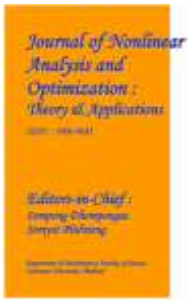
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 2 : 2024
ISSN :**1906-9685**



SECURE AND EFFICIENT COMMUNICATION FOR INTERNET OF THINGS USING LIGHT WEIGHT CRYPTOGRAPHY AND MQTT PROTOCOL

[1] **SHAIK MAHAMMED HANEEF,** PG SCHOLAR IN DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES, KADAPA, A.P. mahammedhaneefshaik8@gmail.com
[2] **Dr. C. VENKATA SUBBAIAH,** ASSOCIATE PROFESSOR, HEAD OF DEPARTMENT IN DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, ANNAMACHARYA INSTITUTE OF TECHNOLOGY AND SCIENCES, KADAPA, A.P. cvenkatasubbaiah@gmail.com

Abstract

The Internet of Things (IoT) is rapidly growing and silently transforming many industries achieving seamless communication between devices. IoT has become an integral part of our daily lives with a wide range of applications from smart phones to smart home systems. As these IoT devices are efficient in their operation, but they are not secure. The security is a major concern as they come with limited computing, memory and power resources. These limitations make the Internet of Things, a resource constrained environment. This paper proposes a method that combines light weight cryptographic algorithms like AES, PRESENT, SPECK, RECTANGLE along with MQTT protocol providing robust data encryption and decryption with secure and efficient data transmission. This combination enhances the overall security and performance of IoT systems by addressing the limitations of resource constrained environments.

Keywords: IoT, Lightweight Cryptography, DTC, GSC, MQTT, AES, PRESENT, RECTANGLE, SPECK, MQTT.

1. INTRODUCTION

The Internet of Things is a ubiquitous computing technology that has devices with various sensors, computing power, software and other technologies, all interconnected through the internet to enable seamless communication. These devices come with variety of sensors which collect wide range of data from fitness trackers to smart home systems. This rapid growth of IoT has led to a large number of interconnected devices which make them insecure and vulnerable to various threats. As these devices come with limited computing power, memory and battery life, the traditional cryptographic algorithms are very inefficient and not suitable to implement in IoT Systems. Hence, there is a need to look for light weight cryptographic algorithms to provide security for IoT devices.

This paper proposes the combination of lightweight cryptographic algorithms like AES, PRESENT, SPECK, and RECTANGLE along with Message Queuing Telemetry Protocol (MQTT) to provide secure and efficient communication for IoT devices. This approach makes the efficient use of low computing, memory and power resources of IoT devices.

2. RELATED WORK

In this section, we discuss about the existing systems and algorithms that were proposed to address security issues in the IoT environments. We highlight the key findings and limitations of the existing system that motivates us to proposed approach.

The resource constrained environment of IoT devices has led to usage of algorithms like Dynamic Tree Chaining (DTC) and Geometric Star Chaining (GSC) algorithms. These algorithms are used to efficiently utilize the less resources of IoT devices and provide secure communication. The Dynamic Tree Chaining (DTC) is uses hierarchal tree structure for communication. Even though DTC reduces high utilization of resources, it is still vulnerable to various attacks and has delay issues. The Geometric Star Chaining (GSC) algorithm is uses star topology to interconnect IoT devices and provide secure communication. The GSC algorithm reduces the delay and complexity of the data communication but is still energy inefficient as it utilizes high amount of resources and it still vulnerable to DDOS attacks. The Traditional Cryptographic algorithms require high computational power. Hence, it is difficult to implement traditional cryptographic algorithms in IoT devices due to the limited resources.

The IoT systems are vulnerable to attacks like:

1. **Eavesdropping attack**: In this attack, the attacker is able to intercept the data that is being transmitted. These usually include sensitive information like passwords, authentication credentials etc.
2. **Replay attack:** In this attack, the attacker is able to intercept the data and transmit the same data again to the receiver.
3. **Man in the Middle attack:** In this attack, the attacker is able to intercept the data between the transmitter and the receiver.
4. **Denial of Service attack:** In this attack, the attacker is able to fool the IoT device with malicious traffic thereby making the device inaccessible to legitimate users.

To address these issues, it is necessary to implement light weight cryptographic algorithms that can provide secure and efficient communication while utilizing minimum resources.

3. PROPOSED APPROACH

To address the security and performance challenges of IoT systems, we propose a combination of light weight cryptographic algorithms and MQTT protocol.

3.1 Light Weight Cryptographic Algorithms

The lightweight cryptographic algorithms that we discuss in this paper are AES, PRESENT, SPECK and RECTANGLE.

3.1.1 Lightweight Advanced Encryption Standard (AES)

The AES algorithm is a widely used symmetric-key algorithm for encryption and decryption. The AES algorithm has 3 different key lengths: 128-bit, 192-bit and 256-bit. The lightweight variant of AES is used for most of the IoT applications as it guarantees good security and performance in resource constrained IoT devices.

Algorithm 1: Encryption in Lightweight AES

Input: Plaintext P, Key K
Output: Ciphertext C
Data: RK: Round Keys (derived from Key K), S: State of Decryption
Begin
Expand the Key K into multiple Round Keys RK_0, RK_1, …, RK_{10}
$S \leftarrow P \oplus RK_0$
Perform 9 Rounds (1<= i <= 9)
Replace each byte in S using the S-Box
Perform left cyclic shifts on rows of S
Mix columns of S using Galois field multiplication
$S \leftarrow S \oplus RK_i$
Final Round
Replace each byte in S using the S-Box
Perform left cyclic shifts on rows of S
$S \leftarrow S \oplus RK_{10}$
Output: C←S
End

Algorithm 2: Decryption in Lightweight AES

Input: Ciphertext C, Key K
Output: Plaintext P
Data: RK: Round Keys (derived from K), S: Current state of the decryption process
Begin
Expand the Key K into multiple Round Keys RK_0, RK_1, …, RK_{10}
$S \leftarrow C \oplus RK_{10}$
Perform 9 Rounds ($1 \leq i \leq 9$):
Perform right cyclic shifts on rows of S
Replace each byte in S using the Inverse S-Box
$S \leftarrow S \oplus RK_{10-i}$
Reverse column mixing of S using Galois field multiplication
Final Round:
Perform right cyclic shifts on rows of S
Replace each byte in S using the Inverse S-Box
$S \leftarrow S \oplus RK_0$
Output: P←S
End

3.1.2 PRESENT

PRESENT is a lightweight block cipher that has 31 round Substitution-Permutation Network structure. The key length can be 80-bit or 128-bit. The PRESENT algorithm runs efficiently in hardware constrained environments like IoT devices.

Algorithm 3: Encryption in PRESENT

Input: 64-bit plaintext P, 80-bit key K
Output: 64-bit ciphertext C
Key Schedule: Generate 32 round keys RK_0, RK_1, …, RK_{31} from the 80-bit key K
Begin
Initial XOR
$S \leftarrow P \oplus RK_0$
Perform 31 Rounds ($1 \leq i \leq 31$):
Apply the S-Box substitution to each 4-bit nibble of S
Apply the Permutation to rearrange the bits of S
XOR S with the round key RK_i
Final Round:
Apply the S-Box substitution to each 4-bit nibble of S
XOR S with the last round key RK_{31}
(No permutation is applied in this step)
Output: $C \leftarrow S$
End

Algorithm 4: Decryption in PRESENT

Input: 64-bit ciphertext C, 80-bit key K
Output: 64-bit plaintext P
Key Schedule: Generate 32 round keys RK_0, RK_1, …, $RK_{31}$from the 80-bit key K
Begin
Reverse Final Round
$S \leftarrow C \oplus RK_{31}$
Apply the inverse S-Box substitution to each 4-bit nibble of S
Reverse 31 Rounds ($31 \geq i \geq 1$):
XOR S with the current round key RK_i
Apply the inverse Permutation to rearrange the bits of S
Apply the inverse S-Box substitution to each 4-bit nibble of S
Final XOR:
$S \leftarrow S \oplus RK_0$
Output: $P \leftarrow S$
End

3.1.3 SPECK:

SPECK is a lightweight block cipher that uses an Feistel network structure. SPECK offers different block sizes and key lengths. SPECK is efficient in both hardware and software environments.

Algorithm 5: Encryption and Decryption in SPECK

Input: Key K, word size w, rotation parameters α, β, number of rounds r
For encryption: Plaintext block P = (x, y)
For decryption: Ciphertext block C = (x, y)
Output: Encrypted block C (for encryption), Decrypted block P (for decryption)

Key Generation:
Generate r round keys RK_0, RK_1, …, RK_{r-1} using:
Perform circular rotations on the key.
Use XOR and modular addition operations to derive the round keys.

Encryption:
Input: Plaintext block P = (x, y), round keys RK_0, RK_1, …, RK_{r-1}
For each round i ($0 \leq i < r$):
Rotate x right by α, add y (mod 2^w), and XOR the result with RK_i
$x \leftarrow ((x \gg \alpha) + y) \oplus RK_i$
Rotate y left by β, XOR the result with the updated x:
$y \leftarrow (y \ll \beta) \oplus x$
Output: Encrypted block C = (x, y)

Decryption:
Input: Ciphertext block C = (x, y), round keys RK_0, RK_1, …, RK_{r-1}
For each round i in reverse ($r-1 \geq i \geq 0$):
Reverse XOR on y, then rotate y right by β:
$y \leftarrow ((y \oplus x) \gg \beta)$.
Reverse XOR on x, subtract y (mod 2^w), and rotate x left by α:
$x \leftarrow (((x \oplus RK_i) - y) \ll \alpha)$.
Output: Decrypted block P = (x, y)

3.1.4 RECTANGLE

RECTANGLE is a lightweight block cipher that has 25 round Substitution-Permutation Network structure. The key length can be 80-bit or 128-bit. RECTANGLE is designed to run efficiently in resource constrained environments.

Algorithm 6: Encryption and Decryption in SPECK

Input: Plaintext P, Key K, Number of Rounds R, SBOX, SBOX_INV (Substitution boxes for encryption and decryption)**,** P-layer, P-layer inverse (Permutation functions)
Output: Encrypted Ciphertext C, Decrypted Plaintext P

Key Schedule:
Rotate the key K
Extract round keys RK_0, RK_1, …, RK_{R-1}
Generate 25 round keys for the entire encryption and decryption process

Encryption:
Input: Plaintext P, Round keys RK_0, RK_1, …, RK_{R-1}
Initialize the state $S \leftarrow P$
For each round i ($0 \leq i < R-1$):
XOR the state with the round key:
$S \leftarrow S \oplus RK_i$
Substitute nibbles using SBOX
Permute the state using the P-layer
Final round:
XOR the state with the last round key:
$S \leftarrow S \oplus RK_{R-1}$
Apply SBOX substitution again
Output: Ciphertext $C \leftarrow S$

Decryption:
Input: Ciphertext C, Round keys RK_0, RK_1, …, RK_{R-1}
Initialize the state $S \leftarrow C$
Reverse the final round:
Substitute nibbles using SBOX_INV.
XOR the state with the last round key:
$S \leftarrow S \oplus RK_{R-1}$
For each round i in reverse ($R-2 \geq i \geq 0$):
Reverse the P-layer permutation using the P-layer inverse
Substitute nibbles using SBOX_INV
XOR the state with the round key:
$S \leftarrow S \oplus RK_i$
Output: Plaintext $P \leftarrow S$
Note: If multiple blocks need to be processed, repeat the encryption or decryption steps for each block.

3.2 Message Queuing Telemetry Transport Protocol

MQTT Protocol is lightweight network protocol that uses publish subscribe method. It is designed for resource constrained IoT devices and unreliable networks with low-bandwidth, high-latency. MQTT Protocol provides secure communication using TLS/SSL protocol.

The MQTT protocol has three main components:
a) Publisher: The Publisher is a device or application that publishes message to the broker.
b) Subscriber: The Subscriber is a device or application that receives the messages from the broker.
c) Broker: The Broker is an intermediary that receives all messages from publishers and route them to the relevant subscribers.

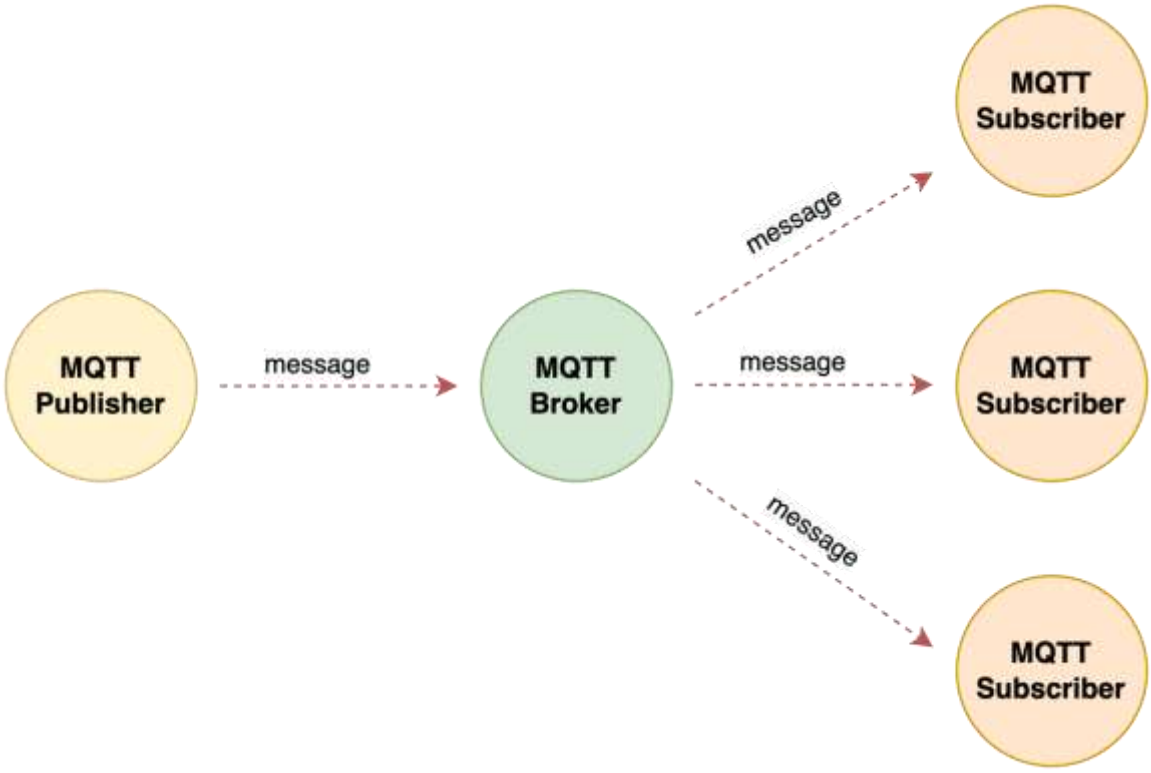


Fig.1 Message Transmission between Publisher, Broker and Subscriber in MQTT Protocol

These following features make the MQTT Protocol best suitable for resource constrained IoT Environments.

1. MQTT has a small overhead which makes it best suitable for resource constrained IoT environments.
2. MQTT ensure the reliable delivery of messages.
3. MQTT supports one-to-many publish-subscribe messaging pattern, making it suitable for IoT applications.

3.3 Working of Proposed Approach

The proposed approach combines the lightweight cryptographic algorithms and MQTT protocol to provide secure and efficient communication for IoT systems.

The steps involved are:

1. The IoT devices encrypt the sensor data using one of the lightweight cryptographic algorithms like AES, PRESENT, RECTANGLE, SPECK.
2. The encrypted data is then published to the MQTT broker using the MQTT protocol.
3. The publisher sends the data to the MQTT broker using an MQTT topic.
4. The subscriber receives the encrypted data from the MQTT broker and decrypts it using the same lightweight cryptographic algorithm.
5. The decrypted data is then used by the subscriber application.

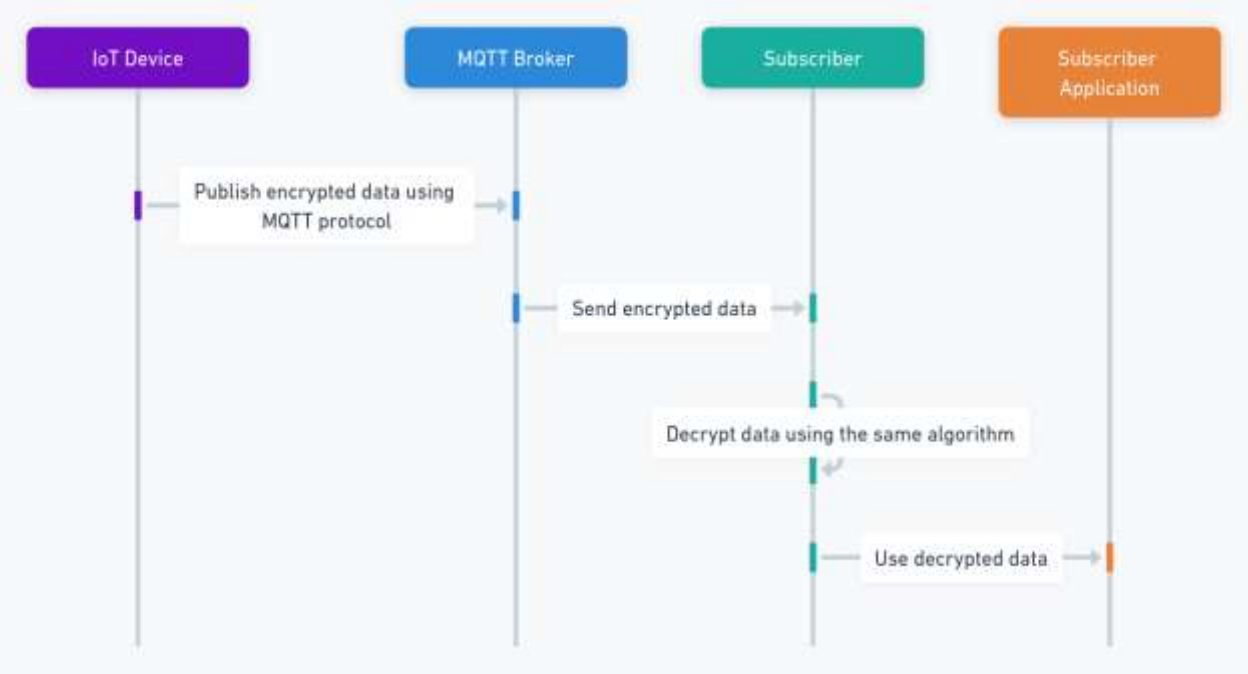


Fig.2 Secure Data Encryption, Decryption and Transmission using Lightweight cryptographic algorithms and MQTT protocol

4. RESULTS AND DISCUSSION

The proposed approach of using lightweight cryptographic algorithms and MQTT protocol has the following advantages:

1. The use of lightweight cryptographic algorithms provides confidentiality and integrity of the IoT sensor data.
2. The lightweight cryptographic algorithms have low computational and memory requirements, making them suitable for resource constrained IoT devices.
3. The MQTT protocol is also lightweight in nature, reducing the overall overhead on the IoT devices.

The simulation results show that the proposed approach can provide secure and efficient communication for IoT systems with minimal overhead on the IoT devices.

Algorithm	**Confidentiality and Integrity**	**Computational Requirements**	**Memory Requirements**	**Suitability for IoT**
AES	High	Moderate	Moderate	Suitable
PRESENT	High	Low	Low	Highly Suitable
RECTANGLE	High	Low	Low	Highly Suitable
SPECK	High	Low	Low	Highly Suitable

Table 1: Performance Analysis of Lightweight Cryptographic Algorithms

Metric	**Proposed Approach**	**Advantage**
Computational Overhead	Minimal	Efficient for resource-constrained IoT

Memory Usage	Low	Supports lightweight IoT devices
Protocol Overhead	Minimal	Reduced with MQTT
Communication Security	High	Ensures confidentiality and integrity
Overall Efficiency	Secure and efficient communication system	Suitable for IoT applications

Table 2: Simulation Results and Protocol Efficiency of Proposed Approach

5. CONCLUSION

In this paper, we have proposed a secure and efficient communication solution for IoT systems by combining lightweight cryptographic algorithms and the MQTT protocol. The use of lightweight cryptographic algorithms like AES, PRESENT, RECTANGLE, and SPECK provides confidentiality and integrity of the IoT sensor data, while the MQTT protocol enables scalable and reliable data transmission. The simulation results demonstrate the effectiveness of the proposed approach in providing secure and efficient communication for IoT applications.

REFERENCES

1. X. Li, M. Wang, H. Wang, Y. Yu and C. Qian, "Toward Secure and Efficient Communication for the Internet of Things," in IEEE/ACM Transactions on Networking, vol. 27, no. 2, pp. 621-634, April 2019.
2. El-hajj, M., Mousawi, H., & Fadlallah, A. (2023). Analysis of Lightweight Cryptographic Algorithms on IoT Hardware Platform. Future Internet, 15(2), 54.
3. Yarali, Abdulrahman, Manu Srinath, and Randal G. Joyce. "A Study of Various Network Security Challenges in the Internet of Things (IoT)." (2018).
4. Sri Ramya Siraparapu, S.M.A.K. Azad, Securing the IoT Landscape: A Comprehensive Review of Secure Systems in the Digital Era, e-Prime - Advances in Electrical Engineering, Electronics and Energy, Volume 10, 2024, 100798, ISSN 2772-6711.
5. Ch. Jnana Ramakrishna, D. Bharath Kalyan Reddy, B.K Priya, P.P Amritha, K.V Lakshmy, Analysis of Lightweight Cryptographic Algorithms for IoT Gateways, Procedia Computer Science, Volume 233, 2024, Pages 235-242, ISSN 1877-0509.
6. Radhakrishnan, I., Jadon, S., & Honnavalli, P. B. (2024). Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. Sensors, 24(12), 4008.
7. O. Sadio, I. Ngom and C. Lishou, "Lightweight Security Scheme for MQTT/MQTT-SN Protocol.
8. Thakor, V. A., Razzaque, M. A., & Khandaker, M. R. A. (2020, June 24). Lightweight Cryptography for IoT: A State-of-the-Art. arXiv.org.
9. A. Baneasa, R. Donca, S. Besoiu and D. Buleandra, "Lightweight Implementation of the AES Encryption Algorithm for IoT Applications Constrained by Memory and Processing Power," 2024 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 2024.
10. Sleem, Lama & Couturier, Raphaël. (2021). Speck-R: An ultra light-weight cryptographic scheme for Internet of Things. Multimedia Tools and Applications.
11. A. A. Zakaria, A. H. Azni, F. Ridzuan, N. H. Zakaria and M. Daud, "Extended RECTANGLE Algorithm Using 3D Bit Rotation to Propose a New Lightweight Block Cipher for IoT," in IEEE Access
12. A. Mhaouch, W. Elhamzi, A. B. Abdelali and M. Atri, "Efficient Serial Architecture for PRESENT Block Cipher," 2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Hammamet, Tunisia, 2022.