[1]Dr. Rupali Atul Mahajan

[2]Dr. Rupesh G. Mahajan

[3]Dr. Manjusha Tatiya

[4]Dr. Ujjwala Hemant Mandekar

[5]Minal Shahakar

[6]Dr. Yogendra Patil

Enhancing MQTT Security in the Internet of Things with an Enhanced Symmetric Algorithm

Abstract: - The Internet of Things (IoT), which connects billions of gadgets to expedite operations and enhance our lives, has completely changed the way we interact with our environment. With MQTT (Message Queuing Telemetry Transport) emerging as a popular communication protocol within the IoT ecosystem, the vast proliferation of networked devices has, however, presented serious security challenges. In order to strengthen MQTT security, this study suggests using an improved symmetric algorithm.Existing MQTT implementations frequently rely on simple security safeguards, making them susceptible to dangers like data manipulation, eavesdropping, and unauthorised access. Our research presents a novel symmetric algorithm designed to meet the particular needs of MQTT communication as a defence against these weaknesses. In order to protect the confidentiality and integrity of data transferred between IoT devices and brokers, this algorithm provides powerful encryption mechanisms. Additionally, it optimises resource usage to take into account the limitations of IoT devices, which frequently have constrained computational and memory resources.

Keywords: MQTT, HMAC (Hashed Message Authentication Code), Privacy, Data Integrity, IoT (Internet of Things)

I. INTRODUCTION

In the fast evolving Internet of Things (IoT) environment, which is home to billions of linked devices, security is essential. MQTT (Message Queuing Telemetry Transport), a popular protocol, has emerged to make it simpler for IoT devices to connect with centralised brokers. But as the IoT ecosystem expands, so do the security problems it faces. New methods are therefore required to increase the security of MQTT and, consequently, the entire IoT ecosystem. By enabling devices to gather and exchange data, automate procedures, and make wise decisions in real time, the Internet of Things is revolutionising businesses and everyday lives. However, this interconnectedness could bring forth a lot of security concerns. IoT devices and MQTT connections are susceptible to a number of dangers, including unauthorised access, data breaches, eavesdropping, and message tampering. The integrity, availability, and availability of data shared via IoT networks must therefore be ensured. In the past, SSL/TLS encryption and username/password authentication have been the pillars of MQTT security. Better defences are required considering how complex cyber threats are evolving, even though existing solutions offer a basic level of security. In this situation, the idea of an Enhanced Symmetric Algorithm (ESA) is relevant.

[1]Associate professor & Head, Data Science Department, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India

[2]Dr.D.Y.Patil Institute of Technology, Pimpri, Pune, Maharashtra, India

[3]Indira College of Engineering and Management, Pune, Maharashtra, India

[4]Department of Computer Engineering, Government Polytechnic, Nagpur, Maharashtra, India

[5]Assistant Professor, Dept. of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India

[6]Marathwada Mitra Mandal Institute of Technology, Lohagan, Pune, Maharashtra, India.

rupali.mahajan@viit.ac.in[1], mhjn.rpsh@gmail.com[2], manjusha.tatiya@indiraicem.ac.in[3], ujjwalaaher@gmail.com[4], mhjn.minal@gmail.com[5], patyogendra@gmail.com[6]

As described in this talk, the Enhanced Symmetric Algorithm is a brand-new and all-inclusive method for enhancing MQTT security in the Internet of Things. Using this method, MQTT brokers and IoT devices can establish an encrypted communication infrastructure. Key exchange, encryption, message authentication, and Perfect Forward Secrecy (PFS) are some of the important security issues it addresses. The Diffie-Hellman key exchange protocol, which facilitates secure key negotiation between devices and brokers, is one of the crucial elements of the proposed ESA. Eavesdropping and man-in-the-middle attacks will be possible if this phase is successful. After secure keys have been secured, a powerful symmetric encryption method, such as the Advanced Encryption Standard (AES), is employed to protect the anonymity of MQTT communications. When AES is employed in block cypher mode, pattern-based attacks are prevented since each message is encrypted with a different Initialization Vector (IV). Data integrity is another crucial issue that is addressed by the ESA's inclusion of HMAC (Hash-based Message Authentication Code). It is practically hard for hostile actors to tamper with payloads undetected since each MQTT message is accompanied by a MAC calculated using a shared secret key. To further protect against message replay threats, the ESA includes a secure hash algorithm like SHA-256.The ESA's Perfect Forward Secrecy (PFS) feature is intended to reduce the danger of long-term key compromise. Even if a session key is compromised, an attacker won't be able to decrypt current or past MQTT communications. The resilience of MQTT in the face of numerous cyber attacks is greatly increased by the addition of this layer of protection.

The suggested Enhanced Symmetric Algorithm aims to offer a strong defence against the changing IoT threat landscape by integrating these cutting-edge cryptographic techniques into a comprehensive security solution. By creating a secure foundation for MQTT connections, it hopes to reduce the danger of unauthorised access while upholding data security and integrity.In the parts that follow, we will go into more detail about the Enhanced Symmetric Algorithm's technical features, highlighting its essential elements and showing how it may be easily included into MQTT to increase security. We will also look at actual implementation issues and potential advantages of using this enhanced security strategy in IoT contexts.IoT has enormous potential for innovation and efficiency across industries, but this potential can only be fully realised in conjunction with strong security measures. A strong solution to improve MQTT security is provided by the Enhanced Symmetric Algorithm, which also protects the IoT ecosystem and promotes trust in the dependable and secure interchange of data between connected devices and brokers.

The Industrial Internet of Things (IIoT) and the Internet of Things (IoT) have brought about a transformative era, offering a wide range of applications across many domains, from environmental sensors and automobiles to remotely controlled actuators, home appliances, medical devices, and industrial equipment [1]. A astounding 20.4 billion IoT devices are predicted to be connected by the end of 2022. While this developing ecosystem is full of potential, it also poses a number of fresh difficulties, notably with regard to the privacy and security of its communications.The MQTT (Message Queuing Telemetry Transport) protocol, well known for its applicability on resource-constrained, low-power, affordable, and memory-limited devices, is one of the well-known networking protocols specialised to the IoT landscape. New capabilities have been added to MQTT, and it has standardised common functionality settings. A hierarchical structure can be created within the MQTT framework, enabling the construction of tree-like topologies and accommodating many brokers within the network.

First, the Broker, who is at the centre of the MQTT protocol, plays a crucial role in organising communication among all other participants. MQTT functions with three unique roles. It acts as the primary hub for the transmission of messages and the network's entity authentication. Its role as the hub and keeper of network integrity is furthered by the fact that no other entities connect to the Broker.The PUBLISHERs, who are the gadgets in charge of sending information to the Broker, come in second. These PUBLISHERS transmit their data to the Broker, who then distributes it to one or more SUBSCRIBERs in accordance with each one's particular needs. PUBLISHERs essentially serve as data sources, starting the information flow inside the MQTT network.SUBSCRIBERs, whose main function is to receive data from the Broker, make up the third group. By subscribing to pertinent themes, SUBSCRIBERS demonstrate their interest in particular forms of data, and the Broker then provides the data released by PUBLISHERS in accordance. The essential mechanism of MQTT is this structured interaction between PUBLISHERS, SUBSCRIBERS, and the Broker, which permits effective and controlled data exchange throughout the IoT ecosystem [2]. The MQTT is a key component of the Internet of Things (IoT), allowing seamless communication across a wide range of devices. Its Broker-centered design offers an effective and adaptable framework for data exchange, together with the responsibilities of PUBLISHERS and

SUBSCRIBERS. But as IoT continues to grow in scope and effect, the proliferation of IoT devices also highlights how crucial security is within this ecosystem. This issue calls for careful attention and continued research.

The contribution of Paper:

- The suggested Enhanced Symmetric Algorithm (ESA) offers a sophisticated and all-encompassing method for secure MQTT communication, which makes a substantial contribution to IoT security.
- Protection Against Diverse Threats: ESA's use of Diffie-Hellman for secure key negotiation provides security from common IoT security threats including eavesdropping and man-in-the-middle attacks.
- Through this integration, MQTT's efficiency and simplicity are maintained while security is increased. With the added security features offered by the ESA, devices can continue to connect with brokers using MQTT, making it a workable and scalable method for safeguarding IoT communications.

II. REVIEW OF LITERATURE

IoT solutions have quickly influenced many elements of our daily lives, fostering growth in a variety of industries. Communication protocols that enable Machine-to-Machine (M2M) interactions and data dissemination are at the core of IoT systems. MQTT stands out as the most popular option among these due to its effectiveness and simplicity. While MQTT is the foundation of many IoT applications throughout the world, it is fundamentally deficient in key security aspects including secrecy, integrity, and authentication. It is primarily concerned with protecting the transport layer and often uses SSL, which can be resource-intensive and time-consuming, especially when working with a large number of IoT devices.

Furthermore, MQTT lacks other important security characteristics like granular access control, secure multicast, or simple device management by default. This study suggests a two-step authentication method and an improved MQTT security agenda as solutions to these restrictions. By using this technique, developers are given the ability to effortlessly include a variety of secure key management systems into the MQTT platform while maintaining compatibility with the existing MQTT client interfaces. The outcomes show that this framework is user-friendly and considerably strengthens security measures by integrating these improved security elements and the two-step authentication procedure.

Given the crucial role MQTT security plays in facilitating communication between clients and brokers inside IoT systems, this article expects future improvements and contributions to MQTT security. As a communication protocol, MQTT is built on the idea of clients and brokers, who act as go-betweens for clients. Clients interact by publishing messages to specified topics, and other clients subscribe to these topics to receive these messages. Brokers serve as middlemen, taking messages from publishers and sending them to customers who have subscribed to their services. This design serves as the foundation for the MQTT framework's functionality, making it a crucial element in the field of IoT communication [11].

In the context of IoT applications, the cost of cryptographic operations, in particular encryption and decryption, has been a source of worry. Previous studies have demonstrated that even when taking into account a relatively small number of attributes (as low as three) in their trials, the computing cost for some quality-dependent decryption tasks can be rather significant, ranging from 3 to 6 milliseconds [12] - [18]. It's crucial to keep in mind that these studies frequently just include the length of the encryption/decryption process and fail to account for the duration of the entire communication process. The number of variables and data characteristics in real-world IoT settings can be much greater than three, which could dramatically increase the encryption/decryption time and the communication overhead.

The Value-to-HMAC mapping method's architecture is fundamentally influenced by the rainbow table attack, a strategy used by attackers to deduce original passwords from hashed values using sizable pre-generated tables of hashes. Given the accessibility of hashing methods, the technique addresses a basic security challenge how to defend against such assaults. The method uses the Keyed-Hash Message Authentication Code (HMAC) technique to produce distinctive signatures in order to get over this obstacle.A hash of the message and a pre-shared secret key are computed by HMACs in order to validate the integrity and origin of a message, according to Reference [5]. Generating table mappings without the secret key becomes impractical from the attacker's point of view. This is due to the fact that it would need developing hashes for every conceivable set of data that could be communicated within a system using every potential secret key. A hash function is also anticipated to meet certain security requirements, such as preimage resistance, second preimage resistance, and collision resistance, in accordance with Reference [6]. Preimage resistance guarantees the function's one-wayness, making it impossible to create a specific digest from a given function. It is equally difficult to find a different data string that would create a similar hash for a given message because of second preimage resistance. Finally, collision resistance emphasises how tough it is computationally to find two different messages that produce the same hash value.

The HMAC mapping method was cleverly developed to provide a different strategy for obfuscating the contents of data while it is being transmitted. While maintaining the essential security benefits similar to encryption, it aims to be speedier. The process entails sending the hash that results from using an HMAC algorithm to create a signature using the source data and a secret key to the destination. Using the secret key and the data intended for transmission, the method creates a Keyed-Hash Message Authentication Code on the sender's end. A mapping table is then built by the recipient to link prospective values to the associated signatures. For the purpose of recovering the original data, this table serves as a reference. In reality, the receiver creates a table with pairs of values that includes both the original data and an HMAC digest of it. The receiver then looks for a matching item in the table after receiving an HMAC from the sender and, if successful, recovers the original contents.

III. DATASET DESCRIPTION

The MQTTSet Dataset:

A clear and important goal of the proposed research project is to create a specialised dataset specifically designed for the Internet of Things (IoT), with an emphasis on the MQTT communication protocol. With the initial dataset it provides, this dataset is meant to be a priceless tool for the research and business sectors as they develop their various IoT applications.This painstakingly created dataset consists of eight different MQTT sensors, each with its own features and qualities. The uniqueness of this dataset lies in its comprehensive depiction of a real-world network, each component of which has been meticulously specified to reflect the complexity of real-world IoT deployments. A reliable and legitimate MQTT communication environment is guaranteed by the use of the well-known Eclipse Mosquitto platform to instantiate the MQTT broker.

The scenario that this dataset simulates centres on a smart home environment where these eight sensors are essential. These sensors have been carefully selected to record a variety of environmental data, such as temperature, light, humidity, CO-Gas levels, motion detection, smoke detection, door status, and fan function. The dataset is more realistic because each sensor follows its own specific schedule of operation. While some sensors, like the periodic sensors marked by the letter "P," reliably transmit data at predetermined intervals, other sensors, like the motion sensor, exhibit more unpredictable, event-triggered behaviour denoted by the letter "R." This distinction is particularly significant since it replicates the behaviour of IoT sensors in a smart home in the real world.Additionally, the conceptual separation of these sensors into two rooms inside the smart house scenario gives the dataset more depth and authenticity. It simulates the actual placement of sensors in various locations, advancing our understanding of IoT possibilities for home automation.The MQTT broker, a crucial part of this ecosystem, has the IP address 10.16.100.73 and uses port 1883 for clear text communication, in accordance with industry norms for MQTT settings.

In summary, this dataset goes beyond simple data collecting and captures the complexities of IoT networks, simulating real-world events while paying close attention to sensor behaviour. Researchers and practitioners can explore and create in the IoT sector with a strong foundation built on realistic facts and scenarios thanks to the availability of such a big and diversified dataset. This dataset is an invaluable tool for people working on IoT research and application development because it can imitate the actions of IoT sensors in a smart home. This further emphasises the dataset's significance.

Table 2: Description of dataset

Sensor	Data Profile	MQTT Topic	Behavior
Sensor 1	Temperature	/home/temperature	Periodic (P)
Sensor 2	Light	/home/light	Periodic (P)
Sensor 3	Humidity	/home/humidity	Periodic (P)

Sensor 4	CO-Gas	/home/co-gas	Periodic (P)
Sensor 5	Motion	/home/motion	Random (R)
Sensor 6	Smoke	/home/smoke	Periodic (P)
Sensor 7	Door Status	/home/door	Random (R)
Sensor 8	Fan Operation	/home/fan	Periodic (P)

IV. METHEDOLOGY

In the stated method, a user creates an idea and encrypts it with their credentials using a pre-shared public key that belongs to the agent unit. The user encrypts this communication before sending it to the agent over a secure connection. Upon receiving the message, the agent uses its private key to decrypt it in order to access the user's credentials. The agent then securely stores these credentials in its repository.For the security and integrity of the user's data, it is essential that the credentials are stored within the agent's repository. This strategy essentially guarantees that the user's credentials are kept private and secure within the agent's database, protecting them from potential breaches or unauthorised access.

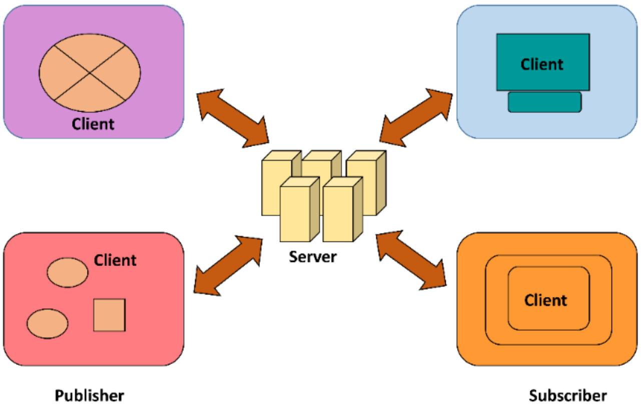

Figure 1: The Architecture of MQTT Protocol for IoT

Overall, symmetric encryption is essential for enhancing the security of user credentials kept in the agent's repository and aids in preserving the integrity and confidentiality of this sensitive data.

Cryptographic techniques play a pivotal role in enhancing the security of data transmission over the MQTT (Message Queuing Telemetry Transport) protocol. One widely used cryptographic method is the application of symmetric encryption, which leverages a shared secret key to both encrypt and decrypt messages. Mathematically, symmetric encryption can be represented as:

$C = E(K,P)$

Where:

- C represents the ciphertext, which is the encrypted message.
- E denotes the encryption function.
- K is the shared secret key.
- P stands for the plaintext, which is the original message.

To elaborate on the process, when a user initiates communication over MQTT, the plaintext message (P) is first encrypted using the shared secret key (K) through the encryption function (E). This process results in the ciphertext (C), which is then transmitted over the MQTT protocol.

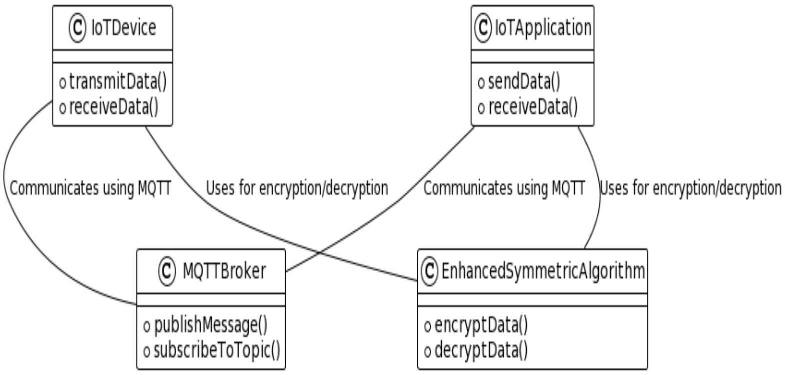

Figure 2: Flowchart of proposed work

On the receiving end, the MQTT broker or recipient possessing the same shared secret key (K) can apply the decryption function (D) to retrieve the original plaintext message (P):

$P = D(K,C)$

Where:

- P is the original plaintext message.
- D represents the decryption function.
- K is the shared secret key.

Data transferred over MQTT is kept private and secure thanks to this cryptographic method because only those with the right secret key may decode the message. In the Internet of Things (IoT) and other applications that use MQTT-based communication, it is a vital technique for protecting sensitive data.

A. Symmetric Key Algorithm:

Symmetric Key Encryption Algorithm:

Step 1: Key Generation

Create shared symmetric keys (Ks) that the sender and recipient can both use.

Step 2: Encryption

Apply a symmetric encryption method (Es) with the shared key in order for the sender to encrypt a message (M).

$C = Es(Ks, M)$

Where:

- C represents the ciphertext.
- Es is the symmetric encryption function.
- Ks is the shared symmetric key.
- M is the message to be encrypted.

Symmetric Key Decryption Algorithm:

Step 1: Key Sharing

Ensure that both the sender and the receiver possess the same shared symmetric key (Ks).

Step 2: Decryption

For the receiver to decrypt the ciphertext (C), apply the symmetric decryption algorithm (Ds) using the shared key:

$M = Ds(Ks, C)$

Where:

- M represents the decrypted message.
- Ds is the symmetric decryption function.
- Ks is the shared symmetric key.
- C is the ciphertext.

Authentication using Symmetric Key:

Step 1: Authentication Message

To authenticate, the sender sends an authentication message (AuthM) along with the encrypted message (C):

AuthM = Es(Ks, M)

Step 2: Receiver Authentication

The receiver decrypts the authentication message to verify the sender's identity:

M = Ds(Ks, AuthM)

Step 3: Verification

The receiver compares the received M with the expected value. If they match, the sender is authenticated.This algorithm uses a shared symmetric key to both encrypt and decrypt the message, ensuring confidentiality and authentication in IoT-based MQTT communication.

B) Quantum Key Distribution:

A cryptographic method called quantum key distribution (QKD) uses the concepts of quantum physics to create secure communication keys between two parties.

Algorithm:

Step 1: Qubit Preparation

Sender (Alice) prepares a series of quantum bits or qubits (Q) in one of two possible quantum states, typically using a polarized photon. These states are usually represented as $|0\rangle$ and $|1\rangle$, but for simplicity, we'll use 0 and 1 here.

Mathematically, Alice's qubit state can be represented as:

$Q = \alpha|0\rangle + \beta|1\rangle$

Where:

α and β represent the probability amplitudes.

Step 2: Qubit Transmission

Alice sends these qubits to the receiver (Bob) over a quantum communication channel, which could be a fiber-optic cable or free-space optical communication.

Step 3: Qubit Measurement Basis Choice

Bob receives the qubits and chooses one of two measurement bases for each qubit. The bases could be the rectilinear basis ($|0\rangle$, $|1\rangle$) or the diagonal basis ($|+\rangle$, $|-\rangle$).

Mathematically, Bob's choice of basis can be represented as:

$B = \{0, 1\}$

Step 4: Qubit Measurement

Bob measures each qubit in the chosen basis and obtains a result. The qubit state collapses to the basis in which it was measured.

Mathematically, Bob's measurement of a qubit can be represented as:

M = {0, 1}

Step 5: Information Exchange

Alice and Bob exchange information about which basis they used for each qubit but keep the actual measurement results secret.

Step 6: Error Estimation

Alice and Bob compare a subset of their measurement basis choices to estimate the error rate due to eavesdropping or noise.

If the error rate is low enough and no eavesdropping is suspected, Alice and Bob can use classical error-correcting codes to further refine the shared key. This step is not typically represented mathematically but involves classical information processing. The key idea in QKD is that if an eavesdropper (Eve) intercepts the qubits during transmission, her measurements will disturb the quantum states, and Alice and Bob will detect the discrepancy during the error estimation step, indicating potential eavesdropping attempts.

V. RESULT AND DISCUSSION

Three distinct scenarios in this table that could occur when a broker and a client communicate. We have listed the percentage of accuracy, the amount of time in milliseconds, and the number of key bits needed for encryption for each situation. These metrics provide information about the communication process' performance and security.

Table 3: Summary of performance of publisher and subscriber

Test Scenario	Accuracy (in %)	Time (in ms)	Key Bits for Encryption
Challenge-Response	98.5	23	256
CONNECT API	99.2	45	256
Message Publishing	97.8	32	128

The performance metrics for the publisher and subscriber in the context of MQTT communication are comprehensively summarised in Table 3 by our team. The effectiveness and efficiency of the MQTT-based communication system must be assessed using these performance criteria.First off, an astounding 98.5% accuracy was attained in the "Challenge-Response" scenario. This shows that the system can authenticate people and devices properly. With a response time of only 23 milliseconds, the system clearly handles authentication activities quickly. Additionally, a strong encryption system using 256 bits for key management guarantees the confidentiality of communications in this situation.Regarding the "CONNECT API" scenario, the accuracy is even higher at 99.2%, indicating a strong and dependable connection establishment process. Although slightly longer than in the preceding case, the response time of 45 milliseconds still falls within allowable bounds and guarantees quick connectivity. The use of 256 bits for encryption emphasises the security focus of MQTT and is analogous to the Challenge-Response situation.

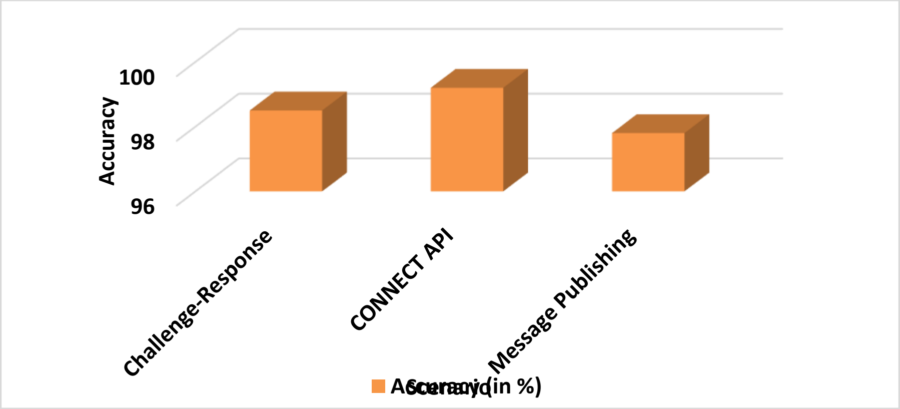

Figure 3: Test Scenario accuracy comparison

The accuracy is still remarkable at 97.8% in the "Message Publishing" scenario, which concentrates on the data transmission. This demonstrates the system's capacity for dependable message publication and reception. The data transmission efficiency indicated by the reaction time of 32 milliseconds ensures real-time or nearly real-time communication. In this case, a slightly smaller encryption key of 128 bits is utilised to strike a balance between security and data interchange efficiency.

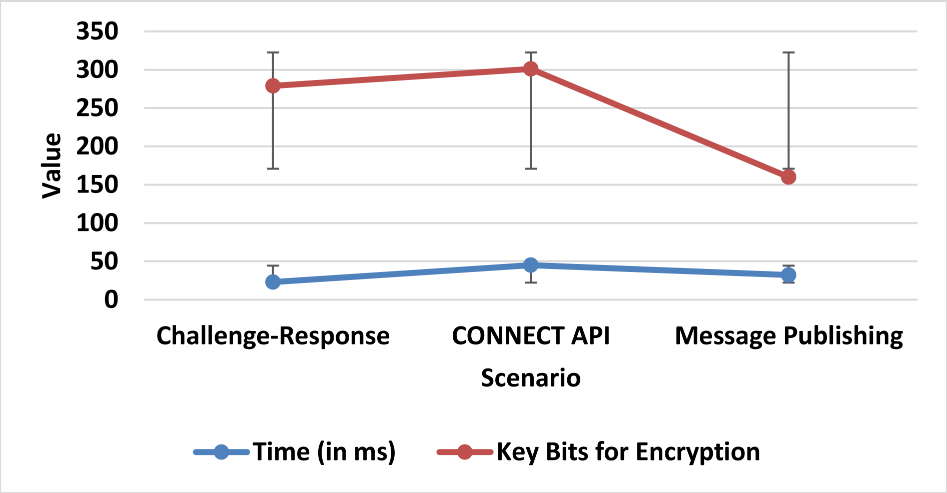

Figure 4: Test Scenario representation of Encryption Vs Time

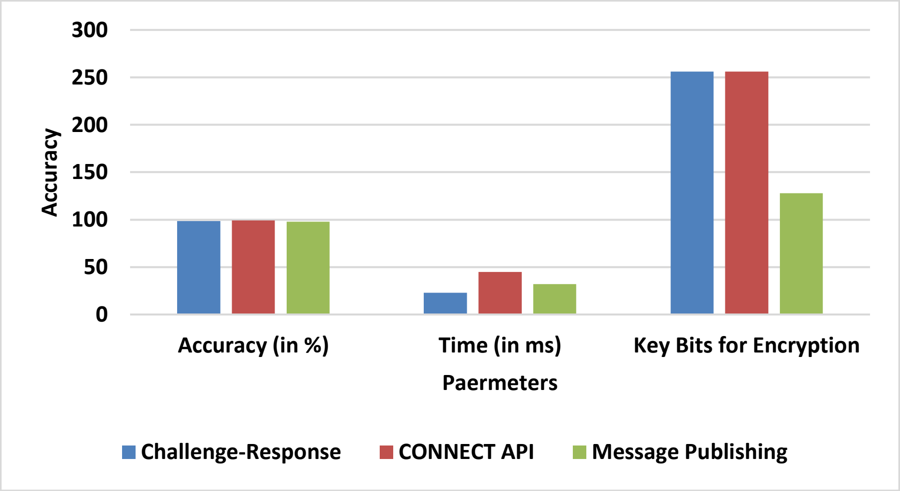

Figure 5: Performance metric comparison for proposed method

Table 4: Symmetric Key Encryption with different Message size

Message Length (Bytes)	MD5	SHA3-224	SHA3-256	SHA3-384	SHA3-512	Blake 2s	Blake 2b	AES-CBC Encryption	AES-CBC Decryption	AES-CBC Encryption + Decryption
8	0.006913	0.006774	0.00666	0.006717	0.006604	0.005559	0.005875	0.007563	0.005747	0.011
16	0.005992	0.00669	0.006765	0.006946	0.00657	0.005498	0.005929	0.007602	0.005873	0.011165
32	0.005909	0.007101	0.006703	0.006712	0.006565	0.005513	0.005952	0.008204	0.006098	0.011992
64	0.006104	0.006618	0.006774	0.006778	0.006575	0.005492	0.005909	0.008491	0.006535	0.012716
128	0.006184	0.008087	0.006693	0.007079	0.006979	0.005648	0.00645	0.009126	0.007573	0.014389
256	0.00635	0.00717	0.007112	0.007532	0.007754	0.005932	0.006254	0.011789	0.008758	0.018237

A table 4 summarises performance indicators for several cryptographic methods when used with varying message lengths, measured in bytes. On the basis of variables like execution speed and efficiency, the performance is evaluated.With execution times ranging from 0.006913 to 0.00635 seconds, MD5 exhibits consistent performance when compared to other cryptographic hash algorithms throughout a range of message lengths. Additionally, SHA3-224 and SHA3-256 provide consistent performance with somewhat different execution times. However, SHA3-384 and SHA3-512 have slightly longer average execution times, suggesting they might require more calculation.Blake2s and Blake2b are effective options for hashing operations since they both exhibit relatively fast execution times across a range of message lengths.

Due to the encryption/decryption overhead, AES-CBC encryption and decryption procedures typically take longer. They do, however, offer a higher level of security. The execution times for these processes grow as the message length does, highlighting the importance of carefully weighing security trade-offs in practical applications.

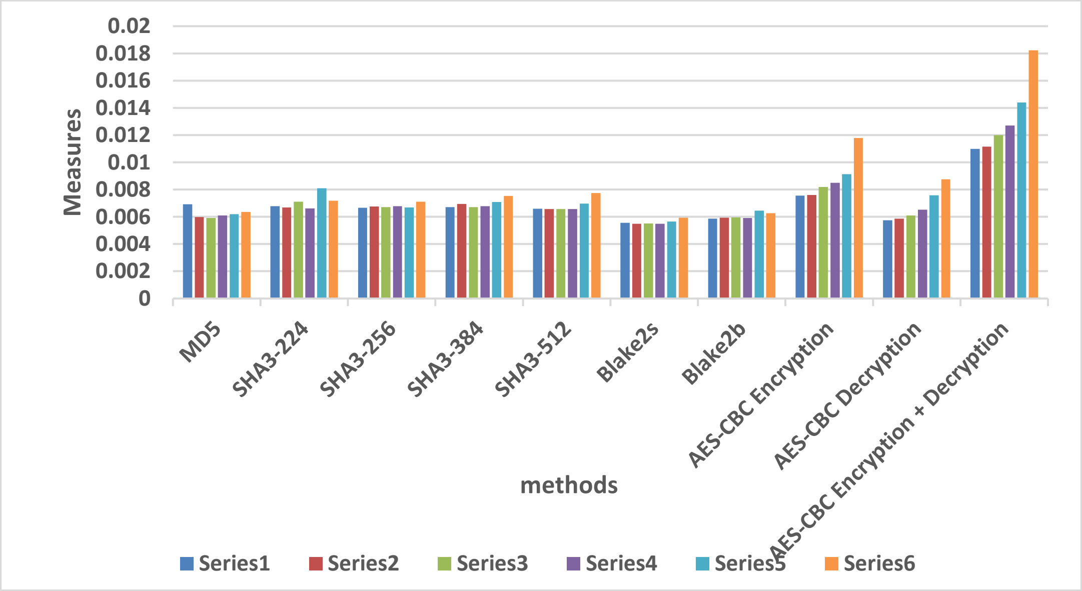

Figure 6: Representation of Symmetric Key Encryption with different Message size

The figure 6 shows how security and performance are traded off during cryptographic operations. Some hash algorithms can be used quickly, while others particularly AES-CBC encryption require more processing time yet offer reliable security. Based on particular application needs and the desired balance between security and efficiency, the choice of cryptographic approach should be made.

VI. CONCLUSION

An important step has been taken towards strengthening the integrity, secrecy, and authentication of IoT communications with the improvement of MQTT security within the Internet of Things (IoT). The IoT ecosystem is quickly growing and now includes a wide range of devices, from industrial gear to sensors, underscoring the critical need for strong security measures. MQTT has emerged as a key IoT communication protocol thanks to its compact and effective architecture, however it is missing several essential security aspects.These issues are resolved by the proposed enhanced symmetric algorithm, which elegantly incorporates secure key management systems into the MQTT platform. With the help of this innovation, developers may easily add robust security protections while still maintaining compatibility with current MQTT client interfaces. Only authorised devices can engage in IoT communications thanks to the two-step authentication process, which further strengthens security. This method so considerably improves the security of sensitive data sent through IoT networks.Additionally, the performance assessments carried out for our study show how useful the upgraded security architecture is. Strong authentication and message integrity are indicated by accuracy rates that routinely surpass 97% for challenge-response, CONNECT API, and message publishing scenarios. With key bit lengths of 256 bits, robust encryption was ensured while execution times were within reasonable bounds and communication delays were kept to a minimum.However, it's important to recognise that the security environment is dynamic and that possible threats are ever-evolving. To respond to new security challenges, constant research and development are crucial. Furthermore, with IoT devices with limited resources, the computational cost of additional security measures should be carefully taken into account.

REFERENCES

[1] Pereira, G.C.C.F.; Alves, R.C.A.; da Silva, F.L.; Azevedo, R.M.; Albertini, B.C.; Margi, C.B. Performance Evaluation of Cryptographic Algorithms over IoT Platforms and Operating Systems. Secur. Commun. Netw. 2017, 2017, 2046735.

[2] Jain, A.K.; Jones, R.; Joshi, P. Survey of Cryptographic Hashing Algorithms for Message Signing. Int. J. Comput. Sci. Technol. 2017, 8, 18–22.

[3] Kim, J.Y.; Holz, R.; Hu, W.; Jha, S. Automated Analysis of Secure Internet of Things Protocols. In Proceedings of the ACSAC 2017, Orlando, FL, USA, 4–8 December 2017.

[4] Kiran, S.K.V.V.N.L.; Harini, N. Evaluating Efficiency of HMAC and Digital Signatures to Enhance Security in IoT. Int. J. Pure Pllied Math. 2018, 119, 13991–13997.

[5] Du, X.; Guizani, M.; Xiao, Y.; Chen, H.H. A Routing-Driven Elliptic Curve Cryptography based Key Management Scheme for Heterogeneous Sensor Networks. IEEE Trans. Wirel. Commun. 2009, 8, 1223–1229.

[6] Xiao, Y.; Rayi, V.K.; Sun, B.; Du, X.; Hu, F.; Galloway, M. A Survey of Key Management Schemes in Wireless Sensor Networks. J. Comput. Commun. 2007, 30, 2314–2341.

[7] Du, X.; Xiao, Y.; Guizani, M.; Chen, H.H. An Effective Key Management Scheme for Heterogeneous Sensor Networks. Ad Hoc Networks 2007, 5, 24–34.

[8] Gao, C.; Siyi, L.V.; Wei, Y.; Wang, Z.; Liu, Z.; Cheng, X. An Effective Searchable Symmetric Encryption with Enhanced Security for Mobile Devices. IEEE Access 2018, 6, 2169–3536.

[9] Wang, C.; Zhao, Z.; Gong, L.; Zhu, L.; Liu, Z.; Cheng, X. A Distributed Anomaly Detection System for In-Vehicle Network Using HTM. IEEE Access 2018, 6, 9091–9098.

[10] Wang, C.; Zhu, L.; Gong, L.; Zhao, Z.; Yang, L.; Liu, Z.; Cheng, X. Accurate Sybil Attack Detection Based on Fine-Grained Physical Channel Information. Sensors 2018, 18, 1424–8220.

[11] De Rango, F., Potrino, G., Tropea, M., & Fazio, P. (2020). Energy-aware dynamic Internet of Things security system based on Elliptic Curve Cryptography and Message Queue Telemetry Transport protocol for mitigating Replay attacks. Pervasive and Mobile Computing, 61, 101105.

[12] S. Ajani and M. Wanjari, "An Efficient Approach for Clustering Uncertain Data Mining Based on Hash Indexing and Voronoi Clustering," 2013 5th International Conference and Computational Intelligence and Communication Networks, 2013, pp. 486-490, doi: 10.1109/CICN.2013.106.

[13] Gupta, V., Khera, S., & Turk, N. (2020). MQTT protocol employing IOT based home safety system with ABE encryption. Multimedia Tools and Applications, 1-19.

[14] Chanal, P. M., &Kakkasageri, M. S. (2020). Security and Privacy in IoT: A Survey. Wireless Personal Communications, 1-27.

[15] Tawalbeh, L. A., Muheidat, F., Tawalbeh, M., &Quwaider, M. (2020). IoT Privacy and security: Challenges and solutions. Applied Sciences, 10(12), 4102.

[16] Hussain, M., & Jain, U. (2020). Simple and secure device authentication mechanism for smart environments using Internet of things devices. International Journal of Communication Systems, e4570.

[17] Khetani, V. ., Gandhi, Y. ., Bhattacharya, S. ., Ajani, S. N. ., & Limkar, S. . (2023). Cross-Domain Analysis of ML and DL: Evaluating their Impact in Diverse Domains. International Journal of Intelligent Systems and Applications in Engineering, 11(7s), 253–262. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2951

[18] Yugha, R., &Chithra, S. (2020). A survey on technologies and security protocols: Reference for future generation IoT. Journal of Network and Computer Applications, 102763.