Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 233 (2024) 235–242

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2024)

Analysis of Lightweight Cryptographic Algorithms for IoT Gateways

Ch. Jnana Ramakrishna[a], D. Bharath Kalyan Reddy[a], Priya B.K[a,*], Amritha P.P[b], Lakshmy K.V[b]

[a]*Department of Electronics and Communication Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India*
[b]*TIFAC - CORE in Cyber Security, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India*

bk/priya@blr.amrita.edu, chjramakrishna25@gmail.com, dbkrbharath@gmail.com, pp amritha@cb.amrita.edu, kv_lakshmy@cb.amrita.edu

Abstract

This research work explores the growing role of IoT devices in daily life and the increasing demand for secure data transmission in different sectors like healthcare, military, etc. Moreover, this study focuses on the recent advancements in lightweight cryptography and compares various encryption algorithms in an IoT environment using real-time data. Employing Message Queuing Telemetric Transport (MQTT) protocol, this research work assess the time of encryption, memory usage, and CPU usage as key parameters. By standardizing iterations, key width, and block length, this study aims to identify optimal encryption solutions for IoT applications.

Keywords: Internet of Things (IoT); Lightweight cryptography; Network Security; Machine-to-Machine protocols; Integrity

1. Introduction

The Internet of Things (IoT) is a distributed interconnection of devices embedded with sensors, computing power and software, allowing them to connect and share data with other devices and networks over the internet. Given IoT's widespread implementation, addressing security concerns becomes vital for protecting data from a variety of cyber attacks. There are several IoT applications and solutions. Its applications include smart watches, smart cities, smart homes, and industrial IoT [15]. The Cisco study projected that the IoT would encompass nearly 23 billion IoT devices by 2025 [4].

However, when it comes to effectively generating both dynamic and secure identification of devices with limited resources and their associated resources, this transforming landscape creates significant challenges. Enabling them to be intelligent involves cognitive capacities, which include logical decision-making, networking and memory [7].

10.1016/j.procs.2024.03.213

IoT devices have evolved into critical components for many organizations, municipalities, and urban environments, emphasizing the critical importance of protecting shared data. Furthermore, objects connected to IoT networks have severe resource constraints, such as constrained power, memory constraints, and processing capacity. As a result, implementing the complex arithmetic operations required by encryption algorithms on such devices is challenging. In the IoT environment, there is an urgent need for a security mechanism that is streamlined and resource-efficient. Traditional authentication and encryption methods place a significant demand on equipment, resulting in inefficiencies. Therefore, this research centres on developing a security approach characterized by its lightweight nature, aiming to both enhance communication within IoT networks and alleviate the resource utilization challenge on constrained IoT devices.

Numerous threats that pose risk to IoT devices have been categorized, encompassing network, physical, environmental, cryptanalysis and software-based attacks [5]. Within the realm of network attacks, several types have been identified, such as man-in-the-middle (MITM) assaults, replay and distributed denial of service (DDoS) attacks [19]. The utmost importance of security in IoT cannot be overstated, as a single successful attack has the capacity to bring an entire sector to a standstill, be it manufacturing, transportation, healthcare, or others.

Derived from these prerequisites, a communication stack arises as a necessity to furnish a protocol that is simultaneously energy-efficient, secure and lightweight. The IoT communication stack encompasses various protocol types. In this study, the focus was directed toward enhancing the security of IoT networks by encrypting data using lightweight ciphers and Message Queuing Telemetric Transport (MQTT) application layer protocol. The reason for choosing MQTT is that MQTT is constructed atop the TCP protocol, exhibiting minimal power consumption and a lighter overhead when compared to alternative IoT protocols [18]. Various ciphers are implemented in a network and are compared to get an analysis to understand which algorithm suits the environment perfectly.

The remaining part of the paper is as follows: Section 2 provides insights on earlier works about lightweight cryptography and IoT security, Section 3 gives details on the implementation of the proposed method, Section 4 discusses results and analysis and Section 5 gives the conclusion of the paper and future work.

2. Review of Earlier Works

2.1. Lightweight ciphers

2.1.1. HIGHT Cipher

A highly lightweight algorithm is employed, designed to handle 64-bit data through 32 iterations with a 128-bit key. This algorithm executes a streamlined round function devoid of S-boxes, relying on straightforward computational processes. The most condensed variant of this algorithm demands 2608 gate equivalents to achieve a throughput of 188 Kbps [9], [13].

2.1.2. Simon & Speck Ciphers

Simon represents a cluster of block ciphers, which encompass key sizes as 32, 48, 64, 96, 128 and 64, 72, 96/128, 96/144, 128/192 as block size respectively. The numerical labels in the titles denote the bit quantities for block size and key size. Employing a Feistel network architecture, Simon ciphers employ bitwise logical operations, rotations and XOR operations to establish both confusion and diffusion, as well as handle key expansion and encryption. Remarkably, despite their straightforward design, Simon ciphers offer a substantial level of security and efficiency, rendering them well-suited for devices with limited resources [1], [17].

Speck emerges as an additional series of compact block ciphers created by the same authors as Simon. In a manner similar to Simon, Speck ciphers are available in an assortment of dimensions, spanning 32/64, 48/72, 64/96, 64/128, 96/96, 96/144, 128/128 and 128/192. Constructed through a fusion of bitwise logical operations, rotations and XOR operations, Speck ciphers aspire to strike a harmonious equilibrium between security and efficiency, especially tailored for energy-efficient devices [1].

2.1.3. PRESENT Cipher

PRESENT employs a Substitution-Permutation Network (SPN) structure and operates on a 64-bit block, utilizing either 80-bit or 128-bit keys. The gate equivalents (GE) required for these variants are 1570 and 1886, respectively

[3]. Notably, PRESENT is designed with a focus on hardware efficiency, featuring 4-bit S-boxes within its substitution layer (which replaces eight S-boxes with a single one). In contrast, its software implementation (permutation layer) involves more extensive cycles, prompting the need for an enhanced version [2], [8].

2.1.4. RECTANGLE Cipher

RECTANGLE stands out as an exceptionally lightweight block cipher, adaptable for diverse applications. By making slight modifications to the SPN architecture, the number of rounds has been minimized to 25. This adjustment is aimed at ensuring competitiveness within the evolving landscape [20].

2.1.5. KLEIN Cipher

Klein, as detailed in [6], operates on a 64-bit input while employing 64-bit, 80-bit and 96-bit keys. These key sizes correspond to 12 (requiring 1220 gate equivalents), 16 (needing 1478 gate equivalents) and 20 (with 1528 gate equivalents) iterations, respectively. The design ethos behind Klein emphasizes software implementation, primarily targeting sensor-based applications.

2.1.6. Camellia Cipher

Camellia, garners acknowledgment from respected organizations including ISO/IEC, IETF, NESSIE and CRYPTREC. The collaborative effort of Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation, Camellia delivers a level of security on par with AES. This equivalence is attained using identical key and block sizes as AES, along with the availability of two round variants: 18 and 24 rounds. It's worth noting that Camellia is renowned for its efficient software implementations, even though its hardware implementation demands 6511 gate equivalents [16].

2.1.7. ITUbee Cipher

ITUbee stands as a cipher prioritizing software efficiency, featuring a compact code size of 586 bytes and requiring 2937 cycles for encryption. Operating with an 80-bit key and block size, ITUbee adopts a unique approach by substituting key scheduling with round-dependent constants. This substitution is aimed at diminishing software overhead while maintaining security [11].

2.2. Message Queuing Telemetric Transport Protocol (MQTT)

MQTT serves as an IoT connectivity protocol, operating atop the TCP protocol. Originally created by IBM to facilitate lightweight machine-to-machine (M2M) communications, MQTT gained recognition for its efficiency. Notably, in 2014, MQTT version 3.1.1 received approval as an OASIS standard and it was also standardized under ISO/IEC 20922 [10]. It operates as a message-focused protocol, embodying a publish/subscribe model of interaction where client devices are not obligated to actively request updates. This design minimizes the drain on node resources, making it particularly suitable for deployment on networks with high latency or unreliability. Employing a server/client architecture, the MQTT protocol designates the server as the broker and the scenario is shown in Figure 1. Notably, direct communication between clients is absent; all messages pass through the broker. Each message is associated with a specific topic and clients can subscribe to multiple topics. These topics are hierarchically organized through various levels. In this framework, the broker receives publish messages from a client and is responsible for forwarding them to all other clients subscribed to the relevant topic [14]. MQTT incorporates message buffers and Quality of Service (QoS) tiers that are managed by the broker. These QoS levels are:

- Level 0 – referred to as “at most once”, which can be thought of as a “fire and forget” approach.
- Level 1 – denoted as “at least once”, ensuring the message is delivered at least once.
- Level 2 – labelled as “exactly once”, guaranteeing the message is delivered precisely once.

To ensure QoS levels greater than zero, ACK (acknowledge) packets are utilized between the publisher and the broker. This mechanism provides a means of confirming the successful transmission of messages. Emphasizing MQTT as a lightweight protocol suitable for industrial use, authors of [12] conducted an evaluation of various security mechanisms aimed at protecting MQTT enabled wireless sensor motes. Additionally, they assessed the practicality of the

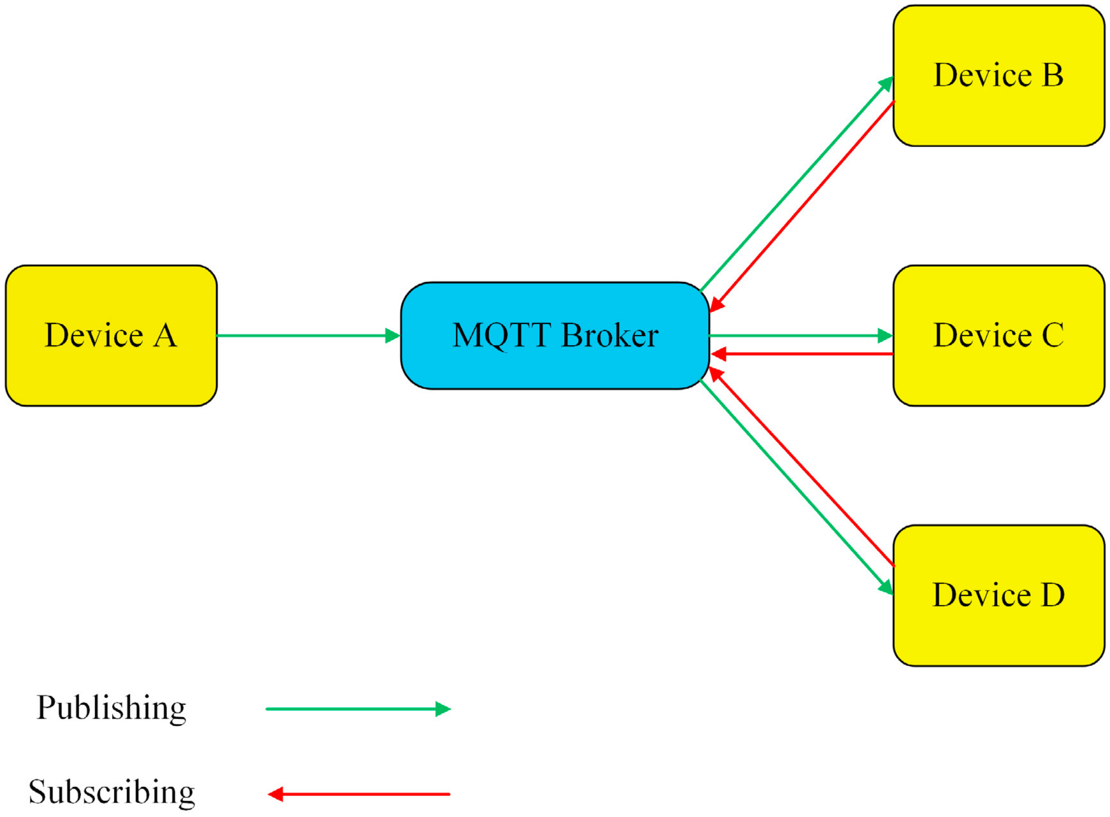

Fig. 1. MQTT working scenario

solutions within the context of a real-world industrial application, while also scrutinizing the associated network characteristics.

3. Proposed Method

A sensor is linked to a Raspberry Pi, serving as a data aggregator for the sensor's output. This accumulated data is logged at specific intervals and then subjected to encryption through the Lightweight cipher. The resultant encrypted file is then dispatched to a secondary Raspberry Pi, functioning as a gateway via MQTT protocol.

A gateway is a pivotal node facilitating communication between two networks that employ distinct transmission protocols. Within this gateway setup, the encrypted file undergoes decryption and partial data analysis. The analyzed data is once again encrypted before being forwarded to fog computing via the Internet Protocol (IP). The gateway essentially functions as a bridge, linking the Machine-to-Machine (M2M) protocol with the Internet Protocol.

Within the fog computing framework, the received file is decrypted and subjected to advanced data analysis. The results of this sophisticated analysis are then encrypted and transmitted to the cloud. This enables authorized personnel to access the data from any location. Figure 2 shows an ideal architecture of a real-time environment.

A comprehensive set of trails were undertaken to assess the performance of various ciphers across IoT environment and quantify their influence on the overall system operation. These experiments were conducted on actual daily basis applications, including a network of Raspberry Pi devices interconnected using Wi-Fi, situated within a well-regulated lab setting.

The Raspberry Pi Zero model with a 1GHz processor, 512 MB of RAM is used for testing. Total of three Raspberry Pis are used in the process, one Pi as MQTT broker and the other two Pi's as publisher and subscriber respectively.

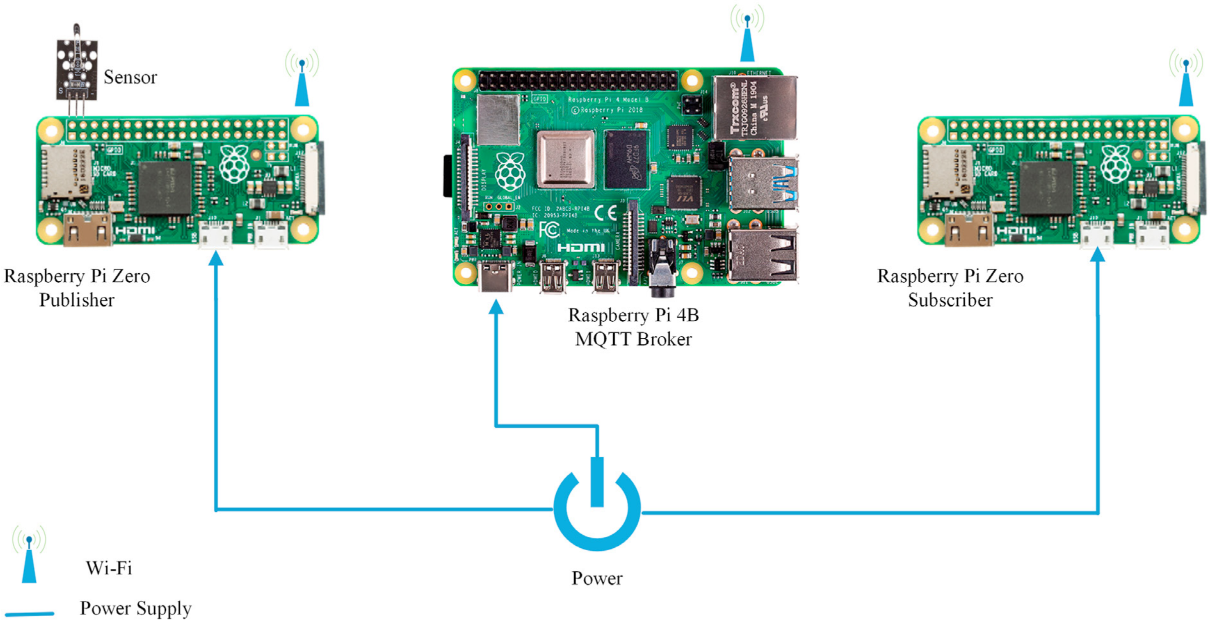

Fig. 2. Ideal Experiment Network

Table 1. Testing devices

Devices	Parameters	Quantity
Raspberry Pi Zero	Publisher & Subscriber	2
Raspberry Pi 4B (2GB RAM)	MQTT broker	1
COREMOD01 Sensor	Analog temperature sensor	1

All Pi's are connected to Wi-Fi, the analog temperature sensor of the COREMOD01 model is used for the collection of sensor data. Based on roundtrip timing, the collected results encompass the entire time necessary for encrypting, sending, receiving, decrypting and re-encrypting, sending it back and decrypting again. Collected time data helps in the understanding of encryption algorithms in an IoT environment. The plaintext data consists of a combination of alphanumeric and special characters. The data is segregated as 128, 512 and 1024 bytes of length for testing. Table 1 shows the configurations of hardware devices used in the experiment.

For the quantification of comparison parameters, a range of analyzer tools and software were utilized. The PERF tool was employed to monitor the time taken for the completion of the roundtrip. Utilization of the NMON tool was implemented to track CPU utility. In parallel, the Wireshark was utilized for packet monitoring. On the Pi entities functioning as MQTT publishers/subscribers, the Paho MQTT software was installed. Meanwhile, the Raspberry Pi 4B was equipped with Mosquitto, serving as the MQTT broker. MQTT protocol is used for the transmission of data from one device to another in an IoT framework. Following the completion of all trials, an analysis was conducted. Data collected is transmitted from gateway to cloud, where it can be accessed from any location.

4. Results and Analysis

The suggested implementation was executed within a real-world setting. For each cipher, the process involved sending messages of 128 bytes in length from the publisher to the subscriber through the broker. This sequence was repeated for 25 iterations for each cipher and the resulting averages were computed from these trials.

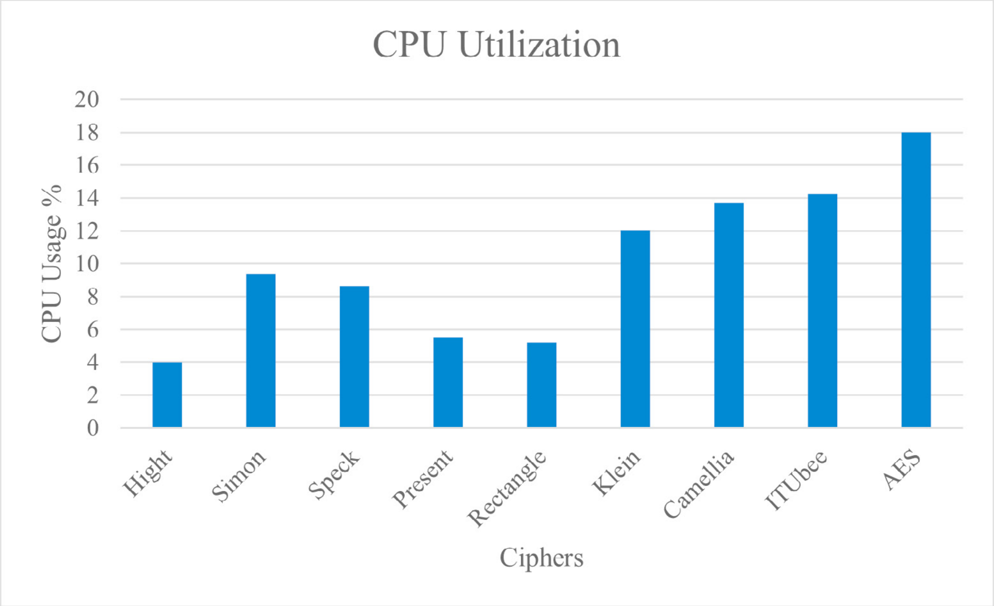

Fig. 3. Average CPU usage

The evaluation encompassed performance metrics such as CPU usage, RAM usage and roundtrip time. These metrics were assessed across all the ciphers under consideration.

4.1. CPU Usage

The CPU utilization percentages associated with each cipher is displayed in Figure 3. Observing Figure 3, it becomes evident that AES engaged 18% of the CPU, given its extensive internal computations for encryption. In contrast, the HIGHT cipher demonstrated the most frugal CPU consumption at a mere 4%, attributed to its lightweight characteristics.

4.2. RAM Usage

The graph depicted in Figure 4 indicates that the ITUbee cipher utilizes the most RAM, consuming 12.4KB. In comparison, both the PRESENT and RECTANGLE ciphers require around 6KB of RAM for their execution.

4.3. Roundtrip Time

This analysis provides the mean duration encompassing encryption, transmission, reception, decryption, re-encryption, return transmission and final decryption for a single message. Figure 5 visually presents the average time taken. Specifically, AES exhibited an average time of 38.77 milliseconds, whereas HIGHT, PRESENT and RECTANGLE ciphers demonstrated significantly quicker durations of 2.83, 3.57 and 3.03 milliseconds, respectively. Data transmitted has been received without any corruption.

5. Conclusion

We examined diverse lightweight encryption methods in a real-world context using Raspberry Pi devices and subsequently, we compared their suitability for the presented IoT environment. Our analysis revealed that HIGHT, PRESENT and RECTANGLE ciphers exhibit quicker execution times, in accordance with the considered parameters.

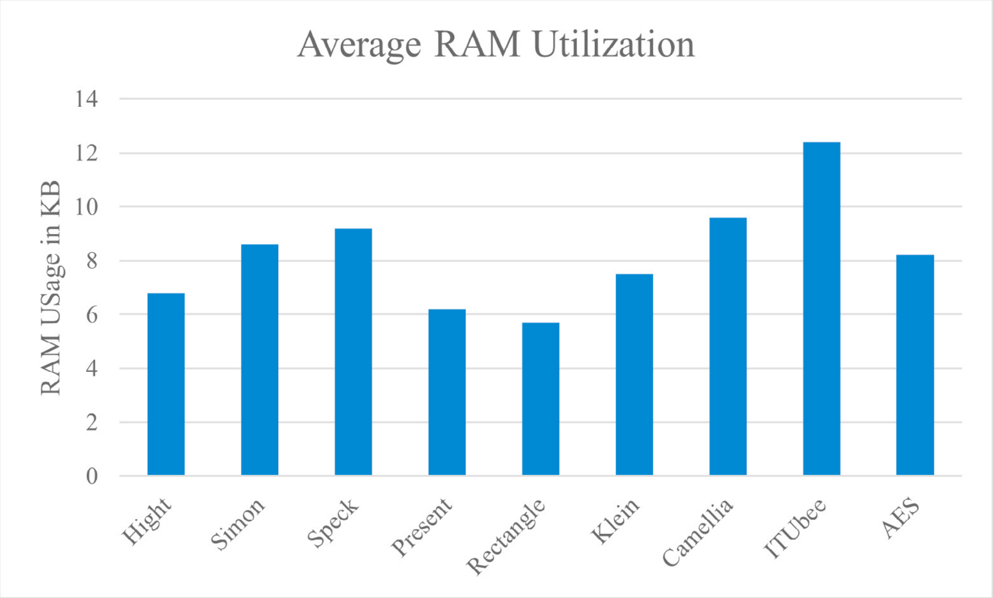

Fig. 4. Average RAM usage

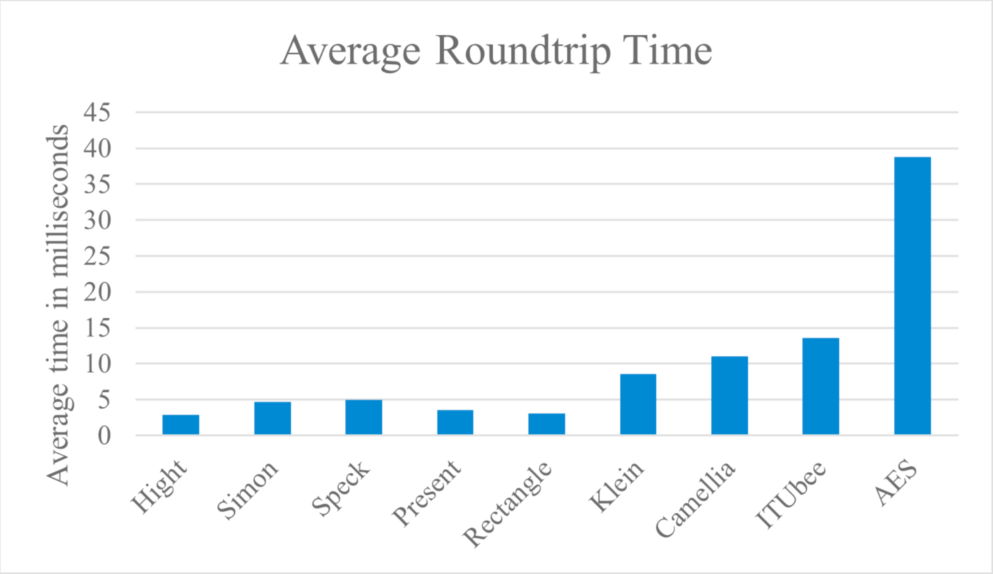

Fig. 5. Average time of roundtrip

In contrast, AES demonstrated a more intricate nature due to its extensive internal computations, which could render its implementation in IoT challenging due to device constraints and specifications.

Researchers need to ensure the efficiency and compatibility of cryptographic algorithms before integrating them into IoT. Future work involves investigating the performance and measures of consolidating multiple nodes into a single gateway or broker, with the aim of enhancing the overall performance of IoT systems. In addition to lightweight cipher, there are Authenticated Encryption schemes which are also known as AHEAD which performs device authentication as well as data encryption simultaneously. These methods are required when multiple entities are performing numerous data transmissions across the IoT network. The integration of enhanced IoT frameworks with authenticated encryption algorithms, coupled with advanced M2M protocols, results in a streamlined and well-protected IoT network.

References

[1] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L., 2013. The SIMON and SPECK families of lightweight block ciphers. *cryptology* eprint archive .

[2] Bhardwaj, I., Kumar, A., Bansal, M., 2017. A review on lightweight cryptography algorithms for data security and authentication in iots, in: *4th International Conference on Signal Processing, Computing and Control* (ISPCC), IEEE. pp. 504–509.

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

[4] Evans, D., 2011. The internet of things. *How the Next Evolution of the Internet is Changing Everything, Whitepaper, Cisco Internet Business Solutions Group*(IBSG) 1, 1–12.

[5] Gloukhovtsev, M., 2018. IoT security: challenges, solutions & future prospects. *Proceedings of the Proven Professional Knowledge Sharing Article* , 1–44.

[6] Gong, Z., Nikova, S., Law, Y.W., 2011. KLEIN: a new family of lightweight block ciphers, in: *International workshop on radio frequency identification: security and privacy issues*, Springer. pp. 1–18.

[7] Gupta, V., Khera, S., Turk, N., 2021. MQTT protocol employing IOT based home safety system with ABE encryption. *Multimedia Tools and Applications* 80, 2931–2949.

[8] Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., Manifavas, C., 2018. A review of lightweight block ciphers. *Journal of cryptographic Engineering* 8, 141–184.

[9] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.S., Lee, C., Chang, D., Lee, J., Jeong, K., et al., 2006. HIGHT: A new block cipher suitable for low-resource device, in: *Cryptographic Hardware and Embedded Systems*-CHES 2006: 8th International Workshop, Yokohama, Japan, Proceedings 8, Springer. pp. 46–59.

[10] ISO/IEC, 2016. Information technology – Message Queuing Telemetry transport (MQTT).

[11] Karakoç, F., Demirci, H., Harmancı, A.E., 2013. ITUbee: a software oriented lightweight block cipher, in: *Lightweight Cryptography for Security and Privacy: Second International Workshop*, LightSec 2013, Gebze, Turkey, Revised Selected Papers 2, Springer. pp. 16–27.

[12] Katsikeas, S., Fysarakis, K., Miaoudakis, A., Van Bemten, A., Askoxylakis, I., Papaefstathiou, I., Plemenos, A., 2017. Lightweight & secure industrial IoT communications via the MQ telemetry transport protocol, in: *IEEE Symposium on Computers and Communications* (ISCC), pp. 1193–1200.

[13] Lim, Y.I., Lee, J.H., You, Y., Cho, K.R., 2009. Implementation of HIGHT cryptic circuit for RFID tag. *IEICE Electronics Express* 6, 180–186.

[14] Naik, N., 2017. Choice of effective messaging protocols for iot systems: MQTT, CoAP, AMQP and HTTP, in: *IEEE international systems engineering symposium* (ISSE), IEEE. pp. 1–7.

[15] Oracevic, A., Dilek, S., Ozdemir, S., 2017. Security in internet of things: A survey, in: *international symposium on networks, computers and communications* (ISNCC), IEEE. pp. 1–6.

[16] Satoh, A., Morioka, S., 2003. Hardware-focused performance comparison for the standard block ciphers aes, camellia, and triple-des, in: *Information Security: 6th International Conference*, ISC 2003, Bristol, UK, Proceedings 6, Springer. pp. 252–266.

[17] Shanmugam, D., Selvam, R., Annadurai, S., 2014. Differential power analysis attack on SIMON and LED block ciphers, in: *Security, Privacy, and Applied Cryptography Engineering*, Springer International Publishing, Cham. pp. 110–125.

[18] Soni, D., Makwana, A., 2017. A survey on mqtt: a protocol of internet of things (iot), in: *International conference on telecommunication, power analysis and computing techniques* (ICTPACT-2017), pp. 173–177.

[19] Yang, Y., Wu, L., Yin, G., Li, L., Zhao, H., 2017. A survey on security and privacy issues in internet-of-things. *IEEE Internet of things Journal* 4, 1250–1258.

[20] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I., 2014. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Cryptology ePrint Archive .