Lightweight Cryptographic Algorithms: A Position Paper

Gabriela Mendes Corrêa de Miranda[1], José Antônio Moreira Xexéo[1] and Renato Hidaka Torres[2]
[1]*Military Institute of Engineering, Rio de Janeiro, RJ, Brazil*
[2]*Federal University of Pará, Belém, PA, Brazil*

Keywords: Cryptographic Algorithms, Lightweight Cryptography, Symmetric Cryptography, Internet of Things, Metrics.

Abstract: The massive connection of devices and huge data traffic on networks have made information susceptible to different type of attacks in Internet of Things (IoT) ecosystem. Devices utilized in these settings introduce complexities in implementing traditional cryptographic solutions, given their constraints such as physical size, processing requirements, memory limitations, and energy consumption. This position paper motivates ongoing and future research on this topic by providing a review to identify, analyze, and classify design components of lightweight cryptographic algorithms. It emphasizes a need to define a set of evaluation metrics and gives a further research progress in lightweight algorithms implementations in order to contribute design robust solutions and architectures.

1 INTRODUCTION

IoT refers to a growing network of everyday physical objects connected to the Internet (Harbi et al., 2021). A main goal of IoT is to evolve Internet-enabled devices into an interconnected ecosystem, allowing digital data to be accessible anywhere and anytime.

The number of connected IoT devices is predicted to grow to 1 trillion by 2025 (Manyika et al., 2015). These resource constrained devices have inherently limited memory space, low processing capacity, and computation power.

Cryptography enables information confidentiality and integrity. Due to their poor processor and memory capacities, low end devices cannot support conventional cryptography (Farhan and Kharel, 2019). This clearly outlines the need to develop Lightweight Cryptography (LWC).

LWC is a field dealing with algorithms or protocols specially designed for the usage in restricted environments (Kouicem et al., 2018). Lightweight cryptographic algorithms are preferred for providing low energy consumption, processing, storage capacity, and memory usage.

In this paper, an in-depth research work has been conducted. The contributions obtained from this research are to discuss: i) the evolution of LWC ciphers; ii) the design components of existing algorithms and iii) an analysis between hardware vs software metrics.

This work is structured as follows. Section II presents the background of security for IoT context and lightweight ciphers. Section III presents a literature review for lightweight primitives. Section 4 discusses research issues. And Section 5 presents the conclusion.

2 BACKGROUND

2.1 Security for Low-Resource Devices

Security aims to preserve, restore and guarantee the protection of information in computer systems from malicious attacks and threats (Kouicem et al., 2018).

IoT enables to improve several applications in various fields such as healthcare, smarts grids, smart cities, smart homes as well as other industrial applications (Sevin and Mohammed, 2021). However, introducing constrained IoT devices and technologies in such sensitive applications leads to new security and privacy challenges.

Low-resource devices should resist against some security challenges as vulnerabilities and heterogeneity of communication and information system technologies, data sensitivity and privacy, resources limitations, mobility, lack of standardization and safety.

Most of IoT applications operate in highly distributed environments with the use of heterogeneous smart objects, sensors and actuators that are limited in terms of power and computation resources.

Corrêa de Miranda, G., Xexéo, J. and Torres, R.
Lightweight Cryptographic Algorithms: A Position Paper.
DOI: 10.5220/0012792900003767
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 764-770
ISBN: 978-989-758-709-2; ISSN: 2184-7711
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

Table 1: Characteristics of LWC algorithms.

Characteristics	Lightweight
Physical (Cost)	Smaller block size
	Smaller key size
	Simple rounds
Performance	Simple key scheduling
Security	Adoption of one of the six structures (SPN, FN, GFN, ARX, NLFSR, Hybrid)

2.2 Lightweight Ciphers

The existing cryptographic primitives contain two main categories: asymmetric and symmetric key cryptography (Stallings, 2013). Symmetric key algorithms as stream ciphers, block ciphers and authenticated encryption algorithms. Asymmetric algorithms can be divided into encryption algorithms and key distribution algorithms.

LWC is a group of cryptographic primitives, methods, and ciphers intended to provide solutions for resource-limited devices such as IoT (Patel and Mistry, 2015).

There are two criteria used to determine the lightweight of a cryptographic algorithm (Thakor et al., 2021). The first is the software weight of the cipher defined by the cipher's time and memory complexities. The second criterion is the hardware weight of the cipher and it is defined by the cipher's area and power consumption. The cipher's area is represented by the number of gate equivalencies (GE) used to implement the cipher and the power consumption is the power demanded during the cipher's execution.

In order to fulfill security needs lightweight ciphers adopts one of the six internal structures: Substitution Permutation Networks (SPN), Feistel Networks (FN), Generalized Feistel Network (GFN), Add-Rotate-XOR (ARX), NLFSR-based or Hybrid to immune against the security attacks (Thakor et al., 2021). The algorithm needs to meet the lightweight standards while having a similar performance for security attacks and security standards. The main characteristics of lightweight cryptographic algorithms are listed in Table 1.

3 LITERATURE REVIEW

3.1 Standardization

Several initiatives have been conducted for standardization for LWC. In 2000 the New European Schemes for Signatures, Integrity, and Encryption (NESSIE) proposed a open call for a broad set of primitives providing confidentiality, data integrity, and authentication. Between 2004 and 2008, the ECRYPT Stream Cipher Project (eSTREAM) was an effort to promote the design of efficient and compact stream ciphers (Biryukov and Perrin, 2017).

The National Institute of Standards and Technology (NIST) had been investigating cryptography issues for constrained environments. In 2023 NIST announces the selection of the ASCON family (Dobraunig et al., 2021) for LWC standardization.

The International Organization for Standards (ISO) and the International Electrotechnical Commission (IEC) has defined standards that include architectures and techniques for information and communication security. The *ISO/IEC 29192 - Lightweight cryptography* specifies block ciphers, stream ciphers and hash functions suitable for lightweight cryptography.

NIST defined LWC as a cryptosystem whose features have been optimized to meet the requirements of resource-constrained devices (Abujoodeh et al., 2023). Any cryptographic primitive can be considered lightweight if it can sufficiently justify its resource requirements to achieve the intended result.

3.2 Lightweight Cryptographic Primitives

3.2.1 Asymmetric Encryption

Public key cryptography algorithms such as Rivest Shamir Adleman (RSA) (Rivest et al., 1978), Elliptic Curve Cryptography (ECC) (Miller, 1986) and Hyperelliptic Curve Cryptography (HECC) (Ranganatha Rao and Sujatha, 2023) provides security services and mechanisms such as data confidentiality, data integrity, access control and non-repudiation.

Asymmetric ciphers require significantly more computational resources. An optimized asymmetric algorithm such as ECC performs 100 to 1,000 times more slowly than a symmetric cipher such as the Advanced Encryption Standard (AES) algorithm (Eisenbarth et al., 2007).

ECC emerged as a preferred cryptographic style due to its shorter key length and reduced power consumption while maintaining a similar level of security.

(Khan et al., 2020) and (Aswathy and Nandagopal, 2021) are examples of proposed methods.

Enhancing the execution speed of ECC operations beside reducing energy consumption and memory requirements can improve ECC implementations, becoming more feasible for IoT devices.

3.2.2 Hashing

Cryptographic hash functions have been implemented in different cryptographic mechanisms as digital signatures, pseudorandom number generators, key generation, password security, and blockchains.

Conventional hash function typically has a sizable internal state size and high power consumption. A lightweight hash function needs smaller output size and smaller message size, thus optimized hash functions for short messages might be better suited for lightweight applications. (Singh et al., 2017).

QUARK (Aumasson et al., 2010) is a lightweight hash function family designed for resource-constrained hardware environments, as RFID tags.

Hashing methods such SPONGENT (Bogdanov et al., 2011) and PHOTON (Guo et al., 2011) produces a much smaller memory footprint.

ASCON-HASH (Dobraunig et al., 2021) is a member of the ASCON family of cryptographic algorithms proposed in the NIST LWC competition.

While a smaller hash digest results in faster computations due to its reduced output size, it's important to balance this reduction since a smaller message digest can compromise data integrity by collisions.

3.2.3 Signing

Message Authentication Code (MAC) symmetric ciphers are a class of keyed functions used to ensure that a message has been sent by the true sender and received without having been altered during transmission (Duval and Leurent, 2020).

GRAIN-128A (Ågren et al., 2011) is stream cipher proposed as an improvement by enhancing the security and optional message authentication.

Hummingbird-2 (HB-2) (Engels et al., 2012), optionally produces a MAC for each message processed and is developed with both lightweight software and hardware implementation constraints.

The Chaskey (Mouha et al., 2014) cipher is a permutation-based LWC method for signing messages. It is patent-free and standardized in ISO/IEC 29192:2015.

The LightMAC (Luykx et al., 2016) cipher offers compact authentication for resource-constrained platforms, and also allows high-performance parallel implementations.

ACORN (Shi and and, 2019) is a lightweight authenticated encryption cipher finalist on the Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR).

3.2.4 Streaming

A stream cipher encrypts and decrypts data bit by bit which means there is continuous transmissions, and is simpler and quicker compared to block ciphers. This type of cipher is mainly used in applications where plaintext size is unknown (Sehrawat and Gill, 2019).

Enocoro (Watanabe et al., 2008) is a family of hardware-oriented pseudo-random number generator. It is a lightweight cipher and it has a 128-bit key and a 64-bit initialization vector value.

Trivium (De Cannière, 2006) is a hardware oriented synchronous stream cipher which aims to provide a flexible trade-off between speed and area. It is a construction based on block cipher design principles.

Enocoro and Trivium are specified as standards for stream cipher methods within ISO/IEC 29192-3:2012.

Stream ciphers have a speedy implementation in hardware but due to lengthy initialization phase and to some of the communication protocols that do not utilize stream ciphers, these are less preferred over block ciphers (Sehrawat and Gill, 2019).

Lightweight block ciphers are taking priority over stream ciphers also due to their lower development complexity which is an important feature of a cipher.

3.2.5 Block

Key size, block size, structure type, and the encryption/decryption rounds are the primary considerations to evaluate a lightweight block cipher (Hatzivasilis et al., 2018).

Data Encryption Standard (DES) is one of the first ciphers to be investigated for LWC. The DES variant DESX (Leander et al., 2007) uses key whitening to increase the security level and prevent brute force attacks.

CLEFIA (Shirai et al., 2007) makes functions sharing between data scheduling and key scheduling parts resulting reduction in the gate size and low cost.

PRESENT (Bogdanov et al., 2011) is a milestone in the evolution of lightweight block ciphers and is used along with AES as a benchmark for newer proposals.

The SIMON and SPECK (Beaulieu et al., 2015) family of block ciphers have multiple instantiations. The author claimed that SPECK has the highest throughput in software compared with any block ciphers in the literature and SIMON have the best performance in hardware performance.

LEA (Lee et al., 2014) is a software-oriented ARX which provides a high-speed software encryption on general-purpose processors.

The block ciphers PRESENT, CLEFIA and LEA are part of the series of standards ISO/IEC 29192 for lightweight cryptographic implementations.

3.3 Performance Metrics

Metrics for software and hardware implementations of lightweight cryptographic algorithms are not identical because complexity of implementing ciphers operations are different in software and hardware.

3.3.1 Software Metrics

One of the main goals of software implementations is to keep memory and CPU needs as low as possible. Memory restrictions, however, are bound to negatively affect performance. As small memory elements are utilized, more cycles are needed to execute an operation.

There are some metrics to check the efficiency of the ciphers but in general the relevant metrics are the memory consumption, the code size and the throughput. Optimized software implementations result in fast speed thereby utilizing low power consumption. There is a strong correlation between energy consumption and cycle count and software implementations are conditioned by the coding style.

The software latency is measured in clock cycles and is technology independent. However, one related metric, throughput, may be technology dependent if the maximum throughput is required since the maximum frequency of a design is technology dependent. Power consumption is technology dependent metric but it is a much less important metric than area (Matsui and Murakami, 2014).

Authors in (Arora, 2012) introduced a Combined Metric (CM) indicating a trade-off between implementation size and performance. A better cipher implementation results a smaller metric value. CM is given by using (1):

$$CM = codesize[bits] - encryptioncyclecount[cycles] \quad (1)$$

The authors in (Baysal and Sahin, 2015) propose a new metric called *ST/A*, which is *Security times Throughput over Area*. In this new metric given by using (2), the key size is inserted to the efficiency metric formula where *KeySize* is the bit size of key used in the cipher, *Throughput* is given in bit-per-second, and *Area* is gate equivalent (GE) in hardware or memory usage in software. Hence increase in the key size increases the efficiency of a cipher.

$$ST/A = \frac{KeySize \times Throughput}{Area} \quad (2)$$

The software implementations are categorized based on the ROM and RAM requirements. Ultra-lightweight implementations require up to 4KB ROM and 256 bytes RAM, low-cost implementations require up to 4KB ROM and 8KB RAM, and lightweight implementations require up to 32KB ROM and 8KB RAM (Hatzivasilis et al., 2018).

Standardization of metrics like throughput, latency, and software efficiency is appropriate for LWC algorithms. Effective evaluation of their characteristics is required and might be done utilizing mathematical analysis.

The software implementation performance metrics are listed in Table 2.

3.3.2 Hardware Metrics

Hardware lightweight implementations try to reach the required functionality with the minimum amount of hardware real-estate (Hatzivasilis et al., 2018).

The basic performance metrics for hardware designs are area, timing, and energy (Blanc et al., 2022). Chip area is a critical factor and should have a small value.

CMOS technology is also essential in the hardware implementation of the ciphers affecting both gate equivalence and energy consumption (Dinu et al., 2019). It is unfair to make a comparison between two ciphers without taking into account their CMOS technologies once the technology also influences the chip area required in cipher's implementation.

Power is important as it is related to the power consumption of a device and attacks related to power analysis. When frequency is fixed at a low value, power consumption is directly correlated with the chip area. A small area predisposes that the circuit will consume low power.

Figure Of Merit (FoM) given by (Badel et al., 2010) was used to compare different ciphers. It was introduced considering the limitations of the efficiency metric for hardware implementation. The FoM metric is an important parameter that can use different weight factors for execution time, RAM footprint, and code size, and may even consider security aspects.

In (Dobraunig et al., 2021) is proposed a new comparison metric that allows comparison of security, time and area. The Figure of Adversarial Merit (FoAM) combines the security provided by cryptographic structures and components with their implementation properties providing a new perspective in building hardware-friendly cryptographic primitives according to area or FoAM metrics.

Hardware implementations metrics are generally reflective of the application constraints. These metrics are relative, meaning that it is usually possible to

Table 2: Software implementation performance metrics.

Metric	Definition
Memory Consumption	The amount of data written to memory during each evaluation of the function
Code Size	The fixed amount of data which is needed to evaluate the function independently
Throughput	Measures the average quantity of data which is processed during each clock cycle
Cycles/byte	Cycle count in encryption and decryption, one block Cycles/byte
Energy consumed	Given in μJ
Latency	The time taken for the computation of one block of either plain-text or cipher-text
Efficiency	Requiring little storage and consuming little energy
Power	Processing time x Device average power
Combined Metric (CM)	A tradeoff between implementation size and performance (code size x cycle)
ST/A	Security times Throughput over Area

optimize a single metric eventually, if the other ones can be compromised. The main consequence of this relativity is that a fair comparison of hardware implementations is always specific to a particular set of constraints.

The hardware implementation performance metrics are listed in Table 3.

4 DISCUSSION

Given the lightweight cryptographic algorithms challenges, this position paper highlights the following points:

- Block sizes, key sizes, and key scheduling should be taken into consideration and need to be small. Small blocks and short-key length can simplify the encryption and decryption process.
- The use of elementary operations as addition, AND, OR, exclusive or shift are efficient because simple operations can be applied to all elementary platforms.
- Increasing the number of iteration rounds and the length of the key enhances system security, but faster and stronger ciphers typically come with higher costs. However, more rounds mean slowness in algorithms.
- For some applications either energy or power consumption are critical whereas for other applications a low latency is much more important.
- Synthetic metrics are used to combine two or more non-correlating metrics to capture various aspects of the performance.
- The energy per bit is an appropriate metric for energy-constrained low-resource device applications.
- Only software latency is independent of technology, while power and area depend on technology.
- Throughput may depend on technology when the maximum throughput is needed, as the maximum frequency depends on the technology being used.
- It is important for a lightweight algorithm occupy small chip area for its implementation.

In general none of the lightweight algorithms meets all the criteria for both hardware and software performance metrics.

A well-defined lightweight metric of cost and performance might check the efficiency of a lightweight cipher or assess its feasibility for a target application. There are plenty of implementation choices and the designs are optimized for specific evaluation metrics.

In contrast from classical algorithms LWC differ by assuming that lightweight primitives aren't designed for wide range application usage and that there is no need to encrypt a great number of data. For these environments lightweight implementations might provide a better balance between security and performance.

LWC should not be associated with weak cryptography but for classic cryptographic contexts conventional approaches continue to be recommended to guarantee data security.

Metrics for evaluating the security performance and hardware and software implementations vary widely. As mentioned in the previous discussion, because of this condition, fair comparisons of different algorithm implementations are a hard issue. Therefore, standard hardware and software security and performance metrics should be developed to analyze LWC security and implementations.

5 CONCLUSION

This position paper presented a research to identify the lightweight cryptographic algorithms and their design components in terms of parameters, performance

Table 3: Hardware implementation performance metrics.

Metric	Definition
Area	The chip area occupied is measured in the gate equivalence (GE)
Area/bit	The area cost to a single bit
Throughput	Measures the average quantity of data which is processed during each clock cycle
Efficiency	The ratio of the throughput calculated at a fixed clock frequency over the area
FoM	FoM = Throughput/GE^2
FoAM	Combines the security provided by cryptographic structures and components with their implementation properties
Power	Indicates the rate of energy consumption. Power is dependent on clock frequency
Energy per bit	Normalizes energy with respect to the number of bits in a cipher block
Energy x Area/Bits	Combines the two constraints in one expression

and metrics. The objective is to contribute to a better understanding of the relationship between algorithm structures, evaluation metrics and implementations to improve the construction of lightweight algorithms.

Evaluating an appropriate lightweight cipher for a specific application is multidimensional issue and difficult to be completely characterized by isolated metrics like throughput or key size, and there is no consensus on which one is more appropriate.

Numerous metrics were mentioned and a collection of the presented metrics might be taken into consideration in order to scale down the device resource consumption without overlooking the provided security.

In this context, we concluded that there is a need for design a set of optimal metrics that will allow to propose suitable encryption algorithms to overcome IoT devices resource constraints. This set must also integrate qualitative dimensions in order to make proper comparisons. As a future work we aim to design this environment for performance evaluation.

New developments constantly emerge, with novel techniques and algorithms being proposed. Post-quantum cryptography research is an important field for IoT networks since LWC primitives and protocols are insecure against quantum attacks. Even though it is not our scope, we see it as an important future work.

This work intended to help researchers to improve IoT security by designing robust solutions and architectures for resource-constrained environments.

REFERENCES

Abujoodeh, M., Tamimi, L., and Tahboub, R. (2023). Toward lightweight cryptography: A survey. In Dekoulis, G. and Yadav, J., editors, *Computational Semantics*, chapter 6. IntechOpen, Rijeka.

Arora, A. (2012). A survey of cryptanalytic attacks on lightweight block ciphers.

Aswathy, R. and Nandagopal, M. (2021). A design of lightweight ecc based cryptographic algorithm coupled with linear congruential method for resource constraint area in iot. *Journal of Ambient Intelligence and Humanized Computing*, 14:1–10.

Aumasson, J.-P., Henzen, L., Meier, W., and Naya-Plasencia, M. (2010). Quark: A lightweight hash. In Mangard, S. and Standaert, F.-X., editors, *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Badel, S., Dagtekin, N., Nakahara, J., Ouafi, K., Reffé, N., Sepehrdad, P., Susil, P., and Vaudenay, S. (2010). Armadillo: A multi-purpose cryptographic primitive dedicated to hardware. volume 6225, pages 398–412.

Baysal, A. and Sahin, S. (2015). Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors. volume 9542.

Beaulieu, R., Treatman-Clark, S., Shors, D., Weeks, B., Smith, J., and Wingers, L. (2015). The simon and speck lightweight block ciphers. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6.

Biryukov, A. and Perrin, L. (2017). State of the art in lightweight symmetric cryptography. *IACR Cryptol. ePrint Arch.*, 2017:511.

Blanc, S., Lahmadi, A., Gouguec, K., Minier, M., and Sleem, L. (2022). Benchmarking of lightweight cryptographic algorithms for wireless iot networks. *Wireless Networks*, 28.

Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., and Verbauwhede, I. (2011). spongent: A lightweight hash function. In Preneel, B. and Takagi, T., editors, *Cryptographic Hardware and Embedded Systems – CHES 2011*, pages 312–325, Berlin, Heidelberg. Springer Berlin Heidelberg.

De Cannière, C. (2006). Trivium: A stream cipher construction inspired by block cipher design principles. In Katsikas, S. K., López, J., Backes, M., Gritzalis, S., and Preneel, B., editors, *Information Security*, pages 171–186, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dinu, D., Corre, Y., Khovratovich, D., Perrin, L., Großschädl, J., and Biryukov, A. (2019). Triathlon of lightweight block ciphers for the internet of things. *Journal of Cryptographic Engineering*, 9.

Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. (2021). Ascon v1.2: Lightweight authenticated encryption and hashing. *J. Cryptol.*, 34(3).

Duval, S. and Leurent, G. (2020). Lightweight macs from universal hash functions. In Belaïd, S. and Güneysu, T., editors, *Smart Card Research and Advanced Applications*, pages 195–215, Cham. Springer International Publishing.

Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., and Uhsadel, L. (2007). A survey of lightweight-cryptography implementations. *IEEE Design and Test of Computers*, 24(6):522–533.

Engels, D., Saarinen, M.-J. O., Schweitzer, P., and Smith, E. M. (2012). The hummingbird-2 lightweight authenticated encryption algorithm. In Juels, A. and Paar, C., editors, *RFID. Security and Privacy*, pages 19–31, Berlin, Heidelberg. Springer Berlin Heidelberg.

Farhan, L. and Kharel, R. (2019). *Internet of Things: Vision, Future Directions and Opportunities*, pages 331–347. Springer International Publishing, Cham.

Guo, J., Peyrin, T., and Poschmann, A. (2011). The photon family of lightweight hash functions. In Rogaway, P., editor, *Advances in Cryptology – CRYPTO 2011*, pages 222–239, Berlin, Heidelberg. Springer Berlin Heidelberg.

Harbi, Y., Aliouat, Z., Refoufi, A., and Harous, S. (2021). Recent security trends in internet of things: A comprehensive survey. *IEEE Access*, 9:113292–113314.

Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., and Manifavas, H. (2018). A review of lightweight block ciphers. *Journal of Cryptographic Engineering*, 8:1–44.

Khan, M. A., Quasim, M. T., Alghamdi, N. S., and Khan, M. Y. (2020). A secure framework for authentication and encryption using improved ecc for iot-based medical sensor data. *IEEE Access*, 8:52018–52027.

Kouicem, D. E., Bouabdallah, A., and Lakhlef, H. (2018). Internet of things security: A top-down survey. *Computer Networks*, 141:199–221.

Leander, G., Paar, C., Poschmann, A., and Schramm, K. (2007). New lightweight des variants. volume 4593, pages 196–210.

Lee, D., Kim, D.-C., Kwon, D., and Kim, H. (2014). Efficient hardware implementation of the lightweight block encryption algorithm lea. *Sensors*, 14(1):975–994.

Luykx, A., Preneel, B., Tischhauser, E., and Yasuda, K. (2016). A mac mode for lightweight block ciphers. In Peyrin, T., editor, *Fast Software Encryption*, pages 43–59, Berlin, Heidelberg. Springer Berlin Heidelberg.

Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., and Aharon, D. (2015). Unlocking the potential of the internet of things. *McKinsey Global Institute*, 1.

Matsui, M. and Murakami, Y. (2014). Minimalism of software implementation. pages 393–409.

Miller, V. S. (1986). Use of elliptic curves in cryptography. In Williams, H. C., editor, *Advances in Cryptology — CRYPTO '85 Proceedings*, pages 417–426, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., and Verbauwhede, I. (2014). Chaskey: An efficient mac algorithm for 32-bit microcontrollers. In Joux, A. and Youssef, A., editors, *Selected Areas in Cryptography – SAC 2014*, pages 306–323, Cham. Springer International Publishing.

Patel, S. T. and Mistry, N. H. (2015). A survey: Lightweight cryptography in wsn. In *2015 International Conference on Communication Networks (ICCN)*, pages 11–15.

Ågren, M., Hell, M., Johansson, T., and Meier, W. (2011). Grain-128a: a new version of grain-128 with optional authentication. *Int. J. Wire. Mob. Comput.*, 5(1):48–59.

Ranganatha Rao, B. and Sujatha, B. (2023). A hybrid elliptic curve cryptography (hecc) technique for fast encryption of data for public cloud security. *Measurement: Sensors*, 29:100870.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126.

Sehrawat, D. and Gill, N. (2019). A review on performance evaluation criteria and tools for lightweight block ciphers. *International Journal of Advanced Trends in Computer Science and Engineering*, 8:630–639.

Sevin, A. and Mohammed, A. (2021). A survey on software implementation of lightweight block ciphers for iot devices. *Journal of Ambient Intelligence and Humanized Computing*, 14:1–15.

Shi, T. and and, J. G. (2019). Cryptanalysis of the authentication in acorn. *KSII Transactions on Internet and Information Systems*, 13(8):4060–4075.

Shirai, T., Shibutani, K., Akishita, T., Moriai, S., and Iwata, T. (2007). The 128-bit blockcipher clefia (extended abstract). In Biryukov, A., editor, *Fast Software Encryption*, pages 181–195, Berlin, Heidelberg. Springer Berlin Heidelberg.

Singh, S., Sharma, P., Moon, S., and Park, J. (2017). Advanced lightweight encryption algorithms for iot devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, 15:1–18.

Stallings, W. (2013). *Cryptography and Network Security: Principles and Practice*. Prentice Hall Press, USA, 6th edition.

Thakor, V. A., Razzaque, M. A., and Khandaker, M. R. A. (2021). Lightweight cryptography algorithms for resource-constrained iot devices: A review, comparison and research opportunities. *IEEE Access*, 9:28177–28193.

Watanabe, D., Ideguchi, K., Kitahara, J., Muto, K., Furuichi, H., and Kaneko, T. (2008). Enocoro-80: A hardware oriented stream cipher. In *2008 Third International Conference on Availability, Reliability and Security*, pages 1294–1300.