Encryption Algorithms in IoT: Security vs Lifetime

Ievgeniia Kuzminykh [1,2,*], Maryna Yevdokymenko [2], Volodymyr Sokolov [3]

[1] Department of Informatics, King's College London, London, WC2R 2ND, UK
[2] Department of Infocommunication Engineering, Kharkov National University of Radio Electronics, Kharkov, 61000, Ukraine, marina.ievdokymenko@nure.ua
[3] Department of Information and Cyber Security, Borys Grinchenko Kyiv University, Kyiv 04212, Ukraine; v.sokolov@kubg.edu.ua
* Correspondence: ievgeniia.kuzminykh@kcl.ac.uk

Abstract: IoT devices are inherently limited by their processing capabilities and power capacity. While aiming to maximise their lifespan, one of the biggest challenges they face is to reduce the computational burden, especially for tasks such as encryption, data transmission, or compression. This paper investigates the lifespan of an IoT device transmitting encrypted data as a function of the encryption algorithm used and the packet length. We focus the analysis particularly on lightweight algorithms popular in IoT ecosystems, such as AES, XTEA, HIGHT, KLEIN, ECC, PRESENT, Serpent, Piccolo, Blowfish, and Twofish. The results of the study indicate that the type of data encryption used for transmission has a significant impact on the IoT device lifetime, together with the data length and the input parameters used. To summarise, the Piccolo algorithm is the most energy-efficient, leading to maximum lifetime and low power consumption, followed by AES, XTEA, and KLEIN. At the other end of the spectrum, ECC, Blowfish, Twofish, PRESENT, and Serpent have high power consumption, hence they should be less preferred for the device-to-device or device-to-gateway IoT communication. Aside from the acknowledged energy efficiency of ciphers based on substitution-permutation operations versus Feistel ones, the results show that algorithms of first group, such as Serpent and PRESENT, require significant encryption and decryption times, while Feistel ciphers such as Piccolo, XTEA and HEIGHT are notably fast.

Keywords: IoT communication model, lightweight cipher, encryption, power consumption, energy efficiency.

1. Introduction

With the growth of IoT technology, more embedded devices, sensors, and other physical objects are becoming connected to the Internet, making available the information that they collect, transmit and store for subscribers of IoT services, analytical companies, or municipalities. Embedded and IoT systems are becoming pervasive, deployed using various communication technologies as Zigbee, LoRaWAN, Sigfox, WiFi, 3G/LTE [1,2] across various domains, including home automation systems [3,4], healthcare [5], automotive [6], industrial installations [7], municipality services [8], critical infrastructure [9], private and public space [10].

With its popularity, IoT also brings issues for IoT service providers and deployers related to compliance with different IoT platforms and integrating devices with various technical characteristics and from different manufacturers into a single application or system. This issue is apparent as soon as a customer starts to address security. The use of the IoT system should be characterized not only by the efficiency of data transmission, device characteristics, communication protocol, but also for data protection since, in most cases, personal data and leakage may compromise the entire IoT system.

In this context, providing confidentiality, integrity, and availability of the data transmitted in IoT remains an ongoing challenge, as fundamental information security principles have always been a major concern when deploying real-world applications. Although many research efforts focused on improving the security of IoT systems, the continuous increase in number of attacks shows that there is still significant work to be

done yet. This is due to a combination of factors, starting with the risks associated with IoT, which tend to be unknown and/or hard to identify, and ending with the effectiveness of security measures, which is typically difficult to assess. In addition to these inherent challenges, a significant number of IoT devices continue to be deployed without using a sufficient level of security exposing their devices to risks [11, 12].

The most convenient alternative to provide confidentiality and integrity is to use encryption of the data transferred between IoT devices. A recent survey determined that most companies are concerned about IoT security but, in addition 24% of companies are even more concerned about the functionality of the product [13]. Over 50% of companies are actively implementing an IoT security strategy on their IoT devices and processes, and 85% of companies use encryption to send the data from the end device to the gateway. The choice of encryption algorithm must be approached wisely, as the use of long-key algorithms, while it does increase the level of security, also is computationally intensive and reduces the lifetime of the device. Traditional encryption algorithms require significant resources and embedded devices are bound by their limited computation capacity, memory, and battery life. Hence, IoT device developers are faced with the problem of balancing security and energy capacity [14]. Due to the limited power and computation capacity of IoT devices, not all the encryption algorithms used in computer networks are suitable for IoT [15,16]. To address this issue, lightweight block ciphers were specially designed to work on devices with limited resources [17–20]; such algorithms are generally characterized by smaller block sizes, smaller key size, lower memory usage (due to minimal encryption/decryption overhead), and shorter execution rounds.

The limited energy capabilities in low-resource devices are the most critical challenge [20]. Energy issues dramatically affect the lifetime of the IoT ecosystem that, in turn, affects the lifespan of the IoT elements and processes as showed in [21], and optimize the management policy in the company. The estimation of the device lifespan can help to predict the Mean Time Between Failures (MBTF) and survivability of the wider IoT ecosystem and inform maintenance timeframes. In this study we propose a model for estimating device lifetime with secure data transmission and analyse ten lightweight encryption algorithms that are suggested to use for IoT. We will show how the type of data encryption and the data length affect the lifespan and power consumption of the device.

1.1. Contributions

Our contributions are summarized as follows:

- We review the security challenges for different communication models in IoT systems and provide a taxonomy of IoT end devices and communication schemes, explaining what device types and protocols are in use for each of the models.
- We offer an up-to-date approach of estimating the maximum lifetime of an IoT device with secure data transmission, achieved by using a series of encryption algorithms.
- We provide a set of recommendations, based on the results of our study, for choosing the optimal cryptographic algorithm and size of payload for deployment of IoT systems, calculate the maximum battery life for nodes in the network and vary the level of security depending on the sensitivity of transmitted data.

1.2 Paper Outline

The paper is organized as follows: Section 2 presents a background of importance the implementing security in the IoT system. Section 3 presents related work information on evaluating of the IoT device lifetime. Section 4 discusses lightweight block ciphers design. Section 5 develops model for evaluating the lifetime of an IoT device. The model results are then compared for different lightweight cryptography algorithms in terms of lifetime and power consumption in Section 6. Concluding remarks are discussed in Section 7.

2. Background

The recent significant success and deployment of IoT are posing an ever-increasing challenge for security. IoT devices are used in many areas of services and industries, across both personal and commercial applications, and the nature of threats varies in each domain and depends on the implementation scenario. The typical security level of IoT technology remains quite low despite its pervasiveness and popularity.

2.1. IoT Security

Back in 2015, Symantec conducted a study on the security of 50 IoT devices and concluded that 19% of the tested devices communicate without encryption, for example, Transport Layer Security (TLS), and none of the devices provided mutual authentication [11]. Another study by Dragoni et al. [22] analyzed the security breaches of 21 smart devices and found that only 9 of them had any security mechanisms in place, while the rest were easily breakable due to their weak security protection mechanisms, including encryption. Similarly, the authors of [23] tested the security level of 28 devices and concluded that 39% of them did not use TLS for communication.

According to a Gartner report from 2018 [24], most companies secure IoT not as part of their business strategy but as line-of-business units. The poor "security by design" and the limited control over the technology within connected devices are direct consequences of this strategy and led to the growing number of cyberattacks on the Internet of Things. Between 2015 and 2018, 20% of the organizations were exposed to the attacks on their IoT systems, as reported by Gartner [24]. According to a study from 2020, only 60% of breaches are detected and can prevent their cybersecurity measures, with the other 40% of the incidents being 'hidden' ones originating from their IoT ecosystem [25]. Therefore, providing security is an important task at all system levels and, with the correct implementation of cybersecurity measures, the probability of hacking remains negligible, especially given the fact that billions of users use IoT systems every day [26].

2.2. IoT Communication Models and Security

Encryption remains the most effective approach to protect data in transit. It may be applied on any segment along the data path, either device-device, device-gateway, and device-cloud or gateway-cloud levels. The choice of encryption algorithm depends on the *architecture* and *communication* models of the IoT system. By definition, in an IoT environment, all devices must be connected to the Internet, and, for this, it is necessary to provide interfaces for connecting "things" to the world. Based on their level and ability to communicate, not all "things" are equal. The IoT device scope is vastly heterogeneous considering the variety of hardware options, operating systems, and communications protocols between end devices and services. Most IoT device life cycles show that devices are manufactured and deployed to a variety of locations worldwide. Some "things" can have Internet access and implement cryptography mechanisms natively while for others that are computation and power-constrained, implementing encryption might be challenging. The types of devices are shown in Fig. 1. They are divided into resourceful, resource-constrained, and resourceless devices.

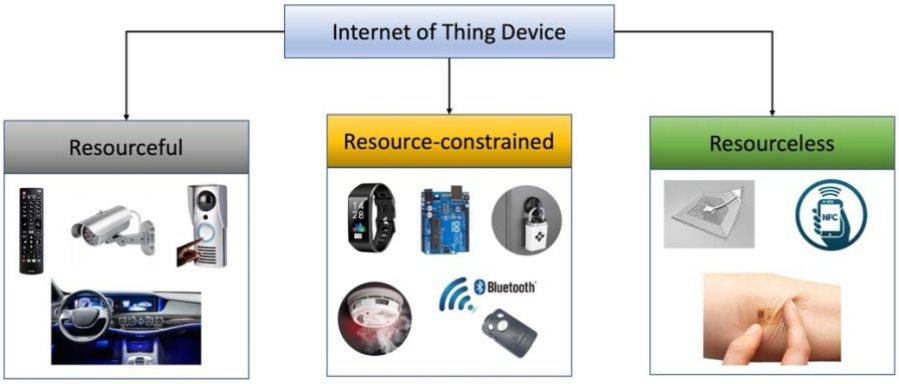

Figure 1. Classification of IoT end devices.

Resourceful end devices have an unlimited power supply, they are autonomous and do not require additional infrastructure elements to run. Such devices can simultaneously act as sensors, gateway, HTTP server, storage, they might perform fog-computing and operate as a platform for Web of Things with all the services it provides. Unlike their resourceful counterparts, resource-constrained "things" have limited resources of power and computation capabilities, but they can operate as sensors, perform basic calculations, and communicate with other entities (gateway, cloud, or web servers). Finally, resourceless end devices are passive objects that might be detected using unique identifiers, such as RFID tags and QR codes. They cannot perform calculations, do not have storage or memory capabilities, they display only an identifier or a set of characters, which require an additional device and software. In some cases, they can be expanded using software deployed in the cloud or on the local gateway.

This paper focuses on the resource-constrained objects, as it is the category which presents the most significant risk – devices that have a level of processing power that may be of interest to an attacker but do not have sufficient resources to mount a complete set of protection mechanisms. We will investigate what encryption algorithms can be applied to such devices and how they may affect the lifetime of the device. To evaluate the efficiency and effectiveness of traditional encryption algorithms used in computer networks versus the light encryption algorithms for IoT, we need to consider and understand the communication model of a typical IoT system. The communication model represents the fundamentals of an IoT that is connecting things regarding information exchange protocols, network protocols, and software. It is useful in the design stage of the IoT architecture solution to understand the interoperability of the elements and required software.

The Internet of Things paradigm relies on two interaction/communication models: *direct* and *transit.* When using a direct connection, an IoT device transmits information either to another IoT end device (for example, a sensor communicating with an actuator) or to a cloud service, that processes the data and generates a response action. For transit interactions, a dedicated device or gateway plays the role of the intermediary by receiving information from other IoT devices and sending the collected data to the application service provider for processing or, in the case of fog computing, communicate with a dedicated device which handles the local requests.

As mentioned above, direct interaction models include device-to-device and device-to-cloud data exchanges [27]. Fig. 2 presents a high-level classification of the communication models for IoT.

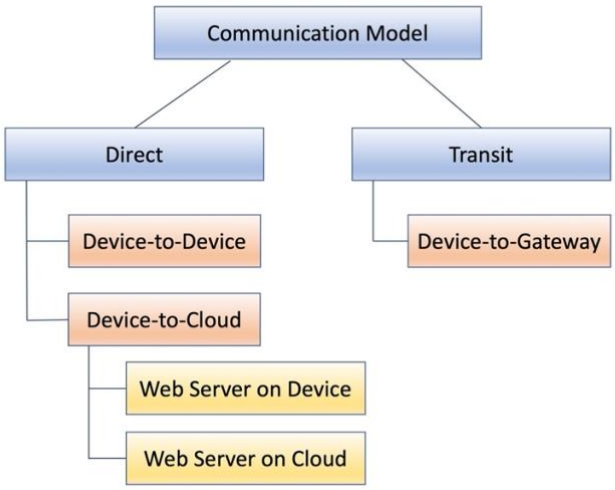

Figure 2. Classification of the communication models in IoT.

2.2.1. Device-to-Device

In this model, two or more devices are directly connected and exchange data with each other, and not through an intermediate device (Fig. 3).

The device-to-device model is particularly popular for home automation systems, HVAC systems, and personal health monitoring, characterized by low-rate, small packet size exchanges, where data does not necessarily have to be shared with multiple people.

The device-to-device objects belong to the second type of the IoT devices, resource-constrained, and include portable and wearable devices, light bulbs, light switches, thermostats and door locks, a heart rate monitor connected to a smart-watch.

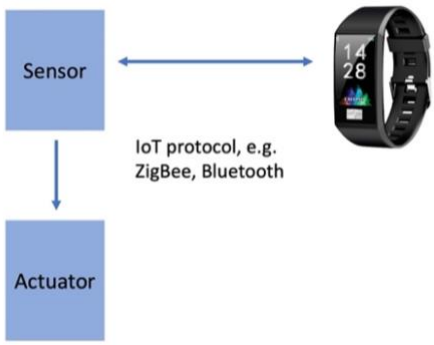

Figure 3. Device-to-device communication.

In most cases, connectivity is provided by short-range low-power communication protocols such as Bluetooth, including the energy-efficient variants (Bluetooth Smart or Bluetooth version 4.0+, Z-Wave, or ZigBee), depending on the device capabilities. Low power communication protocols allow the devices to work for months or years on a single battery. Its lower complexity can also reduce its size and cost.

In the device-to-device model, security is simplified because of the short-range radio technologies employed and the proximity of communicating devices. In the case of targeted attacks, it is possible to intercept traffic remotely using electromagnetic waves, an avenue that was thoroughly explored [28–30]. Missing or weak encryption within IoT device-to-device communication will lead to the illegal interception, modification, or tracking of the data, which can further impact the security of all IoT systems, the network, and the operational level.

2.2.2. Device-to-Cloud

This model involves devices that support HTTP and TCP/IP, provide a REST interface, and can directly connect to the Internet via the web API to exchange data and control message traffic, using, for example, Wi-Fi, cellular or Ethernet, such as shown in Fig. 4. This communication model is suitable for relatively powerful devices which can run a lightweight web server. Typically, embedded web servers have more limited resources than the clients who access them, such as browsers or mobile phones. Due to efficient cross-layer HTTP and TCP stack optimizations and relocation of computational-intensive tasks to the server-side, such web servers with advanced features may have a memory footprint as small as 8KB. There are plenty of examples of such embedded web servers, including GoAhead, Barracuda Embedded Web Server, Lighttpd, or Slinger by Neutrino.

Figure 4. Direct (Device-to-cloud) communication model.

The device-to-cloud objects belong, primarily, to resourceful, sometimes to resource-constrained, and include Raspberry Pi and Photon based devices, smart bulbs or surveillance cameras connected to Wi-Fi, system-embedded consumer IoT devices such as Samsung Smart TV and Nest Learning Thermostat, smart tracking tags, or doorbells that use publish/subscribe communication with a smartphone that listens for events. A cloud connection allows the user to remotely interact with their IoT device using the web application, from accessing the home surveillance camera or remotely update the software of the device to attaching additional services, such as voice assistants or behavior analytics. In these cases, the device-cloud model extends the capabilities of the IoT device, adding convenience to the end-user.

From a security point of view, this model presents more challenges than the device-to-device alternative because objects on a home network need to deal with tunneling, NAT, or TCP for passing a firewall. It also exposes the device to security threats directly from the Internet. Moreover, this model requires two types of credentials, one for device access level (for example, the SIM card of a mobile device), and one for cloud access. Security in this type of communication relies on the transport level encryption provided by transport protocols such as DTLS [31] and on the underlying technology, e.g. WiFi or 3G/LTE. Usually, the IoT device and cloud service are owned by the same provider and any attempt to integrate devices made by different manufacturers to the one cloud service would also bring in increasing security concerns.

One of the modifications of the device-to-cloud model is when the IoT device cannot run an HTTP server and requires a powerful and scalable cloud platform to run it and act as a gateway. This model is cloud-powered, whereby the REST API is provided by the cloud server and the device uses some other protocol such as CoAP, MQTT, to communicate with the server. This modified model is suitable for IoT applications that are deployed over a wide geographic area with many devices that need to be centrally coordinated (for example, air pollution sensors). Both models are presented in Fig. 5.

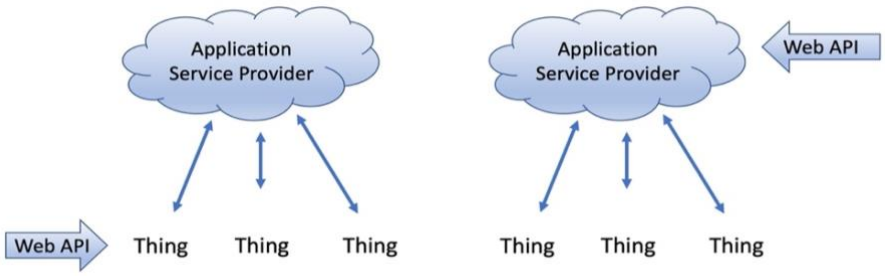

Figure 5. Examples of the webserver deployment on the device-to-cloud model.

2.2.3. Device-to-Gateway

This model relies on proxy communication, as the IoT device uses an intermediate device to access the application service provider, as shown in Fig. 6. In this model, the IoT device is not capable to support HTTP requests directly and needs a more powerful device, such as a smartphone, that connects health wearable devices or activity trackers [32]. To address this, an application gateway acts as a proxy for the device by offering a REST API and the external application can communicate with the proxy using a simple HTTP client. The gateway can be used to connect all kinds of existing IoT devices to the network. The role of the local gateway is typically provided by a smartphone that runs an application to communicate with the device and transfer data to the cloud service.

This model is typically used by a range of popular consumer products, such as personal fitness trackers, powered by batteries, that cannot use Wi-Fi or Ethernet because of their high-power requirements but can use low-power protocols like ZigBee or Bluetooth. In this context, the "thing" can be accessed via a non-http-based protocol.

The device-to-gateway objects belong to the second type of the IoT devices, resource-constrained, and include battery-powered door sensors, sensors, and actuators in home automation systems that are bridged to the gateway device with ZWave transceivers, Zigbee or other transmission technology that, in turn, connect to a cloud service, such as the SmartThings ecosystem promoted by Samsung.

Figure 6. Transit (Device-to-gateway) communication model.

In the device-to-cloud model, the local gateway provides security services, such as securing the transmission and delivery of data. A gateway can add a level of security or authentication, temporarily collect and store data, provide semantic descriptions for "things", and so on. Thereby, the resource-constrained devices that represent the focus of this study are deployed in device-to-device, device-to-gateway, and cloud-powered device-to-cloud model. For all these scenarios, the challenge of implementing security is

of significant interest given that resource-constrained devices, as mentioned before, do not have the computational or power resource required by traditional security methods [4,33,34]. This challenge has triggered a series of studies in the field of light cryptography, which provides a compromise security solution using low cost, low latency implementations that consume less memory [18].

Various parameters are used to evaluate the effectiveness of Light Weighted Algorithms (LWA) or ciphers (LWC) and allow a comparison; such parameters may include security level, power, and energy consumption, execution time, chip area, a figure of merit (FOM), or delay. While all these are of relevance for the research community, a more robust and encompassing measure of the device abilities is the lifetime of a device; as a result, the scope of our paper is to investigate how the choice of security algorithm affects the lifetime of the device and system.

The next section provides an overview of existing approaches for evaluating the effectiveness of cryptography in IoT and the evaluation of the IoT device lifetime.

3. Related Works

Given the prior research relating to IoT device lifetime, it should be noted that no studies considered both the energy consumption for data transmission and for ensuring data protection. Many papers present separate estimates for specific encryption/decryption execution times of different ciphers, energy, and power consumption during encryption/decryption, or the energy consumption during transmission using various communication technologies. While these are indeed critical, the actual battery/device lifetime is a far more informative parameter, as it clearly indicates for how long the system will potentially work autonomously without requiring elements replaced.

All studies can be divided into two major groups: (a) studies about device lifetime under various network topologies, physical conditions, or/and data processing demands; and (b) studies investigating the security of data transmission.

A typical example is a work presented in [16], which compared different LWA (Twofish, Blowfish, DES, 3DES, AES, RC2, RC4, and ChaCha20) in terms of block size, key size, execution time, CPU and memory consumption. These ciphers were tested on the IoT devices by running them on different file sizes ranging from 1 MB to 128 MB. Results showed that the Twofish algorithm had the highest execution speed amongst all the block ciphers and the ChaCha 20 stream cipher showed the best performance and efficiency by all parameters amongst both block and stream ciphers.

In [35], an evaluation of different block ciphers was carried out and their performance was ranked based on memory, CPU usage, and computational cost. The algorithms were XXTEA, RC5, AES, CGEA. Authors concluded that RC5 is the most memory-efficient cipher but recommended to use AES-256 or CGEA because of their long key size.

Batra et al. [36] dealt with various existing lightweight solutions in IoT. The study provides an analysis of the algorithms based on the key size, block size, number of rounds, and attacks possible using the literature review method. Authors considered ciphers such as PRESENT, HIGHT, RSA, ECC, AES and concluded that, for an IoT scenario, symmetric algorithms are more suitable due to their smaller keys and reduced complexity, which subsequently leads to faster encryption requiring less processing power. They all, however, have the same inherent limitation - secure exchange of the secret key, and hence rely on a more secure secret key distribution method, followed by symmetric cryptography to achieve confidentiality and integrity.

The study [37] presents a review of the modern lightweight cryptography. They made a structural analysis of several cryptography algorithms, including PRINT, SIMON, KATAN, HISEC, OLBCA, PRINCE, PRESENT, KLEIN, TWINE. Some of these algorithms use a Feistel network, the others—the SPN, but each of them has its own properties. The authors made a comparative analysis of algorithms by architecture, key size, block size, number of rounds. The analysis showed that, if the algorithm has enough S-blocks and a

well-developed linear operation, then it will provide high security and the cost depends on the design.

The authors in [19] evaluated the performance of different ciphers in terms of energy consumption. For modelling, they chose HEIGHT and KATAN algorithms because of their in-depth analysis of design options and results in the literature. The result showed that the optimum energy is achieved when the block size is between 48-bit and 96-bit and the number of rounds is 16 or less.

Though many surveys reportedly included various encryption algorithms for IoT, most of them focused on the analysis of one algorithm or comparison of only a few of them. Several works give performance evaluation for a bigger list of LWC, but an in-depth analysis has not been carried out. The most recent study that provides a comprehensive cross-comparison of LWC ciphers, including 54 LWC implementations, was performed in [18].

The second group of studies focused on estimating battery/device lifetime and investigated how it can be improved. Several studies [38–41] present an analysis of the lifetime only during data transmission, depending on the wireless technologies used. Other studies [42–45] consider how structural characteristics of the IoT system (heterogeneity of IoT devices, network topology, etc.) impact on the lifetime. The works [46–48] investigated the impact of device characteristics, such as duration of sensing cycle, data gathering, activity modes, on the lifetime of the device.

The authors in [49] proposed a smart rescheduling of the duty-cycles for increasing the lifetime of the device and evaluated their solution using a simulation tool. The security algorithm and transmission technology were not accounted for during the simulation.

The power consumption and device lifetime were also investigated in [42] specifically for various battery powered sensors for home automation systems. The characteristics were measured long term for different modes of sensors (sleep, active) and from different locations within a home (toilet, bedroom, kitchen), using 6LoWPAN communication technology. The study aimed to provide a baseline be used for prediction. The authors did not mention whether the transmission was encrypted or not.

The authors in [48] covered the issue of device lifetime only from the perspective of wireless transmission technology. They investigated energy consumption and device lifetime under a various duration of cycle and size of the transmitted packet for 6LoWPAN, 802.15.4, 802.11ah, Bluetooth low energy, LoRa, and SIGFOX. Results showed that, for small data sizes, BLE achieves the longest lifetime, but 802.15.4 performance is not far behind, for ultralow traffic intensity, LoRa and SIGFOX achieved the best lifetimes, while for high data traffic intensity it was better to use 802.11. The authors did not consider the security overhead for each technology, except SIGFOX, they represented security for this by adding 2 bytes to the packet size for representing the using HMACs for message authentication in SIGFOX.

In [50], the authors outlined a list of parameters that affect the energy consumption and lifetime of IoT devices. Among them, there was a radio duty cycle on the MAC layer, header size, application protocol, communication protocol. During their experiment, the authors determined the minimum and maximum values of a lifetime (in hours) for each parameter. The authors made no references to any security mechanisms in relation to energy consumption.

In [51], the authors solve a topology optimization problem to prolong the lifetime of the IoT network. Their mathematical approach uses predefined energy values of each node and then balance these values. Any particular network or device parameters, e.g. CPU and memory usage, the size and frequency of data transmitted, as well as the ability to use encryption or other security methods, were not included in the research.

In [52], the authors investigated power consumption and device lifetime under various duration of the sensing cycle. The proposed method calculates the best sensing period using the learning data that consist of the remaining time, the remaining power, and the amount of power consumed of each node.

Based on the analysis of existing solutions for estimating the lifetime of the IoT device, we can distinguish three main areas that can affect it:

- Technical characteristics, such as chip area, critical transmission range, signal level, wireless adapter, hardware manufacture, CPU usage, and memory usage.
- Functional characteristics, including wireless communication protocols, data transmission protocols, the nature and frequency of data transmitted, the ability to use encryption, and other security methods.
- Structural characteristics, in particular network topology and heterogeneity of IoT devices.

This brief review shows that, in the context of the limited resources of IoT devices, it is challenging at best to use traditional communication technologies both for data transmission and for data protection in IoT systems. Therefore, this stage of implementing IoT systems is characterized by a variety of solutions in this area that also proves the importance and relevance of research on the IoT devices lifetime. Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

4. Overview of Cryptographic Algorithms for IoT

Most of the algorithms currently used in IoT belong to the group of LWC, symmetric block ciphers. Such ciphers are based on two types of structure: Substitution-Permutation network (SPN) and Feistel [53]. The Feistel structure splits a data block into two equal pieces and uses rounds for encryption, with each round having two separate processes, one to encrypt the plain text and one substitution technique. The decryption process is similar to encryption, the difference is that the keys are used in reversed order. As it can easily be inferred, security is directly proportional to the number of rounds, but that increase comes at the expense of higher latency associated with the implementation. The slow encryption and decryption are the disadvantage of this structure, making it unsuitable for small latency networks. The main advantage of the Feistel structure is low memory usage due to the use of the same program code for the encryption and decryption processes. This can be implemented in battery powered IoT devices with low average power. There are several Feistel ciphers, available, including DES, TEA, Camellia, SEA, CLEFIA, TWINE, LBlock, Piccolo, Blowfish, HIGHT.

The SPN structure uses a single round function that is applied to the whole data block. It is based on Shannon's principle of confusion implemented through the substitution and principle of diffusion with the linear transformation. The security depends on the complexity of the linear function. The decryption process is done by simply reversing the encryption and provides faster computation in comparison with the Feistel ciphers. Its main advantage is the low resource implementation as SPN needs less energy than other structures for offering the same level of security, because it requires a lesser round of execution. The disadvantage is a high level of attacks on SP-based algorithms related to differential and linear cryptanalysis, due to the absence of a key schedule. SPN ciphers include AES, PRESENT, Klein, Serpent.

Asymmetric ciphers are less used in IoT but very popular for resourceful devices, as they are more computationally demanding than symmetric algorithms. Such ciphers can also be used for key exchange and to authenticate an IoT device before data encryption and transmission. Typical such algorithms are RSA and ECC. By design, ECC requires a lower key size than RSA to provide an equivalent security level. The trend is to move from RSA to ECC, a popular choice for the IoT security developers and recommended by Symantec [11], as security requirements mandate key size increase. Moreover, they are included in most cryptographic key management systems used by small and medium businesses that makes them attractive for companies [54]. The disadvantage of symmetric ciphers is a large key size, higher memory consumption, slow speed of execution. This is

a challenge for developers and researchers to improve ECC by reducing memory requirements and computation complexity.

To estimate the lifetime of an IoT device with applied cryptography mechanism, we chose, following an extensive review of current IoT security research, several symmetric and ECC encryption algorithms: AES, TEA, XTEA, HIGHT, KLIEN, Blowfish, Twofish, Serpent, and ECC.

Tiny Encryption Algorithm (TEA), proposed by [55] in 1994, is a LWC with a Feistel structure implemented in a very short program code with simple operations of XOR, ADD, and shifting on a 64bit block size and using 128bit keys. The algorithm is considered secure but, due to its low complexity, is vulnerable to fkey-related attacks. To enhance its security, the TEA algorithm was modified into extended TEA (XTEA) [56] and Block TEA (XXTEA) [57]. XTEA uses a block size of 64 bits, has 32 full cycles, in each complete cycle of two rounds of the Feistel structure. XXTEA ha 64-bit block size and 128-bit key size with a variable number of rounds of Feistel network (6 to 32 full cycles). The cryptoanalysis of XXTEA showed that it is resistant to a plaintext attack based on differential cryptanalysis using 259 queries [58,59].

Advanced Encryption Standard (AES) [60] is an SPN based block cipher standardized by NIST, also known as Rijndael. AES has a fixed block size of 128 bits and operates on a 4×4 bytes matrix. The algorithm uses 4 transformations to convert the plain text into cipher text: arranging data into an array or matrix by SubBytes operation, shifting rows by ShiftRows, then mixing and combining the four bytes in each column by MixedColumns and finally simple XOR in AddRoundKey operation. The available key sizes are 128, 192, 256 bits, with 10, 12 and 14 rounds of repetitions respectively. The higher the key size, the stronger the encryption but AES is still vulnerable to man-in-middle attack [61]. Typically, sensors have the implementation of AES-128 but for resource-constrained devices, AES could be too expensive.

PRESENT [62] is based on SPN based ultralightweight block cipher standardized by ISO/IEC. PRESENT has a block size of 64 bits and a key size of 80 or 128 bits, and a number of rounds are 31. The code size is very small (1000 bytes) and only a single S-box is used instead of eight S-boxes. Low memory consumption and low complexity allow implementing this cipher in RFID tags, which is not possible using the standard AES encryption. This cipher is hardware-based and served as the base for many other ultralightweight ciphers. While appealing given its compact code footprint, PRESENT is vulnerable to differential side-channel attacks [63] and key-related differential attacks on reduced key rounds of 17-26 out of the 31 rounds [64-66], as well as all short key size ciphers.

KLEIN [67] is a typical SPN based ultralightweight cipher with a block size of 64 bits and 64, 80, and 96 bits keys size with 12, 16, and 20 rounds of repetition respectively. The cipher can be implemented as software on sensor platforms that gives more flexibility and lower costs of deployment, and at the same time, its hardware implementation can also be compact. The operation used for achieving compactness is the multiplication operation of bytes, e.g. MixColumns, avoiding bit-shifting operations. KLEIN is resistant to various cryptanalysis techniques as claimed by itsdevelopers but as all short key size ciphers vulnerable to key-related attacks for up to 8 rounds out of 12 in Klein-64, exploits weaknesses of the diffusion layer and key schedule [68, 69].

Elliptic curve cryptography (ECC) [70] is an asymmetric cipher based on the algebraic structure of elliptic curves over finite fields. It uses scalar multiplying which involves point adding and doubling operation. The size of the key of elliptic curves is the size of the field over which the elliptic curve is defined. ECC is another option for lightweight cryptography because it requires less key size and less storage as compared to RSA, therefore, it can work faster and be implemented in the resource-constrained devices [71]. To optimize the use of low power devices ECC uses bit-shifting operation instead of complex multiplication operation [72]. The security level provided by the 1024-bit key in RSA cipher can be obtained with a 160-bit key in ECC.

Blowfish [73] is a Feistel based LWC that uses a block size of 64 bits and a varying key size between 32 and 448 bits. It has also 16 rounds of Feistel structure, makes use of large key-dependent S-boxes. A small key is perfect for encryption in resource-constrained devices. It is an open-source algorithm developed by Schneier with no effective attack against it. Several researchers tested the security of Blowfish and were successful in breaking key but no more than on the fourth round of algorithm, and key can be detected but no more than on 14th round [74].

Twofish [75] is a block cipher with a Feistel structure similar to AES, DES and Blowfish algorithms. Twofish has a block size of 128 bits and uses key sizes of 128, 192 and 256 bits with 16 rounds of repetition. The plain text is broken in-to two 32-bit words and then encrypted into the F-boxes. Each of them contains a layer of 4 S-boxes that depends on different keys and a 4-byte linear transform based on a Maximum Distance Separable (MDS) matrix. The Pseudo-Hadamard Transform (PHT) is used to combine the two 32-bit words. Then two additional 128-subkeys are XORed with the data. Twofish cipher has not been broken yet, any linear attack requires at least 2120.8 chosen plaintexts, a successful differential attack can break up to 5 rounds.

HIGHT [76] is a Feistel based cipher uses with 64 bits block size and 128 bits key size through 32 rounds. HIGHT was designed specially to operate in a low resource environment, a parallel implementation allows improves speed and implements an algorithm in RFID systems. All operations are simple computations such as mod28 or XOR, without the usage of S-boxes. From a security perspective, HIGHT provides sufficient security but it is vulnerable to differential and saturation attacks [65,77].

Piccolo [78] is a Feistel based ultra-lightweight block ciphers which are suitable for extremely constrained environments such as RFID tags. It uses 64-bit block cipher supporting 80 and 128-bit keys with 25 and 31 rounds of repetition respectively. One round of Piccolo consists of two functions: an AddRoundKey that XORs the output with the round key; and Round Permutation functions that groups the 64 bits of a block into 8 bytes and permutes the bytes. Each of these functions includes three operations: SubNibble, MixColumn, and SubNibble. Piccolo, as well as PRESENT, are hardware-oriented lightweight ciphers. Piccolo offers efficient security level and is resistant to key-related attacks and man-in-the-middle attacks, as testing showed that key size can be retrieved if the number of rounds is reduced down to 14 and 21 rounds without whitening for Piccolo-80 and -128, respectively.

Serpent [79] is an SP-network based block cipher that uses a block size of 128 bits and a key length of 128, 192, and 256 bits with 32 rounds of repetition. Each round consists of such operations as key mixing operation, substitution through S-boxes, and a linear transformation. In the last round, this linear transformation is replaced by an additional key mixing operation. All 32 rounds use 32 different S-boxes each of which maps four input bits to four output bits. Decryption is different from encryption, for inverse linear transformation and key reverse order is used. The Serpent is considered to be the most secure, even more than the Rijndael algorithm but has a slow speed of implementation. Since IoT devices send data infrequently, the algorithm is one of the ideal candidates for encryption in IoT.

5. Estimation of Power Consumption and Lifetime of IoT Device

The lifetime of an IoT device is one of the critical features when deploying IoT technology. It is important to estimate in advance the approximate operating time of each node until the battery exhaustion. This indicator directly depends on the device life cycle which includes several factors such as device operating modes (the periods of device active and passive mode), the topology of the Internet of things network, the communication technology, type and amount of data transferred and security mechanism [21]. The life cycle of the IoT end device is shown in Fig. 7.

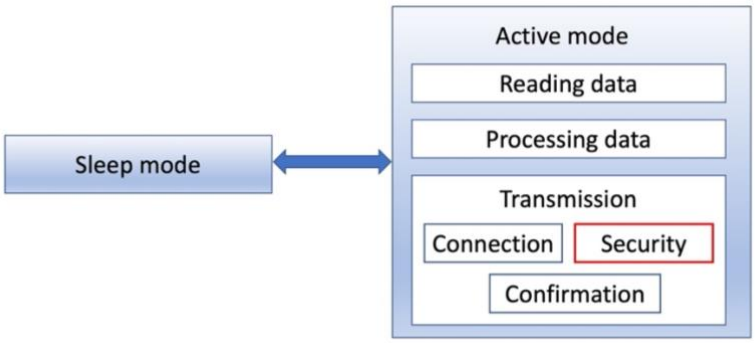

Figure 7. IoT device operation cycle.

During the data transmission process, the system should apply the security measures to achieve the confidentiality and integrity of data. The applied security measures significantly affect the lifetime of the device, subject to the authentication mechanism and the choice of data transmission encryption algorithm. In the latter case, the encryption/decryption of data traffic is the most energy-consuming activity, proportional to the volume of the traffic itself.

To estimate the lifetime of the device, we use the improved model proposed in [4,21] which allows estimating the lifetime of the device as a function of the encryption algorithm and the length of the transmitted packet. To do this, as part of an improved model for IoT device, we introduce the following designation: E_{IoT} is the initial energy of the device's battery (J), P_{IoT} is the total power that the device consumes in one cycle (W), and LT_{IoT} is a lifetime of IoT devices.

Then, following [3], the lifetime of the IoT device can be calculated using the following equation:

$$LT_{IoT} = \frac{E_{IoT}}{P_{IoT}} \quad (1)$$

Further, according to [80], the traditional power consumption equation for an IoT end device is:

$$P_{IoT} = \frac{P_A \cdot t_A + P_S \cdot t_S}{t_c}, \quad (2)$$

where P_A is power consumption in an active mode, mW; t_A is the total time spent in an active mode, sec; P_S is power consumption in a sleep mode, mW; t_S is the total time spent in sleep mode, sec; t_c is the duration of one cycle of the IoT device, sec.

As shown in Fig. 7, the active mode includes the processes of reading and processing data, data transmission and confirmation, as well as encryption and decryption. Accordingly, each one of these processes is characterized by their duration. Then, given the time spent in active mode, the power consumption during the active phase is calculated using the following equation:

$$P_A \cdot t_A = P_D \cdot t_D + P_T \cdot t_T + P_{sec} \cdot t_{sec} \ , \quad (3)$$

where P_D is average power during reading and processing of data, mW; t_D is time spent on reading and processing of data, sec; P_T is average power during data transfer and subsequent confirmation, mW; t_T is time spent on data transmission and receiving of confirmation, sec; P_{sec} is the total power consumption for data encryption and decryption, mW; t_{sec} is the total time spent on data encryption and decryption, sec (which depends on the type of cryptography algorithm).

In equation (2) and (3), the quantities P_S and P_D are constant, determined by the features of the specific hardware implementation of the IoT device. The P_T power

depends on the wireless transmission standard used, the frequency of outgoing packets F, the packet size (in bits) L_{packet} of the transmitted data, as well as the energy spent on transmitting one bit E_T and depending on the characteristics of a particular transceiver, and is calculated using the following equation:

$$P_T = E_T \cdot L_{packet} \cdot F \quad (4)$$

According to [20], the time t_T spent on the data transmission and receiving the confirmation is equal to:

$$t_T = t_{wait} + t_{ch} + t_{tr} + t_{conf}, \quad (5)$$

where t_{ch} is the constant channel listening time that determines its occupancy, equal to 8 symbol periods or 128 μs; t_{conf} is time to transfer confirmation, s; t_{wait} is waiting time for data transfer, s, calculated as:

$$t_{wait} = W \cdot H, \quad (6)$$

and depends on random time interval W, which is an integer selected randomly each time which determines the channel occupancy; for example, in all editions of the IEEE 802.15.4 standard for the 2.4 GHz frequency band, one symbol period is $W = 3$ μs for the best case and the worst-case $W = 7$ μs; H is constantly equal to the period of 20 symbols.

Time spent on data transfer t_{tr} can be calculated by the following equation:

$$t_{tr} = \frac{(L_{packet} + L_{field}) \cdot 8}{Ch}, \quad (7)$$

where L_{field} is the size of the service fields of the packet, bytes; Ch is a channel data transfer rate, kbps.

In turn, the quantity $P_{sec} \cdot t_{sec}$ from equation (3) includes two processes of encryption and decryption, each taking a different time, depending on the type of encryption algorithm. The total power consumption for the data encryption and decryption $P_{sec} \cdot t_{sec}$ is calculated as:

$$P_{sec} \cdot t_{sec} = P_{sec(enc)} \cdot t_{sec(enc)} + P_{sec(dec)} \cdot t_{sec(dec)} \quad (8)$$

where $P_{sec(enc)}$ and $P_{sec(dec)}$ are the power consumption for data encryption during time $t_{sec(enc)}$ and decryption during time $t_{sec(dec)}$, respectively, mW.

Thus, variables $P_{Sec(enc)}$ and $P_{Sec(dec)}$ are defined using the following equations:

$$P_{Sec(enc)} = V_{enc} \cdot I_{enc}, \quad (9)$$

$$P_{Sec(dec)} = V_{dec} \cdot I_{dec} \quad (10)$$

where V_{enc} and I_{enc} are the voltage and current for encryption process respectively; V_{dec} and I_{dec} are the voltage and current for the decryption process respectively.

Then, based on equation (2)–(10), equation (1) can be rewritten as follows:

$$LT_{IoT} = \frac{E_{IoT} \cdot t_c \cdot Ch}{P_D \cdot t_D \cdot Ch + (E_T \cdot L_{packet} \cdot F) \cdot t_T \cdot 8 + P_{sec} \cdot t_{sec} \cdot Ch} \quad (11)$$

6. Calculation of the IoT Device Lifetime using the Proposed Model

We used Arduino Mega 2560 as an exemplary IoT device to calculate the lifetime of IoT device. This platform was chosen due to its performance and compatibility with a range of expansion cards. As this research is focused on the power consumption of the external battery connected to this device, the capacity of the batteries is a defining factor in the calculation.

To evaluate the performance of different ciphers, it is necessary to perform two steps:

1. An experimental study of the selected encryption algorithms deployed on two Arduino Mega 2560 boards acting as end devices, transmitted and receiver, communicating via the wireless RF 433 MHz module running the selected encryption

algorithms implemented using github libraries. A total of ten encryption algorithms were investigated (AES, XTEA, HIGHT, KLEIN, ECC, BLOWFISH, Twofish, Serpent, Piccolo, and PRESENT) with various packets sizes within one cycle. For consistency and similarity of the security level of the algorithms, most ciphers were set with a key size of 128 bits, except for ECC (160 bits), and KLEIN (96 bits). To measure the encryption/decryption time and power usage we used the *micros ()* from the Arduino library which returns the number of microseconds since the Arduino board began running the current program. To measure power consumption, we used values for voltage V and current I obtained from the METRAHit 16S – analogue-digital multimeter.

2. Calculation of the lifetime according to equation (11), using the values obtained from Step 1 as input and the parameters listed in Table 1. We use the longest interval possible of 8 seconds to set the device into power down sleep mode. The CPU frequency is the Arduino clock speed that defines how many operations the board can execute per second. The default clock speed for most Arduino microcontrollers is 16 MHz that equals 16 million instructions per second.

Table 1. Input data for transmission.

Parameter, measurement unit (designation)	**Value**
The initial energy of the node (E_{IoT}), kJ	20
Packet size (L_{packet}), bytes	50, 100, 200
CPU frequency (F), MHz	16
Power consumption in active mode (P_{A}), mW	0.02
Power consumption in sleep mode (P_{S}), mW	0.003
Channel bit rate (Ch), kbps	250
The size of the service fields of the packet (L_{field}), bytes	17
Time sleep (t_s), s	8

Step 1 represented the experimental study of the given encryption algorithms to reflect the dependence between the encryption/decryption time and the power consumption as a function of the packet length. The results of these experiments are summarised in Table 2 and Table 3.

Table 2. Encryption and decryption time for various length of packets.

Algorithm	**Length of packets, bytes**					
	L = 50		L = 100		L = 200	
	Encryption time, ms	Decryption time, ms	Encryption time, ms	Decryption time, ms	Encryption time, ms	Decryption time, ms
AES	5.76	6.76	8.38	8.44	14.23	14.87
XTEA	6.24	6.24	9.36	9.67	16.34	16.53
HIGHT	8.32	8.34	12.20	12.30	18.44	18.45
KLEIN	5.84	5.88	9.06	9.11	15.40	15.47
ECC	10.84	10.80	13.12	13.17	18.78	18.99
Blowfish	27.70	41.20	43.00	52.80	62.00	68.70
Twofish	25.40	37.40	41.30	51.56	58.80	66.00
Serpent	21.40	38.30	30.70	48.50	65.40	70.64
Piccolo	5.62	3.35	7.74	6.50	13.56	12.82
PRESENT	14.58	30.12	22.46	43.65	34.76	92.70

The results showed that some algorithms, such as PRESENT and Serpent, required more time for decryption due to the usage of different transformation functions for encription and decrytion. The algorithms with Feistel stucture, i.e. XTEA, HIGHT, Piccolo, in vast majority showed similar times. The fastest algorithm was Piccolo, which require significant less time than the others. AES, XTEA.HIGHT, Klein can be clustered in a

second group, given their medium encryption/decryption time. Finally, the slowest algorithms were PRESENT, Twofish, Blowfish, and Serpent.

Table 3. Total power consumption for cryptography algorithms, mW.

Algorithm	Length of packets, bytes		
	L = 50	L = 100	L = 200
AES	0.56	0.71	1.20
XTEA	0.61	0.85	1.32
HIGHT	1.21	1.71	2.21
KLEIN	0.65	0.90	1.47
ECC	6.37	8.46	9.30
Blowfish	5.96	7.02	10.64
Twofish	5.57	7.87	10.10
Serpent	5.68	7.99	10.20
Piccolo	0.16	0.66	1.22
PRESENT	4.24	7.60	11.20

Regarding the power consumption (Table 3), the results showed that some of the ciphers with Feistel structure (Piccolo and XTEA) require less or similar energy when compared to the algorithms using an SP structure, which contradicts the theoretical assumption that ciphers with an SP network structure spend less energy than the Feistel ones. Aside from being the fastest algorithm, Piccolo also achieved the lowest power consumption. In addition, although the SPN based ciphers are considered faster in execution than Feistel ciphers, the results show that some of the SPN based algorithms, Serpent and PRESENT in particular, have higher encryption and decryption times, while Piccolo, XTEA, and HEIGHT work relatively fast among Feistel ciphers.

Based on the obtained values of power consumption (Fig. 8) and the encryption/decryption time, the lifetime of the IoT device can be calculated according to equations (2)–(11). The values are presented in Table 4 and Fig. 9.

Table 4. Results of calculation of lifetime IoT devices, days.

Algorithm	Length of packets, bytes					
	L = 50		L = 100		L = 200	
	P_{IoT}	LT_{IoT}	P_{IoT}	LT_{IoT}	P_{IoT}	LT_{IoT}
AES	1.12	318	1.46	266	2.17	176
XTEA	1.21	316	1.72	254	2.22	165
HIGHT	1.86	269	2.04	198	3.21	130
KLEIN	1.30	312	1.60	253	2.37	163
ECC	12.10	113	17.23	77	19.84	51
Blowfish	11.40	149	14.52	93	19.30	62
Twofish	11.03	152	15.37	86	19.10	65
Serpent	11.10	151	15.41	85	19.20	63
Piccolo	0.22	445	1.30	372	2.18	279
PRESENT	10.20	178	15.10	90	20.70	46

The results show that ECC-160 is the most energy-demanding cypher, hence it should be the least preferred for the resource-constrained devices, therefore should be used only if data security is critical for the respective process, given that ECC-160 corresponds to the 1024-bit key in RSA, which means it is highly secure. Given the values from Table 1, the lifetime of a node with ECC encryption will be 51-113 days (2-3.5 months), depending on the packet size.

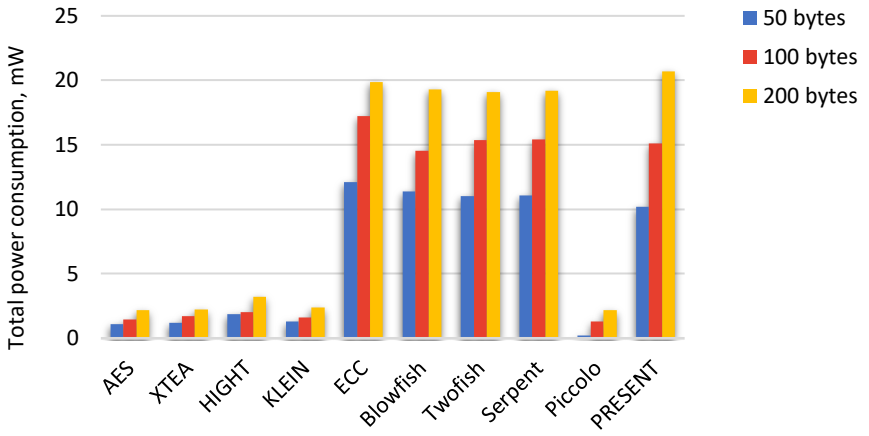

Figure 8. Total power consumption for the IoT device with providing security techniques.

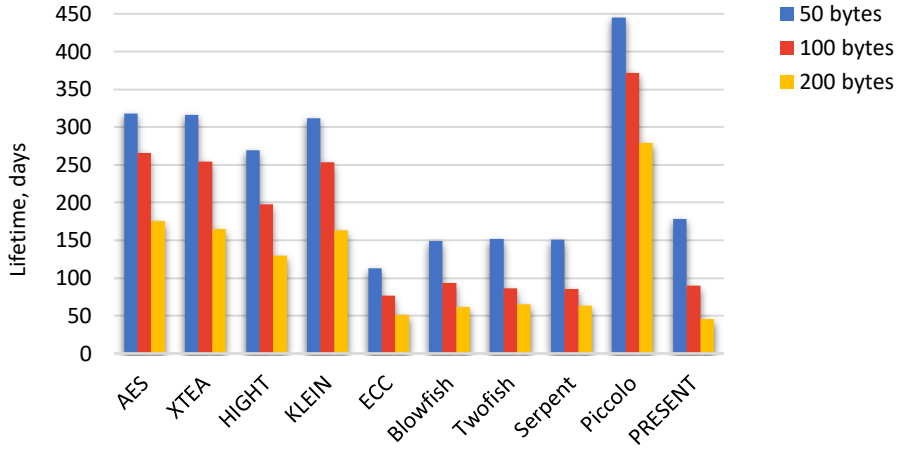

Figure 9. The lifetime for the IoT device with providing security techniques.

Although the HIGHT algorithm was specially designed for RFID tags, with emphasis on low resource communication, the results indicate that devices using it will have a lower lifetime than the AES, XTEA and KLEIN algorithms when applying a 128bit key.

AES, Piccolo-128, XTEA, and KLEIN are the preferred choices when combining power requirements and maximum lifetime. Due to their simple structure, employing an involutive S-block, the AES and KLEIN algorithms are mainly intended to reduce the implementation costs and are considered relatively flexible and providing an average level of security. The lifetime of a node with an encryption algorithms from this group will be 6-10 months, depending on the size of the payload.

Amongst the group, the Piccolo Feistel based cipher scored highest and is the cheapest choice in terms of power consumption. It allows node to operate about 15 month without changing the battery for 50 byte packet length. Piccolo is considered as an ultra-lightweight cipher that can be used even for RFID with key size of 80 and 128 bits; a possible follow-up experiment may investigate the benefits of using an 80bit key, for applications that permit it, in order to extend device lifetme.

7. Conclusions

The lifetime of the IoT device indicates the potential lifecycle of the whole system; at the same time security represents one of the essential building blocks of an IoT system, as it provides confidentiality and integrity of data. This paper proposes a model for estimating the lifetime of an IoT device with an implemented security mechanism given

an encryption algorithm used and the packet length. The study also performed an in-depth analysis of the lightweight block cipher algorithms in terms of their lifetime and power consumption, based on the cipher specification and security.

The results of lifetime can help developers during deployment phase of the IoT system and provide a better maintenance schedule for the system.

As discussed in this paper, the lightweight cryptography algorithms should be used in certain communication IoT models, as device-to-device, device-to-gateway and cloud-powered device-to-cloud model, but the device-to-cloud model can also work with more powerful and secure algorithms.

To benchmark the various algorithms, we used the Arduino Mega platform, to perform a comparative analysis of the AES, XTEA, HIGHT, KLEIN, Twofish, Blowfish, PRESENT, Serpent and Piccolo algorithms in terms of power consumption and maximum lifetime. This ultimately allowed us to characterize the energy efficiency of these solutions for further use on the end IoT devices.

The paper also considered the properties of Feistel network based and SPN based ciphers. Although ciphers with SPN structure do spend less energy and are faster than Feistel based ones, the obtained experimental results showed that some SPN based algorithms, Serpent and PRESENT in particular, have significant encryption and decryption time, while Piccolo, XTEA, and HEIGHT work quite fast among Feistel ciphers.

The results showed that ECC-160 consumes maximum power hence it should be least preferred for the resource-constrained devices. It is recommended to use this cipher if the security is in very high priority because the ECC-160 corresponds to the 1024-bit key in RSA that means it is high secure.

The Blowfish, Twofish, PRESENT, and Serpent algorithms also have high power consumption and less lifetime but could be considered for implementation in the systems that require high security level.

AES, Piccolo-128, XTEA and KLEIN are preferable choices when look upon the power requirements and maximum lifetime with an average level of security.

Finally, we concluded that the Piccolo algorithm was the most efficient and is therefore the best choice in terms of power consumption.

Author Contributions: Conceptualization, all authors.; methodology of experiment, I.K., M.Y.; theoretical background V.S.; mathematical formalization – M.Y.; experiment – V.S. and I.K.; original draft preparation, I.K. and V.S.; writing—review and editing, I.K. and V.S.; visualization, M.Y.; supervision, I.K. All authors have read and agreed to the published version of the manuscript.

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

References

1. Chaudhari, B.S.; Zennaro, M. *LPWAN technologies for IoT and M2M applications*, 1st ed.; Academic Press, 2020; 456p.
2. Liberg, O.; Sundberg, M.; Wang, E.; Bergman, J.; Sachs, J. *Cellular Internet of Things: technologies, standards, and performance*, 1st ed; Academic Press, 2017.
3. Trasvina-Moreno, C.A.; Blasco ,R.; Casas, R.; and Marco, A. Autonomous WiFi sensor for heating systems in the Internet of Things. *J. Sensors* **2016**, Article ID 7235984, 14 pages.
4. Kuzminykh, I.; Carlsson, A.; Yevdokymenko, M. A performance evaluation of sensor nodes in the home automation system based on Arduino. In Proceedings of IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PICS&T), Kyiv, Ukraine, 8-11 October 2019; pp. 511-516.
5. Jeyaraj, P.R.; Nadar, E.R.S. Smart-monitor: patient monitoring system for IoT-based healthcare system using deep learning. *IETE J. Research* **2019.**
6. Ni, Y.; Cai, L.; He, J.; Vinel, A.; Li, Y. et al (202) Toward reliable and scalable internet of vehicles: performance analysis and resource management. *Proc. IEEE* **2020**, *108(2),* 324-340.
7. Mekki, K.; Bajic, E.; Chaxel, F.; Meyer, F. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express* **2019**, *5(1),* 1-7.
8. Kuzminykh, I. Development of traffic light control algorithm in smart municipal network. In 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, Ukraine, 23-26 Feb. 2016.

9. Hunzinger, R. SCADA fundamentals and applications in the IoT. In *Internet of Things and Data Analytics Handbook*; Geng, H., Ed.; John Wiley & Sons, Inc, 2017, pp. 283-293.
10. Das, A.; Mishra Sharma, C.S.; Ratha, B.K. (2019) The new era of smart cities, from the perspective of the Internet of Things. In *Smart Cities Cybersecurity and Privacy*; Rawat, D.B., Ghafoor, K.Z., Eds.; Elsevier Inc, 2019; pp. 1-9
11. Barcena, M.B.; Wueest, C. (2015) Insecurity in the Internet of Things. Symantec report. Available online: https://docs.broadcom.com/doc/insecurity-in-the-internet-of-things-en (accessed on 11 May 2021).
12. Kuzminykh, I.; Carlsson, A. Analysis of Assets for Threat Risk Model in Avatar-Oriented IoT Architecture. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems,* NEW2AN 2018, ruSMART 2018; Galinina, O.; Andreev, S.; Balandin, S.; Koucheryavy, Y., Eds.; Springer: Cham, LNCS, 2018; Volume 11118, pp. 52-63.
13. Kuzminykh, I.; Ghita, B.; Such, J.M. The Challenges with Internet of Things for business. In Proceedings of the 14th conference on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2021), on-line, August 30-31, 2021.
14. Sheng, Z.; Yang, S.; Yu, Y.; Vasilakos, A.; McCann, J.; Leung. K. A survey on the IETF protocol suite for the Internet of Things: Standards challenges and opportunities. *IEEE W. Comm.* **2013**, *20(6),* 91-98.
15. Vahdati, Z.; Md Yasin, S.; Ghasempour, A.; Salehi, M. Comparison of ECC and RSA algorithms in IoT devices. *J. Theor. and Appl. Inform. Techn.* **2019**, *97(16),* 4293-4308.
16. Singh, P.; Deshpande, K. Performance evaluation of cryptographic ciphers on IoT devices. In Proceedings of International Conference on Recent Trends in Computational Engineering and Technologies (ICTRCET'18), Bangalore, India, 17-18 May 2018.
17. Singh, S.; Sharma, P.K.; Moon, S.Y.; Park, J.H. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *J. Ambient Intell Human Comput.* **2017**.
18. Dhanda, S.S.; Singh, B.; Jindal, P. Lightweight cryptography: a solution to secure IoT. *Wireless Pers Commun* **2020,** *112,* 1947–1980.
19. Mohd, B.J.; Hayajneh, T. Lightweight block ciphers for IoT: energy optimization and survivability techniques. *IEEE Access* **2018,** *6*, 35966-35978.
20. Mohd, B.J.; Hayajneh, T.; Vasilakos, A.V. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. *J Network and Computer App* **2015**, *58*, 73-93.
21. Kuzminykh, I.; Carlsson, A.; Yevdokymenko, M.; Sokolov, V. Investigation of the IoT device lifetime with secure data transmission. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems.* NEW2AN 2019, ruSMART 2019; Galinina, O.; Andreev, S.; Balandin, S.; Koucheryavy, Y.. Eds.; LNCS, Volume 11660, Springer, Cham.
22. Dragoni, N.; Giaretta, A.; Mazzara, M. The Internet of hackable things. In 5th International Conference in Software Engineering for Defence Applications (SEDA 2016); *Advances in Intelligent Systems and Computing*, Volume 717, Springer, Cham.
23. Sivanathan, A.; Gharakheili, H.H.; Loi, F.; Radford, A.; Wijenayake, C., et al. Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Trans on Mobile Comp* **2019**, *18(8),* 1745-1759.
24. Middleton, P.; Contu, R.; Pace, B.; Alaybeyi, S. Forecast: IoT security, worldwide, 2018. Gartner Research, 2018.
25. Bissell, K.; Lasalle, R.M.; Dal Cin, P. Innovate for cyber resilience, 3rd annual state of cyber resilience. Accenture Security Report, 2020.
26. Kuzminykh, I. Avatar Conception for "Thing" Representation in Internet of Things. In Proceedings of 14th Swedish National Computer Networking Workshop, May 31 - June 1, 2018, Karlskrona, Sweden.
27. Tschofenig, H.; Arkko, J.; Thaler, D.; McPherson, D. Architectural considerations in smart object networking. RFC 7452, March 2015.
28. Guri, M.; Bykhovsky, D. aIR-jumper: covert air-gap exfiltration/infiltration via security cameras & infrared (IR). *Comp & Sec* **2019**, *82*, 15-29.
29. Wang, J.; Abari, O.; Keshav, S. Challenge: RFID hacking for fun and profit. In Proceedings of 24th Annual International Conference on Mobile Computing and Networking (MobiCom'18), NY, USA, October 18, 2018, pp. 461-470.
30. Abdul-Ghani, H.A.; Konstantas, D.; Mahyoub, M. A comprehensive IoT attacks survey based on a building-blocked reference mode. *Int J Adv. Comp. Sci and App* **2018**, *9(3*), pp. 355-373.
31. Cho, E.; Park, M.; Lee, H.; Choi, J.; Kwon , T. D2TLS: delegation-based DTLS for cloud-based IoT services. In Proceedings of International Conference on Internet-of-Things Design and Implementation (IoTDI'19), Montreal, Canada, April 2019.
32. TajDini, M.; Sokolov, V.; Kuzminykh, I.; Ghita, B.; Shiaeles, S. Wireless Sensors for Brain Activity - A Survey. *Electronics* **2020**, *9(12),* 2092.
33. Granjal, J.; Monteiro, E.; Silva, J.S. Security in the integration of low-power wire- less sensor networks with the internet: A survey. *Ad Hoc Networks* **2015**, *24*, 264-287
34. Zhao, K.; Ge, L. A survey on the internet of things security. In Proceedings of 9th International Conference on Computational Intelligence and Security (CIS'13), Emeishan, China, 14-15 Dec. 2013, pp. 663-667.
35. Biswas, K.; Muthukkumarasamy, V.; Wu, X.W.; Singh, K. Performance evaluation of block ciphers for wireless sensor networks. In *Advanced Computing and Communication Technologies*; Choudhary, R.; Mandal, J.; Auluck, N.; Nagarajaram, H., Eds.; Advances in Intelligent Systems and Computing, 2016, Volume 452, Springer, Singapore.
36. Batra, I.; Luhach, A.K.; Pathak, N. Research and analysis of lightweight cryptographic solutions for Internet of Things. In Proceedings of the 2nd International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '16), March 2016, pp. 1-4.

37. Nandhini, P.; Vanitha, D. A Study of Lightweight Cryptographic Algorithms for IoT. *Int J Innov & Advanc in Computer Sci* **2017**, *6(1),* 26-35.
38. Wadud, Z.; Javaid, N.; Khan, M.A.; Alrajeh, N.; Alabed, M.S.; Guizani, N. Lifetime Maximization via Hole Alleviation in IoT Enabling Heterogeneous Wireless Sensor Networks. *Sensors (Basel)* **2017,** *17(7),* 1677.
39. Mallick, S.; Bin Habib, A. S.; Ahmed, A. S.; Alam, S. S. Performance appraisal of wireless energy harvesting in IoT. In Proceedings of 3rd International Conference on Electrical Information and Communication Technology (EICT'17), Khulna, Bangladesh, 7-9 Dec. 2017.
40. Casals, L.; Mir, B.; Vidal, R.; Gomez, C. Modeling the energy performance of LoRaWAN. *Sensors* **2017**, *17(10),* 2364.
41. Carlsson, A.; Kuzminykh, I.; Franksson, R.; Liljegren, A. (2018) Measuring a LoRa network: performance, possibilities and limitations. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems,* NEW2AN 2018, ruSMART 2018; Galinina, O.; Andreev, S.; Balandin, S.; Koucheryavy, Y., Eds.; Springer: Cham, LNCS, 2018; Volume 11118.
42. Fafoutis, X.; Elsts, A.; Vafeas, A.; Oikonomou, G.; Piechocki, R. On predicting the battery lifetime of IoT devices: experiences from the SPHERE deployments. In Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks, Shenzhen, China, Nov 2018, pp. 7-12.
43. Galkin, P. Model of reducing the power consumption for node of wireless sensor network in embedded control systems. In Proceedings of Int Sci-Practical Conference Problems of Infocommunications. Science and Technology (PICS&T). Kharkiv, Ukraine, 9-12 Oct. 2018, pp. 252-256.
44. Kim, T.; Kim, S.H.; Kim, D. Distributed topology construction in ZigBee wireless networks. *Wireless Pers Comm* **2018**, *103(3),* 2213-2227.
45. Popov, O.; Kuzminykh, I. Analysis of methods for reducing topology in wireless sensor networks. In Proceedings of 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), Lviv-Slavske, Ukraine, 20-24 Feb. 2018, pp. 529-532.
46. Sokolov, V.; Carlsson, A.; Kuzminykh, I. Scheme for dynamic channel allocation with interference reduction in wireless sensor network. In Proceedings of 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PICS&T). Kharkov, Ukraine, 10-13 Oct. 2017, pp. 564-568.
47. Kuzminykh, I.; Snihurov, A.; Carlsson, A. Testing of communication range in ZigBee technology. In Proceedings of 14th International Conference the Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, Ukraine, 21-25 Feb. 2017, pp. 133-136.
48. Morin, E.; Maman, M.; Guizzetti, R.; Duda, A. Comparison of the device lifetime in wireless networks for the Internet of Things. *IEEE Access* **2017,** *5*, 7097-7114.
49. Li, Q.Q.; Gochhayata, S.P.; Conti, M.; Liu, F.A. EnergIoT: A solution to improve network lifetime of IoT devices. *Pervasive and Mobile Computing* **2017**, *42*, 124-133.
50. Lekidis, A.; Panagiotis, K. Model-based design of energy-efficient applications for IoT systems. In Proceedings of the 1st International Workshop on Methods and Tools for Rigorous System Design (MeTRiD 2018), Thessaloniki, Greece, 15th April 2018.
51. Valls, V.; Iosifidis, G.; Salonidis, T. Maximum lifetime analytics in IoT networks. In Proceedings of IEEE Conference on Computer Communications (INFOCOM 2019), Paris, France, 29 April- 2 May 2019, pp. 1369-1377.
52. Kim, W.; Jung, I . Smart sensing period for efficient energy consumption in IoT network. *Sensors* **2019**, *19(22),* 4915.
53. Katz, J.; Lindell, Y. *Introduction to modern cryptography.* 1st ed.; CRC Press, 2007; 552p.
54. Kuzminykh, I.; Ghita, B.; Shiaeles, S. Comparative Analysis of Cryptographic Key Management Systems. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems.* NEW2AN 2020, ruSMART 2020; Galinina, O.; Andreev, S.; Balandin, S.; Koucheryavy, Y., Eds.; LNCS, Volume 12526, Springer, Cham, 2020.
55. Wheeler, D.J.; Needham, R.M. TEA, a tiny encryption algorithm. In *Fast Software Encryption*. FSE 1994; Preneel, B., Ed.; LNCS, Volume 1008, Springer, Berlin, Heidelberg, 1995.
56. Needham, R.M.; Wheeler, D.J. TEA extensions. Technical report. University of Cambridge, Cambridge, UK, 1997.
57. Wheeler ,D.J.; Needham, R.M. XXTEA: Correction of XTEA. Technical report. Computer Laboratory, Cambridge University, UK, 1998.
58. Kurbanmuradov, D.; Sokolov, V.; Astapenya, V. Implementation of XTEA encryption protocol based on IEEE 802.15.4 wireless systems. *Cybersecur Educ Sci Tech* **2019**, *2(6),* 32-45.
59. Yarrkov, E. Cryptanalysis of XXTEA. Available online: https://eprint.iacr.org/2010/254.pdf (accessed on 12 May 2021).
60. Advanced Encryption Standard (AES). *National Institute of Standards and Technology*. Federal information processing standards publication 197, 26 Nov 2001.
61. Derbez, P.; Fouque, P.A. Exhausting Demirci-Selcuk meet-in-the-middle attacks against reduced-round AES. In *Fast Software Encryption* (FSE 2013); Moriai, S. Ed.; LNCS, Volume 8424 Springer, Berlin, Heidelberg, 2014, pp. 541-560.
62. Bogdanov, A.; Knudsen, L.R.; Leander, G.; Paar, C.; Poschmann, A. PRESENT: An Ultra-Lightweight Block Cipher. In *Cryptographic Hardware and Embedded Systems*. CHES 2007; Paillier, P.; Verbauwhede, I., Eds.; LNCS, Volume 4727, Springer, Berlin, Heidelberg, 2007, pp. 450-466.
63. Yang, L.; Wang, M.; Qiao, S. Side channel cube attack on PRESENT. In *Cryptology and Network Security.* CANS 2009; Garay, J.A.; Miyaji, A.; Otsuka, A., Eds.; LNCS, Volume 5888, Springer, Berlin, Heidelberg, 2009, pp. 379-391.

64. Blondeau, C.; Nyberg, K. Links between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities. In *Advances in Cryptology*, EUROCRYPT 2014; Nguyen, P.Q.; Oswald, E., Eds.; LNCS, Volume 8441, Springer, Berlin, Heidelberg, 2014, pp. 165-182.
65. Özen, O.; Varıcı, K.; Tezcan, C.; Kocair, Ç. Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT. In *Information Security and Privacy.* ACISP 2009; Boyd, C.; González Nieto, J., Eds.; LNCS, Volume 5594, Springer, Berlin, Heidelberg, 2009.
66. Jeong, K.; Lee, Y.; Sung, J.; Hong, S. Improved differential fault analysis on PRESENT-80/128. *Int J of Computer Maths* **2013**, *90(12),* 2553-2563,
67. Gong, Z.; Nikova, S.; Law, Y.W. KLEIN: A new family of lightweight block ciphers. In *RFID. Security and Privacy*. RFIDSec 2011; Juels, A.; Paar, C., Eds.; LNCS, Volume 7055, Springer, Berlin, Heidelberg, 2012.
68. Ahmadian, Z.; Salmasizadeh, M.; Aref, M.R. (2015) Biclique cryptanalysis of the full-round KLEIN block cipher. *IET Information Security* **2015**, *9(5),* 294-301.
69. Aumasson, J.P.; Naya-Plasencia, M.; Saarinen, M.J.O. Practical attack on 8 rounds of the lightweight block cipher KLEIN. In *Progress in Cryptology,* INDOCRYPT 2011; Bernstein, D.J.; Chatterjee, S., Eds.; LNCS, Volume 7107, Springer, Berlin, Heidelberg, 2011, pp. 134-145.
70. Blake-Wilson, S.; Bolyard, N.; Gupta,V.; Hawk, V.; Moeller, B. Elliptic curve cryptography (ECC) Cipher Suites for Transport Layer Security (TLS). RFC 4492, IETF, 2006.
71. Vahdati, Z.; Ghasempour, A.; Salehi, M.; Yasin, S.Md. Comparison of ECC and RSA algorithms in IoT devices. *J Theor. and Appl.Inf. Tech.* **2019**, *97(16),* 4293-4308.
72. Ayuso, J.; Marin, L.; Jara, A.; Skarmeta, A. Optimization of public key cryptography (RSA and ECC) for 16-bits devices based on 6LoWPAN. In Proceedings of 1st Int. Work. Secur. Internet Things, Tokyo, Japan, 2010.
73. Schneier, B. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *Fast software encryption*, FSE'93; Anderson, R.J., Ed.; LNCS, Volume 809, Springer-Verlag, Berlin, 1994, p. 191.
74. Vyakaranal, S.; Kengond, S. Performance analysis of symmetric key cryptographic algorithms. In Proceedings of Int Conf on Comm and Signal Processing (ICCSP), Chennai, India, 3-5 April 2018, pp. 411-415.
75. Schneier, B.; Kelsey, J.; Whiting, D.; Wagner, D.; Hall, C.; Ferguson, N. Twofish: a 128-bit block cipher. Available online: https://www.schneier.com/academic/paperfiles/paper-twofish-paper.pdf (accessed on 12 May 2021).
76. Hong, D.; Sung, J.; Hong, S.; Lim, J.; Lee, S.; et al. HIGHT: a new block cipher suitable for low-resource device. In *Cryptographic Hardware and Embedded Systems*. CHES 2006; Goubin, L.; Matsui, M., Eds.; LNCS, Volume 4249, Springer, Berlin, Heidelberg, 2006, pp. 46-59.
77. Wen, L.; Wang, M.; Bogdanov, A.; Chen, H. Multidimensional zero-correlation attacks on lightweight block cipher HIGHT: Improved cryptanalysis of an ISO standard. *Information Processing Letters* **2014,** *114*, 322-330.
78. Shibutani, K.; Isobe, T.; Hiwatari, H.; Mitsuda, A.; Akishita, T.; Shirai, T. Piccolo: an ultra-lightweight blockcipher. In *Cryptographic Hardware and Embedded Systems.* CHES 2011; Preneel, B.; Takagi, T., Eds.; LNCS, Volume 6917, Springer, Berlin, Heidelberg, 2011, pp. 342-357.
79. Biham, E.; Anderson, R.; Knudsen, L. Serpent: a new block cipher proposal. In *Fast Software Encryption.* FSE 1998; Vaudenay, S., Eds.; LNCS, Volume 1372, Springer, Berlin, Heidelberg, 1998, pp. 222-238.
80. Overview of the Internet of Things. ITU-T Recommendation Y400/Y.2060. Approved 15 June 2012. Available online: https://www.itu.int/rec/T-REC-Y.2060-201206-I (accessed on 02 Nov 2021).