Hindawi
Security and Communication Networks
Volume 2020, Article ID 8837671, 15 pages
https://doi.org/10.1155/2020/8837671

Research Article

Energy Consumption Analysis of Lightweight Cryptographic Algorithms That Can Be Used in the Security of Internet of Things Applications

Bora Aslan,[1] **Füsun Yavuzer Aslan**,[2] **and M. Tolga Sakallı**[3]

[1]*Department of Software Engineering, Kırklareli University, Kırklareli, Turkey*
[2]*Department of Computer Programming, Kırklareli University, Kırklareli, Turkey*
[3]*Department of Computer Engineering, Trakya University, Kırklareli, Turkey*

Correspondence should be addressed to Bora Aslan; bora.aslan@klu.edu.tr

Received 25 June 2020; Revised 24 October 2020; Accepted 30 October 2020; Published 21 November 2020

Academic Editor: Savio Sciancalepore

The Internet of Things (IoT) has begun to acquire place in our lives quietly and gradually thanks to the presence of wireless communication systems. An increasing number of M2M applications, such as smart meters, healthcare monitoring, transportation and packaging, or asset tracking, make a significant contribution to the growth of devices and connections. Within such a large and uncontrollable ecosystem, IoT poses several new problems. Security and privacy are among the most important of these problems. Lightweight cryptography can be used more effectively for small size, low energy, and small footprint such as RFID tags, sensors, and contactless smart cards. Therefore, it can be used to ensure security and privacy in the IoT applications. In this study, PRESENT, CLEFIA, PICCOLO, PRINCE, and LBLOCK lightweight cryptographic algorithms, which can be used to secure data in IoT applications, were analyzed in a test environment. As a result of the tests, the energy consumption of the algorithms, current measurement, active mode working time, and active mode energy consumption were identified and based on this, some inferences have been made.

1. Introduction

ARPANET, which was originally created in 1969 based on the idea that only a few systems are connected, has become an immense separate world where billions of computers and systems come together. Internet speed, capacity, and traffic have increased exponentially and extend into the future. The mobile devices that almost everyone has in their pockets today have superior capabilities than the super computers 20 years ago. Now, human beings can make almost all devices smart thanks to microsensors and smart chips. Smart phones, cars, and heating systems have become easily controllable and programmable.

In the last decade, the IoT has begun to take place in our lives quietly and gradually thanks to the presence of wireless communication systems. On a global scale, the number of objects that can be described as devices and connections is growing faster than the human population. Therefore, this situation accelerates the increase in the average number of devices and connections per household and per person. Every year, new devices in different forms with increasing talent and intelligence are introduced and adopted. An increasing number of M2M applications, such as smart meters, healthcare monitoring, transportation and packaging, or asset tracking, make a significant contribution to the growth of devices and connections.

In this context, various definitions used in the literature on the concept of IoT are given as follows:

(i) It is the network of systems created by connecting devices, vehicles, and living things to each other or other systems [1]

(ii) It is a system of devices that share information and network by connecting to each other thanks to various communication protocols [2]

(iii) It is the application area where different technologies integrated with each other are used in social life [3]

(iv) These are systems where devices connected to the Internet share data over the Internet to meet the needs of people without the need for human intervention [4]

(v) It is a system that enables critical and effective use of services such as critical infrastructure, education, health, security, and transportation related to a settlement by using information and communication technologies [5]

(vi) It is a community and marketplace made up of smart devices that communicate with each other using various communication protocols, produce information, and exchange information with their surroundings thanks to the network they create [6]

The concept of the IoT was first mentioned in a presentation prepared by Kevin Ashton in 1999 for the company Procter and Gamble. In fact, in his presentation, Ashton listed the benefits of the company with the use of RFID (Radio Frequency Identification) technology. However, Ashton may have led the concept of the IoT that attracts attention and many products in this direction by putting forward the idea of connecting all devices to each other. With the "ITU Internet report 2005: Internet of Things" report published by the International Telecommunication Union (ITU) in 2005, the IoT concept was officially announced [7]. Then, in 2009, a report titled "Internet of Things—Action Plan for Europe" was published by the European Commission [8]. Similarly, in the European Union report published in 2013, it was emphasized that the findings of the public survey conducted in 2012 show that the IoT technologies will facilitate the lives of individuals in areas such as health, social life, transportation, environment, and energy [9]. According to the report published by Cisco in 2011, in 2003, 500 million devices were connected to the Internet and the number of devices per person was 0.08. In 2010, the number of devices increased to 12.5 billion and the number of devices per person increased to 1.84. By 2020, it is estimated that the world population will be 7.6 billion and the number of devices connected to the Internet will be 50 billion [10]. According to Cisco Annual Internet Report (2018–2023), it is estimated that approximately two-thirds of the world's population will have Internet access by 2023. In addition, by 2023, it is predicted that the number of devices to be connected to the network by obtaining an IP address will be more than three times the global population. Thus, the number of devices connected to the network in 2018 is expected to increase from 18.4 billion to 29.3 billion in 2023. Within such a large and uncontrollable ecosystem, IoT poses several new problems. These problems that the new IT world has to deal with are listed and briefly explained as follows [11].

(i) Security issues: as people, businesses, and countries have increased loyalty to IoT, hackers and malicious people also have a desire to access and steal data. Therefore, security is the biggest problem that IoT has to overcome.

(ii) Privacy issues: most IoT apps collect and process information to make people's daily lives easier. Since most of this data can be described as personal data, privacy problems arise. Such questions require careful analysis and risk-reducing solutions, especially from a legal perspective.

(iii) Interoperability and standards-related problems: although IoT applications work with the Internet TCP/IP infrastructure and server/client architecture, many nonstandard protocols have been developed to allow objects with low processing capacities to communicate better and to operate data transfers effectively. This diversity raises interoperability problems.

(iv) Legal problems: legislation has recently been on the agenda for the solution of the problems experienced by the ownership of the data collected by IoT applications. The national level studies on this issue may be insufficient for global IoT practices.

(v) Economic development problems: IoT applications and developed technologies significantly change the economy. It is thought that dark factories and unmanned transportation vehicles can cause serious development problems. These technologies make it possible to decrease the workforce based on manpower, that is, to increase unemployment or open new business areas. It is argued that developed countries can overcome these problems partially, but developing or underdeveloped countries are expected to experience serious crises.

2. IoT and Security

Devices manufactured for IoT applications provide many advantages, as well as many disadvantages inherently. These disadvantages and the wide ecosystem are of particular interest to attackers. Below are some of the reasons why these devices were chosen as targets by attackers:

(i) Excessive number of devices: as mentioned at the beginning of the study, the number of devices used in the IoT area is quite above the number of laptops or desktop computers. The large surface area is interesting for attackers, because, from the point of view of the attacker, the more the devices there are, the more the entry points that can be captured.

(ii) Resource use is limited: devices used in the IoT are manufactured only to use resources to the extent that they perform their tasks. Therefore, it is not

possible to apply various security measures such as firewall, antivirus software on computers to such devices. This situation predicts that these devices can be easily targeted by attackers.

(iii) Producers' ability of preventing operability: companies producing devices within the scope of the IoT are primarily aimed at operating the system in a healthy way. These manufacturers often try to create new products as security problems arise. First products are offered to the market with weak security measures due to the target of being the first and most used product in the market.

(iv) Collection of personal information: IoT products, especially the ones for the end user, collect and save a great deal of personal information. For example, health practices, home automation systems can be counted in this category. Certainly, storing the private information of individuals in a system attracts the attention of the attackers and increases the attacks on this subject.

(v) Software updates that are not delivered on time: manufacturers only release updates when there is a problem or when there is a high level of innovation. However, delays in situations where these updates cannot be applied to all systems, or not applied at all, and instead encouraging the consumer to buy new products create particular security problems.

(vi) Manufacturer's back door release: manufacturer companies use various access methods, known as back doors, to interfere with the devices remotely. If this method is detected by attackers, all devices can remain vulnerable.

(vii) Default usernames and passwords: most devices are launched with the default username and password provided at the factory during initial setup to connect to the admin interface. In this way, the devices used without changing the default settings can be easily captured.

The devices used in the IoT are in cyber space, and the components that make up cyber space are not inherently safe. At the heart of the problem lies the lack of security in the TCP/IP communication protocol. The data in IP packets is readable with simple software that most people can use, which proves the insecurity of cyberspace.

In the informatics world, security is the provision of three main principles in general. These three principles are as follows [11]:

(i) Confidentiality: only authorized users can see the information.

(ii) Integrity: only authorized users can change the information.

(iii) Availability: information is always available when authorized users request it.

The most important problem for data transfer systems is privacy. In the IT world, confidentiality means protecting data from anyone except those who have an access right. The most important and functional technical data is encrypted by using cryptographic algorithms to ensure confidentiality. According to the Unit42 report, 98% of the IoT device traffic is transmitted on the network unencrypted. In addition, 57% of these devices are vulnerable to medium or high severity attacks [12]. This means that personal or corporate data transmitted by IoT applications are insecure. Therefore, data transmission must be encrypted.

Encryption algorithms are divided into two groups, symmetric and asymmetric algorithms. Symmetric algorithms use the same key (secret key) for encryption and decryption. Asymmetric encryption algorithms, on the other hand, use a public key for encryption, while using a secret key for decryption. Symmetric encryption algorithms work fast compared to asymmetric encryption algorithms. In these algorithms, plain text is encrypted using a secret key and transmitted to the other party. The encrypted text is decrypted again using the same secret key. Symmetric encryption algorithms can be examined in 2 categories: block ciphers and stream ciphers. Block ciphers cryptographic algorithms process open text in bit groups called fixed-length blocks. The encrypted text is revealed by encrypting the blocks with a key. In the deciphering process, the encrypted text is turned into plain text with the help of the same key. When the literature is analyzed, it is seen that block ciphers are used in IoT applications.

3. Lightweight Cryptographic Algorithms

Cryptographic algorithm solutions to be used in the IoT should be designed and implemented in accordance with the limited resources of the devices used in IoT applications. This necessity has created a new encryption area under the name of lightweight cryptography, which can be used more effectively for small size, low energy, and small footprint such as RFID tags, sensors, and contactless smart cards. In the world of data transmission, AES [13] is generally used as it is a secure standard. Although AES is a safe standard, it is not suitable for hardware restricted devices. It is known that a new solution other than AES is needed especially for applications that need to work with low power. Because in the 16 years since the adoption of AES, many technological innovations have emerged. On the other hand, [14] state that 2000 GE area is reasonable for RFID and similar devices and also integrated circuits for this should be produced. However, AES implementations have a 2400 GE area in the best conditions [15]. The purpose of lightweight cryptography is to provide algorithms that provide information security by using limited resources such as space, power consumption, and energy consumption. Many lightweight algorithms have been produced to achieve this goal. The majority of algorithms developed are block encryption algorithms. Throughout the study, PRESENT [16], CLEFIA [17], PICCOLO [18], PRINCE [19], and LBLOCK [20] lightweight

cryptographic algorithms were analyzed. The reason for choosing these algorithms can be explained as follows. PRESENT and CLEFIA are standardized as lightweight block cipher algorithm with the document ISO/IEC 29192-2: 2012 [21] which specifies the requirements for lightweight cryptography. On the other hand, PICCOLO, PRINCE, and LBLOCK are still in use for IoT applications. Also, all of these algorithms are suitable for hardware applications.

General information for analyzed algorithms is given in Table 1 and in this section some detailed information is summarized about the structures of algorithms.

3.1. PRESENT. PRESENT [16] is a block cipher developed by Orange Labs, Ruhr University Bochum, and Technical University of Denmark in 2007. With the document ISO/IEC 29192-2: 2012 [21], it is standardized as lightweight block cipher algorithm. It is the most known and used lightweight encryption algorithm. It supports 64- or 128-bit key options, having a 64-bit block length. The algorithm is designed with SPN architecture and consists of 31 rounds. The round structure of the PRESENT encryption algorithm is given in Figures 1 and 2. Each round consists of key addition, nonlinear S-Box layers, and linear bitwise permutation layers.

Round key $(K_i = k^i_{63}, \ldots, k^i_0)$ and round input bits $(b_{63}, \ldots, b_0)$ are entered into the XOR operation as follows, with $1 \le i \le 32$ in the key insertion phase. Here, 32. XOR process is used for final bleaching. In the nonlinear displacement process, the following S-box defined as 4-bit S: $\mathscr{F}_2^4 \longrightarrow \mathscr{F}_2^4$ is used.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Linear bitwise permutation process was performed according to the following table. Accordingly, the bit in position i is transferred to position $P(i)$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

As in [22–25], there are successful attacks to reduced-round of PRESENT. However, there is no successful attack published in the literature for full-round PRESENT.

3.2. CLEFIA. CLEFIA [17] is an algorithm developed by Sony Corporation that encrypts 128-bit data blocks with 128-, 192-, and 256-bit key options. It has been standardized as a lightweight block cipher algorithm with ISO/IEC 29192-2: 2012 [21] document like PRESENT. The algorithm using the Feistel architecture encrypts at 18 rounds for 128-bit key length, while it is encrypting at 22 and 26 rounds for 192- and 256-bit key lengths, respectively. Each round consists of 4 buses and two 32-bit F functions. CLEFIA round structure is given in Figure 3. In the encryption process, $P, C \in \{0,1\}^{128}$, with P plain text and C encrypted text.

In addition, 4 pieces in the form of $P = P_0|P_1|P_2|P_3$, with $P_i, C_i \in \{0,1\}^{32}$ $(0 \le i < 4)$, are processed in the data paths to obtain $C = C_0|C_1|C_2|C_3$ which is a C encrypted text. In the first and last round, $\mathrm{WK}_0, \mathrm{WK}_1, \mathrm{WK}_2, \mathrm{WK}_3 \in \{0,1\}^{32}$ is used for key whitening. Round keys from the key generation phase are specified as $\mathrm{RK}_i \in = \{0,1\}^{32}$ $(0 \le i < 2r)$, with r being the number of rounds. As a first step, P_1 and P_3 of the open text are taken to XOR with WK_0 and WK_1.

Then, P_0 block is taken to the function F_0 with the key RK_0 and $F_0(\mathrm{RK}_0, P_0)$ being performed. The result is taken to the XOR transaction with the result of the $P_1 \oplus \mathrm{WK}_0$ transaction. Likewise, the P_2 block is taken to the F_1 function with the RK_1 key and the result obtained by performing the $F_1(\mathrm{RK}_1, P_2)$ operation is taken to the XOR operation with the result of the $P_3 \oplus \mathrm{WK}_1$ operation. By changing the $P_0|P_1|P_2|P_3$ block formed at the end of the round to

Table 1: General information for analyzed algorithms.

Encryption algorithm	Year	Block size	Key size	Number of rounds	Architecture	Application area
PRESENT	2007	64	80, 128	31	SPN	Hardware
CLEFIA	2007	128	128, 192, 256	18, 22, 26	Feistel	Software, hardware
PICCOLO	2011	64	80, 128	25, 31	Feistel	Hardware
PRINCE	2012	64	128	12	SPN	Hardware
LBLOCK	2011	64	80	32	Feistel	Software, hardware

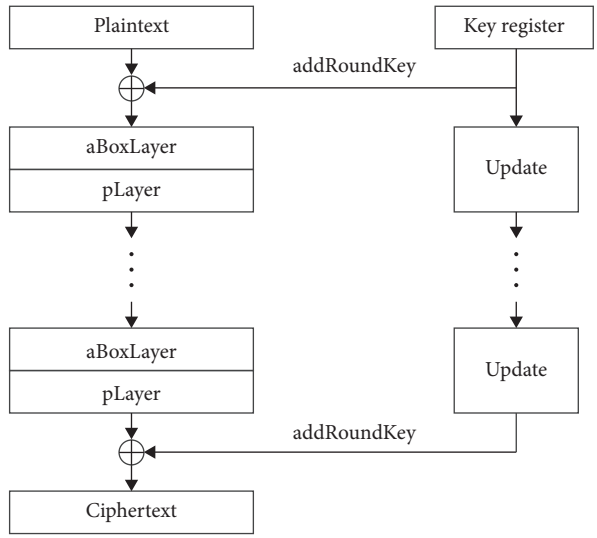


Figure 1: PRESENT encryption algorithm diagram.

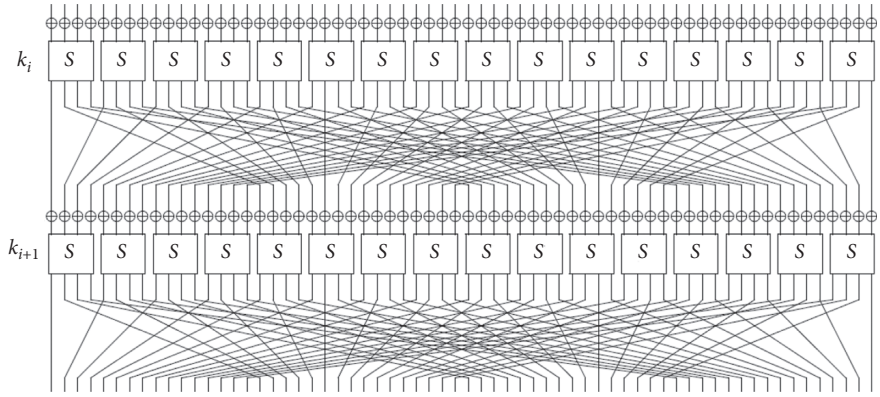


Figure 2: PRESENT encryption round structure.

$P_0 \longrightarrow P_3$, $P_1 \longrightarrow P_0$, $P_2 \longrightarrow P_1$, $P_3 \longrightarrow P_2$, the next round is passed. The P_1 and P_3 parts of the function outputs of the last round are taken to XOR with WK_2 and WK_3.

F_0 and F_1 functions enable F_0, F_1 : $(RK, x) \longrightarrow y$ in the encryption process. The diagrams of the functions are given in Figure 4. S_0 and S_1 which are specified in the functions are nonlinear 8-bit S-boxes. The order of use of S-boxes in F_0 and F_1 functions is different. The M_0 and M_1 matrices used in functions are 4×4 in Hadamard form and are defined by the object formed by the $x^8 + x^4 + x^3 + x^2 + 1$ irreducible polynomial in $GF(2^8)$ as a result of matrix multiplications.

The designers of CLEFIA consider that any attack does not threaten full-round CLEFIA. They analyzed against the differential cryptanalysis, linear cryptanalysis, impossible differential cryptanalysis, and square attack. In [26–31], there are some successful attacks to reduced-round of CLEFIA. However, there is no successful attack published in the literature for full-round CLEFIA.

3.3. PICCOLO. PICCOLO [18] is an algorithm that encrypts 64-bit data blocks, optimized for devices with extremely limited capacity by Sony, such as CLEFIA, with 80- and 128-bit key options. The designers of the algorithm are Kyoji Shibutani et al., Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. The algorithm using Feistel architecture encrypts at 25 rounds for 80-bit key length, while it is encrypting at 31 rounds for 128-bit key

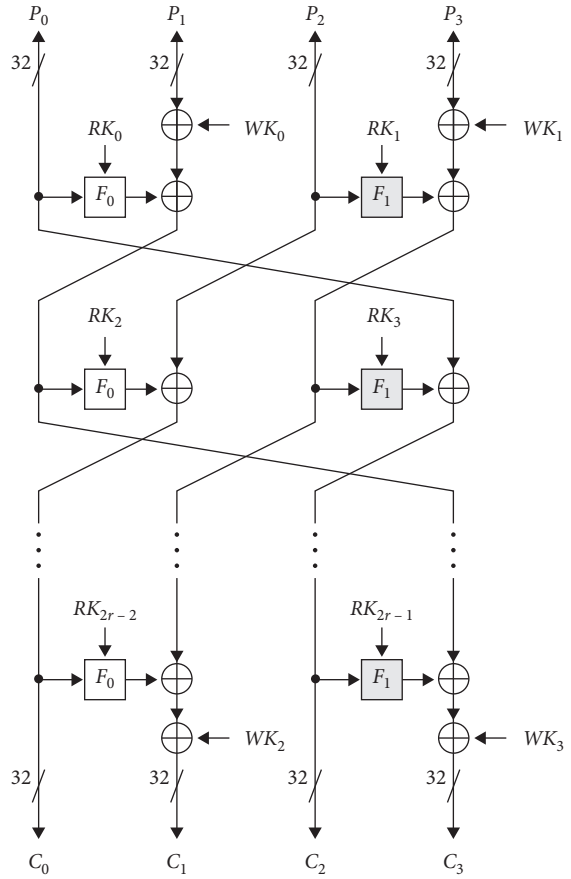


Figure 3: CLEFIA encryption round structure.

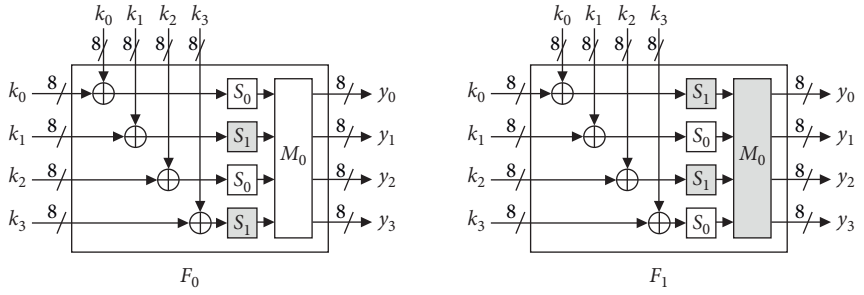


Figure 4: CLEFIA F_0 and F_1 functions.

lengths. In the encryption phase, 64-bit open text is divided into 4 16-bit pieces. After the 1st and 3rd parts whitening switch is put into XOR operation with wk_0 and wk_1, the results are processed with the F function. The function output is subjected to XOR operation with the round switches rk_0 and rk_1. Before the results are transferred to the next round, they are taken to the permutation process with the RP layer. The round structure of the PICCOLO encryption process is given in Figure 5.

The F function defined as F: $\{0,1\}^{16} \longrightarrow \{0,1\}^{16}$ used in the rounds in the PICCOLO algorithm is given in Figure 6.

Accordingly, the 16-bit input value passes primarily through the 4-bit S-boxes as follows:

$$(x_0, x_1, x_2, x_3) \longleftarrow (S(x_0), S(x_1), S(x_2), S(x_3)). \quad (1)$$

The S-box used at this stage is as follows.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	E	4	B	2	3	8	0	9	1	A	7	F	6	C	5	D

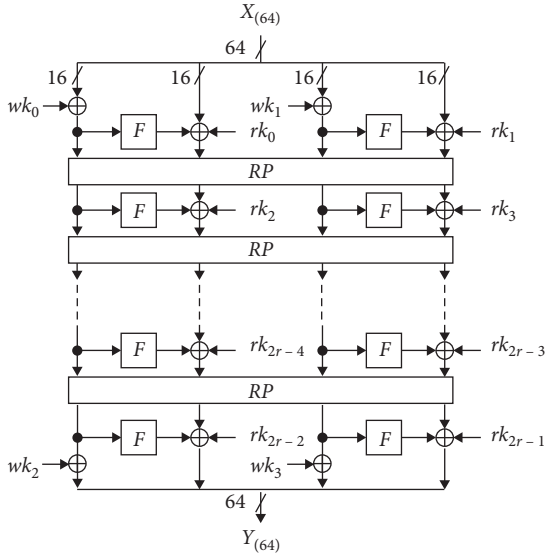


Figure 5: PICCOLO encryption round structure.

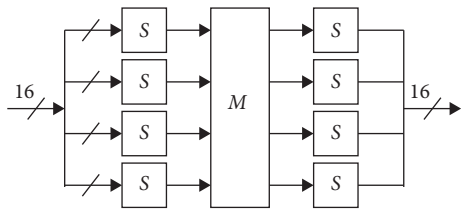


Figure 6: PICCOLO F function.

Then, it is put into diffusion by M linear transformation and finally passed through S-boxes. It is defined in an object created with $x^4 + x + 1$ irreducible polynomial in GF(2^4) as a result of the M matrix multiplication of values. The permutation structure used in the PICCOLO algorithm is defined as RP: $\{0,1\}^{64} \longrightarrow \{0,1\}^{64}$, and it divides the input value of 64-bit $X_{(64)}$ into 8-bit sections as $x_{0(8)}|x_{0(8)}|,\ldots,|x_{7(8)}$ and takes the permutation process as follows:

$$\begin{aligned}\text{RP: } &(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)\\ &\longleftarrow (x_2, x_7, x_4, x_1, x_6, x_3, x_0, x_5).\end{aligned} \tag{2}$$

Then, the bits are combined and transferred to the other round. In the last round, permutation is not performed.

As in [32–36], there are some successful attacks to reduced-round of PICCOLO. However, there is no successful attack published in the literature for full-round PICCOLO.

3.4. PRINCE. PRINCE [19] is an algorithm that encrypts 64-bit data blocks with a 128-bit key. Hardware optimized PRINCE was designed by Borghoff et al. presented in Asiacrypt 2012. The algorithm, which has a structure called FX, performs the encryption by using a different key. According to the designers' suggestion, the algorithm that encrypts in 12 steps divides the 128-bit k key into two 64-bit pieces as $k = k_0|k_1$. Then, $k_0|k_0'|k_1$ conversion is performed so that 192 bits are expanded to $k_0' = (k_0 \ggg 1) \oplus (k_0 \gg 63)$. Here, k_0 and k_0''are used as whitening switches, while k_1 is used as a round switch for 12 steps. Therefore, the algorithm does not have a detailed key generation phase and the same key is used in each round. The key whitening process is done by taking the k_0 value into the XOR operation with the open text m as in Figure 7 and the output value after 12 steps through the XOR operation with the k_0'.

The diagram of the PRINCE encryption algorithm is given in Figure 8. Each step contains XOR'ing with the round key k_1, mapping with S-box, linear conversion process, and XOR'ing with the round constant RC_i. After the first 5 rounds run, S-box, linear transformation, and reverse of S-box are performed as intermediate.

As in [37–42], there are some successful attacks to reduced-round of PRINCE. However, there is no successful attack published in the literature for full-round PRINCE.

3.5. LBLOCK. LBLOCK [20] is an algorithm that encrypts 64-bit data blocks proposed by Wu and Zhang with an 80-bit key. It is designed to be efficient and safe on equipment that works with limited resources. The algorithm that uses Feistel architecture completes the encryption process in 32 rounds. In encryption process, 64-bit M open text is divided into two 32-bit pieces as $M = X_1 \mid X_0$. Then, X_1 and K_1 round keys are taken to the F round function, and XOR operation is taken with the result of $X_0 \lll 8$, which cyclically shifted 8 bits to the left. The result is saved as the X_2 value of the next round. At the end of 32 rounds, M open text is encrypted as $C = X_{32} \mid X_{33}$. The LBLOCK round structure is given in Figure 9.

The F round function, which takes place in the LBLOCK encryption algorithm, includes the steps of passing the input values to the XOR through the S-boxes and then applying the diffusion process. The expression of the F round function is as follows. The diagram is given in Figure 10:

$$\begin{aligned}F\text{: } &\{0,1\}^{32} \times \{0,1\}^{32} \longrightarrow \{0,1\}^{32},\\ &(X, K_i) \longrightarrow U = P(S(X \oplus K_i)).\end{aligned} \tag{3}$$

8 different 4-bit S-boxes are used in the F round function. Accordingly, by XOR'ing with K_i round key, 32-bit X value is divided into 8 pieces of 4 bits in the form of $X = X_7|X_6|X_5|X_4|X_3|X_2|X_1|X_0$ and $s_7, s_6, s_5, s_4, s_3, s_2, s_1, s_0$ is passed through S-boxes and transferred to diffusion layer. The X value passing through the S-boxes in the F round function is taken to diffusion. The diffusion process for the LBLOCK algorithm is designed as a permutation of 8 values of 4 bits. Accordingly, the $X_7|X_6|X_5|X_4|X_3|X_2|X_1|X_0$ entries coming to the diffusion stage are listed as $X_6|X_4|X_7|X_5|X_2|X_0|X_3|X_1$.

As in [43–48], there are some successful attacks to reduced-round of LBLOCK. However, there is no successful attack published in the literature for full-round LBLOCK.

4. Test Environment and Results

In this study, MSP-EXP430FR5994 LaunchPad Development Kit [49] is selected from MSP430 family, which is the product of Texas Instruments (TI) firm used in the industry

Figure 7: PRINCE key whitening process.

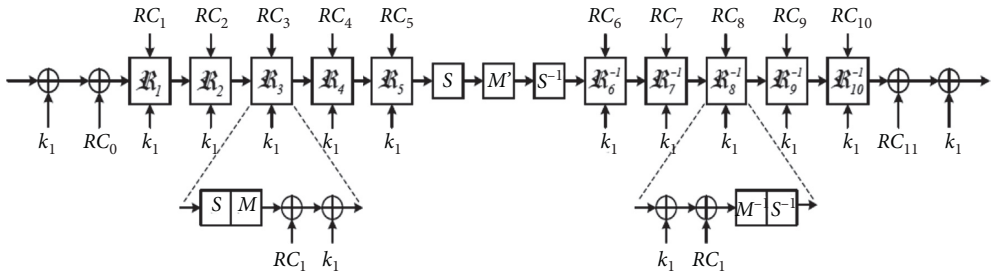


Figure 8: PRINCE encryption algorithm diagram.

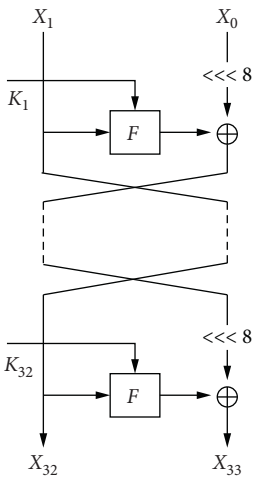


Figure 9: LBLOCK encryption round structure.

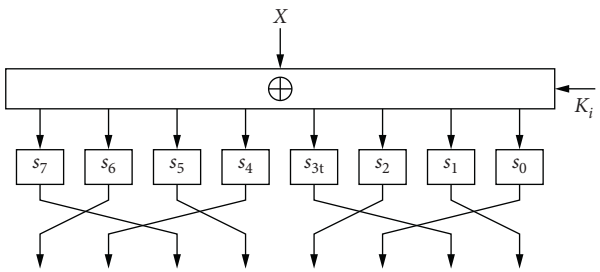


Figure 10: LBLOCK F function.

in the test environment. The kit used is a development platform for the MSP430FR5994 microcontroller. The microcontroller with 16 MHz clock frequency has 256 kB ultra-low power consumption FRAM (Ferroelectric Random Access Memory) permanent memory. MSP430FR5994 has low power consumption and is very efficient thanks to its new technology Low-Energy Accelerator (LEA). It can process analog data in real time. It has built-in eZ-FET debugging feature. With the help of this feature, the performance of the encryption code can be tested and analyzed regardless of any device. Eclipse based Code Composer Studio (CCS) [50], which can be used in devices manufactured by TI, has been used as a development environment. All lightweight encryption algorithms were compiled with C code in CCS and transferred to MSP430FR5994 device. Energy, power, and current measurements of the algorithms were carried out through EnergyTrace software [51]. EnergyTrace technology is an energy-based code analysis tool that measures and displays the energy profile of the application and also helps optimize for ultra-low power consumption. It works integrated within Code Composer Studio. It has two modes, EnergyTrace and EnergyTrace++. Basic energy measurements can be made with the EnergyTrace mode. The supply voltage in the microcontroller is sampled continuously to measure energy and power. This mode can be used to verify the application's energy consumption without accessing the debugger. EnergyTrace++ mode provides basic energy measurements as well as information about the internal status of the microcontroller such as RAM usage and energy modes. MSP430FR5994 used in the test environment is given in Figure 11.

For a secure communication environment, the task of edge devices such as MSP430 can be summarized as encrypting the data they receive from the sensors, transmitting them to cloud, server or broker role devices, and deciphering and processing data from such devices. In other words, usually these devices only encrypt or decrypt data. For this reason, encryption and decryption are considered separately for each algorithm in the scenario used in the test environment. Since the energy consumption of data transmission differs according to communication technologies such as Bluetooth, Wi-Fi, Zigbee, Z-Wave, 6LoWPAN,

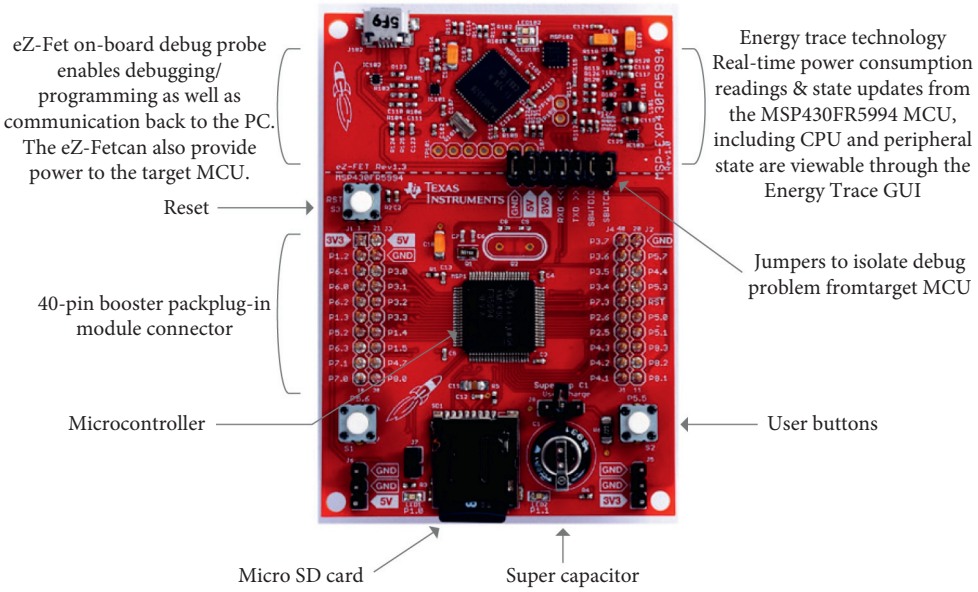


Figure 11: MSP430FR5994 overview.

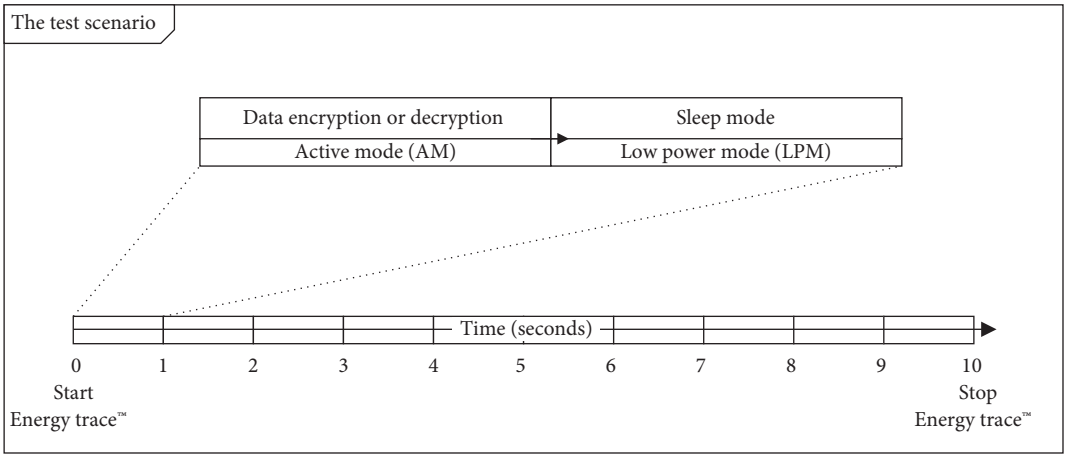


Figure 12: The timing diagram of test scenario.

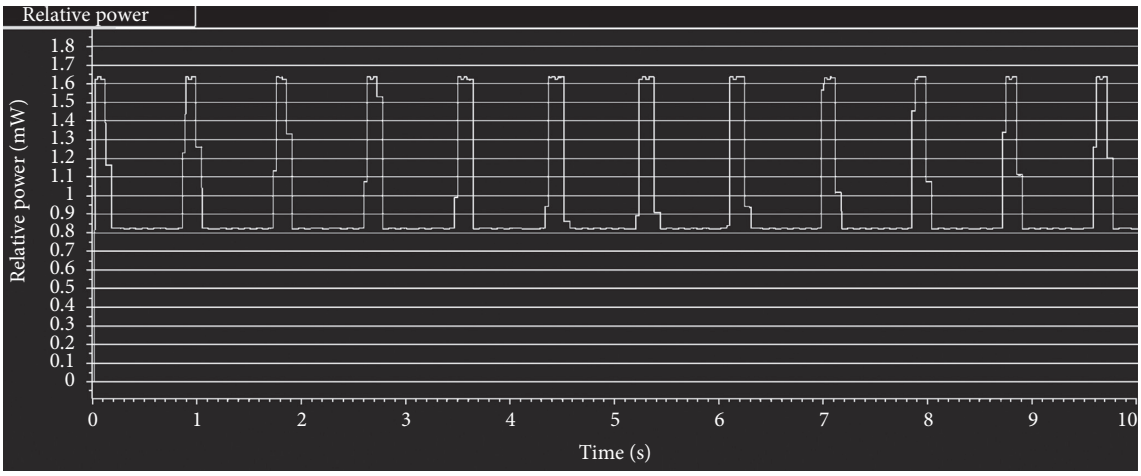


Figure 13: PRESENT encryption power consumption.

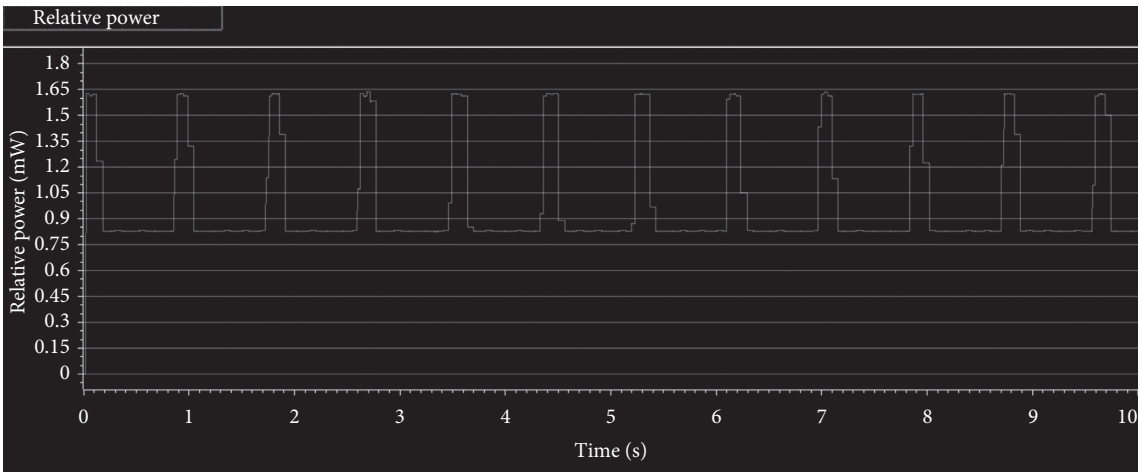
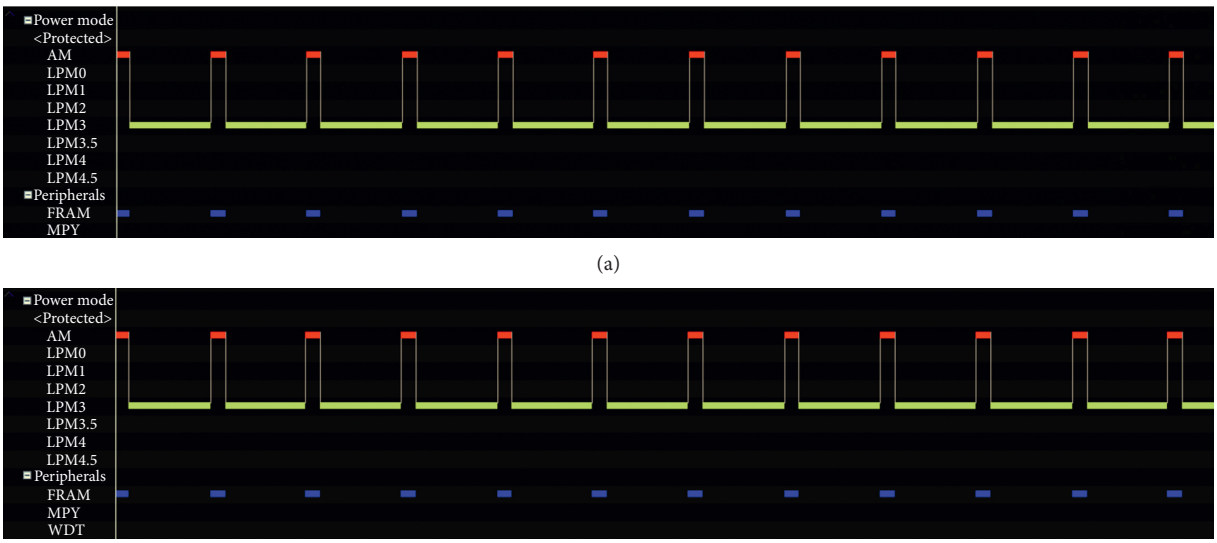


Figure 14: PRESENT decryption power consumption.



(a)

(b)

Figure 15: AM-LPM transitions for PRESENT encryption (a) and decryption (b).

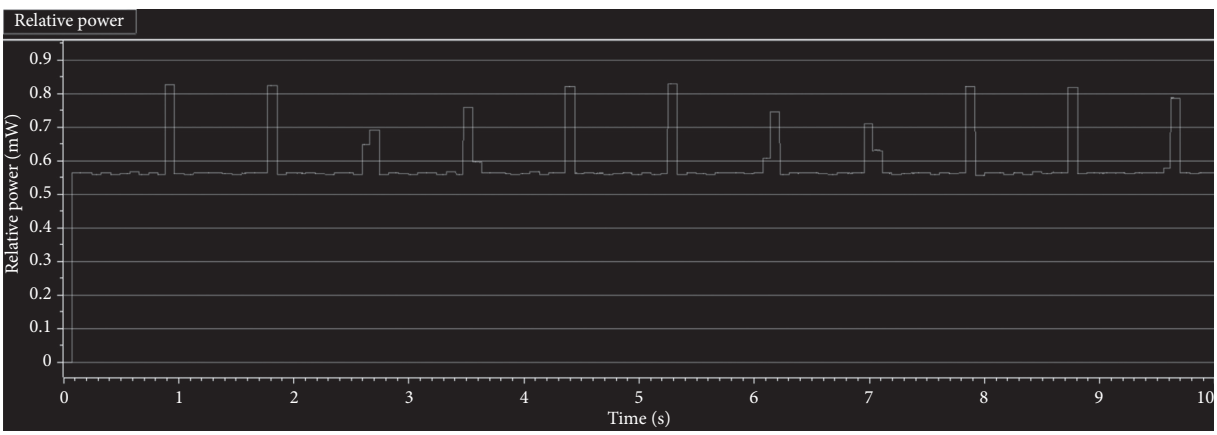


Figure 16: LBLOCK encryption power consumption.

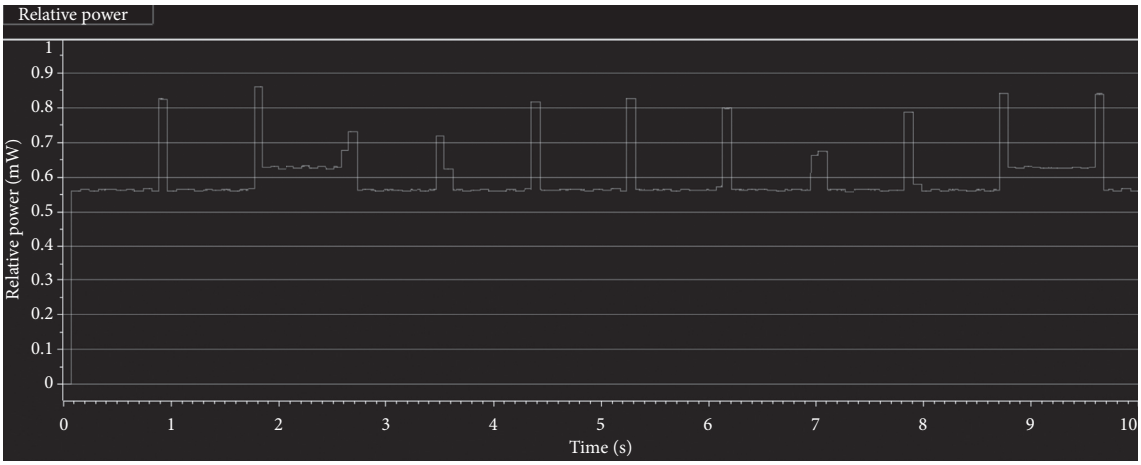
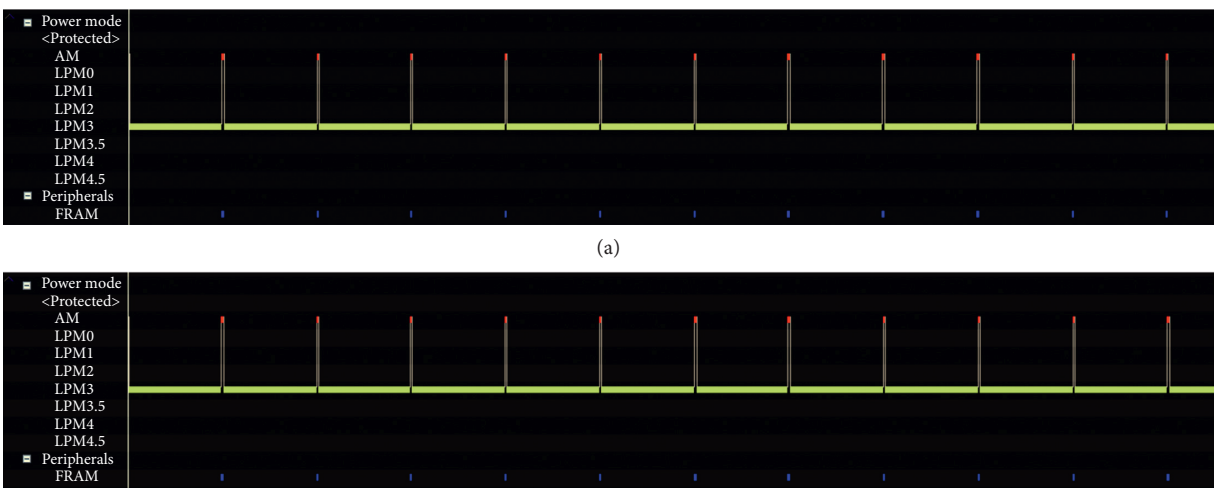


Figure 17: LBLOCK decryption power consumption.



(a)

(b)

Figure 18: AM-LPM transitions for LBLOCK encryption (a) and decryption (b).

Table 2: Energy consumption of the analyzed algorithms.

Cipher	Architecture	Block length	Key length	Number of rounds	Procedure	Energy (mJ)
PRESENT	SPN	64	80	31	Encryption	9.523
					Decryption	9.573
CLEFIA	Feistel	128	128	18	Encryption	9.128
					Decryption	9.128
			192	22	Encryption	9.452
					Decryption	9.494
			256	26	Encryption	9.452
					Decryption	9.575
PICCOLO	Feistel	64	80	25	Encryption	8.487
					Decryption	8.490
			128	31	Encryption	8.533
					Decryption	8.547
PRINCE	SPN	64	128	12	Encryption	7.513
					Decryption	7.549
LBLOCK	Feistel	64	80	32	Encryption	5.812
					Decryption	5.924

Table 3: Power consumption of the analyzed algorithms.

Cipher	Key length	Number of rounds	Procedure	Power (mW)	
				Max	Avg.
PRESENT	80	31	Encryption	1.6383	0.9548
			Decryption	1.6374	0.9591
CLEFIA	128	18	Encryption	1.6036	0.9186
			Decryption	1.6123	0.9187
	192	22	Encryption	1.6040	0.9482
			Decryption	1.6142	0.9527
	256	26	Encryption	1.6040	0.9482
			Decryption	1.6141	0.9579
PICCOLO	80	25	Encryption	1.1419	0.8439
			Decryption	1.1768	0.8449
	128	31	Encryption	1.1886	0.8495
			Decryption	1.2431	0.8516
PRINCE	128	12	Encryption	1.1694	0.7578
			Decryption	1.2797	0.7621
LBLOCK	80	32	Encryption	0.8295	0.5799
			Decryption	0.8605	0.5894

Table 4: Current measurements of the analyzed algorithms.

Cipher	Key length	Number of rounds	Procedure	Current (mA)	
				Max	Avg.
PRESENT	80	31	Encryption	0.4982	0.2905
			Decryption	0.4980	0.2918
CLEFIA	128	18	Encryption	0.4877	0.2795
			Decryption	0.4905	0.2795
	192	22	Encryption	0.4877	0.2885
			Decryption	0.4911	0.2899
	256	26	Encryption	0.4877	0.2885
			Decryption	0.4909	0.2914
PICCOLO	80	25	Encryption	0.3472	0.2568
			Decryption	0.3579	0.2570
	128	31	Encryption	0.3614	0.2585
			Decryption	0.3782	0.2591
PRINCE	128	12	Encryption	0.3863	0.2306
			Decryption	0.3892	0.2319
LBLOCK	80	32	Encryption	0.2522	0.1764
			Decryption	0.2617	0.1793

Table 5: Estimated battery life for the analyzed algorithms.

Cipher	Key length	Number of rounds	Procedure	Battery life ($2 \times$ AA batteries)
PRESENT	80	31	Encryption	9 months and 21 days
			Decryption	9 months and 19 days
CLEFIA	128	18	Encryption	10 months and 2 days
			Decryption	10 months and 3 days
	192	22	Encryption	9 months and 23 days
			Decryption	9 months and 22 days
	256	26	Encryption	9 months and 23 days
			Decryption	9 months and 19 days
PICCOLO	80	25	Encryption	10 months and 27 days
			Decryption	10 months and 26 days
	128	31	Encryption	10 months and 25 days
			Decryption	10 months and 24 days
PRINCE	128	12	Encryption	1 year and 9 days
			Decryption	1 year and 7 days
LBLOCK	80	32	Encryption	1 year and 3 months
			Decryption	1 year and 3 months

Table 6: Active mode ratios for the analyzed algorithms.

Cipher	Key length	Number of rounds	Procedure	Active mode (time)	Active mode (energy)
PRESENT	80	31	Encryption	%15.5	%23.0
			Decryption	%15.8	%23.0
CLEFIA	128	18	Encryption	%9.9	%14.7
			Decryption	%10.0	%14.8
	192	22	Encryption	%14.0	%20.4
			Decryption	%14.0	%20.3
	256	26	Encryption	%15.4	%22.1
			Decryption	%15.4	%22.1
PICCOLO	80	25	Encryption	%1.8	%2.9
			Decryption	%1.9	%3.1
	128	31	Encryption	%1.8	%2.9
			Decryption	%1.9	%3.1
PRINCE	128	12	Encryption	%26.8	%38.8
			Decryption	%26.9	%39.0
LBLOCK	80	32	Encryption	%2.2	%3.9
			Decryption	%2.2	%3.8

and LoRaWAN, it is not considered in this scenario. According to the scenario determined in the test environment, after running MSP430FR5994, it will switch to active mode (AM) in every second, it will perform the encryption or decryption process, and then it will go into the power saving mode LPM (Low Power Mode). The timing diagram of test scenario is shown in Figure 12. Energy, power, and current data of algorithms were measured with EnergyTrace software by operating the device for 10 seconds in this way. The energy consumption of the tested algorithms was determined by compiling the obtained results.

Considering the PRESENT algorithm, the encryption process worked on MSP430FR5994 for 10 seconds, consuming 9.523 mJ of energy. As seen in Figure 13, on average, it consumed 0.9548 mW and at most 1.6383 mW. An average of 0.2905 mA and a maximum of 0.4982 mA current were drawn. In this way, the device can work with 2 AA batteries for 9 months and 21 days. The decoding process consumed 9.573 mJ of energy according to the same scenario. As seen in Figure 14, on average, it consumed 0.9591 mW and at most 1.6374 mW. An average of 0.2918 mA current and the maximum of 0.4980 mA current were drawn. In this way, the device can operate for 9 months and 19 days.

MSP430FR5994 completed the PRESENT encryption process in active mode at 15.5% of the 10 seconds of its operating time and the rest in the LPM mode at 84.5%. This situation can be seen in Figure 15. The device spent only 23% of its total energy consumption when encrypting in active mode. Similar results were obtained in the deciphering process.

On the other hand, considering the prominent LBLOCK algorithm with the lowest results, it spent 5.812 mJ of energy working on the MSP430FR5994 encryption process for 10 seconds. As seen in Figure 16, on average, it consumed 0.5799 mW and the maximum 0.8295 mW. An average of 0.1764 mA and the maximum of 0.2522 mA current were drawn. In this way, the device can work with 2 AA batteries for 1 year and 3 months. The decoding process consumes 5.924 mJ of energy according to the same scenario. As seen in Figure 17, on average, it consumed an average of 0.5894 mW and a maximum of 0.8605 mW. An average of 0.1793 mA and the maximum of 0.2617 mA current were drawn. The device can operate in this way for 1 year and 3 months.

MSP430FR5994 completed 2.2% of LBLOCK encryption process in active mode and spent the remaining 97.8% in LPM mode. This situation can be seen in Figure 18. The device consumed only 3.9% of the total energy consumption while it is encrypting in active mode. Similar results were obtained in the deciphering process.

Energy measurements of the algorithms examined throughout the study are given in Table 2, power measurements are given in Table 3, current measurements are given in Table 4, estimated battery life is given in Table 5, and operating time and active mode rates according to the energy consumed are given in Table 6.

5. Conclusions

The role of energy consumption is emphasized in this study, which was conducted to guide future studies. Access to devices can be difficult, depending on the usage areas of IoT applications. For this reason, parameters such as energy consumption and battery life should be considered when preparing secure communication applications. As mentioned earlier, communication of IoT applications is mostly unsafe. The safest and cheapest method to ensure security is data encryption.

In this study, PRESENT, CLEFIA, PICCOLO, PRINCE, and LBLOCK lightweight cryptographic algorithms, which can be used to secure data in IoT applications, were analyzed in a test environment in terms of energy consumption. The test devices were chosen from the edge devices used in the industry.

PRESENT and CLAFIA algorithms are standardized as lightweight block cipher algorithm with ISO/IEC 29192-2: 2012 document. However, these algorithms emerged at the time when Internet applications of objects had just become widespread. As a result of the tests, the energy consumption of the algorithms, current measurement, active mode working time, and active mode energy consumption were identified. The results are listed in Tables 2–6.

Accordingly, LBLOCK, which encrypts the minimum energy 64-bit block length with an 80 bit key, is used by CLEFIA, which decrypts the 128-bit block length with a 256-bit key. While the LBLOCK algorithm was first in power consumption, other algorithms gave similar values. LBLOCK takes first place in current measurement. Considering the active mode times of the device in encryption and decryption processes, PICCOLO and LBLOCK went ahead, while the PRINCE algorithm had quite bad results. Finally, when active mode energy ratios are examined, it is seen that PICCOLO and LBLOCK algorithms take the first place.

When the results obtained in the study are examined, it can be said that the number of loops and block size of the algorithms make a difference in terms of energy consumption, current measurement, active mode working time, and active mode energy consumption. CLEFIA is the encryption algorithm that has the largest block length among the algorithms examined with 128-bit block length, while, in other algorithms, 64-bit is preferred as block length. This is inherently important for devices operating in the Internet applications of low-capacity objects. It is more efficient to encrypt small size blocks. Also, those that give good results from the studied algorithms use Feistel architecture. On the other hand, increased key size decreases energy efficiency. Of course, the larger the key size is, the better the security is provided. However, in the IoT applications, the keys between 80 bits and 128 bits can be considered ideal. Selecting the structures in the algorithms in a simple way that does not consume too much energy increases efficiency. Energy consuming structures such as reduction processes and mixed bumps used in CLEFIA and PICCOLO algorithms prove this situation. It can be concluded that the reason for the LBLOCK algorithm to come first in these measurements is due to simple operations such as XOR and S-boxes in its structure.

There is no AES-like standard in the industry of IoT for lightweight algorithms. For this reason, it is possible to encounter new encryption algorithms for many new IoT in the near future. Secure data transmission is essential in the field of IoT. However, besides the security, an efficient application is also very important. Therefore, parameters like energy consumption should also be considered for the design of lightweight cryptographic algorithms to be developed in the future.

Data Availability

Data are available upon request to the corresponding author.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] K. Candır, *Technology Breakpoint and Internet of Things*, https://youtu.be/fqR2LcMmISk, 2016.

[2] E. Yetimler, "Internet of things (Nesnelerin İnterneti) nedir? Cihazların etkileşim trendleri," Karel, İstanbul, Turkey, 2017.

[3] T. Büyüktanır and B. Özer, *Internet of Things*, Kodlab, İstanbul, Turkey, 2017.

[4] K. Arslan and İ. Kırbaş, "Developing wireless sensor/actuator node prototype for internet of things applications," *The Journal of Graduate School of Natural and Applied Sciences of Mehmet Akif Ersoy University*, vol. 1, pp. 35–43, 2016.

[5] J. Bélissent, *Getting Clever about Smart Cities: New Opportunities Require New Business Models*, Forrester Research, Cambridge, MA, USA, 2010.

[6] Ö. Erdem, M. Kara, and A. İkinci, "HoneyThing: nesnelerin interneti için tuzak sistem," in *Proceedings of the 8th International Conference on Information Security and Cryptology (ISCTurkey 2015)*, Ankara, Turkey, October 2015.

[7] ITU, *The Internet of Things*, International Telecommunication Union, Geneva, Switzerland, 2005.

[8] C.o.t.E. Communities, *Internet of Things—An Action Plan for Europe*, European Economic and Social Committee, Brussels, Belgium, 2009.

[9] E. Commission, *Report on the Public Consultation on IoT Governance*, European Commission, Brussels, Belgium, 2013.

[10] D. Evans, *The Internet of Things How the Next Evolution of the Internet is Changing Everything*, Cisco, San Jose, CA, USA, 2011.

[11] D. Kim and M. G. Solomon, *Fundamentals of Information Systems Security*p. 548, 3rd edition, Jones & Bartlett Learning, Burlington, MA, USA, 2016.

[12] Unit42, "2020 unit 42 IoT threat report," Palo Alto Networks, Santa Clara, CA, USA, 2020.

[13] FIBS PUB197, "Advanced encryption standard (AES)," p. 0311, Federal Information Processing Standards Publication, Gaithersburg, MD, USA, 2001.

[14] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Proceedings of the Annual International Cryptology Conference*, August 2005.

[15] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: a very compact and a threshold implementation of AES," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, May 2011.

[16] A. Bogdanov, L. R. Knudsen, G. Leander et al., "PRESENT: an ultra-lightweight block cipher," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, September 2007.

[17] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA," in *Proceedings of the International Workshop on Fast Software Encryption*, March 2007.

[18] K. Shibutani, T. Isobe, H. Hiwatari et al., "*Piccolo*: an ultra-lightweight blockcipher," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, September 2011.

[19] J. Borghoff, A. Canteaut, T. Güneysu et al., "PRINCE–a low-latency block cipher for pervasive computing applications," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, December 2012.

[20] W. Wu and L. Zhang, "LBlock: a lightweight block cipher," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, June 2011.

[21] ISO, "Information technology-security techniques-lightweight cryptography—part 2: block ciphers," ISO, Geneva, Switzerland, IEC 29192-2: 2012, 2012.

[22] M. Wang, "Differential cryptanalysis of reduced-round PRESENT," in *Proceedings of the International Conference on Cryptology in Africa*, June 2008.

[23] K. Ohkuma, "Weak keys of reduced-round PRESENT for linear cryptanalysis," in *Proceedings of the International Workshop on Selected Areas in Cryptography*, August 2009.

[24] J. Y. Cho, "Linear cryptanalysis of reduced-round PRESENT," in *Proceedings of the Cryptographers' Track at the RSA Conference*, March 2010.

[25] F. Abed, C. Forler, E. List, S. Lucks, and J. Wenze, "Biclique cryptanalysis of PRESENT, LED, and KLEIN," p. 5912012, Bauhaus-Universität, Weimar, Germany, 2012.

[26] Y. Tsunoo, E. Tsujihara, M. Shigeri et al., "Impossible differential cryptanalysis of CLEFIA," in *Proceedings of the International Workshop on Fast Software Encryption*, Lausanne, Switzerland.

[27] W. Zhang and J. Han, "Impossible differential analysis of reduced round CLEFIA," in *Proceedings of the International Conference on Information Security and Cryptology*, December 2008.

[28] C. Tezcan, "The improbable differential attack: cryptanalysis of reduced round CLEFIA," in *Proceedings of the International Conference on Cryptology in India*, December 2010.

[29] W. Wang and X.-Y. Wang, "Saturation Cryptanalysis of CLEFIA," *Journal of Communications*, vol. 29, no. 10, pp. 88–92, 2008.

[30] X. Tang, B. Sun, R. Li, and C. Li, "Impossible differential cryptanalysis of 13-round CLEFIA-128," *Journal of Systems and Software*, vol. 84, no. 7, pp. 1191–1196, 2011.

[31] Y. Li, W. Wu, and L. Zhang, "Improved integral attacks on reduced-round CLEFIA block cipher," in *Proceedings of the International Workshop on Information Security Applications*, August 2011.

[32] G. Han and W. Zhang, "Improved biclique cryptanalysis of the lightweight block cipher piccolo," *Security and Communication Networks*, vol. 2017, Article ID 7589306, 12 pages, 2017.

[33] J. Song, K. Lee, and H. Lee, "Biclique cryptanalysis on lightweight block cipher: HIGHT and piccolo," *International Journal of Computer Mathematics*, vol. 90, no. 12, pp. 2564–2580, 2013.

[34] T. Ashur, O. Dunkelman, and N. Masalha, "Linear cryptanalysis reduced round of piccolo-80," in *Proceedings of the International Symposium on Cyber Security Cryptography and Machine Learning*, June 2019.

[35] M. Tolba, A. Abdelkhalek, and A. M. Youssef, "Meet-in-the-middle attacks on reduced round piccolo," in *Lightweight Cryptography for Security and Privacy*Springer, Cham, Switzerland, 2015.

[36] M. Minier, "On the security of *piccolo* lightweight block cipher against related-key impossible differentials," in *Proceedings of the International Conference on Cryptology in India*, December 2013.

[37] P. Derbez and L. Perrin, "Meet-in-the-middle attacks and structural analysis of round-reduced PRINCE," *Journal of Cryptology*, vol. 33, no. 1, pp. 1–32, 2020.

[38] P. Morawiecki, "Practical attacks on the round-reduced PRINCE," *IET Information Security*, vol. 11, no. 3, pp. 146–151, 2016.

[39] F. Abed, E. List, and S. Lucks, "On the security of the core of PRINCE against biclique and differential cryptanalysis," Cryptology ePrint Archive, Report 2012/712, , p. 712, 2012 .

[40] S. Rasoolzadeh and H. Raddum, "Cryptanalysis of PRINCE with minimal data," in *Proceedings of the International Conference on Cryptology in Africa*, April 2016.

[41] H. Soleimany, C. Blondeau, X. Yu et al., "Reflection cryptanalysis of PRINCE-like ciphers," *Journal of Cryptology*, vol. 28, no. 3, pp. 718–744, 2015.

[42] A. Canteaut, T. Fuhr, H. Gilbert et al., "Multiple differential cryptanalysis of round-reduced PRINCE," in *Proceedings of the International Workshop on Fast Software Encryption*, March 2014.

[43] H. Soleimany and K. Nyberg, "Zero-correlation linear cryptanalysis of reduced-round LBlock," *Designs, Codes and Cryptography*, vol. 73, no. 2, pp. 683–698, 2014.

[44] Y. Wang, W. Wu, X. Yu, and L. Zhang, "Security on LBlock against biclique cryptanalysis," in *Proceedings of the International Workshop on Information Security Applications*, August 2012.

[45] F. Karakoç, H. Demirci, and A. E. Harmancı, "Impossible differential cryptanalysis of reduced-round LBlock," in *Proceedings of the IFIP International Workshop on Information Security Theory and Practice*, June 2012.

[46] Y. Wang and W. Wu, "Improved multidimensional zero-correlation linear cryptanalysis and applications to LBlock and TWINE," in *Proceedings of the Australasian Conference on Information Security and Privacy*, July 2014.

[47] J. Chen and A. Miyaji, "Differential cryptanalysis and boomerang cryptanalysis of LBlock," in *Proceedings of the International Conference on Availability, Reliability, and Security*, September 2013.

[48] Y. Cui, H. Xu, and W. Qi, "Improved integral attacks on 24-round LBlock and LBlock-s," *IET Information Security*, vol. 14, no. 5, 2020.

[49] "MSP430FR5994LaunchPad™ development kit," in *User's Guide*Texas Instruments, Dallas, TX, USA, 2019.

[50] "Code composer studio™ IDE v10.X forMSP430™ MCUs," in *User's Guide*Texas Instruments, Dallas, TX, USA, 2020.

[51] W. G. BrittanyFinch, "MSP430™ advanced power optimizations: ULP Advisor™ software and EnergyTrace™ technology," in *Application Report*Texas Instruments, Dallas, TX, USA, 2014.