electronics

MDPI

Review

IoT Security Challenges: Cloud and Blockchain, Postquantum Cryptography, and Evolutionary Techniques

Stefan Balogh, Ondrej Gallo, Roderik Ploszek *, Peter Špaček and Pavol Zajac

Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovakia; stefan.balogh@stuba.sk (S.B.); ondrej.gallo@stuba.sk (O.G.); peter.spacek@stuba.sk (P.Š.); pavol.zajac@stuba.sk (P.Z.)

* Correspondence: roderik.ploszek@stuba.sk

Abstract: Internet of Things connects the physical and cybernetic world. As such, security issues of IoT devices are especially damaging and need to be addressed. In this treatise, we overview current security issues of IoT with the perspective of future threats. We identify three main trends that need to be specifically addressed: security issues of the integration of IoT with cloud and blockchains, the rapid changes in cryptography due to quantum computing, and finally the rise of artificial intelligence and evolution methods in the scope of security of IoT. We give an overview of the identified threats and propose solutions for securing the IoT in the future.

Keywords: Internet of Things; cloud; blockchain; postquantum; evolution; artificial intelligence

Citation: Balogh, S.; Gallo, O.; Ploszek, R.; Špaček, P.; Zajac, P. IoT Security Challenges: Cloud and Blockchain, Postquantum Cryptography, and Evolutionary Techniques. *Electronics* **2021**, *10*, 2647. https://doi.org/10.3390/electronics10212647

Academic Editor: Paulo Ferreira

Received: 5 October 2021
Accepted: 25 October 2021
Published: 29 October 2021

1. Introduction

The perceived reality of every person consists not only of physical dimensions but includes a significant virtual presence in cyberspace. The cyberspace dimension, however, is not separate: a huge array of connected sensors brings data from the physical world to cyberspace. These data influence the behavior of people connected to cyberspace, as well as feed back to processes in the physical world, especially in control systems. Similarly, data produced only in cyberspace can influence the physical world either by influencing human minds or control systems connected to cyberspace.

The connected physical and cybernetic world faces many important questions, such as: What if the data are incorrect or even malicious? What if the processes are incorrectly programmed or are outright programmed to produce harmful results? Can people with wrong intentions influence our cybernetic systems and, through them, the physical world in unexpected or outright forbidden ways? We know that the answer is yes, and the potential of physical harm through the virtual world is real. Thus, it is critically important to focus on security aspects of cybernetic reality, especially in domains, where it has a strong interaction with the physical world.

The core of the interactions between physical and virtual worlds is due to the emergence and spread of the Internet of Things. Similar to the classical Internet, the Internet of Things is extremely complex. In the security domain, a core principle is to keep things simple. Complex things are difficult to secure: the attacker has an advantage, as he only needs to find a single chain of exploitable vulnerabilities to achieve his goals. On the other hand, the defender needs to protect all parts of the system, and all interactions between the system, its components, and the rest of the world. The security of complex systems, such as the Internet of Things, requires a combination of partial security solutions, which create a further potential for attacks. Nevertheless, we need to study even these partial solutions, because without them the attackers' task becomes trivial, and security nonexistent.

In this article, we focus on emerging security technologies that are promising to provide security for IoT applications in the (near) future. Cloud storage and blockchain

technologies provide secured distributed databases for transparent and verifiable deployment and device management. Postquantum cryptography provides tools to secure devices against future quantum attackers. Evolutionary algorithms are used to find solutions to hard security problems by solving related optimization tasks. Evolutionary techniques are connected to the development of artificial intelligence that oversees the security of our IoT devices and report and prevents possible security incidents.

2. IoT Security Challenges

In this survey, we focus on the security of IoT devices, products, and technologies. IoT devices have become one of the most common attack targets for cybercriminals. A rapidly rising number of IoT devices acerbates these problems even further. Between 2019 and 2030, the number of IoT-connected devices in the world is predicted to grow from 7.6 billion to 24.1 billion [1]. This includes connected devices, sensors, actuators, GPS- and mobile-enabled devices, and the expected further innovations in smart technologies. These devices are integrated to form hybrid networks based upon concepts such as the Internet of Things (IoT), Smart Grids, sensor networks, etc.

The authors in [2] suggest cooperation between IoT and cloud computing because the devices used in an IoT environment have limitations such as low power, low capacity, and limited performance. Efficient services can be created by combining IoT devices with cloud computing technology. In [3], the concept of the Internet of cloud is discussed. Traditional approaches to the IoT cannot satisfy both demands of low cost and simplicity—either the things become more expensive and complex or limits on their computation resource needs are imposed. However, the cloud presents a solution with the potential to satisfy both demands. Thus, cloud computing provides an alternative solution, presenting the IoT with a virtually limitless source of computing power, easily accessible via the Internet, with better resilience, and at a lower cost.

It is already noted in [3] that the pairing of cloud and IoT has an impact on safety. IoT adopting cloud services has also brought new security challenges.

The security aspects of IoT and cloud computing are discussed in more existing articles. Some of the new and interesting challenges in this area are highlighted in [4]. Security and privacy are becoming key challenges in the deployment of IoT infrastructure. While cloud security is a well-documented challenge, the pairing between the cloud and the IoT presents additional concerns. In [5], a survey of IoT and cloud computing with a focus on the security issues of both technologies is presented. They also connect these areas with another technology, called Mobile Cloud Computing (MCC). Mobile cloud computing is defined as an integration of cloud computing technology and mobile devices to make mobile devices resourceful in terms of computational power, memory, storage, energy, and context awareness.

According to [5], the main security problems are an outdated OS and weak passwords. According to [5], security has not always been considered in product design, due to the idea of networking appliances and other objects being relatively new. IoT products are often sold with old and unpatched embedded operating systems and software. Furthermore, purchasers often fail to change the default passwords on smart devices—or if they do change them, fail to select sufficiently strong passwords.

The security of IoT and cloud systems is becoming more intertwined. According to [6], the first IoT botnet was found in December of 2013, with more than 25 percent of the botnet made up of devices other than computers, including smart TVs, baby monitors, and other household appliances.

Since there are many different technologies and platforms deployed in IoT, it is difficult to create a unified security strategy. Some devices simply do not have sufficient computing capacity and/or memory for implementing security precautions. IoT devices are often powered by batteries with limited capacity and must save energy. Securing an IoT device against some types of attacks causes significant increases in energy consumption;

therefore, it is important to first identify possible threats and then implement appropriate countermeasures for the specific architecture of the developed IoT system.

2.1. Security Model for IoT, Standards and Protocols

Security modeling is an important preliminary for building secure systems. In this section, we summarize the security model for IoT infrastructure we used. In general, we can consider IoT area to suffer from all standard classes of adversaries with attributes typical to internet-based attackers. Thus, we can use the general models of attackers for internet applications, such as one introduced in Chapter 1 of the book [7].

We can point out some specific differences when considering IoT: Specifically for industrial IoT, industrial espionage agents can play a more significant role as threat agents. Their specific objectives include industrial secrets and know-how and a potential disruption of industrial processes.

A dangerous category of attackers with potentially high impact are cyber terrorists. These attackers might target specific IoT devices that have the potential for a physical world impact as a consequence of a cyber attack. Their objectives are thus a real-world (physical) impact, where IoT device represents a transitional asset, instead of a final aim of the attacker.

Another important difference between the general security model and IoT specific model is that attackers' capabilities can be significantly enhanced by a physical access to IoT devices, such as sensors or physically unprotected computing nodes.

A significant problem for IoT security is a lack of standards. The IoT solution integrates various kinds of hardware types, communication protocols, and services. This is a double-edged sword that provides comfort to users but can also create a large number of security threats and attacks.

In Figure 1, we summarize the attack types that can be used by the attacker on various layers of IoT solution. In a specific security modeling, the IoT builders need to map possible attacks to their IoT architecture and assign appropriate risks according to asset and attacker models. The attack-type features in Figure 1 are described in more details in the rest of this section. Note that when modeling attackers and their capabilities, we should consider all parts of an IoT solution, including all parts of the solution on the physical layer, network layer, and application layer. Attackers try to target the weakest part of the system. A security breach on one layer can undermine potential defenses on other layers; thus, it is also necessary to include crosslayer security in the security model.

Survey [8] analyzes various research challenges and open issues related to the security of IoT protocols, on the network and application layers. The survey describes different types of communication protocols, and protocols used for security based on the IoT layer architecture. A significant lack of standards for IoT is also pointed out in [9], which primarily analyzes various types of IoT architectures. Different architectures have been proposed for IoT, such as three-layer [10], middleware-based architecture [11], service-oriented architecture (SOA) [12,13], four-layer [14], and five-layer [11].

In Figure 1, we adopted only the basic three-layer architecture model, with different classification of layers related to security. IoT architecture and layer description are important for security modeling, as each layer of the IoT model is connected to specific security challenges and, at the same time, a possibility to enforce security and privacy standards and protocols (see also [15] for more details). Study [9] surveys advanced features of IoT solutions such as IoT data, machine learning algorithms, and light encryption algorithms, and propose a new compacted and optimized architecture for IoT based on five layers. A more fine-grained model can lead to a better security model but can be more complex to analyze and properly understand all interactions.

Study [15] also proposed new IoT layered models: generic and stretched with the privacy and security components and layers identification. This is a model more suitable for evaluation of solutions that include cloud/edge support. The security protocols and critical management sessions are between each of the architectural layers to ensure the

privacy of the users' information. A categorization of attacks can be also performed relative to standard network layers. E.g., survey [16] summarized the common attacks and security issues according to network layers and protocols.

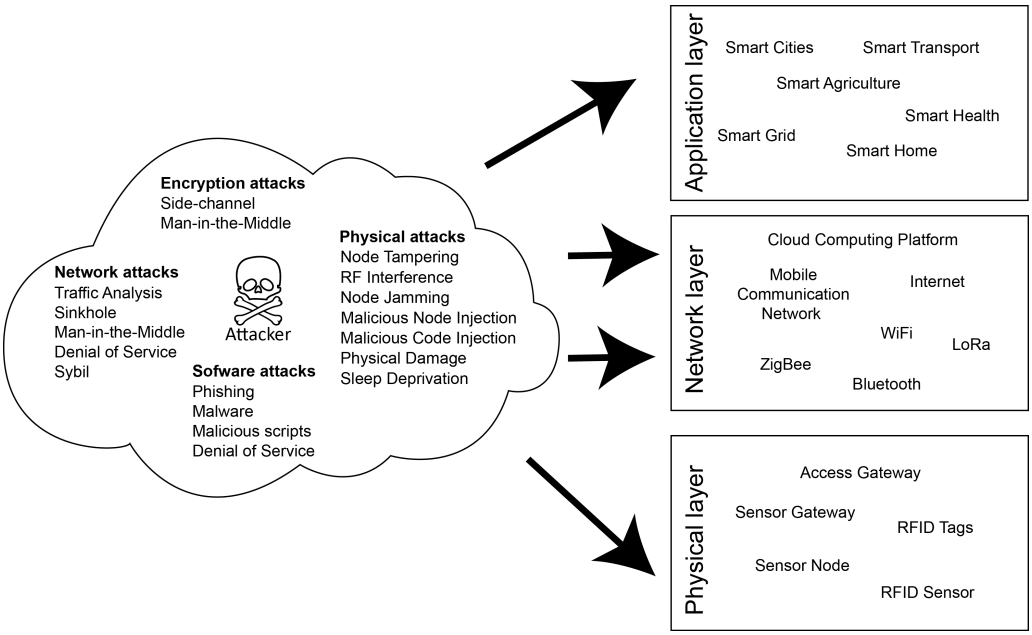

Figure 1. A prerequisite for the attack model for IoT: create a mapping between possible attacks to architectural components of your solution.

A comprehensive study [17] tried to characterize the types of attacks on IoT into four categories: Physical attack, Network attack, Software attack, and Encryption Attack. We adopt the methodology from [17] and characterize the attacks in more details in the following subsections.

2.2. Physical Attacks

This category includes attacks targeting the hardware itself.

- Node Tampering—to perform this attack, the attackers must have physical access to the IoT device. Their goal is to obtain sensitive information such as the encryption key used to communicate with other nodes. According to the authors [18], it is possible to characterize these attacks as invasive and noninvasive. An invasive attack requires expensive equipment because the attacker tries to obtain the contents of the processor's memory by directly observing the semiconductor chip. Noninvasive methods consist of gaining access to the bus, which can be used to access the microprocessor's memory. The JTAG bus is very often harnessed for these purposes. In this way, it is possible to cause great damage, because it is possible, e.g., overwrite the bootloader of the processor with its bootloader and activate reads and writes operations in memory at the request of the attacker. According to [19], it is possible to protect against this attack relatively easily, by detecting an intrusion into the device box. Mechanical switches or additional sensors can be used to detect fluctuations in the supply voltage. A problem with using this countermeasure can be a frequent false alarm.
- RF interference—interferences are caused by transmitting several devices at the same time on the same frequency. An attacker does not have to transmit any data; it is enough to transmit noise on the carrier or subcarrier frequency of a given communication channel. The goal of this attack is to achieve denial of service.
- Node Jamming—this attack is mainly known from Wireless Sensor Networks (WSN). In WSN, the communication between nodes is essential; therefore, rapid attack detection is highly desirable. To successfully execute the attack, the attacker needs to have a high understanding of the communication protocol. Publication [20] describes

this attack in detail. The authors of the article also suggest various countermeasures, e.g., channel hopping, frequency hopping, and spread-spectrum modulation. It is also possible to use software solutions related to the modification of the communication protocol. By adjusting the routing, it is possible to avoid jammed areas. JAM (jammed-area mapping protocol), SAD-SJ (self-adaptive and decentralized MAC-layer), or JAM-BUSTER protocols are suitable.

- Malicious Node Injection—an attacker tries to cause a collision in the network. It is a coordinated attack of several malicious nodes. To perform the attack, the attacker must have certain data of the node to be attacked (e.g., encryption key). The attack consists of two phases. In the first phase, a copy of the node whose data have been compromised is created. This first malicious node has the properties of a legitimate node, but of course, it has other features that make it malicious. The compromised node is isolated from the network (removed or depleted its power). The malicious node creates its copy and attacks another suitable node in a coordinated way. When a legitimate node is requested (either directly or only to forward a message), these two malicious nodes create a collision. The victim never receives or forwards the message, and the other legitimate nodes mark it as malicious or defective. As a result, this node is excluded from the network. It is assumed that the network has certain protection elements to detect malfunctioning nodes. This attack can effectively bypass these bases of protection. A countermeasure could be the MOVE protocol developed by the authors of [21]. It works on the principle of monitoring the transmission of packets in nodes, taking into account the mobility of nodes in the network.
- Physical Damage—this is an attack causing a denial of service. It is necessary to equip IoT devices with quality boxes with simultaneous detection of such an attack in the form of an antitamper technique to mitigate it [19].
- Sleep Deprivation Attack—the IoT device is mostly battery powered and therefore has a limited life. For this reason, IoT devices have implemented sleep modes with varying degrees of energy savings. The purpose of this attack is to prevent IoT devices from going into sleep mode. In this way, the devices run out of power very quickly and switch off permanently. There are several ways to perform this attack. The first way is the so-called barrage attack. In this scenario, the attacker constantly bombs the victim with legitimate requests and thus does not allow it to activate the sleeping mode. This method is simple to implement but can also be easily detected. The second method is based on querying the node in a more sophisticated way. Ultimately, the attack also prevents the IoT device from going to sleep, but it takes longer to drain the battery entirely compared to the previous case. One suitable approach against the sleep deprivation attack is the solution proposed by the authors in [22]. The solution is based on reducing the chance for an attacker to become the central node of the cluster (cluster heads).
- Malicious Code Injection—is a dangerous attack that, if the attacker succeeds, can cause extensive damage. An example is the Stuxnet worm, which has spread to PLC devices controlling various industrial processes. Another type of attack can take control of a large number of IoT devices and launch a large-scale distributed denial of service (DDoS) on the IT infrastructure. An example is the Mirai malware [23]. The attack aims to get full control over the IoT device. An attacker can, for example, steal confidential data from the device or force the victim to carry out the attacker's commands and thus take part in other malicious activities. The attacker exploits the weaknesses of the IoT devices. The most attractive IoT devices for an attacker are those devices that have relatively large computing power and have an operating system, e.g., various IP cameras, routers, or popular hardware platforms such as Raspberry Pi, BeagleBone, or ESP32. Authors in [24] also found a vulnerability in a less powerful platform, Arduino Yún. The main idea of the attack is the so-called memory corruption, specifically buffer overflows and control flow hijacking. A known protection against such attacks is address space layout randomization (ASLR). For low-power

IoT devices, implementing memory randomization can be challenging. The author of the publication [25] managed to implement such protection using external FLASH memory and an additional ATmega processor. Such solutions are possible on less powerful devices but always at the expense of energy consumption and solution price.

2.3. Network Attacks

- Traffic Analysis Attacks—a prerequisite for the realization of this attack is the possibility of interception of communication between the IoT gateway and users who communicate with the gateway via the Internet. Passive eavesdropping allows an attacker to find out the type of IoT devices and the activity of IoT devices connected to the gateway. Communication can also be encrypted. It does not matter for this attack whether the communication is encrypted or not. Traffic analysis provided data that are needed for other dangerous attacks, e.g., Malicious Code Injection. According to [26], there is no perfect protection against this attack, but it is possible to mitigate this attack. The authors in [27] describe a traffic morphing technique that masks real traffic using dummy traffic. This method can significantly reduce the success of the machine learning technique, which is used for analyzing obtained traffic data.
- Sinkhole Attack—the basic idea of the attack is to compromise the data communication of nearby nodes around the malicious node. There are two main types of countermeasures. The first way is to implement an intrusion detection system such as [28,29]. In general, the disadvantage of these systems is the accuracy and thus the relatively high frequency of false alarms. Another option is proper key management [30], in which the identity of each node is secured using an identity-based encryption algorithm.
- Man-in-the-Middle Attacks—this attack is similar to malicious node injection. In a passive attack, the attacker eavesdrops the communication. If the attack is active, the attacker takes control of the communication. They can delay packets, drop packets, or alter their content. The difference is that the attacker does not have to be part of the network because the whole attack takes place exclusively through a given network communication protocol of the sensor network. The most common protection against MITM is a quality intrusion detection system (IDS). In this solution, a compromise is sought between low latency, high detection rate, low CPU load, and the resulting low power consumption of the algorithm. IDS is usually deployed on hierarchically higher and more powerful devices such as gateways for Fog or Edge devices. Publications [31,32] resolve the problematic properties of IDS on these IoT devices.
- Denial of Service—a more accurate description of the attack is given in the publication [21]. An attacker exploits the TCP-based protocol by sending a disproportionate amount of data requests to the victim's device. In this way, all the free resources of the IoT device are gradually occupied. The IoT device thus does not respond to legitimate data requests and ceases to fulfill its function. According to [33], there are three levels of defense against DoS: attack detection, attack mitigation, and attack prevention. Several approaches are known. These are the various classification algorithms, machine learning algorithms, honeypot, IDS, mutual authentication schemes, and many more. To mitigate the DoS attack, a newly developed IOTA protocol may also be used [34]. IOTA protocol was originally developed to verify IOTA cryptocurrency transactions, and it is designed specifically for IoT.
- Sybil Attack—in this attack, the adversary has several identities in the network. They can either create or steal identities. The adversary can then reduce network performance and cause DoS. If data are sent unencrypted, the attacker can steal it and misuse it for other purposes. They can also forward altered data and significantly disrupt the functionality of the proposed system. Protection against this attack is user authentication, encryption of communication, and an efficient Sybil's node detection algorithm [35,36].

2.4. Software Attacks

These types of attacks are implemented at the application layer of the solution. The most common threats are the following:

- Phishing Attacks—most IoT solutions use websites to control IoT devices, collect data, or visualize them. In this attack, the intruder tries to obtain sensitive data from users, such as the name and password. The intruder uses an email with a link to a fake website to lure private user data. The counterfeit website looks similar to the original, so the user submits his login details freely. Suitable antiphishing software [37,38] is a good countermeasure. It can detect suspicious emails and also has a database of suspected websites.
- Virus, Worm, Trojan horse, Spyware, and Adware—the attacker tries to cause damage to the victim through the attacker's malicious code. Typically, an attacker exploits the vulnerabilities of the IoT device and takes control of it. They can then use the device for another type of attack (e.g., phishing, DDoS, and cyber spying) and spread the malware to other devices. More powerful IoT devices can have a full operating system loaded. Attackers often exploit unsecured default settings (e.g., open service ports, a default admin password, etc.). The diversity of operating systems, communication protocols, and installed software is constantly creating new security threats. As the number of IoT devices connected to the network grows, the risk of malware infection specifically directed against IoT devices and their infrastructure increases [9,39]. A specific problem is ransomware, where IoT is an ideal target for attackers [40]. This is growing more serious as the quality of ransomware implementations has improved in recent years [41]. According to publication [17], there are several countermeasures. Depending on the IoT architecture and capabilities, it is advisable to have a strong antivirus system, use a firewall, or use a honeypot to detect dangerous software signatures. Note that these countermeasures are typically applied on devices with full OS support, and parts of IoT infrastructure, such as servers, gateways, edge devices, or cloud infrastructure.
- Malicious Scripts—an attacker can run a malicious script through a website visited on the Internet and gain control over devices in the entire LAN network of the victim [42]. An attacker could gain access to devices that are hidden behind NATs. The suggested countermeasures from [42] are based on the correct configuration of the webserver.
- Denial of Service—it is also possible to attack the application layer of the IoT device. This attack is primarily an attack on a web server that usually has some more powerful IoT devices. An attacker could also target a web server (or cloud) to which IoT devices send messages.

2.5. Encryption Attacks

The goal of this group of attacks is to obtain a private key from an IoT device. An attacker can gain the necessary data through the various techniques mentioned below.

- Side-channel Attacks—a measure of power consumption of the device during cryptographic operations associated with the private key is the most common way to gain a secret parameter. Simple power analysis or differential power analysis is an example of such attacks. There are other techniques: for example, measuring the EM spectrum emitted by the device; acoustic attacks, where the sound generated by the various components of the IoT device is measured; and time attacks, where the time duration of running program is measured at specially selected values on the input. A more detailed description of previous attacks and countermeasures can be found in publications [43–45].
- Man-in-the-Middle Attacks—an attacker eavesdrops on a user's communication by exchanging the public key. The attacker is in the function of an intermediary. They can inadvertently throw their public key and can read and modify encrypted messages between users [46].

2.6. How to Improve Security

To improve security, IoT devices that need to be directly accessible over the Internet should be segmented into their network segment and have other network access restricted. The network segment should then be monitored to identify potential anomalous traffic, and action should be taken if there is a problem.

In [47], authors mainly focus on the security threats for cloud-based IoT, especially in the aspects of secure packet forwarding with outsourced aggregated transmission evidence generation and efficient privacy-preserving authentication with outsourced message filtering. Besides the traditional data confidentiality and unforgeability, the unique security and privacy requirements in cloud-based IoT are presented:

- Identity Privacy: the mobile IoT user's real identity should be well protected from the public; on the other hand, when some dispute occurs in emergency cases, it can also be effectively traced by the authority.
- Location Privacy: If the adversary knows that the target node with pseudonym PID occasionally visits n locations, sets of nodes' real identities passing by these n locations can be observed. The intersection would reveal the target node's real identity and its private activities in other regions.
- Node Compromise Attack: the adversary extracts from the resource-constrained IoT devices all the private information including the secret key used to encrypt the packets, the private key to generate signatures, and so on, and then reprograms or replaces the IoT devices with malicious ones under the control of the adversary.
- Layer Removing/Adding Attack: the attack occurs when a group of selfish IoT users removes all the forwarding layers between them to maximize their rewarded credits by reducing the number of intermediate transmitters sharing the reward.
- Forward and Backward Security: due to the mobility and dynamic social group formulation in IoT, newly joined IoT users can only decipher the encrypted messages received after but not before they join and revoked IoT users can only decipher the encrypted messages before but not after leaving the cluster.
- Semitrusted and/or Malicious Cloud Security: for the convergence of the cloud with IoT, the security and privacy requirements for the cloud should be specially considered. For outsourced computation, the following three security targets should be achieved:
  - Input privacy: The data owner's inputs should be well protected even from collusion between the cloud and authorized data receivers.
  - Output privacy: The computation result should only be successfully deciphered by authorized data receivers.
  - Function privacy: The underlying function must be well protected even from the collusion of the cloud and malicious IoT users.

In [47], a focus is given on providing security mechanisms for complete cloud systems by implementing encryption and intrusion detection systems. They applied hybrid encryption on data at the cloud client level. This means that both data in the medium as well as stored in the cloud server are secured. Security can be improved by implementing an intrusion detection system that detects the anomaly traffic toward the server and blocks unauthorized and unauthenticated traffic. Specific cipher types might be more suitable for IoT applications [48].

The authors in [49] discuss risks if cloud security is not handled properly:

- Privacy and Legal Compliance Risks: such as identity theft resulting in a privacy breach.
- Common Threats and Vulnerabilities: Common threats to both cloud and traditional computing include eavesdropping, fraud, theft, denial of service, logon, abuse, and network intrusion.

According to them, the principal requirement of a secure cloud-based system is to mitigate any known vulnerabilities in the system and make sure that system performance

is not compromised when it is under external malicious attack. The key factors that they recommend for the secure cloud system are:

- Dependability
- Trustworthiness
- Resiliency
- Availability and Fail-Safe
- Sensitive Data handling and Input validation
- Code practices and Language Options

They suggest a way to ensure that requirements of a secure cloud system are captured unambiguously using the S.M.A.R.T.E.R. method (Specific, Measurable, Achievable, Relevant, Time-Oriented, Evaluate, and Revise). They recommend NIST 33 Security principles as guidance for developing any cloud applications. They highlight that the design principles should also be used as a guideline for cloud application security testing to ensure that the cloud applications are built in the right (secured) way to achieve their goals.

IoT and cloud provide a large attack surface and need a significant effort to achieve optimal security. Different authors suggest several core challenges for the security of IoT and cloud systems. The authors in [47] identify these important challenges:

1. Fine-grained ciphertext access control in cloud-based IoT.
2. Besides data confidentiality, location privacy and query privacy for cloud-based IoT users in location-based service (LBS) should also be protected.
3. Increasing batches of data to be processed securely.
4. Privacy-preserving outsourced data mining in cloud-based IoT.

Authors of [49] consider security policy implementation as the most challenging task in cloud computing for service providers. The key challenges include also:

- Virtualization management.
- Remote Management Vulnerabilities.
- Denial of Service.

In [50], the main challenge is that the attacks are becoming more intelligent and diverse as time passes. Conventional security intrusion incident detection and response technologies typically use pattern-based and behavior-based statistical methods. However, an effective intelligent response method is required. An access control technique based on ontology reasoning was suggested as a solution. This can be achieved by adopting a variety of intelligent reasoning technologies for security intrusion incidents. Various reasoning technologies based on ontology and semantic web technology are being actively studied in intelligent systems. Malicious code detection technologies based on an intelligent access control model, text mining, and natural language processing technologies were proposed in [51].

3. Cloud and Blockchain in IoT Security

In recent years, there have been a lot of proposals for using blockchain technologies as a replacement for cloud storage. According to [52]: "Utilizing blockchain can bring increased security and efficiency of network maintenance. The key feature of blockchain, immutability, brings resistance to unauthorized modifications." There is a large number of papers that focus on blockchain and IoT integration. A comprehensive recent survey of blockchain and IoT integration is provided by [53]. In our work, we focus on security issues related to blockchains, which apply to IoT applications.

First, we need to analyze the differences between blockchain and standard cloud solutions for IoT (see e.g., [54] for a recent survey of the topic). In the core of the Internet of Things is the network of physical objects connecting and exchanging data with other objects over the Internet [55]. These objects can potentially be fully autonomous. However, a typical IoT solution requires a management layer, to provide basic configuration, software updates, monitoring, and other noncore functionality. When creating a complex IoT solution, we have three principal options for creating the management layer:

- Solution hosted by IoT owner (or manufacturer). This solution does not scale well and has additional costs associated with maintenance. It is also prone to a single point of failure security problems: any successful attack on the management node can compromise the whole network. We can also use this category for integration platforms such as [56,57].
- Solution hosted in the cloud. There is a large number of examples, surveyed e.g., in [58]. We can include new trends in this, such as serverless computing [59]. Cloud provider provides scalability of the solution and cares for security. Costs of the cloud solution can be lower than maintenance of own servers, depending on the required services and the infrastructure and personnel costs of the IoT solution owner. The security of the solution depends on the quality of the cloud service, and its costs are typically included in the service cost. This requires trust in the cloud provider and does not remove the single point of failure property. However, we can use multiple providers to provide redundancy and attack resiliency (for an increase in operating costs). A recent study focused on security of cloud based solutions is [60].
- Solution based on peer-to-peer decentralized technology, typically a blockchain solution. There are many recent examples, including [61–66], and the number of solutions is growing quickly. Decentralization removes the requirement of trusting the cloud provider. Costs of the decentralized solution, however, can be significant, and, depending on the technology chosen, the current transaction fees in a blockchain network. The core question is, does the blockchain-based solution avoid a single point of failure property, and does it provide required scalability?

The main confusion comes from the fact, that the term blockchain joins multiple technologies under the same name. In the previous paragraph, we have used the term "blockchain" as an antonym to centralized hosting, either owned or rented on a cloud. However, some blockchain technologies (a private blockchain) can be characterized as centralized hosting. Blockchain is sometimes used to denote distributed databases, distributed ledger, or even a distributed virtual machine (such as Ethereum, see e.g., [67] for its security overview).

Proper scientific blockchain taxonomy is still evolving, see e.g., [68] for current definitions. We define a blockchain as a sequence of blocks joined by cryptographic hashes, typically shared by many peers (in the network). Once the hash of the final block is known, then the history of the chain is immutable. It is computationally infeasible to change previous blocks in such a way that the final hash stays the same. However, any peer can append anything to the chain as a new valid block. The extension of the chain requires a consensus protocol, which provides some security guarantees that all members of the network share the same final block and by extent the whole chain. Examples are Nakamoto consensus based on proof of work [69], proof-of-Stake protocols [70], and others [71], with different degrees of resilience against compromised peers in the network. Once the consensus protocol is correctly specified, the blockchain can provide a public bulletin board: peers can append data to the end (and never remove them), and everyone in the network can read the data, with a guaranteed common history. Such a public bulletin board then can be a base for many other solutions, such as transaction ledger for (crypto-)currencies, publicly shared virtual machine (such as Ethereum), and many others.

What the IoT implementer needs to understand is that blockchain, in general, does not equal cryptocurrencies, or a distributed solution. Here, the important part is also the definition of who the network peers are. We can run a blockchain on a single node (e.g., to build an immutable log file). We can build a blockchain for a closed network (private blockchain). If those nodes are fully trusted, a simple first-come-first-appends consensus might be sufficient. A simple voting consensus can be sufficient if no more than 50% of the peers are compromised. For open blockchains (such as Bitcoin), a potential peer is every device connected to the Internet. Nakamoto proof of work [69] requires that no untrusted peer or group of peers can control more than 50% of the computational resources of the whole network. Thus, the cost of the consensus is extremely high. In a private blockchain

(such as [72]), e.g., a chain with defined peers and restricted access, the cost of consensus agreement can be much lower than on public blockchains. However, private blockchain requires additional security solutions to guarantee correct access control. See, e.g., [73] for an overview of these issues.

3.1. Public Blockchains and IoT Security

By public blockchain, we understand a distributed open peer-to-peer blockchain with a consensus mechanism without central trust. There are many examples of public blockchains and their applications, see, e.g., [74]. In the security sense, a public blockchain is a secure public bulletin board: append-only list of items, which everyone can read, and no one can modify. Note that the history of the chain is immutable only if the security prerequisites of the used technology hold, e.g., there is a trusted majority of nodes in PoS types of protocols, or no attacker can obtain more computing power than other nodes combined in PoW protocols.

Note that open public blockchain does not guarantee legal protection or trust. The main principle of blockchain is replacing the trust in some legal entity (e.g., a cloud provider) with trust in technology (blockchain itself, and the software running the blockchain). Blockchain operations might face various regulatory restrictions, see, e.g., [75]. IoT providers should only select public blockchains that fulfill regulatory mandates. A lack of global standards is a significant problem in this area.

While blockchain technology provides some level of integrity protection, in principle, every operation on a public blockchain is public. Confidential data must be encrypted before submitting them to the blockchain. However, blockchain can reveal important metadata, such as who posted which data when. Hiding techniques (see, e.g., [76]) involve additional costs and might not be sufficient for some use cases.

The availability of blockchain access can be a significant problem. Blockchain operations are costly; thus, posting any information on the blockchain is much more expensive than using standard distributed data storage solutions. Blockchain does not solve the problem of denial of service attacks (see, e.g., [77]), which target the network infrastructure of the IoT clients or command centers.

A significant risk related to a public blockchain is that the security of blockchain access is typically fully dependent on the blockchain peer. Access to a blockchain is based on public-key cryptography, and the blockchain peer must secure their private key on their own. Any security breach that leaks this key means a complete takeover of blockchain-based infrastructure, and the loss of the key means complete loss of any further access to the infrastructure.

Despite security problems of the public blockchain technologies, there are some use cases when public blockchains might be useful in providing security solutions for IoT platforms [78]. Inherent integrity protection and public nonrepudiation make public blockchain suitable as a timestamping mechanism: Block hashes in a public blockchain can be used as a control value of a private blockchain in IoT nodes, e.g., for logging and monitoring. Examples include [79–81]. Public blockchain can also be used to publish checksums (hash values) for patches, manuals, and similar public materials, as a replacement for PKI signatures. Examples include [52,82,83]. Note however that the problem of revocation remains: if the private key for blockchain access is compromised, the attacker can push untrusted updates to IoT devices.

3.2. Private Blockchains and IoT Security

When designing a security solution for IoT, we can consider a private blockchain. Private blockchain does not require a complex consensus mechanism, various protocols are resilient even when some IoT nodes are compromised. Note that mechanisms of public blockchains, such as proof of work are not suitable for private solutions, due to their high costs that reflect the lack of trust in the network. Blockchain storage is not suitable for temporary data, as the data structure of blockchain are append-only. To limit the overall

data storage for blockchain, careful consideration is required, what should be stored in the chain. We can save data storage and verification time by using Merkle trees [84] and only store final hashes in the main chain.

Private blockchain requires security mechanisms similar to standard cloud solutions, including access control, administration, backups, etc. Even private blockchain data structure incurs additional operation costs compared to a standard database solution [85]. As such, we recommend using blockchains only as a partial technological solution for the storage of permanent data items, where keeping an immutable linear ordering is required.

4. Postquantum Cryptography Applications

Cybersecurity is now on the edge of a new era. New results in the development of quantum computer [86] lead to serious consequences. The adversaries have more computing power and new threats appear. Algorithms currently used in IoT devices security, especially for key exchange and digital signatures, are vulnerable to new types of attacks, created by the development of quantum computers. In comparison to classical computers (desktops and laptops), we have another factor to consider. The computing power of attackers is increasing, but we have very limited resources on IoT devices.

4.1. Algorithms Used in IoT Security

There are many protocols used to secure IoT communication. However, as we go deeper, and we look at specific cryptographic algorithms they use, we can see that there is only a limited number of ciphers used in these protocols.

The authors of [87] mentioned the most important protocols, used in IoT. For each layer, we show the protocol and used interesting cryptographic algorithms:

- Physical layer—As we see in [88], most of the protocols of physical layer (DASH7, LoRa) use AES-128 for providing confidentiality of the data.
- Data Link layer—the security is provided by IEEE 802.15.4 [89], which specify several cryptographic options, but all are based on AES (AES-32–AES-128)
- Network Layer—IPsec protocol is a requirement for IPv6—allowing for Diffie–Hellman, ECDH, RSA, AES. Another protocol of network layer, 6LoWPAN protocol, only relies on security of transport layer [90].
- Transport Layer—in the transport layer, we can mainly use two types of protocols, TCP or UDP.
  - For TCP, security is provided by TLS, which in version 1.3 allows AES and ephemeral Diffie–Hellman.
  - UDP is secured by DTLS or QUIC. These protocols allow to use ephemeral Diffie–Hellman for key exchange and AES for data confidentiality.
- Application Layer—CoAP protocol proposes to use DTLS to provide security, and AMQP protocol uses TLS. Therefore, the same algorithms are used as in the transport layer.

In all protocols mentioned above, we can see these algorithms: AES, RSA, or Diffie–Hellman (or ECDH). The question here is: are these algorithms secure against the quantum computer?

4.2. Quantum Algorithms That Threaten Our Cryptography

When we talk about a quantum computer as a threat to modern cryptography, we talk mainly about two algorithms:

1. Shor's algorithm is a quantum computer algorithm for finding prime factors of a given number (integer factorization) in polynomial time. This is enough to break modern asymmetric cryptography since it is based on integer factorization or similar problems.
2. In 1996, Lov Grover published a database search algorithm. One interesting consequence is that Grover's algorithm is able to find the n-bit key with time complexity

$\sqrt{n}$. As Grover's algorithm can brute force more or less any black-box function, we need to reconsider the security of symmetric cryptography used in IoT.

4.2.1. Vulnerable Public Key Crypto-Algorithms

All public-key algorithms currently used for key exchange or digital signatures are broken. RSA cipher, which is based on integer factorization problem, is the obvious victim of Shor's algorithm. Other commonly used ciphers are based on discrete logarithm problem, as Diffie–Hellman or its variant based on the elliptic curves over finite fields (ECDH). As mentioned in [91], Shor's algorithm can be used also for computing discrete logarithms. Proos and Zalka [92] have shown that breaking cryptography based on elliptic curves is even easier than breaking RSA.

4.2.2. Vulnerable Symmetric Crypto-Algorithms

Symmetric ciphers are not completely broken with Grover's algorithm. The square root speedup of brute-force attacks requires the change of what is considered to be "secure". As we have shown in Section 4.1, the Advanced Encryption Standard (AES) is widely used for providing data confidentiality in IoT. With Grover's algorithm in mind, the security level of AES-128 is lowered to 64 bits. This means that AES settings with a key length of 128 bits or lower is no longer a secure, and AES needs to be used with 192 or 256 bits for key sizes.

4.3. Postquantum Cryptography in IoT

In recent years, the topic of postquantum security has become more and more discussed. In 2016, NIST began a standardization process to replace the algorithms mentioned above. A new standard is required for two categories: Public-key Encryption and Key-Establishment Algorithms and Digital Signature Algorithms.

4.3.1. Specifics of IoT Postquantum Security

As we have shown in Section 4.1, the IoT world uses the same mechanisms that are used in other applications. The security of IoT devices requires us to keep in mind another, very important factor: the limits of these small devices, namely power, processing, and memory limitations. These limits should be considered when choosing the suitable postquantum mechanisms, as well as when creating postquantum protocols. For example, in many proposals, there is a significant disparity in the difficulty of encryption/decryption or signing/verifying. Ephemeral key generation is considerably slower, and key sizes can grow significantly in comparison to currently used keys. The protocols employed in IoT should reflect the properties of these new underlying algorithms and delegate computationally harder tasks to the server side. A correct selection of a suitable post quantum cryptographic algorithm can lower the price of client devices and provide a competitive advantage to IoT vendor.

4.3.2. Data Confidentiality in Postquantum World

As mentioned in the previous chapters, AES is considered resistant to quantum computers but with a key size of 192 bits or more. In most cases, this is a simple solution to quantum-resistant symmetric encryption. In some cases, however, the limits of the devices force us to search for alternatives. Along with AES, Singh et al. in [93] advise TWINE, HIGHT, and PRESENT for use in IoT, but for postquantum security, the key sizes are too small. We can increase the key size to meet the postquantum requirements or look for quantum-resistant ciphers by design. Li et al. in [94] presented stream encryption scheme with variable plaintext. In addition, interesting solutions are lightweight ciphers families SIMON and SPECK, published by the National Security Agency (NSA) in 2013 [95], that are developed for limited devices. In [96], Jang et al. evaluated and compared SPECK and SIMON in terms of quantum resources. In PQCRYPTO's recommendations of long-

term secure postquantum systems [97], AES-256, Salsa20 with a 256-bit key length and Serpent-256 are advised to use for postquantum security, if the limits of the device allow it.

4.3.3. Key Establishment in Postquantum World

Things are more complicated when we discuss public-key algorithms. Here, we need to replace current algorithms and choose new ones. Good replacement algorithms can be found in the third round of the NIST competition. Here are four candidates for the new standard, but not all are suitable for all IoT applications. It is important to choose the right algorithm according to the limits of the device. Some of the postquantum algorithms are memory intensive, others are computationally intensive, etc.

The first candidate, Classic McEliece [98], is a cryptosystem that takes all the best from classical code-based systems. The first public key system based on a decoding problem was introduced by Robert J. McEliece in 1978 [99]. A random error vector is added to the codeword (ciphertext), and these errors are removed during decryption. The advantage of this method is a relatively high level of security. In the more than 40 years since this system was published, several papers examined its security, and the cipher is still strong. An overview of some of the attacks was written by Zajac and Repka in [100].

McEliece cipher can be also used with symmetric cipher in the dual scheme to provide complete encapsulation of data. In the work of Zajac [101], the symmetric key is embedded into the error vector of the McEliece. If the sender does not want to store the whole message in the memory due to some limitations, the encryption can also be streamed.

The NIST candidate brings a quantum-resistant KEM (key exchange mechanism), based on Niederreiter's dual version of the cipher, and uses the same family of codes as the original design, Goppa codes. The disadvantage of this approach is quite a large size of public keys, which have more than one megabyte at 128 bits of security. In addition, key generation is relatively slow. The advantage is the small size of the ciphertext and the rate of encapsulation and decapsulation.

A better option for IoT devices in terms of saving memory and battery life is a lattice-based cryptosystem. In the third round of the NIST competition, we can find three candidates. NTRU [102], based on finding the shortest vector problem and CRYSTALS-KYBER [103] and SABER [104] based on learning with errors problem (LWE). The memory consumption for NTRU is less than 50 kilobytes. Similarly, the CRYSTALS-KYBER and SABER ciphers have public keys with less than 20 kilobytes. All three lattice-based cryptosystems are a bit slower than Classic McEliece but also range from 10 to 15 ns [105] for both encryption and decryption. Hao et al. [106] also published an implementation of NTRU Prime for IoT devices. An interesting comparison of LWE and Error Correction algorithms focused on lightweight devices can be found in Saarinen's work [107].

4.3.4. Quantum-Resistant Lightweight Digital Signatures

The vulnerability of asymmetric cryptography has also resulted in the need for the updating of algorithms for digital signatures. The NIST [108] competition also includes the standardization of a new digital signature. Two candidates in the third round, CRYSTALS-Dilithium [109] and Falcon [110] are lattice based. The underlying hard problem of CRYSTALS-Dilithium is learning with errors (LWE), and Falcon relies on short integer solution problems (SIS). The third candidate for the new standard is Rainbow [111]. It is based on the problem of solving systems of multivariate polynomial equations.

For the comparison and help with choosing the right algorithm for IoT application, we can find information about energy consumption in the work of Roma et al. [112]. When generating a key pair, CRYSTALS-Dilithium consumes the least amount of energy and can complete the process in less than 1 ms. The Falcon is also good: it can generate key pairs in 22 ms. Signing and signature verification are similar, and all three algorithms did an excellent job. By a small difference, Falcon wins, because it creates a signature in 0.69 ms and verifies it in 0.11 ms. Signature sizes are less than 6 kB in all three cases, as well as

the key size for Falcon and CRYSTALS-Dilithium. However, the size of the keys can reach almost 2 MB with the Rainbow algorithm.

All three algorithms are good for IoT applications, the choice depends on the needs of a specific device or application.

4.4. Group Communication Using Limited Devices

There are many more challenges in securing IoT communication in the postquantum world. In the work of Colombo et al. [113], the authors proposed a new scheme for group communication in the Quantum Era. Ongoing experiments focus on the implementation of this scheme on a small device with a low-power ARM Cortex-M4 processor (seCube).

5. Evolutionary Techniques for Security

Evolutionary algorithms are used to solve optimization problems, where the solution search space is too large for a simple brute-force approach. They take inspiration from biology, where a set of organisms (representing solutions) is *evolved* through various techniques, while the laws of evolution (such as natural selection) apply. The goal is to find the global optimum of a fitness function that evaluates the quality of a solution.

The most popular evolution algorithm used is genetic programming, mostly because it is not difficult to implement and it can provide good results to complex problems.

Evolutionary algorithms may not always find the best global solution. The starting population and definitions of evolution operations (such as crossover and mutation) can greatly influence the ability of the algorithm to find a global optimum.

Artificial intelligence and machine learning are promising solutions to IoT security. They can detect abnormal activities on the network, intrusion, and various malware activities. However, these algorithms have to be trained to successfully detect attacks. This is where GA comes in. Current research focuses on using GA for the optimization of neural network parameters or feature selection. For example, Zhang et al. [114] used specially modified GA to set the parameters of a deep belief network.

Another example where GA is used to optimize the performance of a classification algorithm was presented by Alqahtani et al. [115]. They created a botnet attack classifier using an optimized extreme gradient boosting (GXBoost). GA was used to optimize the parameters of the GXBoost model.

Current solutions using GA achieve very good results. GA allows classifiers to be more efficient and effective. However, GA is still sensitive to the initial population, and the global optimum may not always be found. Future research may show how to choose the initial population and how to evolve it so that global optimum may be found with a very high probability.

IoT devices generate a large amount of data containing numerous data points. Even for an expert, it can be difficult to determine which parameters are important for the detection of harmful activity. GA can be used to select features from this data that can be later used in a classifier. Zhang et al. [116] combined ordinary GA with the GWO algorithm, thus eliminating the shortcomings of both algorithms. The selected features were used to train an SVM model. Intrusion detection using this model performed better than previously available methods. In the future, we can expect machine learning to play a major role in malware and intrusion detection. Therefore, it is important to increase the accuracy of these algorithms. Since combining GA with GWO offered better performance, other combinations of various evolutionary algorithms have to be researched to find out which offers the best results.

The advent of 5G networks will further expand the use of new IoT devices. Such a large number of devices requires careful management of spectrum resources so they can maintain a good level of connectivity. One of the management techniques is *cooperative spectrum sensing*, where devices share sensing information and one control node decides on spectrum assignment. In this configuration, malicious devices sending false information can cause the severe degradation of the performance of the network. Khan et al. [117]

proposed the mitigation to these attacks using GA-based soft decision fusion. This scheme achieved better performance and a lower probability of errors than conventional schemes.

One of the important operations in IoT security is the collection of events for the purpose of detecting security incidents and their subsequent mitigation. The technology to process this large collection of data is called *complex event processing*. This processing consists of filtering, normalization, and subsequent aggregation of information. Since IoT devices do not have large persistent storage space, data are not stored, and these operations are executed on the run. The parallelization and distribution of these operations between network nodes is a complex problem. Kotenko and Saenko [118] used GA to optimize the scheme of aggregation functions. The results were again favorable—the network reached higher throughput, and the CPU load was lower than the scheme without GA.

Intrusion detection systems based on machine learning classifiers use previously recorded data to differentiate between normal traffic and an attack. The obvious disadvantage is that the attackers sooner or later develop an attack that is not classified as an attack and allows them to penetrate the network. Mrugula et al. [119] used the principle of coevolution, where two populations of linked organisms are evolved—predator and prey. In this case, GA is used to evolve new attackers (as predators), and these are then used to train an artificial immune system that detects them. They focused on only one type of attack (*interest cache poisoning*) with good results. This work has shown that exploring the attack space may uncover vulnerabilities sooner than they are exploited by attackers. In the future, we can expect more similar systems that automatically generate new attackers to improve the detection of IDS and other security systems.

Another disadvantage of classifiers based on machine learning are *adversial learning samples*. These samples are created from a malicious sample by a careful small modification. The purpose of this modification is to flip the detector result from positive to negative, and the harmful sample thus enters the system. Liu et al. [120] have created a system that uses GA to create new *adversarial samples* of Android malware. They chose to add various application permissions as the modification. With their adversarial samples, they managed to evade malware detection with almost 100% success. A similar result was obtained in [121], where the adversarial samples of network traffic were created. Artificial intelligence and machine learning are being deployed in an increasing number of security areas, including IoT. Thus, we can expect more research in the field of *adversarial samples* and on refining machine learning algorithms to make such samples correctly identified.

Many vulnerabilities in IoT devices are caused by software bugs such as buffer overflow, etc. Source code of IoT terminals is usually closed and therefore not available for independent review. It is therefore necessary to use a different method to detect vulnerabilities in such code. Zhu et al. [122] presented a method in which firmware instructions from an IoT terminal are extracted in a form of genes. Subsequently, these genes may be compared to other genes representing the instructions that are known to contain vulnerabilities. The authors used a manually constructed distance function that computed a similarity between genes. We think that GA can be useful in providing ways to find an optimum distance function that may provide even better detection rates.

Deep learning, a successor of machine learning, is also showing good results, often exceeding previous techniques with better accuracy of prediction and classification [123]. Interesting results were achieved by the authors in [124]. Researchers noticed similarities between layered architecture of deep neural and IoT networks. They suggested decentralized classifier scheme that took advantage of IoT network architecture. As a result, their framework used just 4% of the original transmission capacity while providing results with just a 2.5% deterioration in inference.

Deep learning techniques can also provide better protection against adversarial attacks and transferability attacks, in which an attacker simulates a model with their own deep learning network [125]. The proposed solution by the author is to use adversarial training that generates adversarial examples during the training procedure.

Other aspects of IoT security can utilize deep learning, including those we already mentioned in this section. This includes intrusion detection [126–129] and other types of anomaly detection [130]. A full coverage of these areas would require a separate article.

At present, IoT security research is mainly conducted in simulated environments mainly because such experiments are convenient and simple. The choice of the environment has a great influence on the results of experiments when a seemingly perfect detector in a simulated environment does not work well on a real IoT network. Zhang et al. [116] mentioned a few problems and mistakes that researchers in this field often make. One of them is the use of an old data set from the KDD Cup 1999 competition. As this data set is more than 20 years old, it is outdated and not suitable for use as an IoT network traffic simulation data. Research in this area could be accelerated and improved by creating high-quality and extensive data sets containing real (not simulated) IoT traffic. GA and machine learning can be trained on these data sets with better results that would allow these technologies to transfer into the real world more smoothly. Furthermore, the developed algorithms have to be optimized, so that they can run on the modest hardware that IoT devices offer.

6. Discussion

The security of IoT applications is becoming a critical factor. Due to the widespread adoption of IoT, attacks in the cybernetic domain can now have significant real-world consequences. IoT devices, especially those connected to the cloud providers can also represent problems with privacy, leaking unintended real-world private data.

There are many security options we can consider in future trends in IoT security. Significant challenges are connected to an interaction between IoT devices and the cloud, with an extra layer added by the emerging integration with blockchain technologies. Careful design and consideration need to be given both to basic security properties such as confidentiality, integrity, and availability. However, we must keep in mind the emerging threats that cross cyber and physical boundaries.

A new asymmetry in potential threat assessment comes from the area of quantum computing. With the rapid development of quantum computing, some of the most used cryptographic algorithms, such as RSA, will become obsolete. When considering physical IoT devices that should stay secure during a longer lifetime, we should consider a way to prepare for migration to quantum-safe algorithms.

We consider the security of IoT as a scope for a cybernetic evolution: attackers evolve new techniques, which are then mitigated by new defense mechanisms. Evolutionary techniques and machine learning have many security applications, especially in processing a large number of network traffic and logs produced by IoT devices. It is an interesting question, whether this type of cybernetic evolution that resembles the natural prey–predator relationship, can lead to the emergence of new artificial intelligence techniques.

Author Contributions: Conceptualization, P.Z.; investigation, S.B., O.G., R.P., P.Š. and P.Z.; writing—original draft preparation, S.B., O.G., R.P., P.Š. and P.Z.; writing—review and editing, S.B., O.G., R.P., P.Š. and P.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was sponsored in part by the NATO Science for Peace and Security Programme under grant G5448. This work was supported in part by the Slovak Research and Development Agency under the Contract no. APVV-19-0220.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hatton, M. The IoT in 2030: 24 Billion Connected Things Generating $1.5 Trillion. Iotbusinessnews. 2020. Available online: https://iotbusinessnews.com/2020/05/20/03177-the-iot-in-2030-24-billion-connected-things-generating-1-5-trillion (accessed on 17 August 2021).
2. Zhou, J.; Cao, Z.; Dong, X.; Lin, X. TR-MABE: White-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 2398–2406. [CrossRef]
3. Cook, A.; Robinson, M.; Ferrag, M.A.; Maglaras, L.A.; He, Y.; Jones, K.; Janicke, H. Internet of Cloud: Security and Privacy Issues. In *Cloud Computing for Optimization: Foundations, Applications, and Challenges*; Springer International Publishing: Cham, Switzerland, 2018; pp. 271–301._11. [CrossRef]
4. Díaz, M.; Martín, C.; Rubio, B. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *J. Netw. Comput. Appl.* **2016**, *67*, 99–117. [CrossRef]
5. Stergiou, C.; Psannis, K.E.; Kim, B.G.; Gupta, B. Secure integration of IoT and Cloud Computing. *Future Gener. Comput. Syst.* **2018**, *78*, 964–975. [CrossRef]
6. Rouse, M. IoT Security (Internet of Things Security). IoT Agenda. 2015. Available online: https://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security (accessed on 1 July 2021).
7. Van Oorschot, P.C. *Computer Security and the Internet: Tools and Jewels*; Springer Nature Switzerland AG: Cham, Switzerland, 2020.
8. Yugha, R.; Chithra, S. A survey on technologies and security protocols: Reference for future generation IoT. *J. Netw. Comput. Appl.* **2020**, *169*, 102763. [CrossRef]
9. Mrabet, H.; Belguith, S.; Alhomoud, A.; Jemai, A. A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. *Sensors* **2020**, *20*, 3625. [CrossRef] [PubMed]
10. Ammar, M.; Russello, G.; Crispo, B. Internet of Things: A survey on the security of IoT frameworks. *J. Inf. Secur. Appl.* **2018**, *38*, 8–27. [CrossRef]
11. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutorials* **2015**, *17*, 2347–2376. [CrossRef]
12. Xu, L.D.; He, W.; Li, S. Internet of Things in Industries: A Survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243. [CrossRef]
13. Hammoudeh, M.; Epiphaniou, G.; Belguith, S.; Unal, D.; Adebisi, B.; Baker, T.; Kayes, A.S.M.; Watters, P. A Service-Oriented Approach for Sensing in the Internet of Things: Intelligent Transportation Systems and Privacy Use Cases. *IEEE Sens. J.* **2021**, *21*, 15753–15761. [CrossRef]
14. Suo, H.; Wan, J.; Zou, C.; Liu, J. Security in the Internet of Things: A Review. In Proceedings of the 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China, 23–25 March 2012; Volume 3, pp. 648–651.
15. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT Privacy and Security: Challenges and Solutions. *Appl. Sci.* **2020**, *10*, 4102. [CrossRef]
16. Litoussi, M.; Kannouf, N.; El Makkaoui, K.; Ezzati, A.; Fartitchou, M. IoT security: Challenges and countermeasures. *Procedia Comput. Sci.* **2020**, *177*, 503–508. [CrossRef]
17. Deogirikar, J.; Vidhate, A. Security attacks in IoT: A survey. In Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 10–11 February 2017; pp. 32–37. [CrossRef]
18. Znaidi, W.; Minier, M.; Babau, J.P. *An Ontology for Attacks in Wireless Sensor Networks*; Technical Report; INRIA: Rocquencourt, France, 2008.
19. Elngar, A.; Bhatt, S. IoT-based Efficient Tamper Detection Mechanism for Healthcare Application. *Int. J. Netw. Secur.* **2018**, *20*, 74–80. [CrossRef]
20. Kirti, S.; Bhatt, S. Jamming Attack—A Survey. *Int. J. Recent Res. Asp.* **2018**, *5*, 74–80.
21. Mohapatra, H.; Rath, S.; Panda, S.; Kumar, R. Handling of Man-In-The-Middle Attack in WSN Through Intrusion Detection System. *Int. J.* **2020**, *8*, 1503–1510. [CrossRef]
22. Pirretti, M.; Zhu, S.; Narayanan, V.; McDaniel, P.; Kandemir, M.; Brooks, R. The Sleep Deprivation Attack in Sensor Networks: Analysis and Methods of Defense. *Int. J. Distrib. Sens. Netw.* **2006**, *2*, 267–287. [CrossRef]
23. Sinanović, H.; Mrdovic, S. Analysis of Mirai malicious software. In Proceedings of the 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 21–23 September 2017; pp. 1–5. [CrossRef]
24. Pastrana, S.; Canseco, J.R.; Calleja, A. ArduWorm: A functional malware targeting arduino devices. In *Actas de Jornadas Nacionales de Investigación en Ciberseguridad*; Universidad de Granada: Granada, Spain, 2016.
25. Habibi, J.; Gupta, A.; Carlsony, S.; Panicker, A.; Bertino, E. MAVR: Code Reuse Stealthy Attacks and Mitigation on Unmanned Aerial Vehicles. In Proceedings of the 2015 IEEE 35th International Conference on Distributed Computing Systems, Columbus, OH, USA, 29 June–2 July 2015; pp. 642–652. [CrossRef]
26. Dyer, K.P.; Coull, S.E.; Ristenpart, T.; Shrimpton, T. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In Proceedings of the 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–23 May 2012; pp. 332–346. [CrossRef]
27. Hafeez, I.; Antikainen, M.; Tarkoma, S. Protecting IoT-environments against Traffic Analysis Attacks with Traffic Morphing. In Proceedings of the 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kyoto, Japan, 11–15 March 2019; pp. 196–201. [CrossRef]

28. Stephen, R.; Arockiam, L. An Enhanced Technique to Detect Sinkhole Attack in Internet of Things. *Int. J. Eng. Res. Technol.* **2018**, *5*, 1–4.
29. Cervantes, C.; Poplade, D.; Nogueira, M.; Santos, A. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 606–611. [CrossRef]
30. Yuan, E.; Wang, L. A key management scheme realising location privacy protection for heterogeneous wireless sensor networks. *Int. J. Sens.Netw.* **2020**, *32*, 34–41. [CrossRef]
31. Owusu Agyemang, J.; Jerry, K.; Acquah, I. Lightweight Man-In-The-Middle (MITM) Detection and Defense Algorithm for WiFi-Enabled Internet of Things (IoT) Gateways. *Inf. Secur. Comput. Fraud.* **2019**, *7*, 1–6. [CrossRef]
32. Aliyu, F.; Sheltami, T.; Shakshuki, E.M. A Detection and Prevention Technique for Man in the Middle Attack in Fog Computing. *Procedia Comput. Sci.* **2018**, *141*, 24–31. [CrossRef]
33. Salim, M.M.; Rathore, S.; Park, J.H. Distributed denial of service attacks and its defenses in IoT: A survey. *J. Supercomput.* **2020**, *76*, 5320–5363. [CrossRef]
34. Attias, V.; Vigneri, L.; Dimitrov, V. Preventing Denial of Service Attacks in IoT Networks through Verifiable Delay Functions. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
35. Pu, C. Sybil Attack in RPL-Based Internet of Things: Analysis and Defenses. *IEEE Internet Things J.* **2020**, *7*, 4937–4949. [CrossRef]
36. Vaishnavi, S.; Sethukarasi, T. SybilWatch: A novel approach to detect Sybil attack in IoT based smart health care. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 6199–6213. [CrossRef]
37. Lam, T.; Kettani, H. PhAttApp: A Phishing Attack Detection Application. In Proceedings of the Proceedings of the 2019 3rd International Conference on Information System and Data Mining, Chiang Mai, Thailand, 26–30 July 2019; pp. 154–158. [CrossRef]
38. Rahim, R.; Murugan, S.; Mostafa, R.; Anil, K.; Dubey, D.A.; Rajan, R.; Kulkarni, V.; Dhanalakshmi, K. Detecting the Phishing Attack Using Collaborative Approach and Secure Login through Dynamic Virtual Passwords. *Webology* **2020**, *17*, 524–535. [CrossRef]
39. Hwang, S.Y.; Kim, J.N. A Malware Distribution Simulator for the Verification of Network Threat Prevention Tools. *Sensors* **2021**, *21*, 6983. [CrossRef]
40. Szücs, V.; Arányi, G.; Dávid, Á. Introduction of the ARDS—Anti-Ransomware Defense System Model—Based on the Systematic Review of Worldwide Ransomware Attacks. *Appl. Sci.* **2021**, *11*, 6070. [CrossRef]
41. Ploszek, R.; Švec, P.; Debnár, P. Analysis of encryption schemes in modern ransomware. *Rad Hrvat. Akad. Znan. Umjet. Mat. Znan.* **2021**, *546=25*, 1–13. [CrossRef]
42. Acar, G.; Huang, D.; Li, F.; Narayanan, A.; Feamster, N. Web-based Attacks to Discover and Control Local IoT Devices. In *Proceedings of the Workshop on IoT Security and Privacy*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 29–35. [CrossRef]
43. Sayakkara, A.; Le-Khac, N.A.; Scanlon, M. A Survey of Electromagnetic Side-Channel Attacks and Discussion on their Case-Progressing Potential for Digital Forensics. *Digit. Investig.* **2019**, *29*, 43–54. [CrossRef]
44. Devi, M.; Majumder, A. Side-Channel Attack in Internet of Things: A Survey. In *Applications of Internet of Things*; Mandal, J.K., Mukhopadhyay, S., Roy, A., Eds.; Springer: Singapore, 2021; pp. 213–222._20. [CrossRef]
45. Prouff, E.; Rivain, M. Masking against Side-Channel Attacks: A Formal Security Proof. In *Advances in Cryptology—EUROCRYPT 2013*; Johansson, T., Nguyen, P.Q., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 142–159._9. [CrossRef]
46. Cekerevac, Z.; Dvorak, Z.; Prigoda, L.; Čekerevac, P. Internet of things and the man-in-the-middle attacks—Security and economic risks. *MEST J.* **2017**, *5*, 15–25. [CrossRef]
47. Zhou, J.; Cao, Z.; Dong, X.; Vasilakos, A.V. Security and Privacy for Cloud-Based IoT: Challenges. *IEEE Commun. Mag.* **2017**, *55*, 26–33. [CrossRef]
48. Grošek, O.; Hromada, V.; Horák, P. A Cipher Based on Prefix Codes. *Sensors* **2021**, *21*, 6236. [CrossRef] [PubMed]
49. Deshpande, V.M.; Nair, M.K.; Bihani, A. Optimization of Security as an Enabler for Cloud Services and Applications. In *Cloud Computing for Optimization: Foundations, Applications, and Challenges*; Springer International Publishing: Cham, Switzerland, 2018; pp. 235–270._10. [CrossRef]
50. Choi, C.; Choi, J. Ontology-Based Security Context Reasoning for Power IoT-Cloud Security Service. *IEEE Access* **2019**, *7*, 110510–110517. [CrossRef]
51. Liang, L. Electric Security Data Integration Framework based on Ontology Reasoning. *Procedia Comput. Sci.* **2018**, *139*, 583–587. [CrossRef]
52. Košťál, K.; Helebrandt, P.; Belluš, M.; Ries, M.; Kotuliak, I. Management and Monitoring of IoT Devices Using Blockchain. *Sensors* **2019**, *19*, 856. [CrossRef] [PubMed]
53. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef]
54. Memon, R.; Li, J.; Ahmed, J.; Nazeer, I.; Mangrio, M.I.; Ali, K. Cloud-based vs. Blockchain-based IoT: A comparative survey and way forward. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 563–587. [CrossRef]
55. Patel, K.K.; Patel, S.M.; Salazar, C. Internet of things-IOT: Definition, characteristics, architecture, enabling technologies, application & future challenges. *Int. J. Eng. Sci. Comput.* **2016**, *6*, 6122–6131.

56. Heimgaertner, F.; Hettich, S.; Kohlbacher, O.; Menth, M. Scaling home automation to public buildings: A distributed multiuser setup for OpenHAB 2. In Proceedings of the 2017 Global Internet of Things Summit (GIoTS), Geneva, Switzerland, 6–9 June 2017; pp. 1–6.
57. Gyory, N.; Chuah, M. IoTOne: Integrated platform for heterogeneous IoT devices. In Proceedings of the 2017 International Conference on Computing, Networking and Communications (ICNC), Silicon Valley, CA, USA, 26–29 January 2017; pp. 783–787.
58. Ray, P.P. A survey of IoT cloud platforms. *Future Comput. Inform. J.* **2016**, *1*, 35–46. [CrossRef]
59. Kjorveziroski, V.; Filiposka, S.; Trajkovik, V. IoT Serverless Computing at the Edge: A Systematic Mapping Review. *Computers* **2021**, *10*, 130. [CrossRef]
60. Chen, F.; Luo, D.; Xiang, T.; Chen, P.; Fan, J.; Truong, H.L. IoT Cloud Security Review: A Case Study Approach Using Emerging Consumer-oriented Applications. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–36.
61. Tapas, N.; Merlino, G.; Longo, F. Blockchain-Based IoT-Cloud Authorization and Delegation. In Proceedings of the 2018 IEEE International Conference on Smart Computing (SMARTCOMP), Taormina, Italy, 18–20 June 2018; pp. 411–416. [CrossRef]
62. Palaiokrassas, G.; Skoufis, P.; Voutyras, O.; Kawasaki, T.; Gallissot, M.; Azzabi, R.; Tsuge, A.; Litke, A.; Okoshi, T.; Nakazawa, J.; et al. Combining Blockchains, Smart Contracts, and Complex Sensors Management Platform for Hyper-Connected SmartCities: An IoT Data Marketplace Use Case. *Computers* **2021**, *10*, 133. [CrossRef]
63. Ajayi, O.J.; Rafferty, J.; Santos, J.; Garcia-Constantino, M.; Cui, Z. BECA: A Blockchain-Based Edge Computing Architecture for Internet of Things Systems. *IoT* **2021**, *2*, 610–632. [CrossRef]
64. Wu, C.H.; Tsang, Y.P.; Lee, C.K.M.; Ching, W.K. A Blockchain-IoT Platform for the Smart Pallet Pooling Management. *Sensors* **2021**, *21*, 6310. [CrossRef] [PubMed]
65. Cho, S.; Khan, M.; Pyeon, J.; Park, C. Blockchain-Based Network Concept Model for Reliable and Accessible Fine Dust Management System at Construction Sites. *Appl. Sci.* **2021**, *11*, 8686. [CrossRef]
66. Meng, Y.; Li, J. Data Sharing Mechanism of Sensors and Actuators of Industrial IoT Based on Blockchain-Assisted Identity-Based Cryptography. *Sensors* **2021**, *21*, 6084. [CrossRef] [PubMed]
67. Wang, Z.; Jin, H.; Dai, W.; Choo, K.K.R.; Zou, D. Ethereum smart contract security research: Survey and future research opportunities. *Front. Comput. Sci.* **2021**, *15*, 152802. [CrossRef]
68. Imteaj, A.; Amini, M.H.; Pardalos, P.M. Introduction to Blockchain Technology. In *Foundations of Blockchain*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 3–13.
69. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, *21260*, 1–9.
70. Gaži, P.; Kiayias, A.; Zindros, D. Proof-of-stake sidechains. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 139–156.
71. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM Sigmetrics Perform. Eval. Rev.* **2014**, *42*, 34–37. [CrossRef]
72. Xue, J.; Xu, C.; Zhang, Y. Private blockchain-based secure access control for smart home systems. *KSII Trans. Internet Inf. Syst. (TIIS)* **2018**, *12*, 6057–6078.
73. Lin, I.C.; Liao, T.C. A survey of blockchain security issues and challenges. *Int. J. Netw. Secur.* **2017**, *19*, 653–659.
74. Johar, S.; Ahmad, N.; Asher, W.; Cruickshank, H.; Durrani, A. Research and Applied Perspective to Blockchain Technology: A Comprehensive Survey. *Appl. Sci.* **2021**, *11*, 6252. [CrossRef]
75. Yeoh, P. Regulatory issues in blockchain technology. *J. Financ. Regul. Compliance* **2017**, *25*, 196–208. [CrossRef]
76. Halpin, H.; Piekarska, M. Introduction to Security and Privacy on the Blockchain. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), Paris, France, 26–28 April 2017; pp. 1–3.
77. Mirkin, M.; Ji, Y.; Pang, J.; Klages-Mundt, A.; Eyal, I.; Juels, A. BDoS: Blockchain denial-of-service. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, 9–13 November 2020; pp. 601–619.
78. Carvalho, K.; Granjal, J. Security and Privacy for Mobile IoT Applications Using Blockchain. *Sensors* **2021**, *21*, 5931. [CrossRef]
79. Ren, Y.; Zhu, F.; Sharma, P.K.; Wang, T.; Wang, J.; Alfarraj, O.; Tolba, A. Data Query Mechanism Based on Hash Computing Power of Blockchain in Internet of Things. *Sensors* **2020**, *20*, 207. [CrossRef] [PubMed]
80. Yang, J.; Onik, M.M.H.; Lee, N.Y.; Ahmed, M.; Kim, C.S. Proof-of-Familiarity: A Privacy-Preserved Blockchain Scheme for Collaborative Medical Decision-Making. *Appl. Sci.* **2019**, *9*, 1370. [CrossRef]
81. Ozyilmaz, K.R.; Yurdakul, A. Designing a Blockchain-based IoT with Ethereum, swarm, and LoRa: The software solution to create high availability with minimal security risks. *IEEE Consum. Electron. Mag.* **2019**, *8*, 28–34. [CrossRef]
82. Lombardi, F.; Aniello, L.; De Angelis, S.; Margheri, A.; Sassone, V. A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart grids. In Proceedings of the Living in the Internet of Things: Cybersecurity of the IoT-2018, London, UK, 28–29 March 2018; pp. 1–6. [CrossRef]
83. Kvarda, L.; Hnyk, P.; Vojtech, L.; Neruda, M. Software implementation of secure firmware update in IoT concept. *Adv. Electr. Electron. Eng.* **2017**, *15*, 626–632. [CrossRef]
84. Li, H.; Lu, R.; Zhou, L.; Yang, B.; Shen, X. An efficient merkle-tree-based authentication scheme for smart grid. *IEEE Syst. J.* **2013**, *8*, 655–663. [CrossRef]
85. Lockl, J.; Schlatt, V.; Schweizer, A.; Urbach, N.; Harth, N. Toward trust in Internet of Things ecosystems: Design principles for blockchain-based IoT applications. *IEEE Trans. Eng. Manag.* **2020**, *67*, 1256–1270. [CrossRef]

86. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef] [PubMed]
87. Jayapal, C.; Sultana, P.; Saroja, M.N.; Senthil, J. Security Protocols for IoT. In *Ubiquitous Computing and Computing Security of IoT*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–28._1. [CrossRef]
88. Chacko, S.; Job, M.D. Security mechanisms and Vulnerabilities in LPWAN. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *396*, 012027. [CrossRef]
89. Sastry, N.; Wagner, D. Security Considerations for IEEE 802.15.4 Networks. In *Proceedings of the 3rd ACM Workshop on Wireless Security*; ACM WiSe: New York, NY, USA, 2004; Volume 2004.
90. Narayanan, R.; Jayashree, S.; Philips, N.D.; Saranya, A.M.; Prathiba, S.B.; Raja, G. TLS Cipher Suite: Secure Communication of 6LoWPAN Devices. In Proceedings of the 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, 18–20 December 2019; pp. 197–203. [CrossRef]
91. Ekerå, M.; Håstad, J. Quantum Algorithms for Computing Short Discrete Logarithms and Factoring RSA Integers. In *Post-Quantum Cryptography*; Lange, T., Takagi, T., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 347–363.
92. Proos, J.; Zalka, C. Shor's Discrete Logarithm Quantum Algorithm for Elliptic Curves. *Quantum Info. Comput.* **2003**, *3*, 317–344.
93. Singh, S.; Sharma, P.K.; Moon, S.Y.; Park, J.H. *Advanced Lightweight Encryption Algorithms for IoT Devices: Survey, Challenges and Solutions*; Springer: Berlin Heidelberg, Germany, 2017. Available online: https://link.springer.com/article/10.1007/s12652-017-0494-4 (accessed on 30 September 2021).
94. Li, Y.; Zhang, P.; Huang, R. Lightweight Quantum Encryption for Secure Transmission of Power Data in Smart Grid. *IEEE Access* **2019**, *7*, 36285–36293. [CrossRef]
95. Beaulieu, R.; Shors, D.; Smith, J.; Treatman-Clark, S.; Weeks, B.; Wingers, L. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404. 2013. Available online: https://eprint.iacr.org/2013/404 (accessed on 15 September 2021).
96. Jang, K.; Choi, S.; Kwon, H.; Seo, H. Grover on SPECK: Quantum Resource Estimates. Cryptology ePrint Archive, Report 2020/640. 2020. Available online: https://eprint.iacr.org/2020/640 (accessed on 10 September 2021).
97. Augot, D.; Batina, L.; Bernstein, D.J.; Bos, J.; Buchmann, J.; Castryck, W.; Dunkelman, O.; Güneysu, T.; Gueron, S.; Hülsing, A.; et al. Initial Recommendations of Long-Term Securepost-Quantum Systems. 2015. Available online: http://pqcrypto.eu.org/docs/initial-recommendations.pdf (accessed on 30 August 2021).
98. Chou, T.; Cid, C.; UiB, S.; Gilcher, J.; Lange, T.; Maram, V.; Misoczki, R.; Niederhagen, R.; Paterson, K.G.; Persichetti, E.; et al. Classic McEliece: Conservative Code-Based Cryptography 10 October 2020. 2020. Available online: https://classic.mceliece.org/nist/mceliece-20201010.pdf (accessed on 24 August 2021).
99. McEliece, R.J. A public-key cryptosystem based on algebraic coding theory. *DSN Prog. Rep.* **1978**, *42*, 114–116.
100. Repka, M.; Zajac, P. Overview of the McEliece cryptosystem and its security. *Tatra Mt. Math. Publ.* **2014**, *60*, 57–83. [CrossRef]
101. Zajac, P. Hybrid encryption from McEliece cryptosystem with pseudo-random error vector. *Fundam. Inform.* **2019**, *169*, 345–360. [CrossRef]
102. Chen, C.; Danba, O.; Hoffstein, J.; Hulsing, A.; Rijneveld, J.; Schanck, J.M.; Schwabe, P.; Whyte, W.; Zhang, Z. NTRU: Algorithm Specifications and Supporting Documentation (2019). Available online: https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions (accessed on 25 August 2021).
103. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber algorithm specifications and supporting documentation. *NIST PQC Round* **2017**, *2*, 4.
104. Vercauteren, I.F. SABER: Mod-LWR Based KEM (Round 2 Submission). Available online: https://www.esat.kuleuven.be/cosic/publications/article-3055.pdf (accessed on 20 August 2021).
105. Basu, K.; Soni, D.; Nabeel, M.; Karri, R. NIST Post-Quantum Cryptography-A Hardware Evaluation Study. *IACR Cryptol. EPrint Arch.* **2019**, *2019*, 47.
106. Cheng, H.; Dinu, D.; Großschädl, J.; Rønne, P.B.; Ryan, P.Y.A. A Lightweight Implementation of NTRU Prime for the Post-quantum Internet of Things. In *Information Security Theory and Practice*; Laurent, M., Giannetsos, T., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 103–119.
107. Saarinen, M.J.O. Ring-LWE ciphertext compression and error correction: Tools for lightweight post-quantum cryptography. In Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security, New York, NY, USA, 2 April 2017; pp. 15–22.
108. NIST. Post-Quantum Cryptography. Round 1 Submissions. 2018. Available online: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions (accessed on 19 August 2021).
109. Soni, D.; Basu, K.; Nabeel, M.; Aaraj, N.; Manzano, M.; Karri, R. CRYSTALS-Dilithium. In *Hardware Architectures for Post-Quantum Digital Signature Schemes*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 13–30.
110. Fouque, P.A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. *Submiss. Nist's-Post-Quantum Cryptogr. Stand. Process.* **2018**, *36*, 1–75.
111. Ding, J.; Schmidt, D. Rainbow, a new multivariable polynomial signature scheme. In *International Conference on Applied Cryptography and Network Security*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 164–175.
112. Roma, C.; Tai, C.E.A.; Hasan, M.A. Energy Consumption of Round 2 submissions for NIST PQC Standards. In Proceedings of the Second PQC Standardization Conference, Oakland, CA, USA, 22–25 August 2019.

113. Colombo, C.; Vasco, M.I.G.; Steinwandt, R.; Zajac, P. Secure communication in the quantum era:(group) key establishment. In *Advanced Technologies for Security Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 65–74.
114. Zhang, Y.; Li, P.; Wang, X. Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. *IEEE Access* **2019**, *7*, 31711–31722. [CrossRef]
115. Alqahtani, M.; Mathkour, H.; Ben Ismail, M.M. IoT Botnet Attack Detection Based on Optimized Extreme Gradient Boosting and Feature Selection. *Sensors* **2020**, *20*, 6336. [CrossRef]
116. Davahli, A.; Shamsi, M.; Abaei, G. Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks. *J. Ambient Intell. Humaniz. Comput.* **2020**, *11*, 5581–5609. [CrossRef]
117. Khan, M.S.; Gul, N.; Kim, J.; Qureshi, I.M.; Kim, S.M. A Genetic Algorithm-Based Soft Decision Fusion Scheme in Cognitive IoT Networks with Malicious Users. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 2509081. [CrossRef]
118. Kotenko, I.; Saenko, I. An Approach to Aggregation of Security Events in Internet-of-Things Networks Based on Genetic Optimization. In Proceedings of the 2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 18–21 July 2016; pp. 657–664. [CrossRef]
119. Mrugala, K.; Tuptuk, N.; Hailes, S. Evolving attackers against wireless sensor networks using genetic programming. *IET Wirel. Sens. Syst.* **2017**, *7*, 113–122. [CrossRef]
120. Liu, X.; Du, X.; Zhang, X.; Zhu, Q.; Wang, H.; Guizani, M. Adversarial Samples on Android Malware Detection Systems for IoT Systems. *Sensors* **2019**, *19*, 974. [CrossRef]
121. Liu, X.; Zhang, X.; Guizani, N.; Lu, J.; Zhu, Q.; Du, X. TLTD: A Testing Framework for Learning-Based IoT Traffic Detection Systems. *Sensors* **2018**, *18*, 2630. [CrossRef] [PubMed]
122. Zhu, X.; Li, Q.; Chen, Z.; Zhang, G.; Shan, P. Research on Security Detection Technology for Internet of Things Terminal Based on Firmware Code Genes. *IEEE Access* **2020**, *8*, 150226–150241. [CrossRef]
123. Malhotra, P.; Singh, Y.; Anand, P.; Bangotra, D.K.; Singh, P.K.; Hong, W.C. Internet of Things: Evolution, Concerns and Security Challenges. *Sensors* **2021**, *21*, 1809. [CrossRef] [PubMed]
124. Du, R.; Magnússon, S.; Fischione, C. The Internet of Things As a Deep Neural Network. *IEEE Commun. Mag.* **2020**, *58*, 20–25. [CrossRef]
125. Lin, T. Deep Learning for Iot. In Proceedings of the 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC), Austin, TX, USA, 6–8 November 2020.
126. Albulayhi, K.; Smadi, A.A.; Sheldon, F.T.; Abercrombie, R.K. Iot Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors* **2021**, *21*, 6432. [CrossRef]
127. Alsoufi, M.A.; Razak, S.; Siraj, M.M.; Nafea, I.; Ghaleb, F.A.; Saeed, F.; Nasser, M. Anomaly-Based Intrusion Detection Systems in Iot Using Deep Learning: A Systematic Literature Review. *Appl. Sci.* **2021**, *11*, 8383. [CrossRef]
128. Apostol, I.; Preda, M.; Nila, C.; Bica, I. Iot Botnet Anomaly Detection Using Unsupervised Deep Learning. *Electronics* **2021**, *10*, 1876. [CrossRef]
129. Ferrag, M.A.; Shu, L.; Djallel, H.; Choo, K.K.R. Deep Learning-Based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0. *Electronics* **2021**, *10*, 1257. [CrossRef]
130. Ahmad, Z.; Khan, A.S.; Nisar, K.; Haider, I.; Hassan, R.; Haque, M.R.; Tarmizi, S.; Rodrigues, J.J.P.C. Anomaly Detection Using Deep Neural Network for Iot Architecture. *Appl. Sci.* **2021**, *11*, 7050. [CrossRef]