

Article

A Novel Methodology for Securing IoT Objects Based on their Security Level Certificates

Hezam Akram Abdulghani , Dimitri Konstantas, Niels Alexander Nijdam

Geneva School of Economics and Management, Geneva University, Switzerland;
{mohammed.akram | dimitri.konstantas | niels.nijdam}@unige.ch

* Correspondence: mohammed.akram

Abstract: [Internet of Things \(IoT\)](#) provides a huge business value for customers, organizations, and governments due to the developments of so many applications in different sectors like energy and healthcare. Nevertheless, as a new emerging technology, [IoT](#) faces several security concerns that are more challenging than conventional Internet because of its limited resources as well as its complex ecosystem. Toward this end, we first highlight [IoT](#) security challenges and briefly discuss its security goals like confidentiality and integrity. Second, we discuss the most common attacks against [IoT](#), along with their violated security goals. We also review the existing frameworks of security and privacy guidelines for [IoT](#) and illustrate their shortcomings. Third, we propose a novel framework for securing [IoT](#) objects, the key objective of which is to assign different [Security Level Certificates \(SLCs\)](#) for [IoT](#) objects based on their hardware capabilities and protection measures. Objects with [SLCs](#), therefore, will be able to communicate with each other or with the Internet in a secure manner. The proposed framework is composed of five main phases. In phase 1, we classify [IoT](#) assets into four components: (i) physical objects, (ii) protocols, (iii) data at rest, and (iv) software, which includes [Operating Systems \(OSs\)](#), middlewares, and applications. We also classify [IoT](#) objects into five categories based on their hardware capabilities. In phase 2, we propose security and privacy guidelines for previously mentioned [IoT](#) assets, along with their protection measures. In phase 3, we classify protection measures into five [SLCs](#), and then we assign different [SLCs](#) for [IoT](#) objects. In phase 4: we develop a communication plan between objects based on their [SLCs](#). In phase 5, we propose a four-step method to seamlessly integrate our objects with legacy objects (objects are not developed according to our framework). Fourth, the feasibility and application of this framework are illustrated using smart homes as a case study. Finally, we investigate how our framework would lessen several attacks and threats against [IoT](#) like routing attacks and physical damage. We also provide qualitative arguments to show that this framework could be utilized to solve some of [IoT](#) security challenges such as tight resource constrains. Moreover, we discuss the shortcomings of our suggested framework

Keywords: Internet of Things (IoT); security goals; security guidelines; IoT assets; IoT security level certificates; countermeasures; IoT attacks; secure IoT frameworks

1. Introduction

The concept of [IoT](#) was proposed in 1999 as the consequence of the development of two emerging technologies, namely [Radio Frequency Identification \(RFID\)](#) and [Wireless Sensor Networks \(WSNs\)](#) [1]. The main objective of [IoT](#) is to seamlessly integrate physical-world objects or things into a digital one using already existing infrastructures (e.g, routers, gateways and switches). To this end, several [IoT](#) objects equipped with sensors, actuators, and connectivity protocols have been used in multiple sectors (e.g.,

1. INTRODUCTION

energy and healthcare) to offer a huge business value for customers, organizations and governments [1]. For instance, smart watches, smart homes, and smart phones are examples of IoT applications designed specifically to make customers' life much easier and more productive [2]. However, previously mentioned applications and IoT in general have faced many security and privacy issues, the common example of which are side-channel attacks, unauthorized conversation, routing attacks and unexpected use of IoT data. Such attacks and threats may jeopardize the existence of IoT, if they are left untouched. That said, securing IoT seems to be very complex and confusing compared to traditional Internet because of two primary factors [3,4]. One is IoT objects vary from tiny and lightweight objects (e.g., light bulbs) to powerful objects (e.g., smart phones), the majority of which is being connected to the Internet or each other to achieve specific tasks. It is, therefore, possible to apply traditional security mechanisms (e.g., **Advanced Encryption Standard (AES)**) directly to powerful objects like smart phones. In contrast, tiny objects, for instance light bulbs, may not be able to apply such techniques directly without some modifications due to their limited resources in terms of battery life, memory storage, and computational power [3,5]. To this end, several research proposals have been proposed in this regard, and they can be classified broadly into three categories: (i) gateway-based solutions [6–8], (ii) IoT stack-based solutions [9–13], and (iii) middleware-based solutions[14–18]. Despite the benefits of using such solutions for addressing some IoT security concerns (e.g., secure communication), they suffer from some shortcomings. For instance, using a gateway for securing IoT objects is a matter of compromise. On one hand, it can be used to address some of security issues like updating objects' firmware and providing a secure key management method between IoT objects and the gateway[19]. On the other hand, it introduces a single point of failure in both security and operation. Furthermore, flexibility and scalability will be decreased and hampered, as the development of a new IoT application or IoT object requires some changes to be implemented into the gateway [20].

The other factor is the lack of frameworks that provide widely agreed upon security and privacy guidelines for IoT assets (physical objects, protocols, data at rest, and software proposed in our earlier work [21]), along with their protection measures[3]. Such guidelines and their suitable implementation techniques would pave the road for IoT stakeholders like developers and manufacturers to build secure IoT systems by integrating such guidelines into their systems from the start [4]. In spite of the importance of such frameworks of security and privacy guidelines for IoT to enhance its security and privacy by design according to [4], a few research studies have been proposed in this regard [3,4,22–24].

Nevertheless, framework-based solutions require more efforts not only to address their limitations (e.g., poor implementation see Section 3), but also to go beyond such limitations and contribute to make them more secure and reliable. To this end, we develop a novel framework by which different SLCs will be assigned to IoT objects based on their hardware capabilities and protection measures. Objects in our framework, hence, will be able to protect themselves autonomously (except object with SLC 1 and SLC 2). In other words, this framework will allow IoT objects to protect themselves either independently for object equipped with higher SLCs like SLC 3, SLC 4, and SLC 5 or dependently for objects armed with lower SLCs such as SLC 1 and SLC 2. A detailed explanation of our methodology is presented in Section 4

The main contributions of this work are the following:

1. To highlight IoT security challenges as well as IoT security goals.
2. To investigate the most common popular attacks against IoT, along with their violated security goals such as confidentiality, integrity, and availability.
3. To review the current frameworks of security and privacy guidelines for IoT, and present their limitations.
4. To suggest a secure IoT framework that will assign different SLCs to IoT objects based on their hardware capabilities and implementation techniques in order to be able to communicate securely with each other or the internet.

2. IOT SECURITY CHALLENGES AND GOALS

5. To utilize a smart home as a case study to illustrate the feasibility of our proposed framework.
6. To discuss how our framework can be used to address some of IoT security challenges as well as IoT attacks, and to present its limitations and future work.

The remainder of this paper is structured as follows. In Section 2, we illustrate IoT security challenges and briefly discuss IoT security goals. In Section 3, we review the existing frameworks of security and privacy guidelines for IoT and discuss their limitations. In Section 4, the proposed methodology for securing IoT objects is presented. In section 5, a case study is introduced to illustrate the benefits of our proposed framework. In Section 6, we describe how our framework can mitigate some attacks and threats against IoT and solve some IoT security challenges. We also present limitations as well as future works of our framework.

2. IoT Security Challenges and Goals

In this section, we first discuss IoT security challenges, and then we briefly highlight IoT security goals.

2.1. IoT Security Challenges

In this part, different security challenges which may hinder and threaten IoT to reach its full potential and growth have been investigated. Such challenges include, but not limited to, the lack of a secure IoT development, tight resource constraints, and uncontrolled environments. A summary of the most popular IoT security challenges is presented in Table 1.

Table 1. IoT Security challenges

ID	Name	A brief description
SC1	Lack of a secure development	Both traditional software and IoT systems engineering processes focus primarily on functional requirements. Security, however, was never a primary consideration of the software development process in both traditional and IoT systems, since they concentrate mainly on achieving their functional requirements, leaving security requirements as an after-thought to be addressed at the end of software development [25]. This kind of practice therefore is insufficient, and IoT systems need to integrate security requirements or guidelines from ground up. To this end, the authors in [3] proposed a comprehensive set of security and privacy guidelines for IoT assets, particularly for physical objects and protocols.
SC2	Tight resource constraints	IoT objects may have different hardware constraints in terms of computational power, storage and battery life. Therefore, traditional security mechanisms like AES can be applied directly to some IoT objects like cell phones and tablets due to their hardware capabilities. For example, the authors in [26] stated that window 10 mobile uses the same protection measures (e.g., Windows Hello mechanism) that have been utilized by the Windows 10 OS personal computer to offer protection against new security threats. while for other simpler objects(e.g., presence sensors and smoke detectors) these mechanisms are not applicable[5].
SC3	Designed for specific Tasks	Most of IoT objects have been designed to achieve specific tasks in a particular environment. Hence, it is impractical to build common defensive mechanisms for different IoT objects, operating in heterogeneous environments and offering different functions and services. To this end, the authors in [5] have defined different mitigation techniques for IoT objects based on three main factors: (i) functionality, (ii) attributes, and (iii) capabilities.

2. IOT SECURITY CHALLENGES AND GOALS

SC4	Changes in security requirements	The security requirements for an IoT object can be changed, depending on the status of the overall system to which it participates. For example, let assume that a smart car has several embedded objects. So, deciding which one of these embedded objects is the most essential to secure depends heavily on the status of the car, in our opinion. If the car is moving, the most crucial one is the an anti-lock braking object. In contrast, if the car is not moving, the most important one is a glass break detector object.
SC5	Update mechanisms	Security of IoT objects depends heavily on their update mechanisms. In other words, an IoT object designed to get its updates remotely (e.g., through a server) requires more security mechanisms than the object designed to get its updates locally (e.g, using a USB cable), according to [3]. This is because any object willing to remotely update its firmware in a secure manner must be able first to establish a secure channel with the server, and also be able to check the integrity of a new firmware image. On the other hand, only the authenticity of a person, who will install newly released firmware into the object, must always be checked in the local firmware updates [27].
SC6	Objects' mobility	One of the main attributes of IoT objects is its mobility. Security of an IoT object, in our perspective, depends heavily on its location either static or dynamic. To this point, we do believe that a dynamic object requires more security mechanisms than static one for many reasons. One is that the dynamic object may be connected to unknown objects appearing in different environments. Therefore, according to [28], such object must be armed with different protection measures like an end-to-end security to secure its communications with other objects, tamper-proofing techniques to prevent physical attacks, side-channel analysis to prevent data leakage, and a secure firmware update method. While the static one may always be connected to trusted objects, and its security might be provided by the trusted objects.
SC7	The Importance of IoT objects	Security of an IoT object depends on its importance. For instance, in a WSN, a sink node requires more defensive techniques when compared to sensor nodes, since the sink node is responsible for collecting, aggregating and processing data coming from sensor nodes as well as managing the whole network [28]. To this point, the authors in [29] have stated that the malicious nodes in WSN that persistently emit unwanted signals towards the sink node or a base station could stop or paralyze the whole network.
SC8	Uncontrolled environment	As some of IoT objects may be deployed on remote environments and left unattended, such objects therefore are vulnerable to physical attacks (e.g., malicious modification of Integrated Circuits (ICs) [30]. For example, having placed IoT objects in uncontrolled areas without proper protection measures, an attacker could take such objects to a lab or a home for further analysis in order to discover their security parameters (e.g., private keys).

2.2. IoT Security Goals

In the state of the art, conventional security goals have been divided into three main groups: (i) Confidentiality, (ii) Integrity and (iii) Availability, referred to as the **Confidentiality, Integrity, and Availability triad (CIA-triad)**. Confidentiality achieves through a set of rules that limit access to only authorized objects or users. Integrity, in the context of IoT, is also of paramount importance, as it assures the accuracy and completeness of IoT data. IoT availability is also an indispensable requirement, since it ensures the availability of IoT objects along with their data to its users. In spite of the popularity of

3. RELATED WORK

Table 2. IoT Security goals[4]

Security Requirements	Definition	Abbreviations
Confidentiality	Only authorised objects or users can get access to the data	CONF
Integrity	Data completeness and accuracy is preserved	INTG
Non-repudiation	IoT system can validate the occurrence of any event	NREP
Availability	Ensuring accessibility of an IoT system and its services	AVAL
Privacy	Presence of privacy rules or policies	PRIV
Auditability	Monitoring of the IoT object activity	AUDI
Accountability	End users can take charge of their actions	ACNT
Trustworthiness	Reliability on IoT object identity	TRST

Table 3. The violated security goals by AT1

CONF	INTG	NREP	AVAL	PRIV	AUDI	ACNT	TRST
▲		▲		▲			

CIA-triad, it fails to deal with novel threats appearing in a collaborative environment [31]. Toward this end, the authors in [31] suggest a thorough set of security goals called **Information, Assurance, and Security octave (IAS)** octave, known as the **IAS** octave, by investigating a huge number of information in the literature in terms of security. An overview of the security goals proposed by the **IAS** octave, along with their definitions and abbreviations in link with **IoT** environment is presented in Table 2.

3. Related Work

In this section, we first describe the most popular attacks against **IoT** and investigate their violated security goals. Then, we review the existing frameworks of security and privacy guidelines, along with their shortcomings.

3.1. Attacks against IoT

In this part, we discuss attacks and threats applicable for **IoT** systems and correlate them with **IoT** security goals, identified in Table 2. More specifically, we annotate with '▲' when security goal in question is violated by the described attack.

(AT1) Eavesdropping: Intentionally listening to packets over communication links is called eavesdropping, and it is a powerful attack against communication channels if packets are not encrypted during transmission, according to [3]. The main goal of such attack is to intercept, read, and alter the communication packets. Three security goals, namely CONF, NREP, and PRIV, are affected by this type of attacks (see Table 3). The CONF and PRIV security goals are violated, since the attacker is indirectly revealing some private information by listening to communication channels that are not encrypted nor well protected. The NREP is compromised, as the attacker could recognize a private key of an object or a sender in case of a weak cryptographic algorithm and thus use such key to sign some packets and send them to other objects or recipients without revealing his/her true identity.

(AT2) Physical attacks: Some **IoT** objects may be deployed in hostile areas due to the nature of **IoT**, and such objects, therefore, are susceptible to physical attacks. These attacks include, but not limited to, vandalizing circuit, modifying OS, and extracting valuable cryptographic information. In this type of attack, all security goals are violated (see Table 4), as the attacker directly operates on the **IoT** object.

3. RELATED WORK

Table 4. The violated security goals by AT2

CONF	INTG	NREP	AVAL	PRIV	AUDI	ACNT	TRST
⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠

Table 5. The violated security goals by AT3

CONF	INTG	NREP	AVAL	PRIV	AUDI	ACNT	TRST
⚠	⚠			⚠			

(AT3) Side-channel attacks: As IoT objects execute their normal functions, there is a huge risk that such objects may reveal critical information (e.g., the secret keys). This type of attacks may happen because of the lack of secure techniques of processing and storing IoT data (e.g., storing unencrypted data on IoT objects). It is also worth mentioning that IoT objects may be vulnerable to such attack even when such objects are not equipped with wireless protocols to transmit their data. For example, an electromagnetic wave omitted by an object may reveal sensitive data about both the object and its users, according to [27]. Three security goals (CONF, INTG, and PRIV) are affected by this attack (see Table 5). The CONF and PRIV are violated as the attacker could reveal sensitive data about the object and its users by analyzing its side exposed features such as algorithms and power consumption. Having discovered some security parameters (e.g., encryption keys), the attacker could modify, for instance, the transmitted data. Thus, the INTG is also compromised by this attack.

(AT4) Malicious object insertion: Maliciously adding an object to the existing set of objects by duplicating another object's identification number to either corrupt the packets or misdirect them is the main goal of this attack. Therefore, this type of attack may cause a huge drop in the network performance. Moreover, upon arrival of messages at a replica, an attacker could not only gain access to different security parameters (e.g., encryption keys) but also revoke authorized objects, since the attacker could execute an object-revocation protocol, according to [27]. This attack violates all security goals (see Table 6) as the attacker has capability to misdirect, drop, decrypt, and corrupt the messages.

(AT5) Routing attacks: In [3], the authors illustrate several attacks like Gray hole, sybil, and worm hole designed specifically to target how IoT packets are directed. The consequences of such attacks include, but not limited to, dropping, spoofing, and misdirecting packets. The simplest form of such attacks is known as modifying attack in which routing information is illegally manipulated by an attacker. Apart from modifying attack, several attacks have been identified in the state of the art like sybil, selective forwarding, wormhole gray hole [3], and hello flood. Several security goals (see Table 7) such as CONF, INTG, NREP, PRIV, and ACNT are compromised by this attack. The CONF, INTG and PRIV security goals are violated as the attacker is indirectly capable of disturbing routing path and spoofing packets. ACNT is also affected as the attacker could drop or misdirect some messages.

Table 6. The violated security goals by AT4

CONF	INTG	NREP	AVAL	PRIV	AUDI	ACNT	TRST
⚠	⚠	⚠	⚠	⚠	⚠	⚠	⚠

3. RELATED WORK

Table 7. The violated security goals by AT5

CONF	INTG	NREP	AVAL	PRIV	AUDI	ACNT	TRST
▲	▲	▲		▲		▲	

Table 8. The violated security goals by AT6

CONF	INTG	NREP	AVAL	PRIV	AUDI	ACNT	TRST
▲	▲	▲	▲	▲	▲	▲	▲

(AT6) **Malicious firmware:** Several manufacturers such as Apple and Sony have been using **Over-the-air (OTA)** methods to update their objects which were already being deployed in power grids, smart homes, smart cars, and more. Due to the large number of **IoT** objects that require updates, a trusted server has been used by manufacturers to publish or push newly released updates of their objects. This method, however, is vulnerable to a single point of failure because of **Denial of Service (DoS)** attacks and a huge number of valid update requests sent simultaneously to the server. This attack violates all security goals such as CONF, INTG, and PRIV (see Table 8) as the attacker has full control over **IoT** objects.

3.2. Framework-based solutions

Although the development of a comprehensive set of security and privacy guidelines, covering all **IoT** assets, is nowadays an indispensable requirement for building secure **IoT** systems, a few frameworks equipped with such guidelines have been proposed, which briefly present in the next paragraphs.

In [32], the authors suggest a list of privacy guidelines for **IoT** moddlewares and applications and their data at rest. Such guidelines include, but not limited to, reducing data granularity, blocking repeated queries, and distributing data storage. However, they do not propose guidelines for different **IoT** assets like physical objects (computing nodes and **RFID**), protocols, and **OSs**. Moreover, they do not address attacks and threats against **IoT**, nor do they identify suitable protection measures to implement their guidelines.

In [22], the **Broadband Internet Technical Advisory Group (BITAG)** suggests an abstract list of security and privacy guidelines (e.g., encrypting communications) for some of **IoT** assets (computing nodes, applications, and protocols). That said, **BITAG** neither provides a thorough set of guidelines, nor recognizes proper security mechanisms to carry out its guidelines. Moreover, attacks and threats against **IoT** are left untouched.

In [23], the **Open Web Application Security Project (OWASP)** proposes a list of security and privacy for some **IoT** assets (computing nodes, applications). Nevertheless, the **OWASP** does not identify attacks and threats against **IoT**, nor does it discuss the required security techniques to apply its guidelines.

In [4], the authors propose a comprehensive list of security and privacy guidelines only for **IoT** data at rest, such as searching on encrypted data, ensuring authorized access, encrypting data storage, and minimizing duplicated copies. Moreover, the authors investigate all possible attacks and threats against data at rest and identify a set of protection measures which can be used to implement their guidelines. Moreover, they show the link between their guidelines, attacks, and mitigation techniques.

In [24], the **IoT Security Foundation (IoTTSF)** proposes a complete list of security and privacy guidelines for all **IoT** assets, except **RFID** tags. Nevertheless, **IoTTSF** does not address attacks and threats against **IoT**, nor does it distinguish suitable implementation techniques to accomplish its guidelines.

In [3], the authors propose a comprehensive list of security and privacy guidelines for some **IoT** assets (computing nodes, **RFID**, and protocols), and they also investigate all possible attacks and threats against

4. METHODOLOGY

them. Furthermore, they identify proper protection measures to implement their guidelines. Not only that, they also show the link between their proposed guidelines, attacks, and protection measures.

In [5], the authors first state the importance of defining security requirements for IoT objects based on three factors: (i) functionality, (ii) capabilities, and (iii) characteristics. Then, they investigate security threats as well as vulnerabilities of IoT objects, and more importantly they utilize the classification of IoT objects capabilities into different classes to suggest a list of security requirements suitable for each class.

Table 19 summarizes the recently-published frameworks that suggest several security and privacy guidelines for several IoT assets along with their appropriate implementation techniques.

Shortcomings: It can be noticed that the suggested research proposals presented in Table 19 suffered from one common limitation which is the lack of a list of security and privacy guidelines that cover all IoT assets. Moreover, the authors in [4] have stated that the success of such frameworks of security and privacy guidelines depends heavily on their implementation techniques. Poor implementation of such frameworks, therefore, may lead to develop insecure IoT systems despite of having security and privacy guidelines.

We do believe that framework-based solution is the answer to many security challenges now facing IoT to reach its full potential. This is because such frameworks have suggested a set of security and privacy guidelines along with their protection measures which can be utilized by different IoT stakeholders (e.g, developers and manufacturers) to build secure systems from ground. This kind of practice, for sure, will enhance security and privacy by design for IoT.

4. Methodology

In this section, we will discuss the proposed methodology through which different SLCs will be assigned to IoT objects based on their hardware capabilities as well as their implemented protection measures like Intrusion Detection System (IDS), side channel protection, and secure storage schemes. In our framework, IoT objects equipped with SLC 1 or SLC 2 indicate that they will have weak protection measures (e.g., Data Link Layer Security (DLLS)) and limited hardware resources. Hence, these objects will neither be deployed in unattended areas, nor will be connected directly to the Internet. Such objects will depend heavily on objects with SLC 3 (acting as gateways) to protect them and manage their communication to the Internet. In contrary, IoT objects armed with SLC 3 or SLC 4 or SLC 5 indicate that they will have strong protection mechanisms (e.g., blockchain-based solutions) and powerful hardware resources. These objects, thus, can be deployed in uncontrolled environments and more importantly can be connected directly to the Internet and protect themselves autonomously. Moreover, our framework states the communication plan in which different IoT objects can communicate securely with each other or with the Internet based on their SLCs. The proposed framework consists of five main phases. The main phases are briefly described below, and an overview of the overall methodology is shown in Figure 1.

4.1. Phase 1: classify IoT assets, attacks, countermeasures, and objects

In this phase, we first recognize IoT assets along with their associated attacks and countermeasures, and then we classify IoT objects into different categories based on their hardware resources.

4.1.1. IoT Assets classification

Due to the complexity of IoT ecosystem composed of so many enabler technologies (e.g, WSN and IPv6 networking for Low power Wireless Personal Area Networks (6LoWPAN)), it is essential to recognize precisely IoT assets in order to be able to protect them. To contribute to such objective, many IoT Reference Models (RMs) have been proposed in literature, such as a three-level model [33], a five-level model [34], and a seven-level model [35]. Even though such RMs simplify the complexity of IoT by breaking it into

4. METHODOLOGY

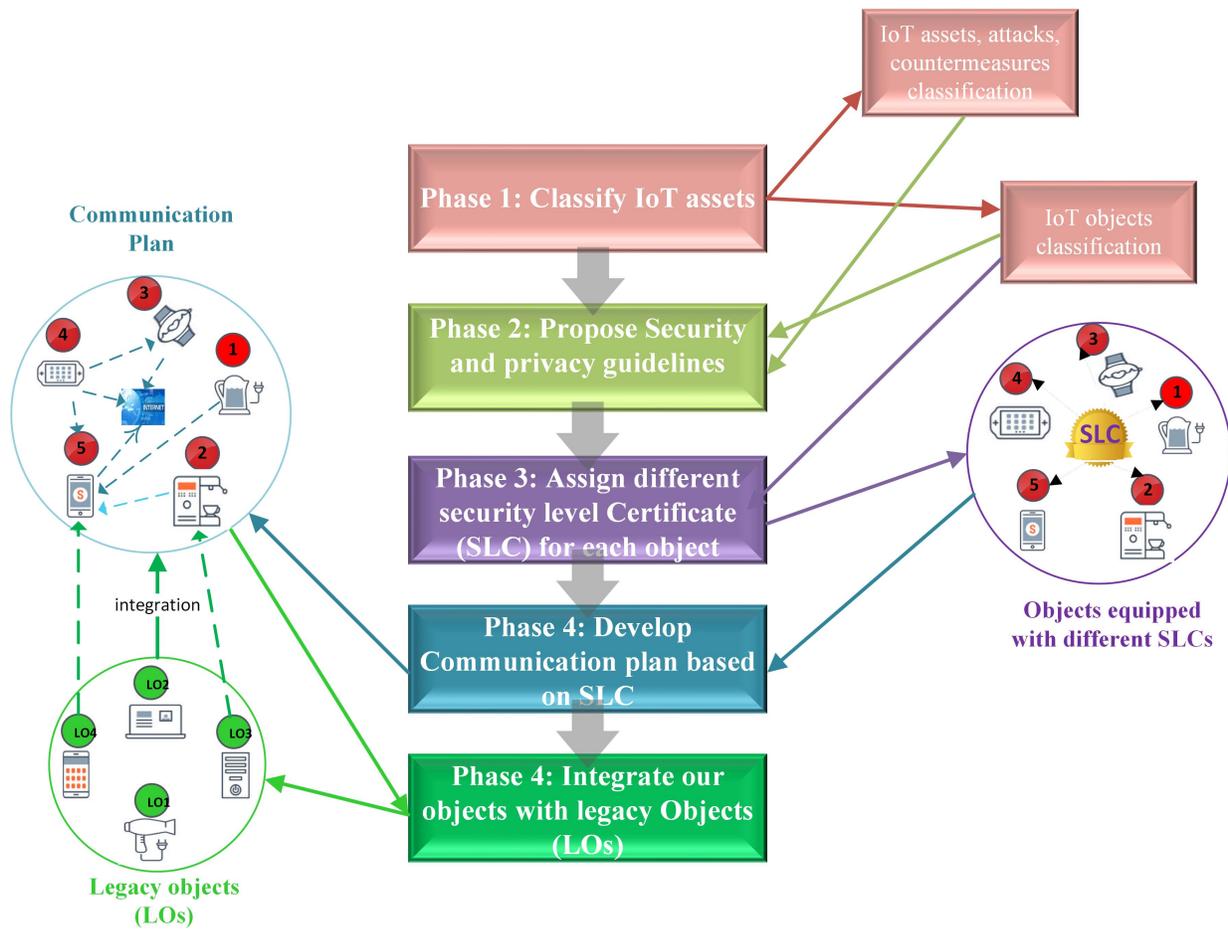


Figure 1. An overview of the proposed methodology

4. METHODOLOGY

Table 9. Classification of IoT objects based on their hardware capabilities

Object Categories	CPU/ Microcontroller	Memory (M)	On-board storage	Power consumption (P)	Example
Category 1	Low CPU like 8- bit Microcontroller 16 MHz	M ≤ 32 KB	None	P ≤ 1W	Arduino Mego
Category 2	Moderate CPU such as 32- bit Microcontroller 80 MHz	32 KB < M ≤ 80 KB	None	P ≤ 1W	NodeMCU ESP-12
Category 3	Single core CPU (e.g., ARM1176 single-core 1 GHz)	80 KB < M ≤ 512 MB	≤ 4GB	1W < P ≤ 2W	Raspberry Pi Zero
Category 4	Quad core CPU (e.g., ARM Cortex-A53 quad-core 1.2 GH)	512 MB < M ≤ 2GB	≤ 8GB	2W < P ≤ 4W	Raspberry Pi 3
Category 5	High(e.g., ARM Cortex-A57 quad-core 2 GHz)	M > 8GB)	High (> 32 GB)	High	NVIDIA Jetson TX2

different layers, they do not address the required building blocks for their layers or levels, which can be used by IoT developers to easily construct their systems. Toward this end, a novel building-blocked RM for IoT was introduced in earlier work in [21], and IoT assets were divided into four main layers (components): (a) physical object layer, (b) communication layer, (c) data at rest layer, and (d) software layer. Physical layer consists of computing nodes (RFID readers and sensor nodes) and RFID tags. Communication layer includes all IoT protocols covering all IoT stack and the existing network infrastructures (e.g., routers and switches). Data at rest layer involves data stored either in IoT objects or on the Cloud Storage Service (CSS). Software layer is composed of IoT middleware, IoT applications, and IoT OSs.

It is worth mentioning that the process of identifying all possible attacks and threats against each IoT asset and also recognizing their suitable protection mechanisms has been investigated in our earlier work in [21]. The whole process can be summarized in Figure 2a, and 2b.

4.1.2. IoT objects classification

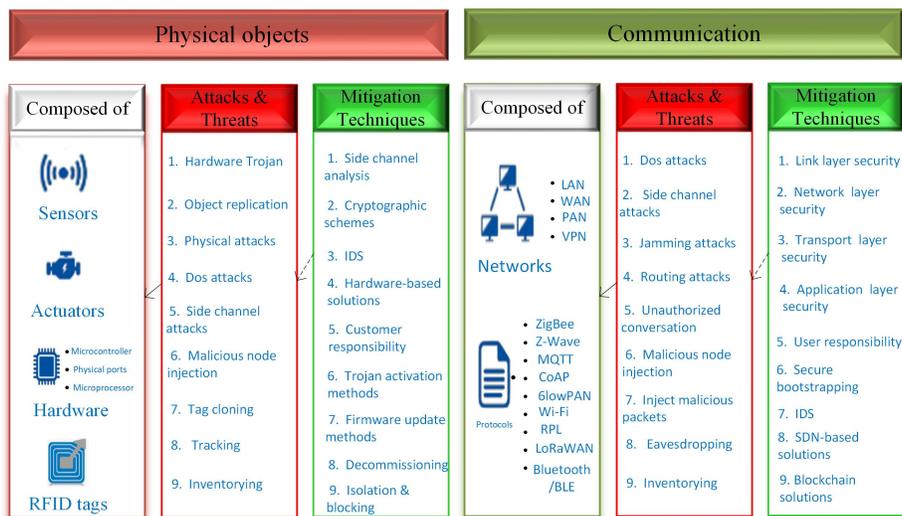
In IoT environment, different types of IoT objects that vary from tiny and lightweight objects (e.g, light bulbs) to powerful objects (e.g, smart phones) are being connected to the Internet or each other to achieve specific tasks. It is therefore unwise to suggest common implementation techniques for all of them, since some objects, for instance light bulbs, would not be able to run them due to their limited resources. Therefore, there is a need to classify IoT objects into different categories based on their hardware capabilities. To this point, we classify IoT objects, in our framework, into five categories based on four primary factors: (a) Central Processing Unit (CPU), (b) memory, (c) power consumption, and (d) on-board storage.

In our framework, if an IoT object is in category 1, it indicates that it has minimal hardware capabilities. While if an IoT object is in category 5, it indicates that it has very powerful hardware capabilities. An overview of objects classification in our framework, along with a real example for each category is presented in Table 9.

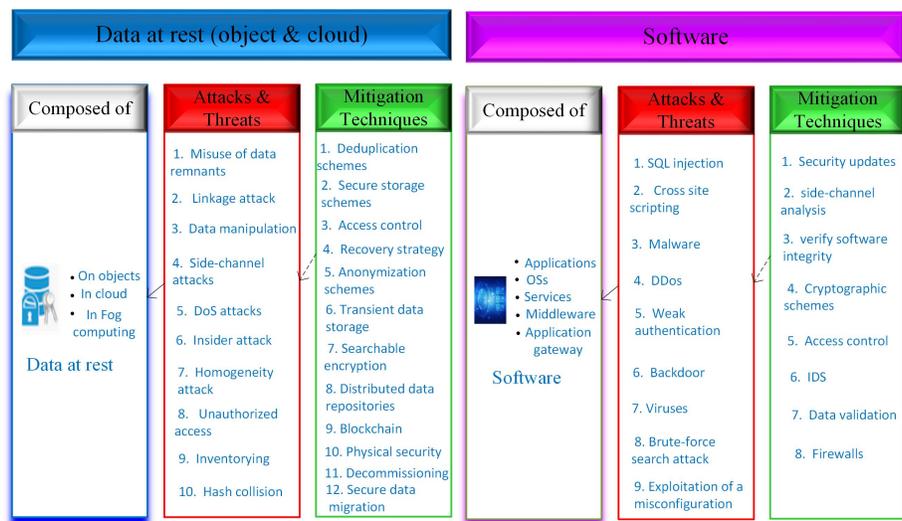
4.2. Phase 2: Proposing security and privacy guidelines for each IoT asset

In this phase, we propose a set of security and privacy guidelines depicted in Figure 3 for all IoT assets mentioned above. Next, we briefly discuss such guidelines for each IoT asset.

4. METHODOLOGY



(a) Physical and communication layers



(b) Data at rest and software layers

Figure 2. A summary of IoT assets, attacks, and countermeasures

4. METHODOLOGY

Physical objects	Communication	Data at rest	Software
Secure boot process	Implement hop to hop security	Minimize data storage	Prevent malicious requests
Update firmware securely	Secure bootstrapping	Minimize data retention	Integrate OS with TLS
Use hardware identifier	Prevent packet modification	Encrypt data storage	Provide process protection
Prevent reverse engineering	Encrypt data communication	Prevent data leakage	Validate and encrypt update
Safe disposal	Support end to end security	Ensure data availability	Reduce raw data intake
Implement hardware trust	Prevent packet duplication	Ensure authorized access	Provide events trace
Detect abnormal nodes	Strong key management	Remove or hide sensitive data	Provide memory protection
Prevent node replication	Hidden data routing	Search on encrypted data	
Prevent unwanted IC modifications	Ensure Continuous monitoring	Proper data destruction	

Figure 3. IoT assets along with their suggested security guidelines

4.2.1. Physical layer:

As shown in Figure 2a, this layer is susceptible to several attacks and threats, the most popular of which are physical attacks, object replication attacks, side-channel attacks, and Hardware Trojan attacks. To contribute to alleviate such attacks, we suggest a list of security and privacy guidelines for the physical layer. These guidelines can be utilized at the early stages of systems development life cycles by IoT stakeholders like developers so that such systems will be resistant to such attacks. For example, one guideline recommends that a hardware secure boot process should be integrated into each IoT object. If such guideline is implemented by developers or manufacturers, it will prevent such object from running malicious code. Table 10 highlights the suggested security and privacy guidelines at this layer, their purposes, and their implementation techniques.

4.2.2. Communication layer

Similarly, this layer is vulnerable to several attacks shown in Figure 2a. These attacks include, but not limited to, side channel attacks, malicious packet modification, routing attacks, malicious node injection, and eavesdropping. To mitigate these attacks, we propose a set of security and privacy guidelines for this layer which can be utilized by IoT stakeholders to build secure system from start. For instance, we suggest that each IoT object should be equipped with techniques, such as Network Layer Security (NLS), Application Layer Security (ALS), and Transport Layer Security (TLS), to prevent malicious packets modification. An overview of our proposed guidelines for communication layer, along with their purposes and their protection measures is presented in Table 11.

4.2.3. Data at rest layer

Data at rest either on IoT objects or in the cloud is also susceptible to different attacks depicted in Figure 2b., such as misuse of data remnants, linkage attacks, data manipulation, insider attacks, side-channel attacks, and homogeneity attacks. To lessen such attacks, we suggest a set of security and privacy guidelines that can be implemented by either IoT developers, manufactures, and providers from ground up so that IoT data stored by their applications is fully protected. The prevention of IoT data leakage, for instance, requires IoT stakeholders to implement a set of techniques into their systems for start like monitoring and auditing schemes, Searchable Encryption (SE), anonymisation schemes, and transient

4. METHODOLOGY

Table 10. A summary of the suggested guidelines for physical layer along with their countermeasures

Guidelines	Purpose	Implementation techniques
Secure boot process [3]	An IoT object should have a fixed hardware secure boot process to prevent it from running a malicious code	Cryptographic schemes [36], hardware-based solutions [37,38]
Update firmware securely [3]	An IoT object should have a mechanism to update its firmware securely to detect malicious firmware	Firmware update methods [39]
Use hardware identifier [3]	An IoT object should be equipped with a unique identity associated with its hardware	Hardware-based solutions
Prevent physical tampering [3]	A proper tamper-resistant measure should be integrated into each IoT object to prevent physical attacks	Hardware-based solutions (e.g., tamper-proofing techniques)
Safe disposal [3]	An IoT object should have a method to destruct its data properly when reaching its end-of-life stage	Decommissioning methods [40]
Implement hardware trust [3]	An IoT object should have a hardware trust method to prevent malicious modification of data	Hardware-based solutions (e.g., a Physical Unclonable Function (PUF)-based authentication [41])
Detect abnormal nodes [3]	An IoT object should have a technique to detect abnormal nodes and activities	IDS techniques (e.g., a method based on Markov model [42])
Prevent node replication [3]	An IoT should have a method to safeguard its identification number	Cryptographic schemes
Prevent unwanted IC modifications [3]	An IoT object should be armed with a mechanism to detect malicious modifications on its IC	Side channel analysis (e.g., dynamic permutation [43]) and hardware-based solutions

Table 11. A summary of the suggested guidelines for communication layer along with their countermeasures

Guidelines	Purpose	Implementation techniques
Implement hop to hop security [3]	In some cases (e.g., lack of resources), an IoT needs to encrypt and decrypt packets to the next object through shared keys	DLLS like IEEE 802.15.4 [44]
Secure bootstrapping [3]	It is a process by which an IoT object joins a network or another object securely	Secure bootstrapping like Diet Host Identity Protocol (HIP) [45]
Prevent packet modification [3]	An IoT object should be integrated with message integrity mechanism to prevent packet injection attacks	TLS in [11], or NLS in [46], or ALS in [47]
Encrypt data communication [3]	A communication link between two IoT objects must be encrypted	TLS, or NLS or blockchain solutions in [48], or Software Defined Network (SDN)-based solutions in [49]
Support end to end security [3]	Each IoT, if it has capabilities, must implement an end-to-end security	TLS, or NLS, or ALS or blockchain
Prevent packet duplication [3]	Each IoT object should be armed with a mechanism to prevent reply attacks	TLS or ALS, or NLS
Strong key management [3]	An IoT object should have a strong key management scheme to protect its data during communication	SDN-based solutions, or blockchain solutions or TLS, or ALS
Hidden data routing [3]	An IoT should be armed with anonymous routing methods to protect routing data	SDN-based solutions, or blockchain solutions, or NLS
Ensure Continuous monitoring [3]	An IoT object should have a technique to monitor abnormal activities	IDS

4. METHODOLOGY

Table 12. A summary of the suggested guidelines for data at rest layer along with their countermeasures

Guidelines		Purpose	Implementation techniques
Minimize storage [4]	data	Minimize the amount of data stored either on objects or in the cloud to reduce IoT breaches by deleting any portion of data not required to achieve a certain task	Deduplication in [50] and anonymisation in [51] schemes
Minimize retention [4]	data	Minimize data retention on IoT objects or in the cloud to prevent data breaches	Transient data storage in [52]
Encrypt data storage [4]		IoT applications should store their data in encrypted format either on the objects or in the cloud	Secure storage schemes in [53] and SE in [54]
Prevent leakage[4]	data	Although IoT data may be stored in encrypted format, it is still vulnerable to side channel attacks	Monitoring and auditing, SE, anonymisation schemes, and transient data storage
Ensure availability[4]	data	Both CSSs and IoT objects must implement efficient methods to ensure availability of their data	Deduplication schemes and Recovery strategy in [55]
Ensure authorized access [4]	authorized	Offering solid mechanisms to control access to IoT data at rest is essential to prevent unauthorized access	Physical security and access control in [56]
Remove or hide sensitive data [4]		IoT applications should first remove a personally identifiable information before storing it	Anonymisation schemes
Search on encrypted data [4]		IoT applications must be equipped with methods to reply to any queries by searching on encrypted data	SE
Proper destruction [4]	data	The secure destruction of IoT data at rest is of importance to mitigate several security concerns (e.g., data leakage)	Decommissioning

data storage. Table 12 outlines our suggested guidelines at this layer, along with their purposes, and their implementation techniques.

4.2.4. Software layer

As depicted in Figure 2b, this layer, composed of IoT application, OSs, and middleware, is also vulnerable to many attacks and threats like DoSs attacks, Structured Query Language (SQL) injection attacks, weak authentications, malicious requests, and viruses. We suggest a set of security and privacy at this layer to alleviate its associated attacks and threats. For example, having integrated access control mechanisms into IoT applications, OSs, and middleware, such software will be able to prevent malicious requests coming from either adversaries or malicious objects. A summary of our proposed guidelines for this layer, along with their purposes and their protection measures is presented in Table 13.

4.3. Phase 3: Assigning different SLC for IoT objects

In this phase, we briefly classify SLCs used in our framework and then discuss how to assign SLCs to IoT objects.

4.3.1. SLCs classification

Classifying SLCs, in our suggested framework, is a fundamental requirement, and it stems from two primary reasons. One is that IoT objects come in different sizes and hardware capabilities. In general, most of IoT objects have limited resources, but this is not always the case since some objects may have powerful hardware resources. Thus, it is impractical to assign common mitigation techniques for all IoT objects. The

4. METHODOLOGY

Table 13. A summary of the suggested guidelines for software layer along with their countermeasures

Guidelines	Purpose	Implementation techniques
Prevent malicious requests	IoT applications should be armed with a technique to prevent and block malicious requests	Access control methods
Integrate OS with TLS	To offer data integrity and privacy two or more communicating objects, TLS should be integrated into an object's OS.	Secure IoT OSs
Provide memory protection	An object's OS should have a strong process management to manage an object's resources	Secure IoT OSs
Validate and encrypt update	To prevent the injection of malware in objects' OSs or IoT applications during their update processes, each update patch must be encrypted and validated	Secure IoT OSs and secure update methods
Provide events trace	IoT objects should be armed with a technique to continuously monitor their logs, processes, and software	
Provide memory protection	An object's OS should be integrated with a memory management technique to properly allocate/deallocate its parts for different threads and processes	Secure IoT OSs

other reason depends on the environment at which IoT objects are being deployed and operated. Objects operating in a controlled area will require less protection measures as they will always be connected to trusted objects and will always be monitored by human beings or security cameras. On the other hand, objects operating in an uncontrolled environment will need more protection measures, since they will neither be connected to trusted objects, nor will be monitored by human beings or security cameras. To this end, we classify all the protection measures suggested to implement our proposed guidelines for previously mentioned IoT assets into five groups known as SLCs, starting from SLC 1 to SLC 5. The number attached to each SLC indicates its security level, which is very weak in SLC 1 and very powerful in SLC 5. This is because SLC 1 includes only two protection measures (DLLS and secure bootstrapping), whereas SLC 5 includes almost all mitigation techniques. It is worth noting that the process of assigning and issuing SLCs depends heavily on entities (e.g., IoT manufacturers or IoT developers) that implement our framework. More importantly, we assume that such entities are trusted and authenticated so that they will neither issue nor assign fake SLCs. In the next paragraphs, we briefly discuss each one of them.

SLC 1: This type of SLC will implement two mitigation techniques, namely DLLS and secure bootstrapping. Thus, IoT objects with SLC 1 will be able to encrypt and decrypt packets only at data link layer because of DLLS. In other words, they will be able to provide hop-to-hop security. Moreover, they will be able to join or rejoin gateways securely (objects with SLC 2) because of their secure bootstrapping techniques. Due to the lack of hardware-based solutions to prevent physical attacks and end-to-end security techniques (e.g., TLS) to provide secure communication channels, objects with SLC 1 will neither be deployed in an uncontrolled environment, nor will be connected to the Internet directly. Furthermore, objects with SLC 1 will not be able to store data locally as they will lack secure methods of storing IoT data, nor will be able to update their firmware autonomously since they will not have secure firmware update methods, and they will depend on objects with SLC 2 to do so. Moreover, objects with SLC 1 will depend on objects with SLC 2 to register their SLCs in objects with SLC 5 (responsible for tracking of SLCs for all objects).

SLC 2: It has 5 mitigation techniques (see Table 14), such as DLLS, TLS, NLS, and firmware update methods. IoT objects with SLC 2 will have capabilities to encrypt and decrypt packets at data link, transport, and network layers. That said, IoT objects with SLC 2 will only be able to communicate with the

4. METHODOLOGY

Internet through objects with **SLC 3**, as objects with **SLC 2** will not have side channel protection measures to prevent data leakage and **IDSs** to detect malicious packets. In other words, such objects will not be able to connect directly to the Internet. Since such objects will have firmware update methods, they will be able to update their firmware by connecting objects with **SLC 4**, which are responsible for managing firmware updates in our framework. Like objects with **SLC 1**, objects with **SLC 2** will not be deployed in an uncontrolled area because they lack required security techniques (e.g., tamper-proofing method) to prevent physical attacks. Furthermore, objects with **SLC 2** will not be able to store data locally due to the absence of secure techniques to do so. Unlike objects with **SLC 1**, objects with **SLC 2** will be able to register their **SLCs** in objects with **SLC 5** and also communicate with other objects securely, as they will have end-to-end security techniques (e.g., **NLS** and **TLS**).

SLC 3: This type of **SLC** will be integrated with 12 mitigation techniques (see Table 14) like **DLLS**, **TLS**, **NLS**, firmware update methods, **IDS**, secure **OS**, and access control methods. **IoT** objects with **SLC 3** will be able to connect directly to the Internet and also manage communication between objects with **SLC 1** and **SLC 2** and the Internet. This is because objects with **SLC 3** will be armed with the required protection measures like end-to-end security techniques, side channel protection methods, secure **OSs**, and more importantly **IDSs** to detect malicious packets. Unlike objects with **SLC 1** and **SLC 2**, **IoT** objects with **SLC 3** will be able not only to deploy and operate in unattended areas, but also to update their firmware by connecting objects with **SLC 4**. Furthermore, **IoT** objects with **SLC 3** will be able to store their data or data coming from objects with **SLC 2** temporarily on their data storage after simple data processing to offer quick response to objects with **SLC 2**, as they will be equipped with transient data storage techniques. Nevertheless, **IoT** objects with **SLC 3** will lack secure data destruction as well as recovery mechanisms. To this end, such objects will be able only to store data just for a short period of time (e.g., per an hour). However, such objects will be able to communicate with objects with **SLC 5** in order to store their data for a long time, as they will have suitable protection measures (e.g., secure storage schemes, recovery strategy, and deduplication schemes) to prevent data at rest breaches.

SLC 4: This type of **SLC** will be equipped with 14 protection measures (see Table 14), such as **DLLS**, **TLS**, **NLS**, hardware-based solutions, and blockchain-based solutions. **IoT** objects with **SLC 4**, in our framework, will be responsible for managing firmware updates of the all **IoT** objects equipped with different **SLCs**. More importantly, objects with **SLC 4** will utilize blockchain-based solutions like smart contracts proposed in [57] to manage secure **IoT** firmware updates for all **IoT** objects participating in our framework. As a consequence, manufacturers, implemented our framework, will be able to create smart contracts for the newly-developed firmware versions and push them to all objects with **SLC 4**. Having pushed the smart contracts to the blockchain network formed, in our framework, by different objects equipped with **SLC 4**, objects with **SLC 2**, **SLC 3**, and **SLC 5** will be able autonomously to query objects with **SLC 4** and therefore download the latest versions of firmware available for them. That said, time latency to register each smart contract to the blockchain network may take a long time (e.g., 10 minutes per transaction), but this is not an issue since objects' firmware updates may be released once per month or week. As objects with **SLC 4** will be used to store the latest versions of objects' firmware, they will be also equipped with secure decommissioning methods to securely destruct their data when reaching their end-of-life stages to prevent data breaches (e.g., misuse of data remnants).

SLC 5: This type of **SLC** includes all our suggested mitigation techniques (e.g., recovery strategy, blockchain solutions, and hardware-based solutions), except transient data storage and **IDS**. As **IoT** objects with **SLC 5** will be equipped with blockchain-based solutions, such objects will be responsible for registering and tracking of all **SLCs** into their chain. Such objects, however, are not responsible for assigning and

4. METHODOLOGY

Table 14. SLCs classification according to their mitigation techniques

ID	Mitigation Techniques	SLC 1	SLC 2	SLC 3	SLC 4	SLC 5
MT1	Link layer security	✓	✓	✓	✓	✓
MT2	Transport layer security	✗	✓	✓	✓	✓
MT3	Network layer security	✗	✓	✓	✓	✓
MT4	Firmware update methods	✗	✓	✓	✓	✓
MT5	Intrusion detection system	✗	✗	✓	✗	✗
MT6	Side channel protection	✗	✗	✓	✓	✓
MT7	Decommissioning methods	✗	✗	✗	✗	✓
MT8	Secure bootstrapping	✓	✓	✓	✓	✓
MT9	Blockchain solutions	✗	✗	✗	✓	✓
MT10	Hardware-based solutions	✗	✗	✓	✓	✓
MT11	Deduplication schemes	✗	✗	✗	✗	✓
MT12	Anonymisation schemes	✗	✗	✗	✗	✓
MT13	Transient data storage	✗	✗	✓	✓	✗
MT14	Secure storage schemes	✗	✗	✗	✓	✓
MT15	Searchable encryption	✗	✗	✓	✓	✓
MT16	Monitoring and auditing	✗	✗	✗	✗	✓
MT17	Recovery strategy	✗	✗	✗	✗	✓
MT18	Access control methods	✗	✗	✓	✓	✓
MT19	Secure IoT OSs	✗	✗	✓	✓	✓

validating SLCs, since this process will be archived by entities implemented our framework from the early stages of their systems development processes. Each object in our framework (except object with SLC 1) will have to register its SLC in object with SLC 5. This step, in our suggested framework, is an indispensable requirement for many reasons. One is that objects will not be able to change their SLCs, since they will be allowed to register their SLCs once in objects with SLC 5. Another reason is that it will ease the communication process among objects with different SLCs, as their public keys will be accessible by all the objects (except objects with SLC 1). The other reason is that it will detect malicious objects with fake SLCs, as each object will be able to verify the SLC of another object by checking its SLC in the chain in objects with SLC 5. More importantly, fake objects will be placed in a revocation list by objects with SLCs 5, and all objects (except object with SLC 1) will be notified by this list so that they will no longer communicate with them. Like objects with SLC 4, objects with SLC 5 will be able to deploy and operate in uncontrolled environments, and will be able to destroy their data in a proper way. Unlike objects with SLC 4, these objects will provide a recovery strategy of their data, and most importantly will be responsible for integrating legacy objects with our objects.

By observing Table 14, it is clear that our framework will provide a common method by which all objects (except objects with SLC 1) will be able to communicate with each other securely. This is because objects with SLC 3, SLC 4, and SLC 5 will implement the same protection measures in TLS and NLS. An overview of SLCs classification used in our suggested framework along their mitigation techniques is presented in Table 14.

4.3.2. Assigning SLC to IoT objects :

As we have classified both IoT objects based on their hardware capabilities into five categories (see Table 9) and SLCs based on their mitigation techniques into 5 levels (see Table 13), it is crucial to present the link between them. In other words, we want to define what types of SLCs are suitable for each category. For instance, as IoT objects in category 1 will have limited hardware resources, they will only be able to implement SLC 1 because it has only two mitigation techniques. However, such objects will not be able

4. METHODOLOGY

Table 15. Assigning SLCs to IoT objects

Object categories/SLCs	SLC1	SLC2	SLC3	SLC4	SLC5
Category 1	✓	✓	✗	✗	✗
Category 2	✓	✓	✗	✗	✗
Category 3	✓	✓	✓	✗	✗
Category 4	✓	✓	✓	✓	✗
Category 5	✓	✓	✓	✓	✓

Table 16. The suggested communication plan

Types of SLCs	SLC1	SLC2	SLC3	SLC4	SLC5	Internet
SLC1	✗	✓	✗	✗	✗	✗
SLC2	✓	✓	✓	✓	✓	✗
SLC3	✗	✓	✓	✓	✓	✓
SLC4	✗	✓	✓	✓	✓	✓
SLC5	✗	✓	✓	✓	✓	✓

to implement SLC 3, or SLC 4, or SLC 5 due their limited resources. Similarly, IoT objects in category 2 will be able to implement either SLC 1 and SLC 2, as they have required hardware capabilities to do so. Unlike objects in category 1 and category 2, objects in category 5 will have powerful hardware resources to implement any one of SLCs.

It is worth stressing that the process of assigning SLCs to IoT objects must be done with caution. This is because the improper assignment of SLCs to IoT objects may lead to have insecure objects despite having strong hardware capabilities, for instance assigning SLC1 or SLC2 to objects in category 5. Table 15 states the SLCs that are suitable for each category.

4.4. Phase 4: Developing communication plan based on SLC

Defining a communication plan between objects, in our framework, depends heavily on their SLCs in order to minimize the risks associated with weak links and also reduce unexpected used of IoT data. To this end, object with SLC 1 will only communicate with objects with SLC 2 due to their weak protection measures (see Table 14). Not only that, such objects will not be able to communicate with objects having the same SLCs as these objects may have different link layer protocols (e.g, IEEE 802.15.4 and Bluetooth). To do so, they will depend on objects with SLC 2 to perform a required translation between these protocols. Unlike objects with SLC 1, objects with SLC 2 will be able to communicate with all objects in our framework as long as objects with SLC 3, SLC 4, and SLC 5 will implement the same algorithms or mechanisms ,implemented by objects with SLC 2, in their TLS and NLS. Nevertheless, such objects will not be able to communicate with the internet without using object with SLC 3. Unlike objects with SLC 1 and SLC 2, object with SLC 3, SLC 4, and SLC 5 will be able to communicate with the Internet and more importantly communicate with all objects easily (except objects with SLC 1). Table 16 summarizes the proposed communication plan among IoT objects equipped with SLCs.

4.5. Phase 5: Integrating our objects with Legacy objects

It is unrealistic to assume that objects developed based on our suggested framework will always be communicated with each other. This is an unreasonable assumption in IoT, since the mobility of objects is one of main features of IoT. Therefore, integrating legacy objects, which are not implemented our framework, with our objects is a fundamental requirement towards achieving compatibility in IoT. Toward

5. CASE STUDY: SMART HOME

this end, we propose a simple method composed of 4 steps, depicted in Figure 4, that will allow legacy objects to communicate with our objects in a secure manner.

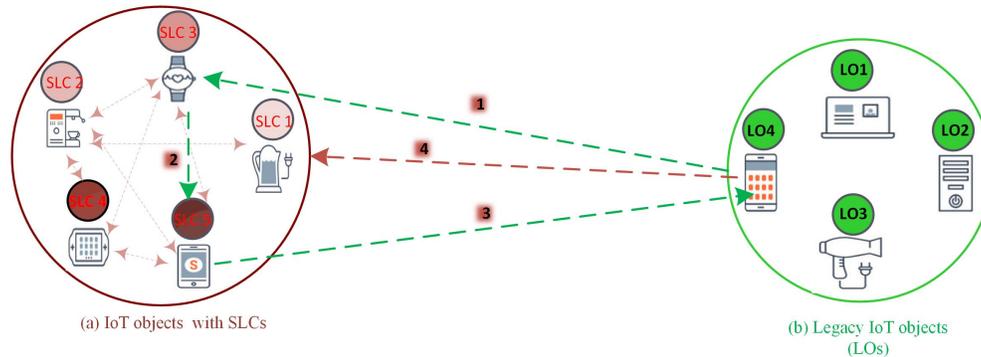


Figure 4. Integration the legacy objects with our framework

First, a legacy object, if it is in the range of our network, will try to communicate with our objects except for objects with **SLC1** and **SLC 2** due to their restricted communication plan (see Table 16). Second, upon receiving the request from the legacy object, our object first checks if that object has a **SLC**. If not, it will send the request to any objects with **SLC 5**, as they are responsible for integrating legacy objects with our objects. Third, an object with **SLC 5** will suggest a set of algorithms that should be implemented in a transport layer of the legacy object to be able to communicate with other objects and will be able to provide an end-to-end security. If the legacy object lacks such algorithms or will not be able to implement them, this legacy object will not be able to integrate with our framework. Fourth, the legacy object will be able to communicate with objects with **SLC 3**, **SLC 4**, and **SLC 5** as long as it will use the same protection measures in its transport layer implemented by those objects.

Figure 4 summarizes the above mentioned steps required to integrate the legacy objects with objects developed according to our suggested framework. Step 1 (Figure 4) shows that the legacy object (lo4), depicted in Figure 4b, will attempt to communicate with an object with **SLC 3**, depicted in Figure 4a. In step 2, the object with **SLC 3** (see Figure 4a) will direct the request coming from the legacy object (lo4) to any object with **SLC 5**. In step 3, the object with **SLC 5** (in Figure 4a) will recommend a set of protection measures that should be implemented in transport layer of the legacy object (lo4), and we assume the legacy object depicted in Figure 4b will have such mechanisms. In step 4, the legacy object (in Figure 4a) will be able to communicate with objects with **SLC 3**, **SLC 4** and **SLC 5**.

Figure 5 highlights the capabilities (shown in green arrows) and the shortcomings (shown in red arrows) of **IoT** objects equipped with our suggested **SLCs**.

5. Case study: smart home

This section illustrates how our proposed methodology can be utilized to develop a secure smart home system as a simplified case study in order to present clearly the benefits of our suggested framework. Even though the feasibility of our framework is presented in the context of a smart home, our methodology is domain agnostic, and thus it can be applied in other **IoT** domains. To this point, a set of steps is required by **IoT** developers or software engineers to apply our framework to smart homes. These steps are briefly presented in the next paragraphs.

Step 1: First, we identify all **IoT** objects operating in smart homes and classify them into different groups. With the advent of **IoT** vision in which most of the physical objects around us will be connected to the Internet, the number of **IoT** objects functioning in smart homes will be almost endless [3]. Such

5. CASE STUDY: SMART HOME

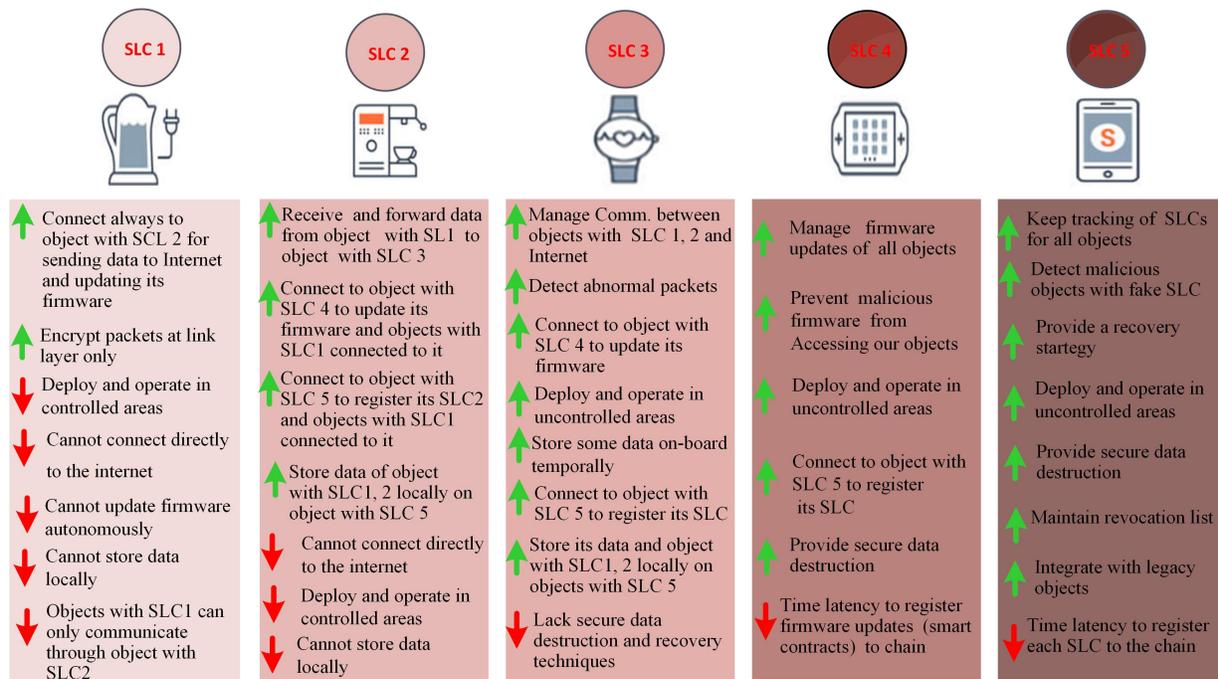


Figure 5. An overview of objects equipped with SLCs

objects include, but not limited to, light-bulbs, smart switches, microwaves, dishwashers, TVs, projectors, and smart phones. To this end, we classify smart home objects into eight groups such as smart detectors, household objects, and consumer objects, depicted in Figure 6. Our classification depends heavily on hardware capabilities of smart home objects as well as their functionalities. For instance, consumer objects have very powerful resources and can be used in multiple purposes (e.g., control other objects), while smart detectors have very limited resources, and their main objectives are to detect changes inside the smart home (e.g., detect motion).

Step 2: We assign a SLC or two SLCs for each group (See Figure 6). For instance smart detectors will have SLC 1, whereas home entertainment objects will have SLC 3. The process of assigning SLCs for each group depends on many factors: (i) hardware capabilities (see Table 9), (ii) location at which such objects are being deployed either inside or outside the home, functionalities, connection to the Internet either direct or indirect, and implemented mitigation techniques (see Table 14). To this end, smart detectors, for example, will be equipped with SLC 1, since such objects will have very limited resources, no direct internet access, their deployment at inside homes and more importantly their weak protection measures (MT1 and MT8). In contrast, consumer objects in smart home according to our framework will have either SLC 4 for smart phones or SLC 5 for laptops. This is because such objects (e.g., phones and laptops) will have very powerful resources, and they will be equipped with most of our suggested mitigation techniques (See Table 14). More importantly, objects with SLC 4 will have unique responsibilities compared to objects with SLC 5, according to our framework. Objects with SLC 5 will be equipped with blockchain solutions to register and track SLCs for all smart home objects, detect fake SLCs, maintain revocation list, and integrate the legacy objects into smart home. While objects with SLC 4 will be responsible for managing firmware updates of the all smart home objects (see Section 4.3.1), as they have required protection measures to do so.

5. CASE STUDY: SMART HOME



Figure 6. Smart home object classification, along with their SLCs

It is worth mentioning that smart home objects in our methodology will not only be used to achieve their specific tasks, but also to carry out other responsibilities due to their SLCs, depicted in Figure 5. For example, the main purpose for a smart TV is to allow its users, without the need to connect the TV antenna, to access to several channels which provide movies, music, and programs. Apart from providing such specific task, the smart TV in our framework will have other responsibilities (See Figure 5) such as managing communication between objects with SLC 2 (e.g., smart health objects) and the Internet, since the TV will have required techniques to carry out such responsibilities.

Table 17 summarizes the process of classifying smart home objects into different categories, along their mitigation techniques required for each SLC. It is worth noting that we assume that the hardware capabilities of each smart home classification in this case study such as smart detectors, smart health, and customer objects match our suggested IoT objects classification, presented in Table 9. By observing Table 13, it is not hard to see that all smart home objects will have a set of mitigation techniques through which previously mentioned attacks against IoT such as AT1, AT2, AT3, and AT5 will be mitigated. A detailed explanation of how our suggested framework will lesson such attacks is presented in Section 6.2.

Step 3: We define a secure communication plan among smart home objects, the main purpose of which is to prevent unexpected use of smart home data. Although we classify smart home objects into several groups, such objects will be able to interact with each according to our suggested communication plan, which depends entirely on objects' SLCs. For example, smart detectors equipped with SLC 1, according to our framework, will be able to communicate with indoor security cameras, energy and

5. CASE STUDY: SMART HOME

Table 17. A summary of smart home object classification, along with SLCs

Object category	Example	Object class	SLC Type	Mitigation Techniques
Smart detectors	Water, gas, motion	Category 1	SLC 1	MT1 and MT8
	Indoor	Category 2	SLC 2	MT1, MT2, MT3, MT4, MT8
Security Cameras	Outdoor	Category 3	SLC3	MT1, MT2, MT3, MT4, MT5, MT6, MT8, MT10, MT13, MT15, MT18, MT19
Energy and lighting	Light bulbs	Category 2	SLC 2	MT1, MT2, MT3, MT4, MT8
Home entertainment	Amazon Echo	Category 3	SLC3	MT1, MT2, MT3, MT4, MT5, MT6, MT8, MT10, MT13, MT15, MT18, MT19
Smart health	Blood pressure monitor	Category 2	SLC 2	MT1, MT2, MT3, MT4, MT8
Household appliances	Dishwasher, refrigerator, dryer	Category 2	SLC 2	MT1, MT2, MT3, MT4, MT8
Consumer objects	Smart phones, tablets	Category 4	SLC 4	MT1, MT2, MT3, MT4, MT6, MT8, MT9, MT10, MT13, MT14, MT15, MT18, MT19
	Laptops	Category 5	SLC5	MT1, MT2, MT3, MT4, MT6, MT7, MT8, MT9, MT10, MT11, MT12, MT14, MT15, MT16, MT17, MT18, MT19
Gateways/hubs	Samsung Smart things	Category 3	SLC3	MT1, MT2, MT3, MT4, MT5, MT6, MT8, MT10, MT13, MT15, MT18, MT19

6. DISCUSSION AND FUTURE WORK

Table 18. The suggested communication plan for smart home objects

Smart home objects	Smart detectors	Indoor Security Cameras	outdoor Security Cameras	Energy and lighting	Smart health	Household appliance	Smart phones, tablets	Laptops	Gateways	Internet	legacy objects
Smart detectors	X	✓	X	✓	✓	✓	X	X	X	X	X
Indoor Security Cameras	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	X
Outdoor Security Cameras	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Energy and lighting	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	X
Smart health	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	X
Household appliance	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	X
Smart phones, tablets	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Laptops	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Gateways	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

lighting, smart health, and household appliance, as long as such objects are located within the coverage area of smart detectors signals. Nevertheless, smart detectors will not be able to communicate with smart phones and tablets, laptops, gateways, and the Internet.

An overview of the communication plan in which smart home objects presented above (see Figure 6) will be able to communicate securely with each other, the Internet, and legacy objects is presented in Table 18.

Step 4: Finally, we illustrate how the smart home that will implement our framework will be able to seamlessly and securely integrate legacy objects. A legacy object inside the smart home will attempt to interact with any objects such as gateways and consumer objects in smart home except objects with **SLC1** and **SLC 2** (e.g., smart detectors and smart health) due to their limited communication plan (See Table 18). However, the legacy objects will not be able to communicate with smart home objects with **SLC 3** (home entertainment), **SLC 4** (phones and Tablets), and **SLC** (laptops), unless they first communicate with objects with **SLC 5** (laptops) and then get their **SLCs** from them. The procedure in which the legacy objects will get their **SLCs** from objects with **SLC 5** is presented in Section 4.5.

6. Discussion and Future Work

A summary of the previously mentioned research proposals is presented in Table 19, along with our intended objectives. It is not so difficult to recognize their limitations while going through them. This article, therefore, is directed to overcome those shortcomings that can be categorised as follows: (i) The lack of a thorough set of security and privacy guidelines for **IoT** assets. (ii) The absence of proper mitigation techniques to carry out such guidelines. (iii) The need of attack investigations related to **IoT** systems. (iv) The necessity of mitigation techniques classification as well as **IoT** objects classification. (v) The need of a communication plan so that **IoT** objects will be able to communicate securely with each other or with the Internet.

In what follows, we illustrate how our framework would alleviate the attacks and threats against **IoT** and solve some **IoT** security challenges, and we discuss its limitations too.

6. DISCUSSION AND FUTURE WORK

Table 19. Comparison of research efforts presented in the literature.

Addressed Features		State-of-the Art Work							This work
		[32]	[22]	[23]	[4]	[24]	[3]	[5]	
IoT asset guidelines	Computing nodes	✗	✓	✓	✓	✓	✓	✓	✓
	RFID tags	✗	✗	✗	✗	✗	✗	✗	✓
	Protocols	✗	✓	✗	✗	✓	✓	✗	✓
	Data at rest	✓	✗	✗	✗	✓	✗	✗	✓
	Applications	✓	✓	✓	✗	✓	✗	✓	✓
	OSs	✗	✗	✗	✗	✓	✗	✗	✓
Addressed IoT SCs	SC1	✓	✓	✓	✓	✓	✓	✓	✓
	SC2	✗	✗	✗	✗	✓	✓	✓	✓
	SC3	✓	✗	✗	✗	✗	✗	✗	✓
	SC4	✗	✗	✗	✗	✗	✗	✗	✗
	SC 5	✓	✓	✓	✗	✓	✓	✓	✓
	SC6	✗	✗	✗	✗	✗	✓	✗	✓
	SC7	✗	✗	✗	✗	✗	✗	✗	✗
	SC8	✗	✗	✗	✗	✗	✓	✗	✓
Addressed IoT attacks	AT1	-	-	-	-	-	✓	✓	✓
	AT2	-	-	-	-	-	✓	✓	✓
	AT3	-	-	-	-	-	✓	✗	✓
	AT4	-	-	-	-	-	✓	✗	✓
	AT5	-	-	-	-	-	✓	✗	✓
	AT6	-	-	-	-	-	✓	✓	✓
Types of guidelines	Privacy	✓	✓	✓	✗	✓	✓	✗	✓
	Security	✗	✓	✓	✓	✓	✓	✓	✓
Communication Plan		✗	✗	✗	✗	✗	✗	✗	✓
Objects classification		✗	✗	✗	✗	✗	✗	✓	✓
Protection measures classification		✗	✗	✗	✗	✗	✗	✓	✓

These symbols ✓, ✗, and – indicate the addressed, not addressed, and not mentioned features, respectively.

6. DISCUSSION AND FUTURE WORK

6.1. Analysis on IoT security challenges

In this part, we offer qualitative arguments to illustrate that our proposed methodology, if it is followed precisely, can be used to address several IoT security challenges (see Table 1), which are briefly discussed below:

(SC1) Lack of a secure development: As our proposed methodology is composed of five phases, one of its phases which is phase 2 is designed specifically to address such challenge. In phase 2, we propose a set of security and privacy guidelines covering all IoT assets (physical objects, protocols, data at rest, and software), along with their appropriate implementation techniques. Integrating such guidelines and techniques by developers or manufacturers into their IoT systems or products from the early stages of development (life cycles) will lead to develop secure system, which in turn improves security and privacy by design for IoT.

(SC2) Tight resource constraints: It is unrealistic to assign common protection measures for IoT objects, since such objects may come in different sizes, varying from resource-limited objects like motion detectors to resource-rich ones such as smart phones. Smart phones, for example, can implement traditional security mechanisms, while it seems to be very difficult to apply such techniques in motion detectors without some modifications. To this point, we first classify IoT objects into five categories (see Table 9) based on their hardware capabilities. Furthermore, we assign different mitigation techniques (SLCs) which are more suitable for each category. For instance, objects in category 1 will implement a few protection measures (see Table 14) suitable to their limited resources, whereas objects in category 5 will almost implement all our suggested protection measures due to their powerful hardware capabilities such as power consumption.

(SC3) Designed for specific Tasks: Being designed to carry out specific tasks and deployed in different environments, IoT objects require different mitigation techniques. In other words, it is wise to assign different protection measures to IoT objects based on their main functions or tasks. To this end, the proposed framework assigns different protection measures to IoT objects based on their tasks and hardware resources. For example, as the main goal of objects with SLC 5 in our method is to register and keep track of all SLCs, such objects thus will be equipped with blockchain solutions to do so. In contrary, objects with SLC 1, 2, and 3 will not be armed with such solutions, as they are not designed to accomplish such goal.

(SC5) Update mechanisms: As the security of IoT objects relies on a method in which such objects receive their newly released updates either locally or remotely, our proposed framework thus assigns different mitigation techniques for IoT objects. For example, objects with SLC 1 will not have firmware update methods, as they depend entirely on objects with SLC 2 to update their firmware, while other objects with SLC 2, 3, 4, and 4 will be equipped with such techniques to independently update their firmware.

(SC6) Objects' mobility: Since the location of IoT objects either static or dynamic plays a key role in defining their security requirements, our framework therefore assigns different mitigation techniques for such objects based on their mobility. For instance, objects with SLC 1 and SLC 2 will have a few protection measures, as they will always interact with each other or with objects with SLC 3. On the other hand, objects with SLC 3, 4, and 5 will have more mitigation techniques due to their communication with each other, the Internet, and legacy objects.

6. DISCUSSION AND FUTURE WORK

(SC8) Uncontrolled environment: The environment at which IoT objects will be deployed and operated plays a key role in determining their proper mitigation techniques. To this point, IoT objects, in our framework, can be classified broadly into two groups: (i) objects operating in controlled environments and (ii) objects operating in uncontrolled areas. Thus, objects operating in controlled areas like objects with SLC1 and SLC 2 will have a few mitigation techniques, as such objects will always be deployed in attended areas and will always be monitored by human beings or security cameras to prevent physical attacks. In contrast, objects with SLC3, SLC 4, and SLC 5 will have more protection measures to prevent physical attacks, as such objects may will be deployed in unattended environments.

Although most of IoT security challenges presented in Table 1 have been addressed in our suggested framework, two security challenges, namely (SC4) "changes in security requirements" and (SC7) "the importance of IoT objects", are left untouched. We do believe that such challenges can be addressed by developers or software engineers during the analysis phase of an IoT system's requirements.

6.2. The mitigated attacks and threats in our methodology

We anticipate that our framework can be used to address several attacks and threats against IoT as long as our methodology is precisely followed. We briefly discuss the mitigated attacks and threats in the next paragraphs.

(AT1) Eavesdropping: To mitigate such attacks, our framework prevents any object from sending and receiving its data or packets over unencrypted channels. This can be observed through mitigation techniques included in all our suggested SLCs. For instance, objects with SLC 1 will implement DLLS to send/receive their data in encrypted format to/from objects with SLC 2. Similarly, objects with SLC 2, SLC 3, SLC 4, and SLC 5 will implement either TLS or NLS to provide end-to-end secure communication channels between them (see Table 14).

(AT2) Physical attacks: To lessen this type of attacks, our framework divides its objects based on their environments into two categories: (a) Controlled environments and (b) Uncontrolled environments. Objects with SLC 1 and SLC will always be deployed in controlled areas to prevent physical attacks despite not having mitigation techniques to do so (see Table 14). This is because such objects will always be monitored by either people or security cameras. On the other hand, objects with SLC 3, SLC 4, and SLC 5 will be deployed in uncontrolled environments, as they will be equipped with hardware-based solutions like tamper-proofing techniques to mitigate physical attacks (see Table 14).

(AT3) Side-channel attacks: To alleviate such attacks, our framework will integrate side-channel protection techniques into objects with SLC 3, SLC 4, and SLC 5 so that such objects will be able to alleviate such attacks. Whereas objects with SLC 1 and SLC 2 will be vulnerable to side-channel attacks, since such objects will not have side-channel protection techniques (see Table 14). However, this will not be an issue in our suggested framework, as these objects will not be connected directly to the Internet (always interact with our objects, see Table 16), nor they will be deployed in uncontrolled environments, according to our methodology.

(AT4) Malicious object insertion: To this end, our suggested framework will force its objects with different SLCs to first register their SLCs in objects with SLC 5 before they will be able to communicate with each other. It is worth repeating that objects with SLC 5 will be shielded with blockchain-based solutions so that other objects like objects with SLC 2, SLC 3, and SLC 4 will be able to track all registered objects and therefore detect the malicious ones. For example, suppose that an object with fake SLC 3 will

6. DISCUSSION AND FUTURE WORK

try to communicate with an object with SLC 2. First, the object with SLC 2 will check if the object trying to communicate with has a genuine SLC 3 by contacting any object with SLC 5. If not, which is the case in this example, the object with SLC 2 will not communicate with it and will notify any object with SLC 5 about this incident.

(AT5) Routing attacks : To lessen such attacks, our framework will compel the majority of its objects to apply NLS to prevent such attacks (see Table 14). For instance, objects with SLC 2, SLC 3, SLC 4, and SLC 5 will have such mitigation techniques against routing attacks. In contrary, objects with SLC 1 will be vulnerable to such attacks, as they will only implement two mitigation techniques (DLLS and secure bootstrapping). However, our suggested communication plan (see Table 16) will play a key role to mitigate such threat, as it will restrict the communication of objects with SLC 1 to only objects with SLC 2. More importantly, communication links or channels between objects with SLC 1 and object with SLC 2 are encrypted using link layer security (DLLS).

(AT6) Malicious firmware To mitigate this types of attacks, our framework will utilize blockchain-based solutions (e.g., A pushed-based firmware update proposed in [58])to update their objects securely. Manufacturers, implemented our framework, will be able to build smart contracts for newly developed firmware versions and will push them to all objects with SLC 4. During the update process, some objects with SLC 4 called miners will verify the integrity of pushed firmware, as they will equip with consensus protocol. Once the smart contracts are verified by miners, objects with SLC 2, SLC 3, and SLC 5 will be able to send requests to objects with SLC 4 and therefore download the latest versions of firmware available for them.

6.3. Limitations of the Study and Threats to Validity

Vulnerable to Insider Threats: The risks associated with the insider threats in our framework can be recognized in two processes: (i) issuing and assigning SLCs and (ii) firmware updates. As the process of assigning and issuing SLCs, in our methodology, will depend heavily on entities (e.g, developers or manufacturers) that will implement it, it is therefore vulnerable to malicious insider threats. It is possible that a developer who is responsible for issuing SLCs could accidentally or intentionally either alter the setting of SLCs or assign SLCs to wrong IoT objects. For instance, a malicious developer could assign SLC 3 to an object whose real SLC is one, which may put the entire system at risk. This is because security of our methodology relies heavily on its communication plan, which in turn depends on SLCs; the object with SLC 1 will always be connected to objects with SLC 2, and it will be deployed in controlled areas, whereas the object with SLC 3 will communicate with all objects (except object with SLC 1) and it will be deployed in uncontrolled environments (see Table 16 and Figure 5).

Similarly, blockchain-based firmware update scheme (smart contract and consensus mechanism) utilized in our framework is also susceptible to malicious insider threats despite its benefits in terms of verifying the firmware integrity and preventing DoS attacks. This is because our methodology assumes that all newly released firmware updates are pushed or published by a trustworthy manufacturer. However, this is not always the case because of two reasons. One is that the manufacturer may be compromised by an attacker, and therefore he could use manufacturer's private keys to sign updates and push them into objects with SLC 4. The other reason is that an employee at manufacturer, attacker, may be able (due to given access rights) to send malicious updates from the manufacturer's server to all objects with SLC 5.

REFERENCES

REFERENCES

6.4. Conclusion

The main goal of this paper is to develop a secure framework for IoT that allows different IoT objects to communicate securely with each other or with the internet based on their SLCs. To this end, we first classify IoT objects into five categories based on their hardware capabilities; objects in category 1 indicate that they will have very limited resources, whereas objects in category 5 indicate that they will have very powerful hardware capabilities. Second, we classify mitigation techniques, suggested to implement our security guidelines, into five SLCs; SLC 1 indicates that it will have weak protection measures, while SLC 5 will have very strong protection measures. Third, we assign SLCs to IoT objects based on their hardware capabilities. Fourth, we propose a communication plan that allows our objects not only to communicate securely with each other but also with the Internet. Moreover, such plan will prevent unexpected use of IoT data. Finally, we propose a four-step method in which the legacy objects can integrate securely with our objects. Our framework also can be used to address several attacks against IoT and also solve some of IoT security challenges, as long as our methodology is precisely followed. To demonstrate the feasibility and application of our suggested framework, we have utilized a smart home system as a case study.

In the future, we will propose a method that provides protection against an insider attacker, since it is the main threat to our methodology. Moreover, we will perform penetration tests on the actual IoT objects equipped with our proposed SLCs to evaluate the performance penalty as well as security benefits of using such framework.

References

- [1] Kouicem, D.E.; Bouabdallah, A.; Lakhlef, H.; Kouicem, D.E.; Bouabdallah, A.; Lakhlef, H. Internet of things security : A top-down survey **2018**.
- [2] Riahi Sfar, A.; Natalizio, E.; Challal, Y.; Chtourou, Z. A roadmap for security challenges in the Internet of Things. *Digital Communications and Networks* **2018**, *4*, 118–137. doi:10.1016/j.dcan.2017.04.003.
- [3] Abdul-Ghani, H.A.; Konstantas, D. A Comprehensive Study of Security and Privacy Guidelines, Threats, and Countermeasures: An IoT Perspective. *Journal of Sensor and Actuator Networks* **2019**, *8*, 38. doi:10.3390/jsan8020022.
- [4] Abdulghani, H.A.; Nijdam, N.A.; Collen, A. A Study on Security and Privacy Guidelines , Countermeasures , Threats : IoT Data at Rest Perspective **2019**. *11*, 1–36. doi:10.3390/sym11060774.
- [5] Seungyong, Yoon; Jeongnyeo, K.Y.J. Security Considerations Based on Classification of IoT Device Capabilities. *SERVICE COMPUTATION 2017 : The Ninth International Conferences on Advanced Service Computing* **2017**, pp. 1–63.
- [6] Chang, C.t.; Chang, C.y.; Dario, R.; Martinez, B.; Chen, P.t.; Chen, Y.d. An IoT Multi-Interface Gateway for Building a Smart Space **2015**. pp. 56–60.
- [7] Rodríguez, J.D.; Schreckling, D.; Posegga, J. Addressing data-centric security requirements for IOT-based systems. *Proceedings - 2016 International Workshop on Secure Internet of Things, SIoT 2016* **2017**, pp. 1–10. doi:10.1109/SIoT.2016.007.
- [8] John Treadway. Using an IoT gateway to connect the Things to the cloud.
- [9] Raza, S.; Trabalza, D.; Voigt, T. 6LoWPAN compressed DTLS for CoAP. *Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2012* **2012**, pp. 287–289.
- [10] Kumar, S.S. draft-keoh-dice-dtls-profile-iot-00 - Profiling of DTLS for CoAP-based IoT Applications **2013**.
- [11] Hartke, K. draft-hartke-dice-practical-issues-01 - Practical Issues with Datagram Transport Layer Security in Constrained Environments **2014**.
- [12] Sethi, M.; Arkko, J.; Keranen, A. End-to-end security for sleepy smart object networks. *Proceedings - Conference on Local Computer Networks, LCN* **2012**, pp. 964–972.
- [13] Kothmayr, T.; Schmitt, C.; Hu, W.; Brunig, M.; Carle, G. A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication. *Proceedings - Conference on Local Computer Networks, LCN* **2012**.

REFERENCES

REFERENCES

- [14] Kefalakis, N.; Aberer, K. OpenIoT: Open Source Internet-of-Things in the Cloud **2017**. 10218. doi:10.1007/978-3-319-56877-5.
- [15] Fremantle, P.; Scott, P. A survey of secure middleware for the Internet of Things **2017**. doi:10.7717/peerj-cs.114.
- [16] Bazzani, M.; Conzon, D.; Scalera, A.; Spirito, M.A.; Trainito, C.I. Enabling the IoT paradigm in E-health solutions through the VIRTUS middleware. *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012* **2012**, pp. 1954–1959. doi:10.1109/TrustCom.2012.144.
- [17] Eisenhauer, M.; Rosengren, P.; Antolin, P. HYDRA: A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems. In *The Internet of Things*; Springer New York: New York, NY, 2010; pp. 367–373. doi:10.1007/978-1-4419-1674-7{_}36.
- [18] Renner, T.; Kliem, A.; Kao, O. The Device Cloud - Applying Cloud Computing Concepts to the Internet of Things. *Proceedings - 2014 IEEE International Conference on Ubiquitous Intelligence and Computing, 2014 IEEE International Conference on Autonomic and Trusted Computing, 2014 IEEE International Conference on Scalable Computing and Communications and Associated Sy* **2014**, pp. 396–401. doi:10.1109/UIC-ATC-ScalCom.2014.106.
- [19] Moosavi, S.R.; Gia, T.N.; Rahmani, A.M.; Nigussie, E.; Virtanen, S.; Isoaho, J.; Tenhunen, H. SEA: A Secure and Efficient Authentication and Authorization Architecture for IoT-Based Healthcare Using Smart Gateways. *Procedia Computer Science* **2015**, 52, 452–459. doi:10.1016/j.procs.2015.05.013.
- [20] Turab, N.M. Internet of Things: A Survey of Existing architectural models and their security Protocols. *IJCSNS International Journal of Computer Science and Network Security* **2017**, 17.
- [21] Akram Abdul-Ghani, H.; Konstantas, D.; Mahyoub, M. A Comprehensive IoT Attacks Survey based on a Building-blocked Reference Model. *IJACSA International Journal of Advanced Computer Science and Applications* **2018**, 9. doi:10.14569/IJACSA.2018.090349.
- [22] Broadband Internet Technical Advisory Group. Internet of things (IoT) security and privacy recommendations: a uniform agreement report. Technical Report November, Broadband Internet Technical Advisory Group, 2016.
- [23] OWASP. IoT Security Guidance.
- [24] IoT Security Foundation. IoT Security Compliance Framework. *IoT Security Foundation: Best Practice User* **2017**.
- [25] El-Attar, M.; Abdul-Ghani, H.A. Using security robustness analysis for early-stage validation of functional security requirements. *Requirements Engineering* **2016**. doi:10.1007/s00766-014-0208-9.
- [26] Ahvanooy, M.T.; Li, Q.; Rabbani, M.; Rajput, A.R. A Survey on Smartphones Security: Software Vulnerabilities, Malware, and Attacks. Technical Report 10, 2017.
- [27] Mohsen Nia, A.; Jha, N.K. A Comprehensive Study of Security of Internet-of-Things. *IEEE Transactions on Emerging Topics in Computing* **2016**.
- [28] Sen, J. Security in Wireless Sensor Networks.
- [29] Yang, G.; Dai, L.; Wei, Z. Challenges, threats, security issues and new trends of underwater wireless sensor networks. *Sensors (Switzerland)* **2018**, 18. doi:10.3390/s18113907.
- [30] Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey **2010**. doi:10.1016/j.comnet.2010.05.010.
- [31] Cherdantseva, Y.; Hilton, J. A reference model of information assurance & security. *Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES 2013*, 2013, pp. 546–555. doi:10.1109/ARES.2013.72.
- [32] Perera, C.; McCormick, C.; Nuseibeh, B. Privacy-by-Design Framework for Assessing Internet of Things Applications and Platforms. *IoT'16* **2016**.
- [33] Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *FGCS* **2013**.
- [34] L. Atzori, A.I.; Morabito, G. The Internet of Things: A survey. *Computer Networks* **2010**, 54, 2787–2805. doi:10.1016/J.COMNET.2010.05.010.
- [35] Cisco. The Internet of Things Reference Model. *Internet of Things World Forum* **2014**.
- [36] Pierce, L.; Tragoudas, S. Multi-level secure JTAG architecture. *Proceedings of the 2011 IEEE 17th International On-Line Testing Symposium, IOLTS 2011* **2011**, pp. 208–209. doi:10.1109/IOLTS.2011.5993845.

REFERENCES

REFERENCES

- [37] Moriyama, D.; Matsuo, S.; Yung, M. PUF-Based RFID Authentication Secure and Private under Memory Leakage, 2014.
- [38] Hristozov, S.; Heyszl, J.; Wagner, S.; Sigl, G. Practical Runtime Attestation for Tiny IoT Devices. *Proceedings 2018 Workshop on Decentralized IoT Security and Standards* **2018**.
- [39] Doddapaneni, K.; Lakkundi, R.; Rao, S.; Kulkarni, S.G.; Bhat, B. Secure FoTA Object for IoT. *Proceedings - LCN Workshops 2017* **2017**.
- [40] Alliance, A.S.C. Embedded Hardware Security for IoT Applications. *A Smart Card Alliance Internet of Things Security Council White Paper* **2016**.
- [41] Mauw, S.; Piramuthu, S. A PUF-based authentication protocol to address ticket-switching of RFID-tagged items. *springer* **2013**, 7783 LNCS, 209–224. doi:10.1007/978-3-642-38004-4{_}14.
- [42] Saiful Islam Mamun, M.; Sultanul Kabir, A.; Sakhawat Hossen, M.; Hayat Khan, M. Policy based intrusion detection and response system in hierarchical WSN architecture. Technical report, 2012.
- [43] Dofe, J.; Frey, J.; Yu, Q. Hardware security assurance in emerging IoT applications. *Proceedings - IEEE International Symposium on Circuits and Systems* **2016**, 2016-July.
- [44] Tomić, I.; McCann, J.A. A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols. *IEEE Internet of Things Journal* **2017**, 4, 1910–1923.
- [45] Moskowitz, R. HIP Diet EXchange (DEX). Technical report, 2011.
- [46] Kamesh.; Sakthi Priya, N. Secure communication for the Internet of Things— a comparison of link-layer security and IPsec for 6LoWPAN. *International Journal of Applied Engineering Research* **2014**, 9, 5968–5974.
- [47] Granjal, J.; Monteiro, E.; Sa Silva, J. Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys & Tutorials* **2015**, 17, 1294–1312.
- [48] Otte, P.; de Vos, M.; Pouwelse, J. TrustChain: A Sybil-resistant scalable blockchain. *Future Generation Computer Systems* **2017**.
- [49] Gonzalez, C.; Charfadine, S.M.; Flauzac, O.; Nolot, F. SDN-based security framework for the IoT in distributed grid. *2016 International Multidisciplinary Conference on Computer and Energy Science, SpliTech 2016* **2016**, pp. 1–5.
- [50] Yan, Z.; Wang, M.; Li, Y.; Vasilakos, A.V. Encrypted Data Management with Deduplication in Cloud Computing. *IEEE Cloud Computing* **2016**, 3, 28–35. doi:10.1109/MCC.2016.29.
- [51] Xu, Y.; Qin, X.; Yang, Z.; Yang, Y.; Huang, C. An algorithm of k-anonymity for data releasing based on fine-grained generalization. *Journal of Information and Computational Science* **2012**, 9.
- [52] Narendra, N.C.; Nayak, S.; Shukla, A. Managing large-scale transient data in IoT systems. 2018 10th International Conference on Communication Systems and Networks, COMSNETS 2018, 2018, Vol. 2018-Janua, pp. 565–568. doi:10.1109/COMSNETS.2018.8328274.
- [53] Bokefode, J.D.; Bhise, A.S.; Satarkar, P.A.; Modani, D.G. Developing A Secure Cloud Storage System for Storing IoT Data by Applying Role Based Encryption. *Procedia Computer Science* **2016**.
- [54] Sun, W.; Yu, S.; Lou, W.; Hou, Y.T.; Li, H. Protecting Your Right: Verifiable Attribute-Based Keyword Search with Fine-Grained Owner-Enforced Search Authorization in the Cloud. *IEEE Transactions on Parallel and Distributed Systems* **2016**, 27.
- [55] Kumar, A.; Narendra, N.C.; Bellur, U. Uploading and replicating internet of things (IoT) data on distributed cloud storage. *IEEE International Conference on Cloud Computing, CLOUD* **2017**, pp. 670–677.
- [56] Yu, Z.; Au, M.H.; Xu, Q.; Yang, R.; Han, J. Towards leakage-resilient fine-grained access control in fog computing. *Future Generation Computer Systems* **2018**, 78, 763–777.
- [57] Yohan, A.; Lo, N.W. An Over-the-Blockchain Firmware Update Framework for IoT Devices. 2018 IEEE Conference on Dependable and Secure Computing (DSC). IEEE, 2018, pp. 1–8. doi:10.1109/DESEC.2018.8625164.
- [58] Yohan, A.; Lo, N.w.; Achawapong, S. Blockchain-based Firmware Update Framework for Internet-of-Things Environment. *Conf. Information and Knowledge Engineering* **2020**, pp. 151–155.

Sample Availability: Samples of the compounds are available from the authors.