Lightweight Cryptography for the Internet of Things

Alaa Hassan(✉)

University of Calgary, Calgary, Canada
eng.alaa.hassan@ieee.org

Abstract. The rate of implementation of the Internet of Things (IoT) devices is increasing drastically. With that, the security issues of these connected devices and their associated network is concerning. In some applications, a security breach in an IoT device can lead to serious ramifications. For instance, hacking into a control system of a manufacturing plant can put the entire production process to a stop; intruding on critical biomedical devices such as a pacemaker or an Implantable Cardioverter Defibrillator can potentially risk the life of the user. Therefore, the security challenges of such devices against cyber-attacks are of paramount importance and critical when it comes to determining the future success of IoT. In this paper, a systemization of knowledge regarding the lightweight cryptographic algorithms area for IoT based devices has been provided to better understand the limitation of IoT devices and their design constraints. We identified in this study not only the real-world applications of IoT devices with their constraint resources but also the security challenges and security threats related to IoT devices. Also, we provided an exhaustive survey of lightweight cryptographic algorithms proposed by various researchers. According to this survey, we recommended two lightweight algorithms to address the security needs of IoT devices.

Keywords: IoT · Lightweight cryptography · SoK · Survey · Security

1 Introduction

Internet of Things (IoT) is an interconnection of two or more small size resource constraint devices. These devices are composed of software, electronics, and sensors that allow them to exchange data with each other and other IP enabled devices. It includes devices and objects connected over the Internet Protocol (IP) such as personal computing devices, laptops, tablets, desktop computers, smartphones, and it also includes devices that are connected through non-IP protocols (LLDP, and Spanning Tree are data link layer protocol) [1]. IoT devices are also called smart devices, such as networked home appliances controlled or monitored remotely (example: smart AC, smart TV, smart washing machine, microwave oven, lighting, heating, smartwatch, personal healthcare device, etc.), Sensor networks used by industry for automation and remote-control cars. IoT devices work without any screen or without any user interface, and it usually consumes power from the battery for a regular task. IoT devices are dedicated to a single

K. Arai et al. (Eds.): FTC 2020, AISC 1290, pp. 780–795, 2021.
https://doi.org/10.1007/978-3-030-63092-8_52

task. Most of the IoT devices use very low power microcontrollers, so, these devices can give a small fraction of their computing power to security [2]. If we use the conventional cryptographic algorithms on these devices to provide security, the conventional algorithms may consume too much latency or too much power on these devices. The conventional algorithm may also consume more portion of RAM and ROM. IoT devices like sensor networks (used to measure average temperature in a particular geographical area, planted at various site in that geographic area) getting power from either battery or solar plant and RFID chips (used to identify places of an item in a mall) getting power from the battery, have very limited energy and implementing modern cryptographic algorithms on these devices to provide security is not feasible. To address the security needs of these devices, we need a lightweight cryptographic algorithm, which consumes the low computing power of these devices to run the algorithm [3]. Lightweight cryptography is a subfield of cryptography that is implemented to run on resource constraint devices by consuming less fraction of the computing power of those devices. To secure the design of a cryptographic algorithm, a balance between performance and resources is required. The performance of a cryptographic algorithm is defined in terms of power/energy consumption, latency, and throughput [2, 4].

If the algorithm is implemented in hardware then the required resources used to run algorithms are defined in terms of gate area (GE), gate equivalents, latency, throughput, power/energy consumption, etc. The area can be defined by the number of slices for FPGA or by gate equivalent for ASIC implementation. On FPGA, a slice is the basic reconfigurable unit, which contains the number of lookup tables, flip-flops, and multiplexers. One GE defines the area needed by a two-input NAND gate in ASIC (application-specific integrated circuit). In a low-cost RFID TAG, which contains approximately 1,000–10,000 gates, only 5% of the total gate may be used for security needs [5]. According to [6], hardware-based cryptographic algorithms are categorized into ultra-lightweight and lightweight on the basis of on-chip area and complexity. The ultra-lightweight hardware-based algorithm occupies 1000–2000 logic gates. And maximum limit in lightweight cryptographic algorithms is 3000 logic gates. According to [6], “the same implementation of PRESENT produces 1075GE on 0.18 μm, 1169GE on 0.25 μm and 1000GE on 0.35 μm CMOS technology”. If the algorithm is implemented in software then the required resources used to run algorithms are defined in terms of the number of registers, required number of bytes of RAM and ROM, execution time, latency, and power/energy consumption. For a software-based ultralightweight cryptographic algorithm, 4 KB ROM and 256 bytes RAM are needed. And maximum limit in the lightweight cryptographic algorithm is 32 KB ROM and 8 KB RAM. According to [6], the factor that has an impact on the implementation characteristics of a cryptographic algorithm is CMOS technology. “Different technologies and standard cell libraries produce a different response”. The main aim of the software-based lightweight cryptographic algorithm is to utilize less amount of memory and CPU cycle to reduce power consumption and compatible devices cost.

The lightweight cryptographic algorithm can be implemented either in hardware or in software. In this project, the main focus will be given on the symmetric lightweight cryptographic algorithm.

The outline of this project is as follows:

Section 2: The lightweight cryptographic primitives.

Section 3: The design constraint of lightweight cryptography.

Section 4: The performance evaluation parameters for the lightweight cryptographic algorithm.

Section 5: IoT based security challenges, threats, and its solution.

Section 6: Some real-world applications of IoT devices that need a cryptographic solution.

Section 7: Propose two lightweight cryptographic algorithms for this area of research.

2 Lightweight Cryptographic Primitives

Lightweight cryptographic primitives define that lightweight cryptographic algorithms should be based on a well-established and low-level protocol. Lightweight cryptographic primitives are the basic building block of lightweight cryptographic algorithms. NIST and CRYPTREC approved the following four types of cryptographic primitives for resource constraint devices like IoT:

2.1 Lightweight Block Ciphers

Lightweight block ciphers are the block ciphers, which are used by resource constraint devices like (devices used in smart home automation systems, smartphones, and remote health monitoring tools). Lightweight block cipher uses smaller block size, smaller key size, simpler rounds, and simpler key schedule mechanism and it supports minimal implementation compare to modern block cipher used for desktop and laptop. It takes a block of data as input (plaintext and key) and produces a block of data as output (ciphertext). According to [7–14], Lightweight block ciphers can be further categorized on the basis of its internal structure as follows:

- Substitution Permutation Networks (SPNs): In substitution permutation-based block cipher, the plaintext is processed through a sequence of permutation and substitution boxes to produce ciphertext. Example: AES, NOEKEON, ICEBERG, mCrypton, PRESENT, PRIDE, LED, PRINCE [8–13].
- Feistel Networks (FNs): In a block cipher which is based on Feistel network, plaintext block is divided into two halves, a diffusion function is applied on one-half of the data of the block in a round and output of this diffusion function is then XORed with the other half of the data in that particular round to apply diffusion in the other half of the block. The XORed operation needs 2.5-3 gate equivalents per bit which are extra overhead in Feistel based block cipher compared to SPN based cipher. Examples: TDES, Camellia, CALEFIA, Piccolo [15–19].
- Add-Rotate-XOR Networks (ARXNs): ARXNs based block cipher uses addition, rotation, and XOR operation without any substitution box. ARXNs based block cipher produces the compact and fast result, but its security property is not analyzed properly compared to SPN and FN based cipher. Example: IDEA, HIGHT, SPECK [20–22].

- Non-Linear Feedback Shift Register based (NLFSR): This block cipher utilized the building block of a stream cipher, which is mainly implemented in hardware. Example: KeeLoq, KATAN, KATANTAN [23, 24].
- Hybrid: Hybrid ciphers are the combination of SPN, FN, and ARXN to take advantage of all three-block cipher, like throughput. An example of Hybrid Cipher is the Hummingbird family [25, 26], which is a special case with a hybrid structure of stream and block cipher.

2.2 Lightweight Stream Ciphers

A lightweight stream cipher is a lightweight cryptographic primitive. In the stream cipher, the encryption-decryption operation is faster compared to block cipher, and the error propagation rate is less compared to block cipher. Stream cipher can be further categorized on the basis of its internal state into the following category [42]:

- Synchronous stream cipher: in synchronous stream cipher, generated keystream is independent of plaintext message (P) and ciphertext message(C). According to [47], "synchronous stream ciphers update the internal state independently from plaintext or ciphertext data". In a synchronous stream cipher, the sender and receiver should be synchronized with each other before starting encryption and decryption operation. If a bit of cipher is modified during communication only corresponding single plaintext bit will be affected. Synchronous stream cipher usually suffered from an active attack. Example: E0, A5/1 [27, 28].
- Asynchronous or self-synchronous stream cipher: In an asynchronous stream cipher, generated keystream is a function of key K and some fixed number of previous plaintext digit or ciphertext digit. In asynchronous stream cipher changes in one bit of ciphertext affects many bits in decrypted plaintext. Asynchronous stream cipher usually suffered from a chosen ciphertext attack. Example: SSS, MOUSTIQUE [29, 30].
- Linear feedback shift register based stream cipher: LFSR is a shift register, which takes input bit from the linear function of its previous state. The most common linear function which is used in LFSR based stream cipher is XOR. LFSR based ciphers are good for hardware-based implementation. It is used to generate a pseudorandom bitstream with good cryptographic properties. Its hardware-based implementation is more economical compared to software-based implementation in terms of money, speed, and simplicity. Example: E0, A5/1 [27, 31].
- Non-linear feedback shift register based stream cipher: Nonlinear feedback shift registers based stream ciphers were implemented to increase the security of LFSR based stream cipher. Nonlinearity can be introduced by adding clock-controlled generators, filter generators. The output of two or more parallel LFSR can be passed to nonlinear combining Boolean function to introduced nonlinearity into the stream cipher. NLFSR based stream ciphers are more resistant against cryptanalytic attacks compared to LFSR based stream ciphers. Implementation of NLFSR based stream cipher in hardware is more complex compared to LFSR based stream ciphers. Example: Grain, MICKEY 2.0, and A2U2 [32–34].

2.3 Lightweight Hash Functions

A hash function is a function, which takes the input of the variable size and produces an output of fixed size (n bits). It is used to verify message integrity, digital signature, and figure prints. A hash function with n bit output size should be collision resistance of $2^{n/2}$ and preimage and secondary preimage resistance of 2^n. Lightweight hash functions are important in resource constraint devices to reduce hardware and energy consumption. The lightweight hash function uses smaller internal state and output size, and smaller message size. The signature of a hash function should be determined by a number of state bits and the size of functional and control logic used in a round function. Example: Keccak, PHOTON, QUARK, and SPONGENT [35–38].

2.4 Lightweight Message Authentication Code and Authenticated Encryption

This is a mechanism that provides performance and resource requirement advantages. This can be done by providing confidentiality, authentication, and integrity protection to the data which is used in a resource constraint environment to reduce the energy consumption and implementation area. Example: SipHash, Chaskey, CCM, GSM [39–45].

3 Design Constraint of Lightweight Cryptography

A design constraint refers to a limitation on the requirement and/or operation conditions under which lightweight algorithm expected to operate.

Lightweight cryptography is optimized in terms of the amount of memory consumption, implementation size, and computation speed of the algorithm. The following Table 1 shows a comparison between design constraints of a lightweight algorithm implemented on hardware vs. software platform [4, 46, 48, 49]:

Table 1. A Comparison Between Design Constraints of a Lightweight Algorithm Implemented on Hardware vs. Software Platform

Design Constraints for Algorithms Implemented in Hardware	Design Constraints for Algorithms Implemented in Software
1. Metrics used for memory consumption is gate area or gate equivalent. It represents the physical size of the circuit used to implement a lightweight algorithm	1. Metrics used for memory consumption are RAM consumption and code size
2. Throughput is defined as the number of plaintext processes in unit time. It is measured in bytes/second	2. Throughput is defined as the number of bytes per CPU cycle

(*continued*)

Table 1. (*continued*)

Design Constraints for Algorithms Implemented in Hardware	Design Constraints for Algorithms Implemented in Software
3. The most expensive part of the hardware-based lightweight algorithm is memory consumption	3. The most expensive part is memory and power consumption during the encryption-decryption operation
4. A Fair Comparison between hardware-based algorithms is difficult, because of the cost factor of the circuit and tools used to simulate those circuits	4. A fair comparison between software-based algorithms on the same hardware platform is more feasible and better compared to the comparison between hardware-based algorithms
5. Energy and power efficiency is the main focus of the hardware-based lightweight algorithm	5. The main focus is given on the processing speed of the algorithm; slow operation drains more energy
6. Low latency (time used in encryption, decryption operation) is one of the desired features of the hardware-based lightweight algorithm	6. Low latency (time used in encryption, decryption operation) is the desired property of a software lightweight algorithm

4 Performance Evaluation Metrics for Lightweight Cryptographic Algorithm

According to [6], the best cryptographic algorithm is one that provides a good level of security and makes a balance between the level of security, cost of security, and performance of IoT devices. According to [6, 50, 51], Lightweight cryptographic algorithm can be evaluated based on the following metrics:

- Security level: It is used to measure the strength of lightweight cryptographic primitives. It is measured in terms of the number of bits. An n bit security level means the attacker needs to perform 2^{n} operations to break the security of cryptographic primitives. Usually, the security level of a cryptographic algorithm is defined by the length of the key used in that algorithm. The security level of the cryptographic algorithm cannot exceed more than its key length, but it can be less than or equal to key length depends on the reported attack on that algorithm.
- Hardware technology: Cryptographic algorithms are implemented by using CMOS technology by an occupied area measured in µm. According to [6], the technologies used in lightweight cryptography research papers are 0.13 µm and 0.18 µm. The complexity and the area occupied by the hardware implementation are measured in gate equivalent (GE) metric and depend on the used technology.
- Throughput: It is measured as the amount of encrypted or decrypted data (in bits) in a unit time at a specific frequency. It is measured in Kb/s. In most of the research papers, the hardware-based lightweight algorithm uses a 100 kHz frequency and a software-based lightweight algorithm uses 4 MHz frequency.

- Latency: It is defined as a number of clock cycles utilized to compute a block of plaintext/ciphertext.
- Power and energy consumption: The power of hardware implementations is measured and evaluated in µW. It is estimated on the basis of gate equivalent and hardware technology. Energy consumption per bit is calculated by using the following formula: Energy = (latency * power)/block size
  Where Energy is measured in µJ, Latency is measured as a number of CPU clock cycle/block (plaintext block/Ciphertext block), Power is measure in µW that are consumed by the hardware or software implementation, and Block size is the size of the data in bits which can be processed in one encryption or decryption operation.
- RAM/ROM memory: It is the amount of RAM and ROM in bytes used by the algorithm. RAM bytes used to store the intermediate operation in encryption-decryption operation and ROM bytes represent the code size which is actually the size of the implementation.
- Efficiency: Usually there is a tradeoff between security and performance in cryptographic algorithms. A higher value of efficiency means better performance.
  For hardware implementations, efficiency is calculated by using the following formula: Hardware Efficiency = Throughput [Kbps]/Complexity [KGE]
  Where throughput is the Kb/s the cipher's encryption operation achieves at 100 kHz frequency and the complexity is the value of the chip area in KGE.
  For software implementations, the efficiency is calculated by using the following formula: Software Efficiency = throughput [Kbps]/Code size [KB]
  Where throughput is the Kb/s the cipher's encryption operation achieves at 4 MHz frequency and code size is the size of the executable code in KB.

5 IoT Security Challenges, Threats, and Solutions

5.1 Security Challenges Related to IoT Devices [52–56]

1. Communication between Constraints and Heterogeneous network: interconnection between resource constraint networks with the internet is a challenging task because of heterogeneity of both networks which complicates protocol design and system operation. Coordination between heterogeneous networks is a challenging task.
2. Lack of human intervention: Due to a lack of human interventions, there are more chances of physical and logical attacks.
3. Active and passive attack: Security vulnerability in an IoT sensor leads to many attacks like DoS/DDoS, replay attack, eavesdropping, and many others.
4. Poor Efficiency: Efficiency of IoT devices are not up to mark because of the limitation of power consumption, battery life, bandwidth, heterogeneous platforms, and complicated security methodology.
5. Secure software update and crypto-agility: according to [54], there are several challenges that make secure software update mechanism difficult, like 1) no incentive for manufacturers, vendors, and others on the supply chain to issue updates for their devices. 2) Sometimes source code of the software running on IoT devices is simply not available. Without the availability of complete source code, patches cannot be written for that piece of code. 3) Sometimes updates are available, but users never

get alert about security updates. Cryptographic algorithms can be outdated due to advances in technology and cryptanalysis techniques.

6. Verifying device behavior: users who are using internet connected IoT devices are not aware that their privacy can be compromised due to these devices. According to many survey reports, it has been found that many IoT Vendors are using these devices to collect a user's personal sensitive data without the user's knowledge.
7. Testing IoT devices to identify unwanted vulnerabilities: due to the resource constraint nature of IoT devices, users and developers cannot perform extensive testing on their IoT devices to identify the unwanted vulnerability. So, vulnerabilities remain with IoT devices and can be exploited in the letter stage.
8. Guaranteed security, protection, data trustworthiness, and user privacy: Even if communication links between IoT devices are encrypted, a static sensor emits a packet based on the presence or absence of the people in its range. Anyone who observes these packets by using some sniffing tools can collect private sensitive information.

5.2 Security Threats Related to IoT Devices [57–59]

Security is an important factor in any communication system. In the case of real-world IoT based communication system; security is a very critical requirement because of the following reasons:

- An attack in IoT systems does not only compromise the privacy and security of a user, but it can also cause physical harm. Because, IoT systems consist of sensors, actuators, and other connected devices in the physical environment of the user, which could badly affect the user if they are compromised.
- A vulnerable IoT system does not only affects the manufacturer's brand image, but it can also leak information that is very sensitive for the business of the manufacturer.
- The impact of attacking an IoT system can go beyond a specific device because compromised IoT systems can be misused anywhere in the network. A compromised IoT system can be used to perform a DDoS attack that affects the availability of other networks and services in the network.
- IoT systems rely on standard IP protocols that allow easier system integration, but it also makes attacks on standard IP protocols.

5.3 Security Solution [59–61]

1. Use lightweight cryptographic primitives to reduce resource consumption of security protocol.
2. Careful protocol design and usage reduce energy consumption during normal operation and under DoS attack.
3. Devices that are not able to verify a cryptographic signature should not be connected to the Internet.
4. Source authentication should be required for secure software and firmware updates.
5. Use commercial devices or software services to learn and monitor the behavior of different IoT devices in each network. Such monitoring devices or services can be configured by the user to restrict network traffic and trigger alarm when an unusual operation of IoT devices detected.

6. User awareness and training program.

6 Applications of Lightweight Cryptography in IoT Devices

Nowadays, IoT applications are used almost everywhere in our surroundings. According to ARM Report, IoT connected devices across all technologies will reach one trillion IoT devices by 2035. With the launch of the 5G network, superior sensors, and revolutionary computing capabilities, the Internet of Things (IoT) could be the next frontier.

6.1 Smart Traffic Camera [62, 63]

IoT based devices can be used to solve common issues in a city like traffic control, clean drinking water problem, air quality, increasing urban population in different areas in a city, waste management. Example: traffic jams in a city is a very common problem nowadays. Smart traffic cameras installed on the road can be used to monitor traffic congestion, accidents, weather condition, and communicate this data to the city traffic management board, where this information is analyzed to manage traffic better way. A smart traffic system can learn and predict traffic patterns in a particular area by using machine learning. The traffic management systems can use this predicted data to analyze and derive the route around the project to avoid bottlenecks. Traffic information about a particular area could be passed to drivers through smart devices to take a better decision about which rout to follow to reach a particular destination. Accuracy and Integrity of traffic data pass from various traffic cameras to traffic management board is very important to take the right decision to divert the traffic on alternative less traffic route. Lightweight message authentication code (MAC) can be used here to maintain the integrity of traffic data.

6.2 Wearable Technologies [64, 65]

Wearable devices are in demand now a day. Companies like Google and Samsung invested heavily in building wearable devices like Fit Bits, heart rate monitoring devices, smartwatches, etc. sensors and software are installed in these devices to collect user data and information to preprocess and extract useful information about the user. These devices are usually highly energy-efficient, ultra-low power, and small-sized.

6.3 Smart Home [66–68]

The smart home market is booming now a day because it provides more convenience and comfort to the user. In the smart home, all devices in a home-like clock, speaker, doorbells, light, cameras, windows, geyser, fridge, TV, AC, washing machine, and many more connected to the internet and communicate with the user in a home, follow user's command. These devices provide convenience and comfort to the user at home but at the same time, it can be the cause of big cyber threats due to the connectivity of the Internet. Since IoT devices used in the home are resource constraints and in-depth security analysis of these devices is not possible due to limited resources. IoT vulnerable devices used in the home can be an attractive target for the attacker. These vulnerable devices can be utilized as a botnet by the cybercriminals to perform DDoS attacks. The vulnerable internet-connected doorbell can be manipulated by a burglar to enter inside the house. According to Kaspersky [77], Fibaro smart home allowed uploading and downloading of smart hub's backup data to and from the cloud server, to anyone without authentication. The most vulnerable point in a smart home is a vulnerable IoT hub that is connected with user phones, computers, or laptops. All devices in the smart home directly communicate with the hub and hub sends an alert signal to the user if any unusual things happen inside the home. If the attacker gets control over the hub, he will get control over all devices used in the home. The attacker can manipulate the camera, door locks, alarm, and many more. According to Scott Robinson [78], the hubs root account password is usually DES-encrypted by default and can be easily cracked by using a brute-force attack. Cyberattacks on smart homes not only leak personal sensitive information of victim users, but it can also be the cause of physical harm to the victim.

6.4 Health Care System [69–73]

The demand for IoT devices in healthcare is growing tremendously due to the following factors: 1) technological advancements, 2) better utilization of high-speed internet, 3) to make collaboration between top IT companies and health care system, 4) decreased price of sensor technology.

Benefits Related to IoT Devices in the Health Care System:

1. Remotely monitor patient
2. Patient drug supply
3. Electronic health implants
4. Hospital and building management
5. Patient engagement
6. Data collection

According to many survey reports, it is found that 60% of the health care organization utilizing IoT in their health care system. Nowadays, approximately 10-15 IoT devices per bed are used in the US health care system. These devices help patients to improve their health, improve the efficiency of the health care system. It helps in innovative medical research. At the same time, it is an attractive target for cyber-criminal.

Risks Related to IoT Devices in the Health Care System:
Internet-connected IoT devices in healthcare can be an attractive target for the attacker for the following reasons:

1. There may be a security gap between IoT devices connected to the network in the health care system.
2. IoT devices used in the health care system may contain valuable information like personally identifiable information and personal health information about high profile patients, which be exploited by an attacker for monetary benefit. Managing IoT

security in the Healthcare system is challenging and headache, because security vulnerabilities in IoT devices could affect people's safety, and even can cause life and death implications.

6.5 Industrial Automation [74]

By using IoT in industry automation, products and their packaging can be reengineered to deliver better performance. We can utilize IoT in the industry for the following tasks: Digitalization of factory, product flow monitoring, inventory management, quality control, logistics, and supply chain optimization, packaging optimization, safety, and security.

7 Lightweight Cryptographic Categories

According to the survey of lightweight cryptographic primitives for IoT, it has been proposed to divide the lightweight algorithms into the following categories to be optimized [75, 76]:

7.1 Ultralightweight Cryptography

This kind of cryptography has been developed to provide very efficient and secure solutions. It has many properties, such as data confidentiality, resistance to disclosure attack, anonymity, and resistance to tracking, resistance to replay attack, resistance to impersonation attacks, explicit key confirmation, and resistance to de-synchronization attack, and forward security. Ultralightweight Cryptography provides one function on one platform with high performance as it deals with very specialized algorithms, Examples of this classification are Grain, Qarma, and Chaskey. For the coming up years, the ultralightweight field appears to be an exciting and challenging research topic.

7.2 Ubiquitous Cryptography

This kind of cryptography deals with primitives that are provided according to the functionality and implementation properties. Also, it has important features, such as providing versatility, establishing high-speed cipher, and achieving strong cipher strength and by this, it can encrypt the full plaintext directly without division. It runs on several platforms, such as microcontrollers, FPGAs, and ASICs. Also, it has many functions, for instance, hashing, encryption, and authentication.

8 Conclusion and Future Plan

This project presented security challenges faced by the IoT devices, their design constraints, the latest development in the field of lightweight symmetric key algorithms, and the performance evaluation metrics for these lightweight algorithms. It has been concluded that Lightweight cryptography is one of the main security solutions for internet connected IoT devices. Because of the efficiency and smaller footprint of the lightweight primitives, we believe that these primitives should be implemented in the smart networks, specifically lightweight block ciphers which are practical to use nowadays. Our recommendation for the designers is to use ultralightweight cryptography and ubiquitous cryptography. By using these two categories, they will be able to classify their algorithms more precisely.

The future plan is to investigate the available cryptographic algorithms feasible for IoT devices further and extract their strengths and weaknesses in more depth to get a better understanding about them and using those insights during the design and implementation of the planning practical system. Also, we would like to optimize the efficiency of the implementation, and the suitability of the lightweight solution for the application in resource-constrained IoT devices.

References

1. Sethi, P., Sarangi, S.R.: Internet of things: Architectures, protocols, and applications. J. Electric. Comput. Eng. **2017**, 1–25 (2017)
2. Trappe, W., Howard, R., Moore, R.: Low-energy security: limits and opportunities in the internet of things. IEEE Secur. Privacy **13**(1), 14–21 (2015)
3. Goyal, T., Sahula, V., Kumawat, D.: Energy efficient lightweight cryptography algorithms for IoT devices. IETE J. Res. 1–14 (2019)
4. Eprint.iacr.org (2020). https://eprint.iacr.org/2017/511.pdf. Accessed 15 April 2020
5. Nvlpubs.nist.gov (2020). https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8114.pdf. Accessed 18 April 2020
6. Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., Manifavas, C.: A review of lightweight block ciphers. J. Cryptographic Eng. **8**(2), 141–184 (2017). https://doi.org/10.1007/s13389-017-0160-y
7. Heron, S.: Advanced encryption standard (AES). Network Secur. **2009**(12), 8–12 (2009)
8. Daemen, J., Peeters, M., Van Assche, G., Rijmen, V.: The NOEKEON Block Cipher, pp. 1–30 (2000)
9. Standaert, F., Piret, G., Rouvroy, G., Quisquater, J., Legat, J.: Iceberg: an involutional cipher efficient for block encryption in Reconfigurable hardware. Fast Software Encryption, pp. 279–298 (2004)
10. Lim, C.H., Korkishko, T.: MCrypton – a lightweight block cipher for security of low-cost RFID tags and sensors. Inf. Secur. Appl. 243–258 (2006)
11. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74735-2_31
12. Albrecht, M.R., Driessen, B., Kavun, E.B., Leander, G., Paar, C., Yalçın, T.: Block ciphers – focus on the linear layer (feat. PRIDE). In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 57–76. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_4
13. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçın, T.: PRINCE – a low-latency block cipher for pervasive computing applications. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_14

14. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED Block Cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23951-9_22
15. Csrc.nist.gov (2020). https://csrc.nist.gov/CSRC/media/Publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf. Accessed 24 April 2020
16. Satoh, A., Morioka, S.: Hardware-focused performance comparison for the standard block ciphers AES, camellia, and Triple-DES. Lecture Notes in Computer Science, pp. 252–266 (2003)
17. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., & Tokita, T. (2001). Camellia: A 128-Bit block cipher suitable for multiple platforms — Design andAnalysis. Selected Areas in Cryptography, 39–56
18. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-Bit Blockcipher CLEFIA (Extended abstract). Fast Software Encryption, pp. 181–195 (2007)
19. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: *Piccolo*: an ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342–357. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23951-9_23
20. Lai, X., Massey, J.L.: A proposal for a new block encryption standard. Advances in Cryptology — EUROCRYPT 1990, pp. 389–404 (1991)
21. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S.: HIGHT: a new block cipher suitable for low-resource device. Lecture Notes in Computer Science, pp. 46–59 (2006)
22. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK lightweight block ciphers. Proceedings of the 52nd Annual Design Automation Conference on - DAC 2015 (2015)
23. Indesteege, S., Keller, N., Dunkelman, O., Biham, E., & Preneel, B. (n.d.). A practical attack on KeeLoq. Advances in Cryptology – EUROCRYPT 2008, 1–18
24. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. Lecture Notes in Computer Science, pp. 272–288 (2009)
25. Engels, D., Fan, X., Gong, G., Hu, H., Smith, E.M.: Hummingbird: ultra-lightweight cryptography for resource-constrained devices. Financial Cryptography and Data Security, pp. 3–18 (2010)
26. Engels, D., Saarinen, M.O., Schweitzer, P., Smith, E.M.: The hummingbird-2 lightweight authenticated encryption algorithm. RFID. Security and Privacy, pp. 19–31 (2012)
27. Biryukov, A., Shamir, A., Wagner, D.: Real time cryptanalysis of A5/1 on a PC. Fast Software Encryption, pp. 1–18 (2001)
28. De Cannière, C., Preneel, B.: Trivium. Lecture Notes in Computer Science, pp. 244–266 (2008)
29. Ecrypt.eu.org (2020). The Estream Portfolio Page. https://www.ecrypt.eu.org/stream/. Accessed 25 April 2020
30. Käsper, E., Rijmen, V., Bjørstad, T.E., Rechberger, C., Robshaw, M., Sekar, G.: Correlated keystreams in moustique. Progress in Cryptology – AFRICACRYPT 2008, pp. 246–257 (2008)
31. Galanis, M., Kitsos, P., Kostopoulos, G., Sklavos, N., Koufopavlou, O., Goutis, C.: Comparison of the hardware architectures and FPGA implementations of stream ciphers. In: Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems, 2004. ICECS (2004)
32. Hell, M., Johansson, T., Meier, W.: Grain: a stream cipher for constrained environments. Int. J. Wireless and Mob. Comput. **2**(1), 86 (2007)
33. Ecrypt.eu.org (2020). http://www.ecrypt.eu.org/stream/p3ciphers/mickey/mickey_p3.pdf. Accessed 27 April 2020

34. David, M., Ranasinghe, D.C., Larsen, T.: A2U2: a stream cipher for printed electronics RFID tags. In: 2011 IEEE International Conference on RFID (2011)
35. Kavun, E.B., Yalcin, T.: A Lightweight Implementation of Keccak Hash Function for Radio-frequency Identification Applications, pp. 258–269. Security and Privacy Issues, Radio Frequency Identification (2010)
36. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON Family of Lightweight Hash Functions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 222–239. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_13
37. Aumasson, J.-P., Henzen, L., Meier, W., Naya-Plasencia, M.: QUARK: A Lightweight Hash. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 1–15. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15031-9_1
38. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., Verbauwhede, I.: SPONGENT: a lightweight hash function. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 312–325. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23951-9_21
39. Nvlpubs.nist.gov (2007). https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf. Accessed 28 April 2020
40. Nvlpubs.nist.gov (2004). https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf. Accessed 28 April 2020
41. Nvlpubs.nist.gov (2005). https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38b.pdf. Accessed 29 April 2020
42. Aumasson, J.: The impact of quantum computing on cryptography. Comput. Fraud Secur. **2017**(6), 8–11 (2017)
43. Aumasson, J., Bernstein, D.J.: SipHash: a fast short-input PRF. Lecture Notes in Computer Science, pp. 489–508 (2012)
44. Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: an efficient MAC algorithm for 32-bit microcontrollers. In: Joux, A., Youssef, A. (eds.) SAC 2014. LNCS, vol. 8781, pp. 306–323. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13051-4_19
45. Mouha.be. 2020. Chaskey - Nicky Mouha. https://mouha.be/chaskey/. Accessed 1 May 2020
46. Buchanan, W.J., Li, S., Asif, R.: Lightweight cryptography methods. J. Cyber Secur. Technol. **1**(3–4), 187–201 (2017)
47. Biryukov, A., Perrin, L.: State of The Art in Lightweight Symmetric Cryptography. Semanticscholar.org (2020). https://www.semanticscholar.org/paper/State-of-the-Art-in-Lightweight-Symmetric-Biryukov-Perrin/532441547d905feae7a65f635594585c96d2987b. Accessed 1 May 2020
48. Manifavas, C., Hatzivasilis, G., Fysarakis, K., Papaefstathiou, Y.: A survey of lightweight stream ciphers for embedded systems. Secur. Commun. Netw. **9**(10), 1226–1246 (2015)
49. Gunathilake, N.A., Buchanan, W.J., Asif, R.: Next generation lightweight cryptography for smart IoT devices: implementation, challenges and applications. In: 2019 IEEE 5th World Forum on Internet of Things (WF-IoT) (2019)
50. Csrc.nist.gov (2020). Lightweight Cryptography | CSRC. https://csrc.nist.gov/Projects/Lightweight-Cryptography. Accessed 1 May 2020
51. Cryptrec.go.jp (2020). https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016en.pdf. Accessed 2 May 2020
52. Stout, W. M., & Urias, V. E.: Challenges to securing the Internet of things. In: 2016 IEEE International Carnahan Conference on Security Technology (ICCST) (2016)
53. Singh, S., Singh, N.: Internet of things (IoT): security challenges, business opportunities & reference architecture for e-Commerce. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (2015)

54. Gupta, K., Shukla, S.: Internet of things: security challenges for next generation networks. In: 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH) (2016)
55. Guarda, T., Leon, M., Augusto, M.F., Haz, L., De la Cruz, M., Orozco, W., Alvarez, J.: Internet of things challenges. In: 2017 12th Iberian Conference on Information Systems and Technologies (CISTI) (2017)
56. Nzabahimana, J.P.: Analysis of security and privacy challenges in Internet of things. In: 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT) (2018)
57. Sezer, S.: "T1C: IoT Security: - Threats, Security Challenges and IoT Security Research and Technology Trends,": 31st IEEE International System-on-Chip Conference (SOCC). Arlington **2018**, 1–2 (2018)
58. Lee, Y., Park, Y., Kim, D.: Security threats analysis and considerations for Internet of things. In: 2015 8th International Conference on Security Technology (SecTech) (2015)
59. Rajendran, G., Ragul Nivash, R.S., Parthy, P.P., Balamurugan, S.: Modern security threats in the Internet of things (IoT): attacks and countermeasures. In: 2019 International Carnahan Conference on Security Technology (ICCST) (2019)
60. Shifa, A., Asghar, M.N., Fleury, M.: Multimedia security perspectives in IoT. In: 2016 Sixth International Conference on Innovative Computing Technology (INTECH) (2016)
61. ITU News. 2020. ARM Predicts 1 Trillion Iot Devices By 2035 With New End-To-End Platform. https://news.itu.int/arm-pelion-iot-end-to-end-platform/. Accessed 2 May 2020
62. ReadWrite (2020). Iot And Cameras: Going from Smart to Intelligent. https://readwrite.com/2016/07/22/cameras-smart-intelligent-dt2/. Accessed 2 May 2020
63. Sharif, A., Li, J., Khalil, M., Kumar, R., Sharif, M.I., Sharif, A.: Internet of things - smart traffic management system for smart cities using big data analytics. In: 2017 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP) (2017)
64. Pasluosta, C.F., Gassner, H., Winkler, J., Klucken, J., Eskofier, B.M.: An emerging era in the management of Parkinson's disease: wearable technologies and the Internet of things. IEEE J. Biomed. Health Inform. **19**(6), 1873–1881 (2015)
65. U.S. 2020. Google Taps Fitness Tracker Market With $2.1 Billion Bid for Fitbit. https://www.reuters.com/article/us-fitbit-m-a-alphabet/google-taps-fitness-tracker-market-with-2-1-billion-bid-for-fitbit-idUSKBN1XB47G. Accessed 3 May 2020
66. Stojkoska, B.R., Trivodaliev, K.: Enabling Internet of things for smart homes through fog computing. In: 2017 25th Telecommunication Forum (TELFOR) (2017)
67. Ur Rehman, S., Gruhn, V.: An approach to secure smart homes in cyber-physical systems/internet-of-Things. In: 2018 Fifth International Conference on Software Defined Systems (SDS) (2018)
68. Mahmud, S., Ahmed, S., Shikder, K.: A smart home automation and metering system using Internet of things (IoT). In: 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (2019)
69. Hu, F., Xie, D., Shen, S.: On the application of the Internet of things in the field of medical and health care. In: 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing (2013)
70. Riazul Islam, S.M., Kwak, D., Humaun Kabir, M., Hossain, M., Kwak, K.-S.: The Internet of things for health care: a comprehensive survey. IEEE Access **3**, 678–708 (2015)
71. Das, S., Ballav, M., Karfa, S.: Application of IoT in detecting health risks due to flickering artificial lights. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (2015)

72. Abouzakhar, N. S., Jones, A., Angelopoulou, O.: Internet of things security: a review of risks and threats to healthcare sector. In: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (2017)
73. Blackberry.com (2020). https://www.blackberry.com/content/dam/blackberry-com/asset/enterprise/pdf/wp-cybersecurity-healthcare.pdf. Accessed 3 May 2020
74. Faul, A., Jazdi, N., Weyrich, M.: Approach to interconnect existing industrial automation systems with the industrial internet. In: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA) (2016)
75. D'Arco, P.: Ultralightweight cryptography. Innovative Security Solutions for Information Technology and Communications, pp. 1–16 (2019)
76. Fukase, M., Sato, T.: Innovative ubiquitous cryptography and sophisticated implementation. In: 2006 International Symposium on Communications and Information Technologies (2006)
77. Kaspersky.com. 2020. Smart Home Hacks. https://www.kaspersky.com/blog/vulnerable-smart-home/27617/. Accessed 4 May 2020
78. Cisco. 2020. Solutions. https://www.cisco.com/c/en/us/solutions/internet-of-things/smart-home-attacks.html. Accessed 4 May 2020