Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 4-B, pp. 1241–1252
DOI : 10.47974/JIOS-1906

DOEROMI : DES oriented encrypted routing of messages in IoT

Sonam *
Rahul Johari †
Security, Wireless, IoT Network Group of Engineering and Research
University School of Information, Communication and Technology
Guru Gobind Singh Indraprastha University
Sector 16C, Dwarka
Delhi 110078
India

Abstract

This paper introduces an algorithm aimed at enhancing secure routing in IoT networks, emphasizing the critical prerequisites of routing and security. The research demonstrates successful message exchange through the implementation of IoT-MQTT compliant code on an MQTT server, employing the Publisher-Broker-Subscriber model. A real-time authentic weather dataset from Uttarakhand state is utilized as a case study. The message encryption is performed using the DES algorithm, and the encrypted messages are stored in a MySQL Server. This approach ensures a high level of security for IoT transactions.

Subject Classification: *68P25 Data encryption, 68M12 Network protocols.*

Keywords: *IoT, Security, DES, DoS, WSN, MQTT, CoAP.*

1. Introduction

The rapid development of mobile computing and wireless communications has sparked significant interest and industrial advancement in a new paradigm called the Internet of Things (IoT). IoT refers to a network where smart objects are interconnected with each other and the Internet [3]. These smart things communicate, exchange information, and process data autonomously, revolutionizing various

† *ORCID-ID:* `0000-0001-7283-4483`
* *ORCID-ID:* `0000-0002-7675-8550`

† *E-mail:* `sonammathur57@gmail.com`
* *E-mail:* `rahul@ipu.ac.in (Corresponding Author)`

aspects of our daily lives. Despite the remarkable progress in IoT devices and applications in recent years, there are significant concerns regarding security and privacy. The extensive data exchange within IoT networks makes them vulnerable to breaches and attacks. This is particularly critical in sensitive fields like healthcare, where the security of networks and data is crucial for both end-users and service providers. Solutions for data communication between devices that are scalable, lightweight, mobile, and secure are essential for the effective implementation and broad adoption of IoT. Maintaining the privacy, availability, and integrity of resources and shared data in Internet of Things contexts is critical.

IoT networks face numerous threats such as Denial of Service (DoS), eavesdropping, and man-in-the-middle attacks, which have become increasingly potent and disruptive to system operations. Therefore, safeguarding IoT networks against potential attacks and vulnerabilities is crucial. Particularly challenging is ensuring secure routing within IoT's dynamic and decentralized environment, primarily due to the diverse nature of smart devices.

2. Literature Survey

The authors of [1] presented a new distributed architecture that combines a peer-to-peer gateway network with a modified Proxy Re-Encryption technique. The goal of this architecture is to reduce the computational load of many encryptions on IoT devices that have limited resources. The simulation results clearly demonstrate the effectiveness of the strategy in reducing latency, network traffic, energy usage, and encryption/decryption times.

[2] provides a detailed examination of security challenges within WSN-IoT, exploring applications, network management leveraging machine learning, and the architecture specific to IoT in Low Power Networks (IoT-LPN). The paper addresses research hurdles in IoT-LPN, identifies network attacks in WSN and IoT infrastructures, and outlines key objectives within current frameworks. The study concludes with a comparative analysis demonstrating the broad scope covered by the proposed work compared to existing literature.

In [5], the authors examined symmetric, asymmetric, and hybrid encryption techniques and gave a summary of the IoT and its security. The authors found that Elliptic Curve Cryptography (ECC) works well than other techniques after evaluating these algorithms according to a number of security standards. ECC was found to offer superior performance due to reduced memory demands and faster encryption/decryption processes. Their findings suggest that ECC is particularly advantageous for securing IoT environments compared to alternative encryption techniques.

In [6], survey of IoT, the architecture of the IoT, applications of it, challenges in it, approaches for secure communication, etc. provided. To broadcast data in the IoT network, the authors categorized the communication mechanism into various scenarios.

3. Key Components of IoT

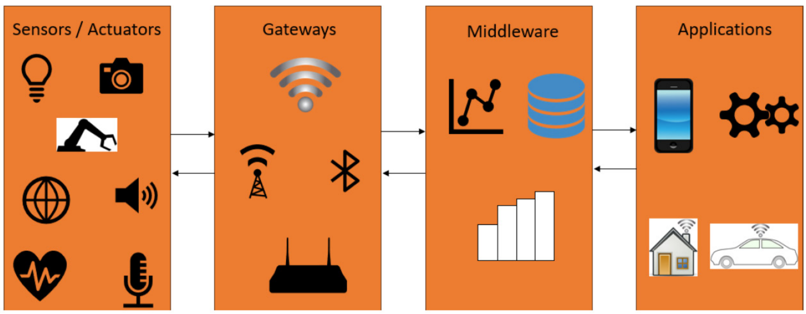

Figure 1

Key components of IoT

The following are the key components of IoT (Figure 1):

- **Sensors/Actuators:** Sensors and devices collect and transmit data while executing actions based on that data. For instance, sensors can measure temperature and humidity among other variables. A variety of sensors are available, such as GPS sensors for tracking location and sensors that measure temperature, humidity, motion, light, pressure, and gas.
- **Gateways:** A gateway in IoT serves as a crucial intermediary between sensors and the central cloud infrastructure. It facilitates communication, manages data, and processes information. Key functions of an IoT gateway include aggregating data, ensuring secure communication, translating protocols, balancing loads, and reducing latency.

  Numerous communication channels, such as Bluetooth, Cellular Networks, Satellites, Wi-Fi, Low-Power Wide-Area Networks (LPWAN), etc., are used to transmit information from sensors to the cloud.

- **Middleware:** IoT platforms, also known as middleware, offer analytical tools, sophisticated processing engines, and data storage spaces. Middleware in IoT refers to software platforms or layers that sit between the IoT devices and the cloud or enterprise applications, providing essential capabilities for managing, processing, and analyzing data generated by IoT devices.

  IoT middleware usually has data storage features made to manage the massive amounts of data produced by IoT devices. This can involve storing data in databases, data lakes, or distributed storage systems that are optimized for scalability, reliability, and fast access. Middleware manages data ingestion, normalization, and aggregation from diverse sources, ensuring that data is structured and prepared for further processing and analysis.
- **Applications:** Applications make it possible for end users to take use of the IoT and alter the physical world. Applications in the context of IoT refer to software programs or solutions that utilize data from IoT devices and systems to provide specific functionalities or services to end users. By modifying the physical world in different ways, these applications are essential to allowing people to engage with and benefit from the IoT ecosystem.

4. Advantages of IoT

The advantages of IoT encompass a broad spectrum of benefits:

- **Enhanced Communication:** IoT facilitates seamless connectivity among devices, ensuring transparency with fewer inefficiencies and higher quality interactions.
- **Automation and Autonomous Control:** IoT automates processes and controls vast amounts of data without human intervention, resulting in faster and more timely outputs.
- **Cost and Time Savings Through Monitoring:** Smart sensors used in IoT monitor various aspects of daily life across different applications, leading to cost and time savings.
- **Creation of New Business Opportunities:** IoT technology creates new economic avenues and jobs, fostering economic growth.
- **Environmental Benefits:** IoT contributes to a smarter, greener, and more sustainable planet by conserving natural resources and promoting environmental stewardship.

- **Enhanced Safety and Security:** IoT enables continuous monitoring of people and assets, thereby boosting safety and security measures.
- **Better Decision-Making:** Information gathered from IoT devices is analyzed to provide insights that improve the way decisions are made.
- **Convenience:** IoT allows for remote control of tasks and operations, simplifying daily activities.

5. Disadvantages of IoT

Following are the disadvantages of IoT:

- **Compatibility Issues:** IoT lacks a universal standard for compatibility among devices from different manufacturers, which complicates the integration and monitoring of equipment.
- **Complexity:** The IoT network is intricate and diverse, making it susceptible to significant consequences from software or hardware failures, including inconveniences caused by power outages.
- **Privacy and Security Concerns:** The involvement of multiple devices and technologies in IoT increases the risk of data breaches and privacy violations due to the transmission of sensitive information collected by smart sensors.
- **Reduction in Job Opportunities:** Automation facilitated by IoT reduces the need for manual labour, leading to potential unemployment issues in society.
- **Increasing Dependence on Technology:** IoT intensifies our reliance on technology, potentially controlling aspects of daily life and deepening generational dependency, prompting reflection on the extent to which we mechanize and surrender control to technology in our routines.

6. Most popular IoT protocols

6.1 *MQTT*

A lightweight protocol called MQTT (*Message Queue Telemetry Transport*) was created especially for sending basic data streams from sensors to middleware and applications. Originally created by Andy and Arlen in 1999, MQTT is a network protocol that uses the publish-subscribe paradigm. It is well-suited for small, cost-effective, low-memory, and energy-efficient devices. MQTT operates atop TCP/IP and consists of three primary elements: the publisher, broker, and subscriber, facilitating efficient communication in IoT and other networked environments.

In MQTT, the publisher, typically a sensor, collects environmental data and disseminates it to subscribers. The MQTT broker functions as a pivotal server responsible for authenticating both publishers and subscribers, meticulously validating their permissions, and safeguarding the entire communication process to uphold robust security standards [4]. This architecture ensures efficient and secure data transmission within MQTT-based IoT networks, supporting reliable information flow across diverse applications and devices.

After registering with the MQTT broker, a publisher can start communicating by sending a connection request. At the same time, customers who are interested in particular subjects also sign up with the broker. Clients can subscribe to topics of interest after registering. When data is released from the database, it is encrypted using the client's chosen encryption method, generating ciphertext before being delivered to the designated subscriber.

6.2 *CoAP*

The CoAP protocol (*Constrained Application Protocol*) is a specialized internet protocol designed for constrained devices, allowing them to efficiently connect to the IoT. It enables communication with minimal resources, catering to small, low-power devices. CoAP operates as a standard client-server protocol over UDP, facilitating a request/response model between application endpoints. This supports interoperability across diverse device types, enabling clients to send requests for web transfers as needed, while servers can respond accordingly.

Known for its reliability, CoAP ensures message delivery even under challenging network conditions or low device power. These features make it particularly suitable for IoT applications where devices encounter limited connectivity and operate in varied network environments.

7. Methodology Adopted

To effectively demonstrate the exchange of messages between the nodes, the IoT-MQTT compliant code was deployed on the MQTT server using the Publisher-Broker-Subscriber model approach. A real-time authentic dataset showcasing a weather dataset of the state of Uttarakhand was taken into consideration. A Cryptographic Algorithm such as DES Algorithm was then used to encrypt the message, the encrypted message was then archived in the MySQL Server. The proposed approach has been showcased with the help of Algorithm 1.

Algorithm 1 : Encryption and Decryption of the Weather Record using IoT MQTT Protocol

Notation

S_r – Server
B_r – IoT Broker; wherein B_r is with in S_r
Pr – Publisher
Sbr – Subscriber
T_p – Topic where valid $T_p \in W_d$
V – Values
VSbr [ ] - Values Table at Subscriber end
VB_r [ ] - Values Table at Broker end
VW_r [ ] - Values Table for holding Weather Record
E_T – Encrypted Text
D_T – Decrypted Text
C_a – Cryptographic Algorithm
D_B– Database
Valid $C_a \in$ **AES** | | DES | | RSA
W_d – Weather Dataset
NRW_d– Number of Records of Weather Dataset; where $NRW_d = 1$ to n

Pre-requisite :

Post successful transfer of data from Sensor- IoT Publisher- Broker - IoT Subscriber

Trigger :-

When Sbr receives W_d from B_r
Choose the Cryptographic Algorithm for Encryption
If($C_{a\ \text{Not Equal to}}$ NULL)
{
    Algorithm chosen : DES Algorithm
    Apply Encryption()
{
    $E_T \leftarrow DES_{Encryption} W_d$ [ ]
}

Apply Decryption()
{
$D_T(W_d[\,]) \leftarrow DES_{Decryption}(E_T)$
}
$D_B[\,] \leftarrow$ Store the Encrypted and Decrypted along with Weather Record{month, avg_temperature, min_temperature, max_temperature, Precipitation, Humidity, Rainy_days, avg_sun_hours, PublisherId, BrokerID, SubsriberId, topic, content, qos, $E_{T,}$ D_T }
}
else
{
  print ("Unable to encrypt the weather record");
}

---

8. Simulation

Software Description:

On the client side, the implementation utilizes Eclipse's Paho Java Client version *1.0.2*, while the broker relies on Eclipse's open-source Mosquitto message broker, version *2.0.11*, which supports MQTT protocol versions *3.1* and *3.1.1* [7]. For data storage and management, MySQL Shell *8.0* is employed, with connectivity to the MySQL database facilitated by the Java-MySQL Connector version *8.0.25*. The entire application has been developed using *JDK version 1.8.0_291*.

9. Result

The simulation was successfully accomplished and the results have been showcased in the snapshots as illustrated in Figure 2 to 6.

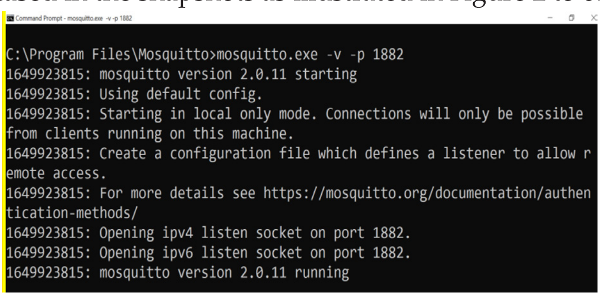

Figure 2

Snapshot illustrating the initiation of the Mosquitto Broker Server operating on port number 1882.

```
mysql> desc uttarakhand_weather;
+------------------+-----------+------+
| Field            | Type      | Null |
+------------------+-----------+------+
| month            | char(50)  | NO   |
| avg_temperature  | char(50)  | NO   |
| min_temperature  | char(50)  | NO   |
| max_temperature  | char(50)  | NO   |
| Precipitation    | char(50)  | NO   |
| Humidity         | char(50)  | NO   |
| Rainy_days       | char(50)  | NO   |
| avg_sun_hours    | char(50)  | NO   |
| PublisherId      | char(50)  | NO   |
| BrokerID         | char(50)  | NO   |
| SubsriberId      | char(50)  | NO   |
| Topic            | char(50)  | NO   |
| Content          | char(50)  | NO   |
| EncryptedText    | char(50)  | NO   |
| DecryptedText    | char(50)  | NO   |
+------------------+-----------+------+
15 rows in set (0.02 sec)
```

Figure 3

Snapshot depicting the schema design of the Uttarakhand Weather Table designed in MySQL

	January	February	March	April	May	June	July	August	September	October	November	December
Avg. Temperature °C (°F)	10.8 °C (51.4) °F	13.5 °C (56.3) °F	18.3 °C (64.9) °F	23.9 °C (75.1) °F	27.2 °C (81) °F	27.9 °C (82.1) °F	25.8 °C (78.5) °F	25.2 °C (77.4) °F	23.9 °C (75.1) °F	20.5 °C (68.8) °F	16.1 °C (61.1) °F	12.1 °C (53.9) °F
Min. Temperature °C (°F)	4.1 °C (39.4) °F	6.4 °C (43.6) °F	10.2 °C (50.3) °F	15.3 °C (59.5) °F	19 °C (66.3) °F	21.8 °C (71.3) °F	22.8 °C (73.1) °F	22.3 °C (72.2) °F	19.7 °C (67.4) °F	14.2 °C (57.5) °F	9.5 °C (49) °F	5.6 °C (42) °F
Max. Temperature °C (°F)	17.4 °C (63.4) °F	20.3 °C (68.5) °F	25.5 °C (77.9) °F	31.4 °C (88.5) °F	33.9 °C (93) °F	32.7 °C (90.9) °F	28.9 °C (84) °F	28.3 °C (82.9) °F	27.9 °C (82.1) °F	26.6 °C (79.9) °F	23 °C (73.4) °F	19.1 °C (66.5) °F
Precipitation / Rainfall mm (in)	43 (1.7)	66 (2.6)	42 (1.7)	28 (1.1)	34 (1.3)	142 (5.6)	435 (17.1)	424 (16.7)	187 (7.4)	17 (0.7)	5 (0.2)	18 (0.7)
Humidity(%)	73%	67%	55%	39%	42%	58%	85%	87%	83%	70%	68%	72%
Rainy days (d)	4	5	4	4	6	11	20	20	14	2	1	2
avg. Sun hours (hours)	8.7	9.4	10.6	11.5	11.9	10.7	8.0	7.5	8.7	9.8	9.4	8.9

Figure 4

Snapshot of the weather dataset of the state of Uttarakhand from January to December with fields such as : Average Temperature, Minimum Temperature, Maximum Temperature, Humidity, Rainy Days and Average Sun Hours et al.

```
C:\IoT>javac MqttPublishSecurityDES.java

C:\IoT>java MqttPublishSecurityDES January 10.8 4.1 17.4 43 73 4 8.7
Generated Publisher Id ispaho3623144138000
Connecting to broker: tcp://127.0.0.1:1882
Sub Connected
Text [Byte Format] : [B@21a06946
Text : January10.84.117.4437348.7
Text Encryted : [B@326de728
Encrytion over
Encrypted Message published
Publisher Disconnected
Received message from Broker is [B@326de728
Text Decryted : January10.84.117.4437348.7
Decrytion over
Thu Jul 04 14:48:22 IST 2024 WARN: Establishing SSL connection without server's ident
ity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ r
equirements SSL connection must be established by default if explicit option isn't se
t. For compliance with existing applications not using SSL the verifyServerCertificat
e property is set to 'false'. You need either to explicitly disable SSL by setting us
eSSL=false, or set useSSL=true and provide truststore for server certificate verifica
tion.
No of rows entered1
Thu Jul 04 14:48:22 IST 2024 WARN: Caught while disconnecting...
```

Figure 5

Snapshot depicting the launch of the Publisher, which publishes the encrypted and decrypted text on the broker window and also inserts the record into the Uttarakhand Weather Table in the Database

```
mysql> select * from uttarakhand_weather;
+---------+-----------------+-----------------+-----------------+---------------+----------+------------+------------
---+----------------------------------------------+-----------------------+------------------------------------------
---------------+---------------+---------------------------+---------------+---------------+
| month   | avg_temperature | min_temperature | max_temperature | Precipitation | Humidity | Rainy_days | avg_sun_hou
rs | PublisherId
                      | BrokerID               | SubsriberId
                      | Topic          | Content                      | EncryptedText | DecryptedText |
+---------+-----------------+-----------------+-----------------+---------------+----------+------------+------------
---+----------------------------------------------+-----------------------+------------------------------------------
---------------+---------------+---------------------------+---------------+---------------+
| January | 10.8            | 4.1             | 17.4            | 43            | 73       | 4          | 8.7
   | org.eclipse.paho.client.mqttv3.MqttClient@5b8dfcc1 | tcp://127.0.0.1:1882 | org.eclipse.paho.client.mqttv3.MqttC
lient@2f9f7dcf | Weather Record | January10.84.117.4437348.7 | [B@326de728   | [B@233c0b17   |
+---------+-----------------+-----------------+-----------------+---------------+----------+------------+------------
---+----------------------------------------------+-----------------------+------------------------------------------
---------------+---------------+---------------------------+---------------+---------------+
1 row in set (0.00 sec)
```

Figure 6

Snapshot depicting a MySQL Command Prompt window depicting the encrypted and decrypted text published by the Publisher has been archived into the Uttarakhand Weather Table of MySQL Database.

10. Conclusion and future work

In the course of this research, message exchange was carried out by deploying IoT-MQTT compliant code on the MQTT server using the Publisher-Broker-Subscriber model approach. A real-time validated dataset showcasing a weather dataset of the state of Uttarakhand was taken into consideration. The DES algorithm was used for carrying out the encryption of the message, and the encrypted message was then archived in MySQL Server. The kind of security thus achieved of the IoT Transactions was quite high. The results obtained have been positive and encouraging. In light of existing knowledge and information, the planned effort is believed to be the first of its sort in this direction.

Acknowledgment

The author(s) express their sincere gratitude to GGSIP University for offering an exceptional academic setting and research facilities that greatly aided in the accomplishment of this work. Their profound gratitude is also extended to AICTE for bestowing the AICTE Doctoral Fellowship (ADF), whose kind assistance has been essential in enabling and promoting their study.

References

[1] Gangadharaiah and L. B. Bhajantri, “Secure data dissemination and routing in Internet of Things,” *International Journal of Information Technology* (Apr. 2024), https://doi.org/10.1007/s41870-024-01848-4.

[2] M. Z. Hussain and Z. M. Hanapi, “Efficient secure routing mechanisms for the low-powered IoT network: A literature review,” *Electronics*, vol. 12, no. 3, p. 482 (2023), https://doi.org/10.3390/electronics12030482.

[3] Sonam and R. Johari, “SEROMI: Secured Encrypted Routing of Message in IoT,” in *International Conference on Advanced Communications and Machine Intelligence* (pp. 339-347). Singapore: Springer Nature Singapore, pp. 339–347 (Dec. 2022), https://doi.org/10.1007/978-981-99-2768-5_32.

[4] Sonam and R. Johari, “Crypto-Economic Model for Data Security in IoT Network,” in *Proc. Int. Conf. Innovative Comput. Commun. (ICICC)*, vol. 3, Singapore: Springer Nature Singapore, pp. 427–434 (Nov. 2022), https://doi.org/10.1007/978-981-19-3679-1_34.

[5] S. K. Mousavi, A. Ghaffari, S. Besharat, and H. Afshari, “Security of Internet of Things based on cryptographic algorithms: A survey,” *Wireless Networks.*, vol. 27, no. 2, pp. 1515–1555 (2021), https://doi.org/10.1007/s11276-020-02535-5.

[6] S. N. Mahapatra, B. K. Singh, and V. Kumar, "A survey on secure transmission in Internet of Things: Taxonomy, recent techniques, research requirements, and challenges," *Arabian Journal for Science and Engineering*, vol. 45, no. 8, pp. 6211–6240 (2020), https://doi.org/10.1007/s13369-020-04461-2.

[7] MQ Telemetry Transport (MQTT) V3.1 Protocol Specification,http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1- os.html

Received October, 2024