cryptography

MDPI

Article

Hardware Limitations of Lightweight Cryptographic Designs for IoT in Healthcare

Kyriaki Tsantikidou * and Nicolas Sklavos *

SCYTALE Group, Computer Engineering and Informatics Department, University of Patras, 26504 Patras, Greece
* Correspondence: k.tsantikidou@upatras.gr (K.T.); nsklavos@upatras.gr (N.S.)

Abstract: Security is an important aspect of healthcare applications that employ Internet of Things (IoT) technology. More specifically, providing privacy and ensuring the confidentiality, integrity and authenticity of IoT-based designs are crucial in the health domain because the collected data are sensitive, and the continuous availability of the system is critical for the user's wellbeing. However, the IoT consists of resource-constrained devices that increase the difficulty of implementing high-level-security schemes. Therefore, in the current paper, renowned lightweight cryptographic primitives and their most recent architecture, to the best of the authors' knowledge, are investigated. Their security, architecture characteristics and overall hardware limitations are analyzed and collected in tables. Finally, all the algorithms are compared based on their effectiveness in securing healthcare applications, the utilized device and the overall implementation efficiency.

Keywords: hardware security; lightweight cryptography; Internet of Things (IoT); healthcare; embedded systems

check for updates

Citation: Tsantikidou, K.; Sklavos, N. Hardware Limitations of Lightweight Cryptographic Designs for IoT in Healthcare. *Cryptography* **2022**, *6*, 45. https://doi.org/10.3390/cryptography6030045

Academic Editor: Jim Plusquellic

Received: 29 July 2022
Accepted: 29 August 2022
Published: 1 September 2022

1. Introduction

The Internet of Things (IoT) is a network of heterogenous devices that are interconnected with each other and can transmit data to the Internet [1]. The utilized devices are resource constrained with computational and energy limits. In recent years, IoT has been adopted by the healthcare domain because of the new services and extensive capabilities it provides, while being low cost and easily accessible. Nonetheless, security requirements for applications in IoT-based healthcare are also expanding [2]. The design and implementation of security mechanisms are essential in smart health because a malicious attack or a device malfunction can negatively affect the users/patients, even endangering their lives. Common cryptographic algorithms that can potentially secure such systems cannot be employed because their complexity and high area/power demands are ill-suited for IoT devices. Thus, lightweight security solutions that can be easily implemented in these devices, without affecting their basic functionality, are needed.

Lightweight cryptography provides algorithms that can be harnessed by IoT-based healthcare implementations. These cryptographic primitives have simple computational needs; thus, they can be area and power efficient. Their designs on hardware are commonly utilized for IoT systems and are suitable for various health applications that focus on either low-area or high throughput. In other healthcare application cases, the trade-off between these implementational traits and the overall balance of the hardware performance is more important. A drawback to the hardware implementation of lightweight algorithms is the reduction in the security, compared to the familiar heavy cryptographic techniques, because of the decrease in computational complexity. This can be described as a hardware limitation in the IoT environment, namely, the lack of high-level security is directly intertwined with the employed resource-constrained devices' efficiency and hardware traits. Nevertheless, it does not negate the fact that these lightweight alternatives can still provide effective security and fulfil, to some extent, the requirements for a trustworthy e-healthcare system despite the hardware limitations. The efficient and flexible architecture of a lightweight algorithm with perhaps additional security components that enhance the main security concepts, namely, confidentiality, integrity and availability, can confront these hardware limitations and fulfil the security and privacy requirements of smart-health systems.

In the present work, the authors' ongoing research of investigating and designing hardware implementations for lightweight cryptographic algorithms applied in healthcare structures is continued. These lightweight cryptographic primitives are divided into five categories: block ciphers, stream ciphers, hash functions, message authentication codes and authenticated encryption schemes. The authors strive to achieve an efficient architecture that provides adequate security according to healthcare standards for IoT systems by utilizing a cryptographic primitive. For that purpose, in this comparative paper, recently implemented designs that deploy lightweight schemes for IoT security and are simulated to a variety of devices, such as application-specific integrated circuit (ASIC) and field-programmable gate array (FPGA) boards, are examined and compared. Specifically, their performance is analyzed based on their potential utilization in IoT-based health applications. Their general security and design optimizations are presented and all their collected hardware attributes, namely, the employed area, total power, throughput and frequency, are inspected. This paper's final objectives and contributions are the comparison of these architectures and the selection of the few, most suitable and beneficial designs to IoT. The authors want to shed light on the research of IoT security structures conducted to date. The research community will grasp the current state of hardware limitations that exist in state-of-the-art lightweight implementation schemes, namely, the level of security and hardware resource traits achieved by current IoT-based devices and innovative computational approaches. This will benefit every researcher inquiring into the level of resource limitations that IoT-based security approaches possess and the most efficient architecture of an appropriate lightweight cryptographic algorithm for their preferred IoT healthcare application.

Various other papers compare the security mechanisms suggested for IoT systems. However, most of them present other related surveys and draw conclusions based on them, rather than examining technical papers and comparing them based on both hardware and security efficiency. Ref. [3] analyzes the IoT structure and compares some lightweight cryptographic protocols based on the achieved level of security. In [4], the IoT layers and their security requirements are examined, while popular lightweight cryptographic algorithms and their characteristics are discussed. Moreover, Ref. [5] presents common attacks and vulnerabilities of IoT layers with implemented security and cryptography solutions. However, the hardware traits of each examined primitive, such as throughput and resource consumption (LUT, Slices, etc.), are not mentioned in these papers and a proper comparison is not displayed. Ref. [6] highlights the most trusted and researched security primitives in the field of IoT without thoroughly providing the hardware and security optimizations of each referenced paper. Ref. [7] also does not compare different lightweight algorithms but presents the results of various survey papers that implement and then compare lightweight approaches. In [8] and [9], a similar approach to this paper is presented with various cryptographic primitives compared based on performance traits. Nevertheless, the first study focuses only on block ciphers, as opposed to this paper in which algorithms from five cryptographic categories are examined. Moreover, the second study does not provide a flexible conclusion that proposes a variety of security-implemented schemes based on different IoT application requirements. Overall, in this paper, both the hardware and security characteristics of proposed lightweight implemented primitives are investigated and then compared based on the security and implementation efficiency. Therefore, unlike [10], this comparative research, following an extensive investigation of the implemented schemes, proposes the best cryptographic algorithms for IoT devices.

2. IoT-Based Healthcare

A general architecture for an IoT-based healthcare application, as it is depicted in Figure 1, consists of IoT devices and sensors that collect health data from the environment and then transmit them to the cloud services through communication networks. These cloud services make the health data easily accessible to authorized users and healthcare providers.

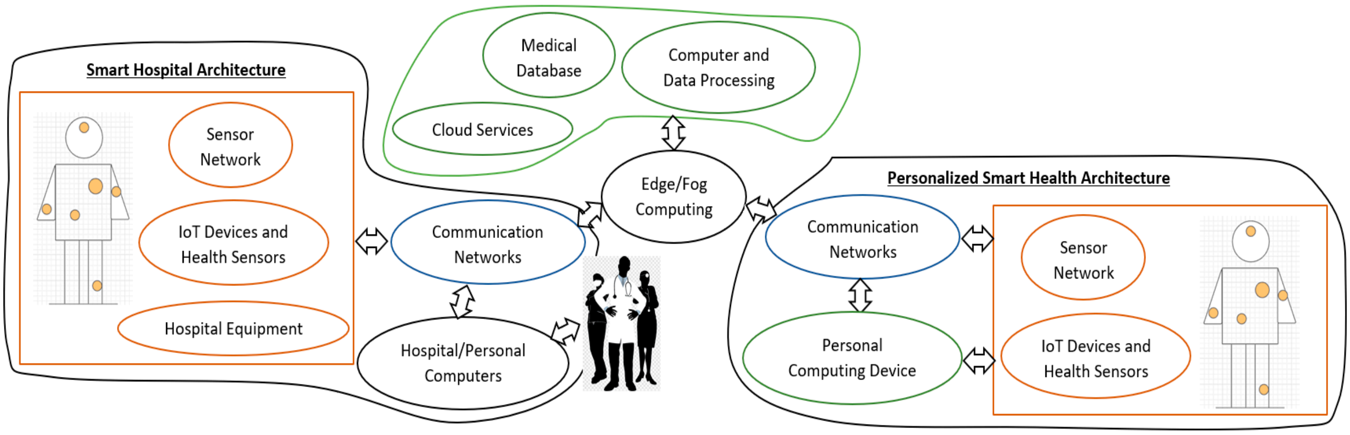

Figure 1. IoT-based healthcare environment (adapted from [2]).

There are two distinct designs, the smart hospital architecture and the personalized smart health architecture, which are connected via the Internet. They both have a similar IoT-based structure. The first layer consists of various IoT devices and perhaps some hospital equipment enhanced with various sensors and actuators. It is mostly composed of health sensors, such as insulin pumps, heart rate sensors and EEG sensors; environmental monitoring sensors, such as air pressure and humidity sensors; position/motion sensors; clinical beds and diagnosis machines. All these devices are interconnected and can directly exchange data via a simple network, such as Bluetooth, ZigBee, Radio Frequency Identification (RFID) and 6LoWPAN. Furthermore, they can communicate with the external environment through the network layer of the architecture, consisting of certain networks, such as Wi-Fi and 5G, and connect to the edge/fog computing and cloud computing layers. In some cases, there is an intermediate or gateway device, namely a personal computing device, such as a mobile phone, between the first and the communication layer. The first-layer devices have very limited power and can sometimes only connect to simple, local networks. Their capabilities and level of security are not fitting for wide-area networks and the Internet. Therefore, via this gateway device, which can easily connect to more power-demanding networks, the resource-constrained IoT system can communicate with the external environment.

The edge/fog computing layer consists of edge and fog computing devices that are connected via the Internet to medical databases and cloud computing servers, namely, the last layer, where an extensive medical history of all patients is maintained and most computations are performed. Edge computing strives to resolve the problem of IoT devices' resource limitations by processing the large amount of collected data at the architecture's network edge before their transmission to the cloud. Thus, it reduces the network load and response latency [3]. In many applications, the hospital/personal computer and the personal computing devices are considered a part of the edge computing layer as they have more computational and memory resources than other IoT devices. They can easily provide solutions to the network edge by executing specific services and operations that are quite resource demanding. Nevertheless, even these devices have some limitations to their resources and speed. Therefore, an additional layer is necessary. The fog computing layer is an extension of the cloud layer and consists of various devices, such as routers, servers and access points (APs). This layer enhances the IoT network edge by offering cloud services and additional resources closer to the computational-constrained devices.

Hence, the latency and communicational load are reduced while allowing the employment of real-time applications and better handling of the IoT's scalability problem [3].

Even though a vast number of health data is collected and utilized by IoT devices and wearables, these components display many security concerns [2]. The IoT network presents many vulnerabilities, being easily exploitable by malicious attacks. Unauthorized parties can potentially gain access and process the data for their own benefits and goals, breaching the data privacy policy that healthcare standards require [11]. Some of the most common attacks that disrupt the confidentiality of IoT systems are eavesdropping, data transmission and traffic-monitoring attacks [12]. Furthermore, the diagnosis of the patient can be distorted by data fabrication and alteration attacks; hence, preventing proper medical treatment that can lead to potentially deadly situations. There can also be cases where a function of the IoT-based healthcare system is interrupted or delayed because of Denial-of-Service (DoS) attacks, endangering the user who heavily depends on these applications. For example, smart health applications that are responsible for instantly responding to patients' health deterioration and providing in real-time first aid or alerting medical services must not be hindered by any kind of attack at any moment of operation. Moreover, due to the scalability and heterogeneity of the IoT environment, common security schemes do not provide different mechanisms in order to cover various application requirements and circumstances at any given time. Conclusively, flexible security mechanisms for IoT in healthcare, which provide and maintain basic security concepts, such as confidentiality, integrity, availability and authentication, must be implemented.

The basic rational for security schemes in IoT-based healthcare applications is explained. This rational follows the new healthcare framework, Health 4.0 [11], which presents creative methods for efficiently implementing the IoT technology by abiding to the security, performance and application requirements a healthcare infrastructure displays. First, a security scheme must utilize an efficient and minimal number of resources for proper employment in resource-constrained IoT devices. Additionally, the performance throughput must be high enough to cope with the high network load and big data transmission. Finally, the security mechanisms must be properly applied and pass formal tests provided by international standards, such as NIST, to ensure their credibility.

Basic IoT Protocols

IoT providers created an IoT protocol architecture based on the TCP/IP protocol Stack, which is divided into five layers: application, transport, network, data link and physical [13]. In each layer, different protocols can be employed depending on the application's requirements. These protocols utilize security mechanisms and particularly implement the cryptographic algorithms that will be analyzed in further sections.

There are three main lightweight protocols that operate and provide security to multiple layers of the IoT's architecture. These protocols are smaller than the Hypertext Transfer Protocol (HTTP) and more suitable for machine-to-machine communications and low-power-bandwidth devices [13]. The first one is the Constrained Application Protocol (CoAP), which includes HTTP features while being suitable for IoT devices. It is composed of two layers, messages and request/response, with each containing a set of methods that achieve Quality of Service (QoS). The CoAP is an application layer protocol that is employed above one of the two basic transport layer protocols: User Datagram Protocol (UDP) or Transmission Control Protocol (TCP). UDP, as a connectionless protocol, is susceptible to the utilized network protocols but has a low overhead and requires minimum memory and computational resources. TCP is a connection-oriented protocol where reliable data transmission is a priority. However, there are no security mechanisms implemented in these two transport layer protocols. Thus, for transmission security, CoAP also employs the Datagram Transport Layer Security (DTLS), which is designed for datagram-based applications and prevents a variety of network attacks, such as eavesdropping and tampering.

The second lightweight protocol utilized in IoT applications is Message Queue Telemetry Transport (MQTT). It is based on the TCP protocol and is suitable for IoT architecture as it requires a minimum bandwidth and less power consumption [13]. Similar to CoAP, MQTT is susceptible to attacks without the addition of any extra security protocols. Transport Level Security (TLS) is a stream-oriented protocol that efficiently secures the TCP and allows three levels of QoS. Moreover, the employment of X.509 certificates further enhances the privacy and client authentication. Nevertheless, the utilization of long strings and even the TCP features can be difficult to implement in a very resource-constrained IoT environment. Many different variations of this protocol have been created in recent years, depending on the performance requirements of each application. Some examples are the Advanced Message Queueing Protocol (AMQP), MQTT Sensor Networks (MQTT—SN) and Secure MQTT.

The last protocol that also utilizes the TLS protocol for security is the Extensible Messaging and Presence Protocol (XMPP). This protocol follows a distributed network architecture and can be widely utilized for instant messaging functionality [14]. Overall, all application and transport layer protocols have their own advantages and drawbacks and are best suited to different IoT environments and circumstances. Therefore, always depending on the smart health application requirements, the most appropriate protocol must be selected.

Lastly, there are some additional protocols that mostly operate in the last layers of the architecture, namely, the physical, data link and network layers. The physical layer is characterized by the IoT devices that collect and transmit health data. Its security is ensured through cryptographic algorithms, which must be lightweight and require a low bandwidth and energy. The data link layer is composed of wireless communication protocols, such as Bluetooth and Wi-Fi, which provide proper security via symmetric cryptographic mechanisms. Finally, an exemplary lightweight protocol that provides security to the network layer is 6LoWPAN. It has been designed for IoT communications and offers various address lengths and a low bandwidth. The more well-known IPv6 network protocol requires more energy resources, deeming it unsuitable for low-powered IoT devices.

3. Architecture of Lightweight Block Ciphers

Block ciphers encrypt the data block by block using a secret key and various mathematical rounds, depending on the lightweight algorithm. The block ciphers, whose security optimizations and hardware designs are discussed, are Advanced Encryption Standard (AES), CLEFIA, KASUMI, Lightweight Encryption Algorithm (LEA), LED, Piccolo, PRESENT, RECTANGLE, SKINNY and SIMON. Finally, the designs' characteristics are collected in Table 1.

Each lightweight cryptographic algorithm has specific characteristics that can provide security for IoT-based applications. In this section, creative optimizations that enhance and verify the security were highlighted. It must be mentioned that a characteristic that reflects the magnitude of the security is the key size. Specifically, larger-sized keys are better than smaller ones.

The most utilized block cipher is the AES symmetric cryptographic algorithm. Three FPGA-based implementations employ the AES primitive with various hardware optimizations. Two of those examine the security of their design through image encryption. In addition to AES, CLEFIA algorithm's FPGA-based architecture increases its security by supporting encryption with three different key sizes. Another cryptographic primitive, which enhances the robustness of its security with a chaotic generator, is the KASUMI algorithm. The design efficiently passes various NIST tests. Additional primitives with interesting designs are the LEA and LED whose hardware implementations are straightforward. The LEA algorithm supports three different key sizes and is evaluated by image encryption, while one of the LED algorithm's designs is improved by the simultaneous utilization of two different key sizes. Lastly, PRESENT, SKINNY and SIMON implementations were analyzed. The first one presents methods for either resisting CPA attacks or detecting side-channel attacks by inserting a threshold implementation-based component. Moreover, SKINNY implements and validates a concurrent error-detection method and SIMON design enhances the security by supporting keys of different sizes, whose utilization depend on the application requirements.

Table 1. Hardware implementation results of block ciphers.

Block Ciphers	Device	Structure	Key Size (bit)	Clock Cycles [1]	Area	LUTs	Freq. MHz	Throughput Mbps	Power
nanoAES [15]	TSMC-65 NM	AES with 8 bit datapath	128	527	11.7(x103 µm3)	-	100	-	245.6 µW
AES-128 [16]	Virtex-5	Loop unrolled/ FSM-based	128	-	-	20402/ 14798	332.34/ 272.33	4342/3485	-
AES-256 [17]	Virtex7	Reusing S-box and mix-column blocks	256	74	-	1814	161	278	0.58 W
CLEFIA [18]	Artix-7	Iterative	128/192/ 256	19/23/ 27	506 slices	1725	147	990/818 /696	-
CLEFIA [19]	ARTIX-7	4 bit architecture	128	526	-	606	115	28	83 mW
KASUMI [20]	CMOS 0.18 µm	Low-area S9/S7 s-box	64	16/59 [2]	2487/2294 gates	-	214	32.4/4.6	-
KASUMI [21]	Virtex-5	Simplification/ chaotic generator	128/ 526	8	468/1112 slices	-	644.33/59.45	5154.64/475.60	-
LEA [22]	UMC 0.09-µm	Unified architecture	128/192/ 256	25/29/ 33	11080 GE	-	740	3788 @100KHz	2.65 mW [3]
LED [23]	CMOS 180nm	Flexible	64/128	33/49	3556 GE	-	100	680.3/ 1010.1	8.751 mW [3]
LED-64 [24]	Kintex7/ Artix7 /Spartan3	Round-based	64/64/64	32/32/ 32	122/91/114 slices	273/191/ 274	485/439/ 167	971.58/879.44/ 334.68	-
Piccolo-80 [25]	Virtex5/Spartan3	Compact	80	26	194/282 slices	372/ 535	280.9/ 132.25	691.54/ 325.54	0.699/0.183 W
Piccolo [26]	Spartan-3	Iterative/serial 4 bits	128/128	31/496	397/265 slices	757/442	81.82/45.85	168.9/5.92	-
PRESENT-16/64 [27]	Kintex-7	Optimized threshold design	80	129/-	197/447 slices	570/ 860	342.83/ 445.33	170.09/ 919.39	-
PRESENT-80/128 [28]	Kintex-7	Using 256–150 slice MUXs	80/128	-	68/101– 75/123 slices	246/205– 271/210	639/741– 624/740	1319.22/1529.80 – 1288.26/ 1527.74	40.93/22.88 — 42.69/24.87 pJ/bit
Rectangle [29]	Virtex-5	Optimized	80	100	81 slices	281	390.78	250.098	721.04 mW
Rectangle [29]	CMOS 180nm	Optimized	80	100	2375.64 GE	-	200	250	5.0876 mW
SKINNY [30]	Virtex-7	Pipelined with fault detection	64/128/ 192	-	-	2965/ 3802 /5176	768/ 691 /597	49150/ 44220 /38210	
SKINNY [30]	Virtex-7	Pipelined with fault detection	128/256/ 384	-	-	10407/ 14072 /16926	560/ 547 /545	71680/ 70020 /69760	-
SIMON [31]	Silicon 40 nm	Round-parallel	64/256	-	0.70E6 F/-	-	530	5302/ 132.53	0.98 mW

[1] For encryption; [2] For substituting input I; [3] @ 100MHz.

Hardware Implementation Analysis of Block Ciphers

All the presented hardware designs of each block cipher displayed better hardware attributes than the conventional implementations of those cryptographic algorithms. Overall, based on the designs' results in both ASIC and FPGA, all these block ciphers can be efficiently utilized for IoT-based applications. However, some algorithms displayed better results than others. For both FPGA and ASIC structures, the algorithm with the least clock cycles and the largest key size was the KASUMI cipher. Furthermore, the algorithm that occupied the smallest area was the RECTANGLE cipher. The highest throughput was achieved by SKINNY-128 and LEA on FPGA and ASIC, respectively, and the lowest power consumption was acquired by AES on ASIC and CLEFIA on FPGA. However, not all designs were implemented in the same FPGA; thus, a comparison of them based on the FPGA board was presented. The RECTANGLE algorithm was the most area-efficient cipher on both the Virtex-5 FPGA board and ASIC CMOS-180nm. Moreover, KASUMI had the highest throughput on Virtex-5. AES-256 and LED-64 utilized the least number of slices on Virtex-7 and Artix-7, respectively. However, CLEFIA had a slightly higher throughput than LED-64 on Artix-7. The LED-64 cipher also had the smallest area and the highest throughput on Spartan-3. Finally, the PRESENT-80 had the least number of total slices and the highest throughput on Kintex-7.

Nonetheless, the efficiency of a cryptographic algorithm's implementation is defined by the performance trade-offs and not only by a single characteristic. Overall, the KASUMI cipher is the best candidate for IoT-based implementations because it displays a high throughput for a high frequency while the area and clock cycles remain low. Even when the security is enhanced and the frequency decreases, it preserves a decent throughput and a logical area utilization. In addition to KASUMI, other block ciphers are also efficient for different scenarios. CLEFIA has better throughput at a low frequency than the rest of the FPGA implementations, thus can be easily applied to IoT devices. Furthermore, the AES and SKINNY ciphers can be effectively employed in high-throughput applications that do not prioritize the area. Nevertheless, IoT is mostly characterized by resource-constrained components with energy limitations. Therefore, such approaches are not advisable. For small-area-demanding applications, the LED and PRESENT ciphers are better suited, while also retaining a high throughput. The rest of the ciphers occupy a good number of slices, but have lower throughputs, apart from LEA that displays a satisfactory throughput for a slightly larger area.

4. Architecture of Lightweight Stream Ciphers

Stream ciphers encrypt data by generating a pseudo-random key bit stream and combining it with the plaintext digit by digit. These primitives are also more appropriate for telecommunication standards than block ciphers because of their resistance to error propagation and efficient hardware implementations [23]. In this section, the stream ciphers whose security and designs were analyzed are A5/3, ChaCha8, Grain-v1, LIZARD, Mutual Irregular Clocking KEY (Mickey) 2.0, Rabbit, SNOW-3G, ZUC, Trivium and Welch-Gong (WG). Lastly, all the implementations are collectively displayed in Table 2.

Table 2. Hardware implementation results of stream ciphers.

Stream Ciphers	Device	Structure	Key Size (bit)	Slices	LUTs	Freq. MHz	Throughput Mbps	Power
A5/3 [32]	Virtex-5	One optimized KASUMI cipher block	128	987	1877	250	2000	1.46 W
ChaCha8 [33]	Virtex 7	Pipeline with DSPs and depth = 1 or 2	-	2867/2819	4556/5633	281.2/356.3	134,090/169,870	-
ChaCha8 [33]	Virtex 7	Pipeline with no DSPs and depth = 1 or 2	-	2982/4075	9138/10,101	368.7/356.3	175,820/169,870	-
Grain v1 [34]	Spartan-7	Serial version1/version2	80	26/35	66/76	250/313	250/313	-
Grain v1 [34]	Spartan-7	Basic/parallel	80	62/111	198/361	333/250	333/4000	-
LIZARD [34]	Spartan-7	Serial v1/v2	100	60/71	106/109	100/208	100/208	-
LIZARD [34]	Spartan-7	Basic/parallel	100	108/150	304/466	277/200	277/1200	-
Mickey 2.0 [34]	Spartan-7	Basic v1/v2	80	78/107	258/370	250/384	250/384	-
Mickey 2.0 [34]	Spartan-7	Serial v1/v2	80	51/70	171/205	250/384	250/384	-
SNOW 3G [35]	Virtex-5	HC-PRNG [1]	128	-	7881	28.84	922.88	1.36 W
SNOW-ZUC [36]	Virtex-5	With chaotic generator	-	-	10,602	21.201	678.432	1.467 W
ZUC-256 [37]	Altera DE2-115	Pipelined	256	-	-	115	3680	
ZUC-256 [38]	Spartan-6	CO-LFSR/SRO algorithms [2]	256	718	2494	209.346	6540	-
Trivium [34]	Spartan-7	Serial v1/v2	80	15/22	42/49	256/385	256/385	-
Trivium [34]	Spartan-7	Basic/parallel	80	71/133	200/446	416/344	416/22,016	-
WG(16,32) [39]	Spartan-6	Algebraic optimizations	-	631	1906	256	-	-

[1] Hyper-chaotic pseudo-random-number generator; [2] LFSR feedback-calculation optimization (CO-LFSR)/S-box replacement optimization (SRO).

The stream ciphers are efficiently implemented on FPGA for the purpose of providing security to IoT-based applications. Few of the designs, specifically SNOW-3G, ZUC-256 and SNOW-ZUC, add extra security elements to their structures and further validate their security performance by image encryption or NIST tests. The implementation of the SNOW-3G cipher achieves an efficient security level against cryptanalysis attacks with the use of a Hyper-Chaotic Pseudo-Random-Number Generator (HC-PRNG). Thus, the architecture enhances its robustness and randomness and also passes all NIST statistical tests. Furthermore, one implementation of the ZUC-256 cipher examines the correctness of the key stream generation and validates its security. Lastly, SNOW-ZUC, which originates from a combination of SNOW-3G and ZUC stream ciphers and is suited for embedded applications, improves the randomness of the design through a chaotic generator while NIST tests verify the security.

The rest of the cryptographic algorithms utilize basic-structure optimizations, such as pipeline, serial and parallel methodologies. A unique technique is used for the implementation of the A5/3 cipher, whose architecture is based upon an optimized KASUMI cipher block. Nevertheless, all designs are verified in FPGA and offer a basic level of security to embedded applications.

Hardware Implementation Analysis of Stream Ciphers

The resulting designs of all the presented stream ciphers, similar to the block ciphers, display better characteristics than the conventional designs of the same ciphers. First, the implementations with the smallest utilized area are the serial versions of Trivium. Other relatively small-area designs are achieved by the serial versions of Grain v1. However, the Trivium cipher remains superior because it has a higher throughput than Grain v1. Second, the highest throughput is achieved by the ChaCha8 cipher, which also occupies the largest area. Thus, it is only suitable for applications that can employ larger IoT devices whose area consumption is not a priority. SNOW-3G architecture achieves a relatively high throughput for a lower frequency than the rest, classifying it as suitable for IoT devices that only operate at low frequencies. The rest of the high-throughput designs, except the parallel version of Trivium, occupy a large area without achieving a good trade-off. The parallel version of Trivium achieves the best ratio between throughput and area. It has the second-highest throughput value, medium area, an effective 80 bit-size key and an adequate frequency. The parallel version of Grain v1 and the basic version of Trivium also have efficient trade-offs between area and throughput. Nonetheless, the parallel version of Trivium is recommended if the area requirements are slightly flexible.

5. Designs of Hash Functions, MACs and Authenticated Schemes

Hash functions, message authentication codes and authenticated encryption schemes are cryptographic methodologies that can provide security to IoT-based applications. The first primitives, hash functions, offer the ability to compress to a specific length the transmitted data. Lesamnta-LW, LHash-96, SPONGENT-88, PHOTON-80/20/16 and sLiSCP are the hash functions whose hardware implementations are examined in the following section with all of their characteristics accumulated in Table 3. The second discussed primitives are MACs that are mainly utilized for the prevention of identity theft or message forgery between two devices. Two MACs designs, Chaskey and LightMAC, were analyzed and presented in Table 4. Finally, the designs of authenticated encryption schemes, ACORN, AEGIS, Ascorn, NORX and KETJE, which enhance the confidentiality, integrity and authenticity of the system, are displayed in Table 5.

The structures of these algorithms follow the same primary optimization techniques as block and stream ciphers, and additional security components, such as chaotic generators, were not utilized. Therefore, further analysis will not be needed. Nevertheless, all the designs are deemed suitable for providing basic security in constrained lightweight applications due to the limited resource utilization they achieve. It must also be mentioned that, usually, these security schemes cannot provide complete security to the systems alone, but they are employed together with other security primitives to properly prevent more attacks and susceptibilities.

Table 3. Hardware implementation results of hash functions.

Hash Functions	Device	Structure	Clock Cycles [1]	Area	LUTs	Freq. MHz	Throughput Mbps	Power
Lesamnta-LW [40]	Artix-7	Serial and shared operations	768	-	434	161	50	99 mW
LHash-96 [41]	Spartan-3	Less CPR [2] and higher I/O rates	414	203 slices	380	97	60.12	28.89 mW [3]
Spongent-88 [41]	Spartan-3	Loop with single register	1980	74 slices	104	227	29.32	28.06 mW [3]
PHOTON-80/20/16 [42]	Spartan-3/Artix-7	Round-based	60/60	265/145 slices	510/363	157.24/376.43	262.07/627.38	27/82 mW
PHOTON-80 [43]	Spartan-3	Optimized mix-column	-	165 slices	-	93.13	9313	-
sLiSCP-192/256 [44]	CMOS 65 nm	Parallel	108/144	2271/3019 GE	-	100 (kHz)	29.62/44.44 (kbps)	4.62/5.88 µW

[1] For encryption; [2] cycles per round; [3] @ 100 KHz.

Table 4. Hardware implementation results of message authentication codes—MACs.

MACs	Device	Message Size	Key Size	Execution Time	Memory KB	Throughput	Power
Chaskey-8/12 rounds [45]	Arduino M0 Pro	344 bit	128 bit	33/42 µs	16.3/ 16.6	-	-
Chaskey [46]	NUCLEO-F401RE	512 bytes	128 bit	99 ms	22	1.308 [1] (Kbits/sec)	3713.32 [1] (µJoules/bit)
LightMAC [46]	NUCLEO-F401RE	512 bytes	128 bit	0.946 ms	34.5	1414.178 [1] (Kbits/sec)	3.434 [1] (µJoules/bit)

[1] S-parameter = 8.

Table 5. Hardware implementation results of authenticated encryption schemes.

Authenticated Encryption Schemes	Device	Structure	Area	LUTs	Freq. MHz	Throughput Mbps	Power
ACORN-1/-32 [47]	Spartan-6	Threshold implementation	-	784/4072	156.6/111.5	78.3/1784	8.6/27.4
AEGIS-128L [48]	Virtex-7	Loop Rolling/pipeline	7726/10610 slices	-	-	64497/88564	-
Ascorn-128 [49]	Spartan-6	Round-based [1] /serialized [2]	-	2.72k /1.41k	147.228 /217.042	392.61/12.05	20/19
Ascorn-128a [49]	Spartan-6	Round-based [1] /serialized [2]	-	2.93k /1.92k	146.163 /218.052	719.53/21.70	22/21
NORX [50]	Virtex-7	Low-area optimization	326 slices	-	250	3 (Gb/Sec)	53 [3]
NORX [51]	TSMC 65nm	Various optimizations	70.13 KGE	-	757.57	83110	-
KETJE [52]	NanGate 45nm	JR/SR/MINOR	18335/35136/73516 GE	-	892.85/892.85/909.1	-	2.08/3.63/7.75

[1] Two permutations per clock cycle; [2] m = 1; [3] dynamic power.

5.1. Hardware Implementation Analysis of Hash Functions

The presented hash functions are efficiently implemented in both FPGA and ASIC. Specifically, the architecture of Spongent-88 has the smallest area and is more suited for constrained-area applications. However, it is the slowest design and has the lowest throughput, even at a high frequency. The highest throughput is achieved by the low-frequency PHOTON-80 implementation, without occupying many slices. Furthermore, the other design of PHOTON-80/20/16 also has good throughput with a smaller number of slices, average frequency, efficient power consumption and fewer clock cycles. Overall, both PHOTO-80 schemes have greater trade-offs than the rest of the FPGA implementations, and even though they do not have the smallest area of all hash functions, they are compact enough to be employed in area-constrained IoT-based applications.

5.2. Hardware Implementation Analysis of MACs and Authenticated Encryption Schemes

Recently implemented MACs are Chaskey and LightMAC. The devices that implement these MACs are the Arduino and the NUCLEO-F401RE board, which vary from the platforms employed for the previous algorithms. Thus, the design traits are slightly different. Between these two algorithms, LightMAC excels at power consumption and throughput, while also achieving a low execution time for an average-sized message. Nevertheless, the Chaskey algorithm occupies less memory space than LightMAC and has a low execution time in Arduino. Overall, LightMAC has a more balanced implementation with efficient features.

In contrast to MACs, authenticated encryption schemes are implemented in ASIC and FPGA boards. The highest throughput, but also the largest area, is achieved by the AEGIS-128L design. Therefore, it can only be utilized in high-throughput applications where area is not a priority. The second-highest throughput and the smallest area in FPGA is achieved by the NORX design. Furthermore, the ASIC implementation of NORX has a high throughput, but a relatively large area. Some of the ASIC designs of KETJE have smaller area than NORX algorithms; nevertheless, the latter display better trade-offs and have more efficient designs in both FPGA and ASIC. The only issue is the power consumption. ACORN and KETJE have the lowest power-consumption values in FPGA and ASIC, respectively; however, the NORX scheme surpasses them in other sectors. Therefore, each scheme can be employed in an IoT healthcare system depending on the available resources and the priorities the application demands.

6. Comparison Results

In this section, all the lightweight cryptographic primitives are compared. Out of all the categories, stream ciphers and mainly block ciphers provided the most efficient security, which in some cases was even enhanced. As for the hardware limitations, the ASIC implementations that displayed better trade-offs were LED and LEA designs. The former is qualified for small-area applications with good throughput and power consumption, and the latter is appropriate for high-throughput applications with an average area and efficient power consumption. The Piccolo-80 and KASUMI implementations on Virtex-5 are the most suitable for area-efficient and high-throughput applications, respectively, without lacking in other sectors. For Virtex-7, the NORX design is the most beneficial. The other designs in this board demonstrate high-throughput results, but also large areas, which are unacceptable in the IoT resource-constrained environment. Furthermore, the LED-64 design had the best trade-offs on the Artix-7 board, while the PRESENT cipher was deemed the most balanced architecture on the Kintex-7 board. PHOTON-80 function on Spartan-3, ZUC on Spartan-6 and Trivium on Spartan-7 exhibited good throughput and overall better characteristics than the rest of the ciphers on the same FPGA boards. Lastly, between the two MACs, LightMAC had better trade-offs than Chaskey.

After extensive research, the most efficient algorithms out of all those presented in this paper were the KASUMI block cipher, PHOTON-80 hash function, PRESENT block cipher and Trivium stream cipher. The most rapid implementation with the most enhanced security was the KASUMI block cipher. It has a larger area than most but compensates in throughput capability. Moreover, the design with the chaotic generator operated efficiently at a low frequency, which is common in IoT. Nevertheless, the other designs had better performance trade-offs. The PHOTON-80 hash function displayed even higher throughput and smaller area for an equally low frequency, deeming it suitable for low-area and high-throughput-demanding applications. An even smaller area was obtained by the structure of the Trivium stream cipher. This architecture has the highest throughput of all with an average frequency. Overall, Trivium had the best ratio between area and throughput. Finally, the smallest area of these four algorithms with a relatively high throughput was achieved by the PRESENT block cipher. The throughput may be the lowest of these four but compared to algorithms with the same number of total slices it was the highest.

7. Conclusions and Outlook

IoT-based healthcare applications have a variety of vulnerabilities and susceptibilities that can potentially endanger the user's wellbeing due to the considerable impact these health services have on the patient's everyday life and the large amount of private health data collected. Many security mechanisms that depend on lightweight cryptographic primitives have been introduced to the literature. These implemented security schemes aim at ensuring the privacy, confidentiality and integrity of the system, while also utilizing few hardware and energy resources and achieving high-throughput and performance efficiencies. Nevertheless, depending on the application's performance, employed resources and security requirements, specific cryptographic algorithms display better trade-offs between hardware efficiency, performance throughput and security. Therefore, in this paper, after extensive research and the thorough analysis of all recent implemented lightweight cryptographic primitives, four algorithms were deemed better suited for IoT devices in healthcare applications. These algorithms were the KASUMI block cipher, PRESENT block cipher, Trivium stream cipher and PHOTON-80 hash function.

Overall, this paper presented an analysis of the IoT-based healthcare design and the research conducted to date on hardware implementations of cryptographic algorithms. Specifically, the capabilities, to date, and therefore limitations of IoT-based health applications based on the hardware designs of lightweight cryptographic primitives were demonstrated. For future directions, these conclusions will be considered with the aspirations of improving and simplifying IoT-based implementations for security purposes in the healthcare domain.

Author Contributions: K.T. and N.S. contributed to the research, investigation, results analysis, resources and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All of the reported data are included in the manuscript.

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

References

1. Fei, H. *Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implemenations*; CRC Press: Boca Raton, FL, USA, 2016; ISBN 9781498723183.
2. Tsantikidou, K.; Sklavos, N. Vulnerabilities of Internet of Things, for Healthcare Devices and Applications. In Proceedings of the IEEE 8th NAFOSTED Conference on Information and Computer Science (NICS'21), Hanoi City, Vietnam, 21–22 December 2021.
3. Khan, M.N.; Rao, A.; Camtepe, S. Lightweight Cryptographic Protocols for IoT-Constrained Devices: A Survey. *IEEE Internet Things J.* **2021**, *8*, 4132–4156.
4. Latif, M.A.; Ahmad, M.B.; Khan, M.K. A Review on Key Management and Lightweight Cryptography for IoT. In Proceedings of the 2020 Global Conference on Wireless and Optical Technologies (GCWOT), Malaga, Spain, 6–8 October 2020; pp. 1–7.
5. Harbi, Y.; Aliouat, Z.; Refoufi, A.; Harous, S. Recent Security Trends in Internet of Things: A Comprehensive Survey. *IEEE Access* **2021**, *9*, 113292–113314.
6. Dutta, I.K.; Ghosh, B.; Bayoumi, M. Lightweight Cryptography for Internet of Insecure Things: A Survey. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 0475–0481.
7. Shah, A.; Engineer, M. A Survey of Lightweight Cryptographic Algorithms for IoT-Based Applications. In *Smart Innovations in Communication and Computational Sciences. Advances in Intelligent Systems and Computing*; Tiwari, S., Trivedi, M., Mishra, K., Misra, A., Kumar, K., Eds.; Springer: Singapore, 2019; Volume 851.
8. Thakor, V.A.; Razzaque, M.A.; Khandaker, M.R.A. Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities. *IEEE Access* **2021**, *9*, 28177–28193.
9. Dhanda, S.S.; Singh, B.; Jindal, P. Lightweight Cryptography: A Solution to Secure IoT. *Wirel. Pers. Commun.* **2020**, *112*, 1947–1980.
10. Rana, M.; Mamun, Q.; Islam, R. Lightweight cryptography in IoT networks: A survey. *Future Gener. Comput. Syst.* **2022**, *129*, 77–89.
11. Al-Jaroodi, J.; Mohamed, N.; Abukhousa, E. Health 4.0: On the Way to Realizing the Healthcare of the Future. *IEEE Access* **2020**, *8*, 211189–211210.
12. Khanam, S.; Ahmedy, I.B.; Idna Idris, M.Y.; Jaward, M.H.; Bin Md Sabri, A.Q. A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things. *IEEE Access* **2020**, *8*, 219709–219743.
13. Cynthia, J.; Parveen Sultana, H.; Saroja, M.N.; Senthil, J. Security Protocols for IoT. In *Ubiquitous Computing and Computing Security of IoT. Studies in Big Data*; Jeyanthi, N., Abraham, A., Mcheick, H., Eds.; Springer: Cham, Switzerland, 2019; Volume 47.

14. Jienan, D.; Xiangning, C.; Shuai, C. Overview of Application Layer Protocol of Internet of Things. In Proceedings of the 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), Chengdu, China, 23–26 April 2021; pp. 922–926.
15. Shahbazi, K.; Ko, S.-B. Area-Efficient Nano-AES Implementation for Internet-of-Things Devices. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2021**, *29*, 136–148.
16. Lata, K.; Saini, S. Hardware Software Co-Simulation of an AES-128 based Data Encryption in Image Processing Systems for the Internet of Things Environment. In Proceedings of the 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Chennai, India, 14–16 December 2020; pp. 260–264.
17. Gunasekaran, M.; Rahul, K.; Yachareni, S. Virtex 7 FPGA Implementation of 256 Bit Key AES Algorithm with Key Schedule and Sub Bytes Block Optimization. In Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 21–24 April 2021; pp. 1–6.
18. Cheng, X.; Zhu, H.; Xu, Y.; Zhang, Y.; Xiao, H.; Zhang, Z. A reconfigurable and compact hardware architecture of CLEFIA block cipher with multi-configuration. *Microelectron. J.* **2021**, *114*, 105144.
19. Pyrgas, L.; Kitsos, P. A Very Compact Architecture of CLEFIA Block Cipher for Secure IoT Systems. In Proceedings of the 2019 22nd Euromicro Conference on Digital System Design (DSD), Kallithea, Greece, 28–30 August 2019; pp. 624–627.
20. Yasir; Wu, N.; Yahya, M.R.; Bi, Q. Area-Efficient Architectures of KASUMI Block Cipher. In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; pp. 1–6.
21. Madani, M.; Tanougast, C. FPGA implementation of an enhanced chaotic-KASUMI block cipher. *Microprocess. Microsyst.* **2021**, *80*, 103644.
22. Mishra, Z.; Nath, P.K.; Acharya, B. High throughput unified architecture of LEA algorithm for image encryption. *Microprocess. Microsyst.* **2020**, *78*, 103214.
23. Rashidi, B. Flexible structures of lightweight block ciphers PRESENT, SIMON and LED. *IET Circuits Devices Syst.* **2020**, *14*, 369–380.
24. Al-Shatari, M.; Hussin, F.A.; Aziz, A.A.; Witjaksono, G.; Rohmad, M.S.; Tran, X.-T. An Efficient Implementation of LED Block Cipher on FPGA. In Proceedings of the 2019 First International Conference of Intelligent Computing and Engineering (ICOICE), Hadhramout, Yemen, 15–16 December 2019; pp. 1–5.
25. Ramu, G.; Mishra, Z.; Acharya, B. Hardware implementation of Piccolo Encryption Algorithm for constrained RFID application. In Proceedings of the 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON), Jaipur, India, 13–15 March 2019; pp. 85–89.
26. Mhaouch, A.; Elhamzi, W.; Atri, M. Lightweight Hardware Architectures for the Piccolo Block Cipher in FPGA. In Proceedings of the 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, 2–5 September 2020; pp. 1–4.
27. Yu, X.; Wu, N.; Zhou, F.; Zhang, J.; Zhang, X. A Compact Hardware Implementation for the SCA-resistant PRESENT Cipher. In Proceedings of the IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 October 2019; pp. 5463–5468.
28. Dalmasso, L.; Bruguier, F.; Benoit, P.; Torres, L. Evaluation of SPN-Based Lightweight Crypto-Ciphers. *IEEE Access* **2019**, *7*, 10559–10567.
29. Pandey, J.G.; Laddha, A.; Samaddar, S.D. A Lightweight VLSI Architecture for RECTANGLE Cipher and its Implementation on an FPGA. In Proceedings of the 2020 24th International Symposium on VLSI Design and Test (VDAT), Bhubaneswar, India, 23–25 July 2020; pp. 1–6.
30. Nallathambi, B.; Palanivel, K. Fault diagnosis architecture for SKINNY family of block ciphers. *Microprocess. Microsyst.* **2020**, *77*, 103202.
31. Taneja, S.; Alioto, M. Deep Sub-pJ/Bit Low-Area Energy-Security Scalable SIMON Crypto-Core in 40 nm. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; pp. 1–5.
32. Madani, M.; Tanougast, C. FPGA implementation of an optimized A5/3 encryption algorithm. *Microprocess. Microsyst.* **2020**, *78*, 103212.
33. Pfau, J.; Reuter, M.; Harbaum, T.; Hofmann, K.; Becker, J. A Hardware Perspective on the ChaCha Ciphers: Scalable Chacha8/12/20 Implementations Ranging from 476 Slices to Bitrates of 175 Gbit/s. In Proceedings of the 2019 32nd IEEE International System-on-Chip Conference (SOCC), Singapore, 3–6 September 2019; pp. 294–299.
34. Li, B.; Liu, M.; Lin, D. FPGA implementations of Grain v1, Mickey 2.0, Trivium, Lizard and Plantlet. *Microprocess. Microsyst.* **2020**, *78*, 103210.
35. Madani, M.; Benkhaddra, I.; Tanougast, C.; Chitroub, S.; Sieler, L. FPGA implementation of an enhanced SNOW-3G stream cipher based on a hyperchaotic system. In Proceedings of the 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT), Barcelona, Spain, 5–7April 2017; pp. 1168–1173.
36. Madani, M.; Tanougast, C. Combined and Robust SNOW-ZUC Algorithm Based on Chaotic System. In Proceedings of the 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Glasgow, UK, 11–12 June 2018; pp. 1–7.

37. Wang, Y.; Wu, L.; Zhang, X.; Xu, K.; Yang, W. A Hardware Implementation of ZUC-256 Stream Cipher. In Proceedings of the 2020 IEEE 14th International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, China, 30 October–1 November 2020; pp. 94–97.
38. Yang, Y.; Zhao, W.; Xiong, L.; Wang, N.; Ma, Y. Optimized Implementations for ZUC-256 on FPGA. *Wirel. Pers. Commun.* **2021**, *116*, 2615–2632.
39. Zidaric, N.; Aagaard, M.; Gong, G. Hardware Optimizations and Analysis for the WG-16 Cipher with Tower Field Arithmetic. *IEEE Trans. Comput.* **2019**, *68*, 67–82.
40. Pyrgas, L.; Kitsos, P. An 8-bit Compact Architecture of Lesamnta-LW Hash Function for Constrained Devices. In Proceedings of the 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 27–29 November 2019; pp. 743–746.
41. Lara-Nino, C.A.; Morales-Sandoval, M.; Diaz-Perez, A. Small lightweight hash functions in FPGA. In Proceedings of the 2018 IEEE 9th Latin American Symposium on Circuits Systems (LASCAS), Puerto Vallarta, Mexico, 25–28 February 2018; pp. 1–4.
42. Al-Shatari, M.O.A.; Hussin, F.A.; Aziz, A.A.; Witjaksono, G.; Tran, X.-T. FPGA-Based Lightweight Hardware Architecture of the PHOTON Hash Function for IoT Edge Devices. *IEEE Access* **2020**, *8*, 207610–207618.
43. Abbas, Y.A.; Jidin, R.; Jamil, N.; Z'aba, M.R.; Al-Azawi, S. Small Footprint Mix-Column Serial for PHOTON and LED Lightweight Cryptography. In Proceedings of the 2018 International Conference on Advanced Science and Engineering (ICOASE), Duhok, Iraq, 9–11 October 2018; pp. 70–74.
44. AlTawy, R.; Rohit, R.; He, M.; Mandal, K.; Yang, G.; Gong, G. Towards a Cryptographic Minimal Design: The sLiSCP Family of Permutations. *IEEE Trans. Comput.* **2018**, *67*, 1341–1358.
45. Carel, G.; Isshiki, R.; Kusaka, T.; Nogami, Y.; Araki, S. Design of a Message Authentication Protocol for CAN FD Based on Chaskey Lightweight MAC. In Proceedings of the 2018 Sixth International Symposium on Computing and Networking Workshops (CANDARW), Takayama, Japan, 27–30 November 2018; pp. 267–271.
46. Saldamli, G.; Ertaul, L.; Shankaralingappa, A. Analysis of Lightweight Message Authentication Codes for IoT Environments. In Proceedings of the 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), Rome, Italy, 10–13 June 2019; pp. 235–240.
47. Diehl, W.; Farahmand, F.; Abdulgadir, A.; Kaps, J.-P.; Gaj, K. Face-off between the CAESAR Lightweight Finalists: ACORN vs. Ascon. In Proceedings of the 2018 International Conference on Field-Programmable Technology (FPT), Naha, Japan, 10–14 December 2018; pp. 330–333.
48. Katsaiti, M.; Sklavos, N. Implementation Efficiency and Alternations, on CAESAR Finalists: AEGIS Approach. In Proceedings of the 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom / DataCom / CyberSciTech), Athens, Greece, 12–15 August 2018; pp. 661–665.
49. Khan, S.; Lee, W.-K.; Hwang, S.O. Scalable and Efficient Hardware Architectures for Authenticated Encryption in IoT Applications. *IEEE Internet Things J.* **2021**, *8*, 11260–11275. [CrossRef]
50. Abbas, A.; Mostafa, H.; Mohieldin, A.N. Low Area and Low Power Implementation for CAESAR Authenticated Ciphers. In Proceedings of the 2018 New Generation of CAS (NGCAS), Valletta, Malta, 20–23 November 2018; pp. 49–52.
51. Kumar, S.; Haj-Yahya, J.; Chattopadhyay, A. Efficient Hardware Accelerator for NORX Authenticated Encryption. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
52. Arribas, V.; Nikova, S.; Rijmen, V. Guards in Action: First-Order SCA Secure Implementations of Ketje Without Additional Randomness. In Proceedings of the 2018 21st Euromicro Conference on Digital System Design (DSD), Prague, Czech Republic, 29–31 August 2018; pp. 492–499.