LIGHTWEIGHT APPROACHES IN BLOCK CRYPTOGRAPHY: A SURVEY

Fernando Ramirez Arredondo
School of Computer Science
Universidad Católica San Pablo
Email: fernando.ramirez@ucsp.edu.pe

Abstract—**Lightweight Cryptography (LWC) is an expanding field of research as a result of its importance in providing security, via encrypting sensitive information, for IoT devices and other systems where traditional cryptography would be impractical due to their resource demands. It is for this reason that several LWC primitives have been proposed. Among these, lightweight block ciphers are the most popular in terms of methods proposed and usage. These symmetric cryptographic algorithms operate on fixed-size blocks of data and transform each block independently into an output block of the same size. This work surveys lightweight block ciphers' algorithms for resource-constrained devices, which have become increasingly important in today's world for their wide range of useful applications, from convenience and efficiency in our daily routines to far-reaching applications in healthcare, industry, and urban planning. The goal of this work is to provide a comprehensible classification of these methods in terms of performance and resources required, followed by an accurate explanation of the tradeoff between these two when working with limited resources.**

Index Terms—**Lightweight cryptography (LWC), Lightweight block ciphers, Resource-constrained devices**

1. Introduction

Back in 2016, World Economic Forum founder and executive chairman Klaus Schwab wrote a book titled The Fourth Industrial Revolution, in which he explained that the way we live, work and relate to one another was about to be fundamentally altered by a digital revolution unlike the ones humanity had experienced before. Back to today, this so-called fourth industrial revolution has been characterized by a fusion of technologies that is blurring the lines between the physical, digital, and biological spheres [1]. The Internet of Things (IoT) paradigm corresponds to the inter-device domain created by devices embedded with electronics, software, sensors, and connectivity, enabling the transmission, receiving, and processing of data through various communication infrastructures. IoT generates more opportunities for direct integration between the physical world and computer-based systems. As of 2019, IoT, which used to operate in smaller network spaces, has expanded to wide area networks, thereby increasing associated risks due to the expected surge in IoT devices in diverse environments. With the rapid growth of IoT applications, a substantial amount of sensitive data is generated and exchanged, leading to the observation of several security and privacy issues. As devices become more interconnected, security and privacy concerns will become more pronounced, continuously exposing additional security flaws and weaknesses. In statistical terms, all exposed errors and weaknesses may be exploited in an environment with billions of devices.

IoT devices can be categorized into two groups based on their resources. The first group includes devices abundant in resources like servers, personal computers, tablets, smartphones, etc. The second group comprises devices limited in resources, such as industrial sensors or sensor nodes, RFID tags, actuators, etc. This last group of devices face multiple constraints, including limitations in resources, memory, power/energy consumption, and speed. The challenge in terms of performance arises when attempting to implement traditional cryptography in devices with limited resources. This difficulty is attributed to the intricate and resource-intensive mathematical operations involved in traditional cryptographic techniques. Such operations require substantial memory space and demand high processing power. Moreover, the implementation cost of traditional cryptography in low-resource devices is notably high. Consequently, lightweight cryptography was introduced to address these issues. Lightweight cryptography, born out of the remarkable expansion in emerging ubiquitous technologies, represents a modern cryptographic technique.

Ciphers are categorized as either asymmetric or symmetric. While asymmetric ciphers provide enhanced security features, they require more computational power and tend to be relatively more costly. It is for this reason that asymmetric ciphers are not popular in the context of resource-constrained devices. Block ciphers and stream ciphers are the two main classifications of symmetric ciphers. This work will focus on block ciphers because most of the research in the LWC field centers around this paradigm. Several lightweight block ciphers have been suggested to gain performance advantages compared to traditional cryptographic methods. Certain ciphers achieved this by simplifying well-established block ciphers, enhancing their efficiency. Notable examples include AES-128 [2], a variant of NIST's Advanced Encryption Standard (AES), and DESL [3], which is a modified version of DES. In DESL, the round function and permutations were reduced to enhance hardware implementation efficiency. On the other hand, some algorithms are dedicated block ciphers crafted from the ground up. An early example is PRESENT [4], specifically designed for resource-constrained hardware environments. Additionally, there are families of lightweight block ciphers such as SIMON and SPECK [5]. These were created to be uncomplicated, adaptable, and to deliver optimal performance in both hardware and software implementations. The advantages in performance seen in lightweight block ciphers compared to traditional block ciphers result from design decisions that prioritize smaller block sizes, reduced key sizes, simpler rounds, more straightforward key schedules, and minimal implementations. These characteristics will be discussed in detail in the upcoming sections.

The rest of the paper is organized as follows: Section 2 delves into the theorical framework of lightweight block ciphers. This section aims to establish a solid foundation by discussing background information on lightweight ciphers and their significance in ensuring security in low-resource devices. Theorical concepts and principles underlying these ciphers will be thoroughly examined. Section 3 presents a survey of related work in the realm of lightweight block ciphers. This includes a comprehensive review of existing literature, comparing and contrasting various approaches. Section 4 introduces our taxonomy of the cipher implementation space. This taxonomy serves as a structured framework for categorizing and understanding different aspects of lightweight block ciphers. It explores variations, modes of operation, and algorithmic choices, providing a roadmap for researchers and practitioners to navigate this complex landscape. In Section 5, we conduct an in-depth analysis of the strengths and weaknesses inherent in lightweight block ciphers. This assessment encompasses scenarios where these ciphers excel in terms of security and performance, as well as potential vulnerabilities or trade-offs that should be considered in their application. Finally, in Section 6, the paper is concluded with a summary of key findings and overarching conclusions. This section encapsulates the contributions of our work, discusses the implications of our taxonomy, and provides closing remarks on the significance of lightweight block ciphers in contemporary cryptographic applications.

2. Theorical framework

2.1. Lightweight Block Ciphers

In Fan et al.'s study, a lightweight cipher was defined as a cryptographic algorithm designed specifically for devices with limited resources. The algorithm needed to tackle three key challenges: minimizing overhead (in terms of silicon area or memory footprint), ensuring low-power consumption, and maintaining a sufficient level of security.

Other researchers have adopted a quantitative approach to the definition of lightweight ciphers. For instance, in the study conducted by Cazorla et al. specific criteria were proposed for categorizing a cipher as lightweight. These criteria encompass attributes such as block size, key size, operations, and key scheduling. In the context of lightweight ciphers, the term "lightweight" denotes a reduced block size (32, 48, or 64 bits) in contrast to conventional ciphers, which typically feature larger block sizes (64 or 128 bits).

Smaller block sizes in lightweight block ciphers, such as 64 or 80 bits compared to AES's 128 bits, are employed to optimize memory. However, it's crucial to recognize that the use of smaller block sizes imposes restrictions on the maximum number of plaintext blocks that can be encrypted.

Smaller key sizes, less than 96 bits, are implemented in lightweight block ciphers for efficiency (e.g., 80-bit PRESENT), measured in terms of power consumption.

Simpler key schedules are a priority in lightweight block ciphers to mitigate the increased memory, latency, and power consumption associated with complex implementations. However, the choice for simplicity introduces potential vulnerabilities to attacks involving related keys, weak keys, known keys, or chosen keys. To address these concerns, it is recommended to employ a secure Key Derivation Function (KDF).

Simpler rounds are featured in lightweight block ciphers with less complex components and operations compared to traditional block ciphers. In designs utilizing S-boxes, a preference for 4-bit S-boxes over 8-bit ones is observed, resulting in significant area savings. For example, the 4-bit S-box in PRESENT required 28 GEs, whereas the AES S-box needed 395 GEs. Hardware-oriented designs may opt for bit permutations (as in PRESENT) or recursive MDS matrices (as seen in PHOTON and LED) instead of intricate linear layers. It's noteworthy that simpler rounds may necessitate more iterations to achieve the desired level of security.

Two lines of action have been employed to offer lightweight cryptographic primitives. The first involves creating efficient, cost-effective implementations for established and trusted algorithms, primarily focusing on block ciphers like AES. The second approach entails developing novel ciphers with the aim of minimizing hardware implementation costs, exemplified by the Simon and Speck families.

2.2. Target Devices

Lightweight cryptography is designed to cater to a diverse range of devices and can be applied across a wide array of hardware and software configurations (See Table 1). At the upper echelon of this device spectrum are servers and desktop computers, succeeded by tablets and smartphones. Traditional cryptographic algorithms typically exhibit satisfactory performance on these higher-end devices, making lightweight algorithms unnecessary for these platforms. However, at the lower end of the spectrum, we encounter devices like embedded systems, RFID devices, and sensor networks. It is in these highly constrained environments that lightweight cryptography takes center stage, addressing the specific needs of devices in this category.

Device	Cryptography
Servers, Desktops, Tablets, and Smartphones	Conventional Cryptography
Embedded Systems and RFID	Lightweight Cryptography

Table 1. Cryptographic Approaches on Different Devices

While lightweight cryptography is primarily designed for devices at the lower end of the device spectrum, it's crucial to acknowledge the potential necessity of implementing lightweight algorithms at the higher end of the spectrum as well. For instance, even though many resource-constrained sensors may transmit data to an aggregator that, by most standards, is not constrained, the aggregator must still accommodate lightweight algorithms to interact effectively with the constrained sensors utilizing lightweight cryptographic methods. In essence, the decision on whether conventional standards are acceptable should consider the specific environment and application. The demand for lightweight cryptography arises not only from the limitations of a particular device but also from the need to ensure compatibility with other devices directly interacting within the application.

2.3. Applications and Architectures

IoT systems offer a wide range of services, including Intelligent Transportation Systems (ITS), smart grids, smart buildings, smart cities, e-Health, intelligent drug delivery systems, and more. Even Cyber-Physical Systems (CPS), such as Nuclear Power Plants (NPP), are encompassed within the IoT framework. The majority of these services are of a critical nature. Each IoT system is tailored to deliver a specific service, and for each application, diverse architectures are in development. Nevertheless, the shared drawbacks among many proposed architectures prevent them from fully meeting all IoT requirements. The applications can be classified into various domains, including Medical, Military, Industrial, Automobile, Environmental, Agriculture, Retail, and Consumer. Each domain offers its unique advantages while simultaneously tackling the IoT-related challenges discussed earlier.

2.4. Performance Metrics

In the domain of designing cryptographic algorithms, there is a delicate equilibrium between the attained performance and the resources necessary to uphold a particular security level. Factors influencing performance include power and energy consumption, latency, and throughput. Power and energy consumption are crucial metrics for constrained devices. Devices relying on harvested power, such as RFID chips utilizing electromagnetic fields, emphasize the significance of power considerations. In battery-operated devices with fixed energy stores, the concept of energy consumption gains prominence, especially when batteries are challenging or impossible to recharge or replace post-deployment. Latency assumes special significance in specific real-time applications, being defined as the time gap between initiating an operation and generating its output. For example, in the realm of encryption, latency represents the interval from the initial request to encrypt plaintext to the reception of the corresponding ciphertext in the response. Throughput pertains to the pace at which new outputs, such as authentication tags or ciphertext, are created. Unlike conventional algorithms, prioritizing high throughput may not be the main focus in lightweight designs. Nevertheless, a moderate level of throughput remains essential for most applications.

In hardware implementations, resource needs are commonly measured in terms of gate area, gate equivalents, or logic blocks. In software environments, these considerations are reflected in the utilization of registers, RAM, and ROM. These resource requirements are frequently labeled as costs, as increasing the number of gates or memory tends to raise the overall production cost of a device.

2.4.1. Hardware-Specific Metrics.

Resource needs for hardware platforms are commonly expressed in terms of gate area, measured in μm^2. This area can be specified in logic blocks for field-programmable gate arrays (FPGAs) or in gate equivalents (GEs) for application-specific integrated circuit (ASIC) implementations. In the realm of FPGAs, a logic block serves as the fundamental reconfigurable unit, housing various components like look-up tables (LUTs), flip-flops, and multiplexers. For ASICs, one GE corresponds to the area required by a two-input NAND gate. The GE count is determined by dividing the area in μm^2 by the area of the NAND gate.

2.4.2. Software-Specific Metrics.

In software applications, the assessment of resource needs involves considering the number of registers, RAM, and ROM bytes required. Functions with fewer registers experience lower calling overhead. ROM stores program code and fixed data, while RAM holds intermediate values for computations. This presents tradeoffs between on-demand value calculation and referencing precomputed values.

2.5. Security Analysis Techniques

Cryptographic algorithms undergo diverse tests to confirm their efficiency. Within this study, we will provide a broad overview of the five most widely used cryptanalysis techniques.

2.5.1. Linear Cryptanalysis.

This attack is also referred to as a plaintext attack, and it is considered a basic attack. It relies on the occurrence of high probability linear expressions where plaintext bits, ciphertext bits, and subkeys are accountable.

2.5.2. Differential Cryptanalysis.

In this attack, the identification of trails is achieved through the utilization of a difference distribution table (DDT). The construction of the DDT is facilitated by high-probability input and output differences.

2.5.3. Zero Correlation Attack. In the zero correlation attack, we modify a single input bit and determine the corresponding ciphertext according to predefined rules.

2.5.4. Biclique Attack. This attack is a theoretical approach that determines the data complexity and computational complexity of the cipher. The Biclique attack is an extension of the meet-in-the-middle attack (MITM).

2.5.5. Avalanche Effect Attack. The avalanche effect demonstrates the randomization nature of the cipher and its robust cryptanalytic properties, ensuring that under any circumstances, predicting the input plaintext is not feasible.

2.6. Classification of Attacks

Attacks on IoT are divided into two modules.

2.6.1. Protocol-Based Attacks. These types of attacks exploit the internal protocol-based structure of IoT components, impacting the communication medium and the forwarding channels of embedded systems.

2.6.2. Data-Based Attacks. Data-based attacks encompass threats related to the original data packets and messages traversing through node sites. Security vulnerabilities such as hash collisions, denial-of-service (DoS) attacks, malicious node virtual machine creation, and data exposure are among the most impactful exploits in this category.

3. Related Work

4. Taxonomy

5. Strengths and Weaknesses

6. Conclusions

References

[1] “World Economic Forum the fourth industrial revolution: what it means, how to respond,” https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/, accessed: 2023-11-07.

[2] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, “Strong authentication for rfid systems using the aes algorithm,” in *Cryptographic Hardware and Embedded Systems-CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings 6*. Springer, 2004, pp. 357–370.

[3] G. Leander, C. Paar, A. Poschmann, and K. Schramm, “New lightweight des variants,” in *Fast Software Encryption: 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers 14*. Springer, 2007, pp. 196–210.

[4] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, “Present: An ultra-lightweight block cipher,” in *Cryptographic Hardware and Embedded Systems-CHES 2007: 9th International Workshop, Vienna, Austria, September 10-13, 2007. Proceedings 9*. Springer, 2007, pp. 450–466.

[5] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The simon and speck families of lightweight block ciphers,” *cryptology eprint archive*, 2013.