:1209.1678*, 2012.

[97] A. D. Wood and J. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[98] M. Zhang and N. K. Jha, "FinFET-based power management for improved DPA resistance with low overhead," *ACM J. Emerging Technologies in Computing Systems*, vol. 7, no. 3, p. 10, 2011.

[99] V. Sundaresan, S. Rammohan, and R. Vemuri, "Defense against side-channel power analysis attacks on microelectronic systems," in *Proc. IEEE National Conf. Aerospace and Electronics*, 2008, pp. 144–150.

[100] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: The case of AES," in *Topics in Cryptology*. Springer, 2006, pp. 1–20.

[101] C. Wachsmann and A.-R. Sadeghi, "Physically Unclonable Functions (PUFs): Applications, models, and future directions," *Synthesis Lectures on Information Security, Privacy, and Trust*, vol. 5, no. 3, pp. 1–91, 2014.

[102] K. Rosenfeld, E. Gavas, and R. Karri, "Sensor physical unclonable functions," in *Proc. IEEE Int. Symp. Hardware-Oriented Security and Trust*, 2010, pp. 112–117.

[103] A. Kanuparthi, R. Karri, and S. Addepalli, "Hardware and embedded security in the context of Internet of Things," in *Proc. Wkshp. Security, Privacy and Dependability for Cyber Vehicles*, 2013, pp. 61–64.

[104] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead," *J. Electronic Testing*, vol. 30, no. 1, pp. 9–23, 2014.

[105] Y. W. Law, Y. Zhang, J. Jin, M. Palaniswami, and P. Havinga, "Secure rateless deluge: Pollution-resistant reprogramming and data dissemination for wireless sensor networks," *EURASIP J. Wireless Communications and Networking*, vol. 2011, p. 5, 2011.

[106] J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (EMA): Measures and counter-measures for smart cards," in *Smart Card Programming and Security*. Springer, 2001, pp. 200–210.

[107] A. Juels and J. Brainard, "Soft blocking: Flexible blocker tags on the cheap," in *Proc. ACM Wkshp. Privacy in the Electronic Society*, 2004, pp. 1–7.

[108] S. Kinoshita, F. Hoshino, T. Komuro, A. Fujimura, and M. Ohkubo, "Low-cost RFID privacy protection scheme," *IPS Journal*, vol. 45, no. 8, pp. 2007–2021, 2004.

[109] A. Juels, "RFID security and privacy: A research survey," *IEEE J. Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[110] M. R. Rieback, B. Crispo, and A. S. Tanenbaum, "RFID guardian: A battery-powered mobile device for RFID privacy management," in *Proc. Information Security and Privacy*. Springer, 2005, pp. 184–194.

[111] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Cryptographic Hardware and Embedded Systems*. Springer, 2004, pp. 357–370.

[112] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "M^2AP: A minimalist mutual-authentication protocol for low-cost RFID tags," in *Ubiquitous Intelligence and Computing*. Springer, 2006, pp. 912–923.

[113] M. Jung, H. Fiedler, and R. Lerch, "8-bit microcontroller system with area efficient AES coprocessor for transponder applications," in *Proc. Ecrypt Wkshp. RFID and Lightweight Crypto*, 2005, pp. 32–43.

[114] E. Y. Choi, S. M. Lee, and D. H. Lee, "Efficient RFID authentication protocol for ubiquitous computing environment," in *Proc. Embedded and Ubiquitous Computing*. Springer, 2005, pp. 945–954.

[115] T. Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks," in *Proc. IEEE 1st Int. Conf. Security and Privacy for Emerging Areas in Communications Networks*, 2005, pp. 59–66.

[116] S. M. Lee, Y. J. Hwang, D. H. Lee, and J. I. Lim, "Efficient authentication for low-cost RFID systems," in *Computational Science and Its Applications*. Springer, 2005, pp. 619–627.

[117] G. Avoine and P. Oechslin, "A scalable and provably secure hash-based RFID protocol," in *Proc. IEEE 3rd Int. Conf. Pervasive Computing and Communications*, 2005, pp. 110–114.

[118] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Hash-chain based forward-secure privacy protection scheme for low-cost RFID," in *Proc. Scandinavian Conference on Information Systems*, vol. 2004, 2004, pp. 719–724.

[119] D. Molnar and D. Wagner, "Privacy and security in library RFID: Issues, practices, and architectures," in *Proc. ACM 11th Conf. Computer and Communications Security*, 2004, pp. 210–219.

[120] I. Vajda and L. Buttyán, "Lightweight authentication protocols for low-cost RFID tags," in *Proc. 2nd Wkshp. Security in Ubiquitous Computing*, 2003.

[121] P. F. Cortese, F. Gemmiti, B. Palazzi, M. Pizzonia, and M. Rimondini, "Efficient and practical authentication of PUF-based RFID tags in supply chains," in *Proc. IEEE Int. Conf. RFID-Technology and Applications*, 2010, pp. 182–188.

[122] D. Moriyama, S. Matsuo, and M. Yung, "PUF-based RFID authentication secure and private under complete memory leakage," *Int. Assoc. Cryptologic Research Cryptology ePrint Archive*, p. 712, 2013.

[123] H.-H. Huang, L.-Y. Yeh, W.-J. Tsaur *et al.*, "Ultra-lightweight mutual authentication and ownership transfer protocol with PUF for Gen2 V2 RFID systems," in *Proc. Int. Conf. Engineers and Computer Scientists*, vol. 2, 2016.

[124] S. Mauw and S. Piramuthu, "A PUF-based authentication protocol to address ticket-switching of RFID-tagged items," in *Proc. Int. Wkshp. Security and Trust Management*, 2012, pp. 209–224.

[125] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad-hoc Networks*, vol. 1, no. 2, pp. 293–315, 2003.

[126] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. B. Royer, "A secure routing protocol for ad-hoc networks," in *Proc. IEEE 10th Int. Conf. Network Protocols*, 2002, pp. 78–87.

[127] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *Proc. SCS Commnication Networks and Distributed Systems Modeling and Simulation Conference*, 2002, pp. 193–204.

[128] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, and M. Rossi, "Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples," in *Proc. IEEE Int. Symp. World of Wireless, Mobile and Multimedia Networks*, 2012, pp. 1–7.

[129] M. Čagalj, S. Čapkun, and J.-P. Hubaux, "Wormhole-based antijamming techniques in sensor networks," *IEEE Trans. Mobile Computing*, vol. 6, no. 1, pp. 100–114, 2007.

[130] A. Abduvaliyev, A.-S. Pathan, J. Zhou, R. Roman, and W.-C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.

[131] Y. Wang, H. Yang, X. Wang, and R. Zhang, "Distributed intrusion detection system based on data fusion method," in *Proc. 5th World Congress on Intelligent Control and Automation*, June 2004, pp. 4331–4334.

[132] A. P. R. da Silva, M. H. Martins, B. P. Rocha, A. A. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proc. ACM 1st Int. Wkshp. Quality of Service & Security in Wireless and Mobile Networks*, 2005, pp. 16–23.

[133] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad-hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[134] C. Liu, J. Yang, Y. Zhang, R. Chen, and J. Zeng, "Research on immunity-based intrusion detection technology for the Internet of Things," in *Proc. IEEE 7th Int. Conf. Natural Computation*, vol. 1, 2011, pp. 212–216.

[135] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-the Advanced Encryption Standard*. Springer Science & Business Media, 2013.

[136] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," in *Proc. IEEE 38th Symp. Foundations of Computer Science*, 1997, pp. 394–403.

[137] M. Katagi and S. Moriai, "Lightweight cryptography for the Internet of Things," *Sony Corporation*, pp. 7–10, 2008.

[138] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA," in *Proc. Fast Software Encryption*. Springer, 2007, pp. 181–195.

[139] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, *PRESENT: An Ultra-lightweight Block Cipher*. Springer, 2007.

[140] R. Kumar and S. Rajalakshmi, "Mobile sensor cloud computing: Controlling and securing data processing over smart environment through mobile sensor cloud computing," in *Proc. IEEE Int. Conf. Computer Sciences and Applications*, 2013, pp. 687–694.

[141] S. Misra and A. Vaish, "Reputation-based role assignment for role-based access control in wireless sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 281–294, 2011.
[142] C. Ozturk, Y. Zhang, and W. Trappe, "Source-location privacy in energy-constrained sensor network routing," in *Proc. ACM 2nd Wkshp. Security of Ad-hoc and Sensor Networks*, 2004, pp. 88–93.
[143] M. Howard and S. Lipner, *The Security Development Lifecycle*. O'Reilly Media, Incorporated, 2009.
[144] H. Mouratidis and P. Giorgini, "Security attack testing (SAT): Testing the security of information systems at design time," *Information Systems*, vol. 32, no. 8, pp. 1166–1183, 2007.
[145] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-H. Lau, S. Rao, N. Taft, and J. Tygar, "ANTIDOTE: Understanding and defending against poisoning of anomaly detectors," in *Proc. ACM 9th SIGCOMM Conf. Internet Measurement*, 2009, pp. 1–14.
[146] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, "Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," in *Multiple Classifier Systems*. Springer, 2011, pp. 350–359.
[147] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
[148] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, "Systematic poisoning attacks on and defenses for machine learning in healthcare," *IEEE J. Biomedical and Health Informatics*, vol. 19, no. 6, pp. 1893–1905, Nov. 2015.
[149] S. Son, K. S. McKinley, and V. Shmatikov, "Diglossia: Detecting code injection attacks with precision and efficiency," in *Proc. ACM SIGSAC Conf. Computer & Communications Security*, 2013, pp. 1181–1192.
[150] V. Luong, "Intrusion detection and prevention system: SQL-injection attacks," Master's thesis, San Jose State University, 2010.
[151] F. S. Rietta, "Application layer intrusion detection for SQL injection," in *Proc. ACM 44th Southeast Regional Conference*, 2006, pp. 531–536.
[152] W. G. Halfond and A. Orso, "AMNESIA: Analysis and monitoring for NEutralizing SQL-injection attacks," in *Proc. 20th IEEE/ACM Int. Conf. Automated Software Engineering*, 2005, pp. 174–183.
[153] "Use Smart Doorbell to Hack WiFi Password," http://thehackernews.com/2016/01/doorbell-hacking-wifi-pasword.html/, accessed: 2-1-2016.
[154] "Hacking into Internet Connected Light Bulbs," http://www.contextis.com/resources/blog/hacking-internet-connected-light-bulbs/, accessed: 2-1-2016.
[155] E. McKenna, I. Richardson, and M. Thomson, "Smart meter data: Balancing consumer privacy concerns with legitimate applications," *Energy Policy*, vol. 41, pp. 807–814, 2012.
[156] Y. Michalevsky, A. Schulman, G. A. Veerapandian, D. Boneh, and G. Nakibly, "Powerspy: Location tracking using mobile device power analysis," in *Proc. USENIX Security Symposium*, 2015, pp. 785–800.
[157] J. Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang, "Accomplice: Location inference using accelerometers on smartphones," in *Proc. IEEE Int. Conf. Communication Systems and Networks*, 2012, pp. 1–9.
[158] L. Cai and H. Chen, "TouchLogger: Inferring keystrokes on touch screen from smartphone motion," in *in Proc. USENIX Wkshp. Hot Topics in Security*, 2011, pp. 9–9.

Arsalan Mohsen Nia received his B.S. degree in Computer Engineering from Sharif University of Technology, Tehran, Iran, in 2012, and M.A. degree in Electrical Engineering from Princeton, NJ, in 2014. He is currently pursuing a Ph.D. degree in Electrical Engineering at Princeton University, NJ. His research interests include wireless sensor networks, Internet-of-Things, computer security, distributed computing, mobile computing, and machine learning.

Niraj K. Jha (S'85-M'85-SM'93-F'98) received his B.Tech. degree in Electronics and Electrical Communication Engineering from Indian Institute of Technology, Kharagpur, India in 1981, M.S. degree in Electrical Engineering from S.U.N.Y. at Stony Brook, NY in 1982, and Ph.D. degree in Electrical Engineering from University of Illinois at Urbana-Champaign, IL in 1985. He is a Professor of Electrical Engineering at Princeton University.

He is a Fellow of IEEE and ACM. He received the Distinguished Alumnus Award from I.I.T., Kharagpur in 2014. He is the recipient of the AT&T Foundation Award and NEC Preceptorship Award for research excellence, NCR Award for teaching excellence, and Princeton University Graduate Mentoring Award.

He has served as the Editor-in-Chief of IEEE Transactions on VLSI Systems and an Associate Editor of IEEE Transactions on Circuits and Systems I and II, IEEE Transactions on VLSI Systems, IEEE Transactions on Computer-Aided Design, and Journal of Electronic Testing: Theory and Applications. He is currently serving as an Associate Editor of IEEE Transactions on Computers, Journal of Low Power Electronics, and Journal of Nanotechnology. He has also served as the Program Chairman of the 1992 Workshop on Fault-Tolerant Parallel and Distributed Systems, the 2004 International Conference on Embedded and Ubiquitous Computing, and the 2010 International Conference on VLSI Design. He has served as the Director of the Center for Embedded System-on-a-chip Design funded by New Jersey Commission on Science and Technology. He has served on the program committees of more than 150 conferences and workshops.

He has co-authored or co-edited five books titled Testing and Reliable Design of CMOS Circuits (Kluwer, 1990), High-Level Power Analysis and Optimization (Kluwer, 1998), Testing of Digital Systems (Cambridge University Press, 2003), Switching and Finite Automata Theory, 3rd edition (Cambridge University Press, 2009), and Nanoelectronic Circuit Design (Springer, 2010). He has also authored 15 book chapters. He has authored or co-authored more than 410 technical papers. He has coauthored 14 papers, which have won various awards. These include the Best Paper Award at ICCD'93, FTCS'97, ICVLSID'98, DAC'99, PDCS'02, ICVLSID'03, CODES'06, ICCD'09, and CLOUD'10. A paper of his was selected for "The Best of ICCAD: A collection of the best IEEE International Conference on Computer-Aided Design papers of the past 20 years," two papers by IEEE Micro Magazine as one of the top picks from the 2005 and 2007 Computer Architecture conferences, and two others as being among the most influential papers of the last 10 years at IEEE Design Automation and Test in Europe Conference. He has co-authored another six papers that have been nominated for best paper awards. He has received 16 U.S. patents.

His research interests include FinFETs, low power hardware/software design, computer-aided design of integrated circuits and systems, digital system testing, quantum computing and secure computing. He has given several keynote speeches in the area of nanoelectronic design and test.