# Security for IOT communications

Apoorva Saxena
University of Massachusetts
Amherst, MA, USA

## *ABSTRACT*

*Industrial internet of things deals with the interconnections of the individual processes involved. By connecting machines, a manufacturer can create intelligent networks along the entire value chain that communicate and control each other autonomously with significantly reduced intervention by operators. The inception of IoT network demand in manufacturing has aggravated the need for protection of the sensors and any new applications at the industrial side. Secure transmission of the value chain is essential as any change to it by an adversary may incur huge losses to the company. If any of the individual devices in the chain are compromised then the threat can acquire access to that communication link, allowing the adversary to push malicious data causing distributive denial of service (DDoS), or introduce malware or virus that could bring down the entire network. The idea behind the project was to devise a format for secure communication over an IOT network and to develop a proof of concept. We first start with defining how we secure our communications using encryption. We will be discussing the encryption techniques used in the proposed system in the 'Approach' section. Further, we test the viability of the communication format over a simulated IOT network developed over python using MQTT (Message Queue Telemetry Transport) protocol and an open source MQTT broker HiveMQ.org. Python library PyCrypto has been used to implement AES encryption, Python library paho-mqtt was used to implement MQTT protocols and rest of the project was developed from scratch. We have performed experiments to test the vulnerabilities of the proposed system against common network attacks such as eavesdropping, man in the middle and replay attacks the result of which have been explained in the 'Discussion' section.*

**KEYWORDS**

IOT, security, MQTT , encryption , RSA , AES , secure communication.

## *INTRODUCTION*

The main components of an IoT network are small embedded devices that have long lifespans, thus making them very efficient. The main objective of security in an IoT setup is to keep these devices from being hacked. Today, industrial organizations are inclined on creating a secure connected infrastructure with IP enabled sensors or IP/IoT enabled Access Gateways. The data that are constantly being given out by the sensors at a machine location are valuable not only to the central control system but also to other machines in the network that communicate with the sensors. As a result, the operators connect these sensors directly to the cloud or backend databases for all authorized systems to access.

Automobile manufacturing being a pioneer industry that has adopted IoT technology for its operation has been considered in this project. Here each station in the manufacturing process is connected to a central server. The stations act as clients and the central server as the main server. Thus, the central server contains all the parameters associated with each station for manufacturing a car within the specified standards and the information about authorized individuals who can access and alter these parameters. The project deals with ensuring the confidentiality and integrity of the value chain to ensure that no unauthorized individual can monitor or modify the parameters. The requirement of the project is the transfer of large files between server and the sub-unit station securely. To provide confidentiality and integrity a strong crypto mechanism is required. This project will use AES (Advanced Encryption Standard) 256-bit encryption and RSA public key encryption to decrypt the symmetric key. By using this method, the system ensures that there can be no way in which company data can be intercepted and monitored by any person outside the company unless the person has authorized access, also the system proposed ensures that the automobiles are within the normal

and safety standards specified preventing any catastrophes due to a manufacturing fault by a cyberattack. This system also prevents any losses to the company due to its accounting details, which are being shared in the procurement stations if being monitored or hacked.

## *BACKGROUND*

Industry 4.0 is the next industrial revolution in automation and data exchange in manufacturing technologies. It has been a long and steady journey from the first-generation water and steam power to the latest fourth-generation knot of cyber-physical systems, the internet of things, cloud and cognitive computing. With the depiction of the Industry 4.0 as the 'smart factory' there comes along various security problems that can lead to denial-of-services, corporate espionage, theft and brand damage. A robust security model is hence required for the Internet of things. The security detailing is still considered an obstacle in setting up smart factories. It needs to be made sure that safeguards are built into the solution that includes the basic security procedures like hardware encryption, physical building security and network security for data in transit. It is also important for the network to allow secure remote network access to systems. The identity and authentication management need to be updated to support both people as well as "things". Device-to-Device communication (D2D) in IoT operation is achieved using the simple and effective protocol MQTT or Message Queue Telemetry Transport. This protocol has limited security features associated with it. A typical MQTT protocol in communication has a MQTT broker to which all the devices/clients, wireless network sensors, localized centralized databases are connected. The protocol ensures an efficient mode of communication between the broker and the connected devices. MQTT broker being bi-directional in terms of communication of data, multiple clients can access and transmit packets or information from it.

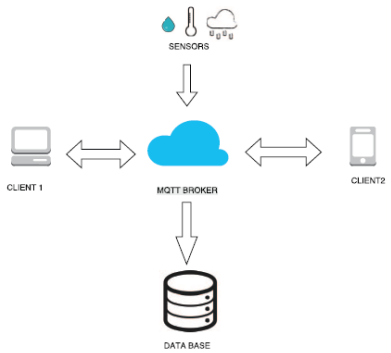


Fig. 1: A general MQTT model used for communication

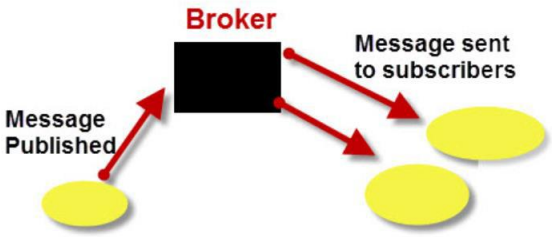


Fig. 2: Basic block diagram of MQTT protocol

**MQTT**

MQTT is a Publish/Subscribe messaging protocol for machine to machine communication developed by IBM. In MQTT protocol we have a publisher, a broker and subscribers. MQTT is called a lightweight protocol because all messages have a small code footprint. Each message consists of a fixed header (2 bytes), an optional variable header, a message payload that is limited to 256 MB. The publisher sends messages to the broker, the broker messages are differentiated based on topic, the broker sends these messages to the subscribers based on the topics the subscribers have subscribed to. The broker does not store the files if the subscriber does not subscribe to the file or is disconnected for

some reason the broker discards the file. MQTT uses TCP/IP to connect to the broker. The MQTT protocol provides authentication in that the broker has is configured such that client ID's are provided for each topic and the topics are sent out to only the client ID's that have been linked to the topic.

The advantages to using MQTT here are:

- It provides an always on connection. Thus, anytime a server updates a file in a certain topic the subscribed station automatically subscribes to the updated file.
- There is no direct connection between the main server and the individual stations.
- If the main server were to go offline for some reason the broker would be able to provide the files to the subscribers if they are subscribed to the topic.
- MQTT offers three Quality of Service(QOS) at most once, at least once and exactly once.

The disadvantages of using MQTT are:

- Encryption must be done separately.
- Scalability is an issue.

## *APPROACH*

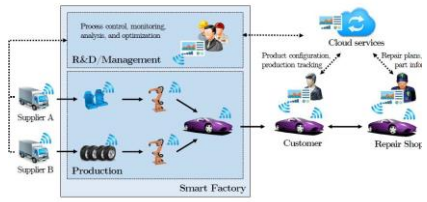


Fig. 3: Manufacturing facility employing IoT

Consider a manufacturing facility as shown above, here each and every station will communicate with each other as well as the main server which in this case is depicted as management. The server will send the appropriate encrypted files to each station when the station sends a request , the station then decrypts the files and performs its operation after which it sends in the result to the server, the server then checks this result and sends to the station a pass or a fail message, the station now performs the same operation with a different value set, once it receives a pass message it sends a done message to the server and the server acknowledges this and the station also sends a done message to the next station so that it is ready to receive the part and this way a station cannot be skipped. These communications between station and the server are encrypted so that even if an adversary can get access to the file, the adversary won't be able to decrypt the file as they won't have the appropriate keys required for decryption. In our case we have used MQTT protocol with AES encryption and an RSA key to ensure secure transmission and to authenticate the parties involved.

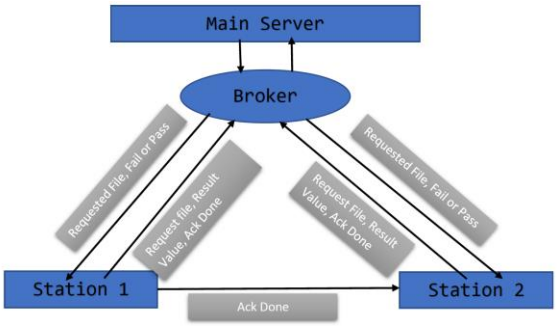


Fig. 4. MQTT protocol in detail

For our system the MQTT protocol is as shown, the main server sends a file to the broker and the broker assigns the file to the corresponding topic, now station 1 is requesting the file for station 1, the broker sends the corresponding file to station 1, this file is encrypted( the encryption/decryption process will be explained later) the station decrypts the file and performs the operation and sends the result to the server, the server checks the result and sends either a pass or a fail, if the part fails, the server sends a fail message to station 1 and thus station 1 tries a new set of values from the same file it had before and tries out various values until the value set for which the part passes is found. If the part passes, station 1 sends an Ack done to station 2 and to the main server. Now once station 2 receives this Acknowledgement it starts with its operations like station 1

## *EXPERIMENTS*

To provide security including Confidentiality and Integrity, a strong cryptographic mechanism is needed to encrypt the data sent. To prevent the key distribution problem, generally Public Key Encryption is used for encrypting data. But, tradeoff using Public Key Encryption to encrypt large files is the Performance Bottleneck to encrypt and decrypt data. So, clearly, a Symmetric key encryption mechanism is required to encrypt these large files before transferring. Main issue of using a Symmetric cipher to encrypt data is protecting the secrecy of the Key which is used for symmetric encryption. To solve both of these performance problem and key distribution problem, AES (Advanced Encryption Standard) 256-bit Encryption for file encryption and RSA public key encryption to encrypt the Symmetric AES key is used.
The AES key is 32 bytes and is generated using a random number generator, the file to be sent is now encrypted using AES with the 32-byte key, this 32-byte key is now encrypted using the receivers public key.
Now the encrypted AES file is prepended to the with the initialization vector (IV) and with the encrypted AES key. The resulting ciphertext is as shown below.

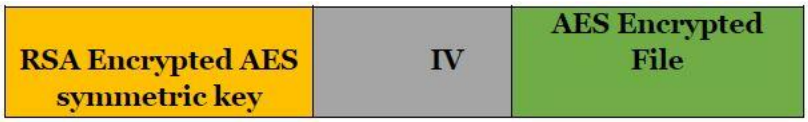

Fig. 5: Ciphertext as obtained

At the receiver side the cipher text is first extracted to get the encrypted AES key and the AES encrypted file. Next the AES key is decrypted using the receivers private key, using this key the encrypted file is decrypted, and the plaintext is obtained. In our system, the server sends the respective encrypted parameter files to the broker, the individual stations subscribe to the respective parameter files, decrypt these and perform their respective operations and sends the encrypted result back to the server, the server now decrypts and checks this result and sends either a pass or a fail message based on which the station either tries the next value set or sends ACK done to server and next station. Thus, using MQTT along with AES encryption we can ensure the Authenticity of the data sent by the server to the stations and vice versa. The robustness of our algorithm and the system has been tested by a display of encryption and decryption of the parameter files by the main server and the subsequent stations respectively. The respective parameter files are encrypted during the transfer to the client side which are then decrypted at the client end. The encrypted and decrypted files for the said “Station 1” are provided as follows:

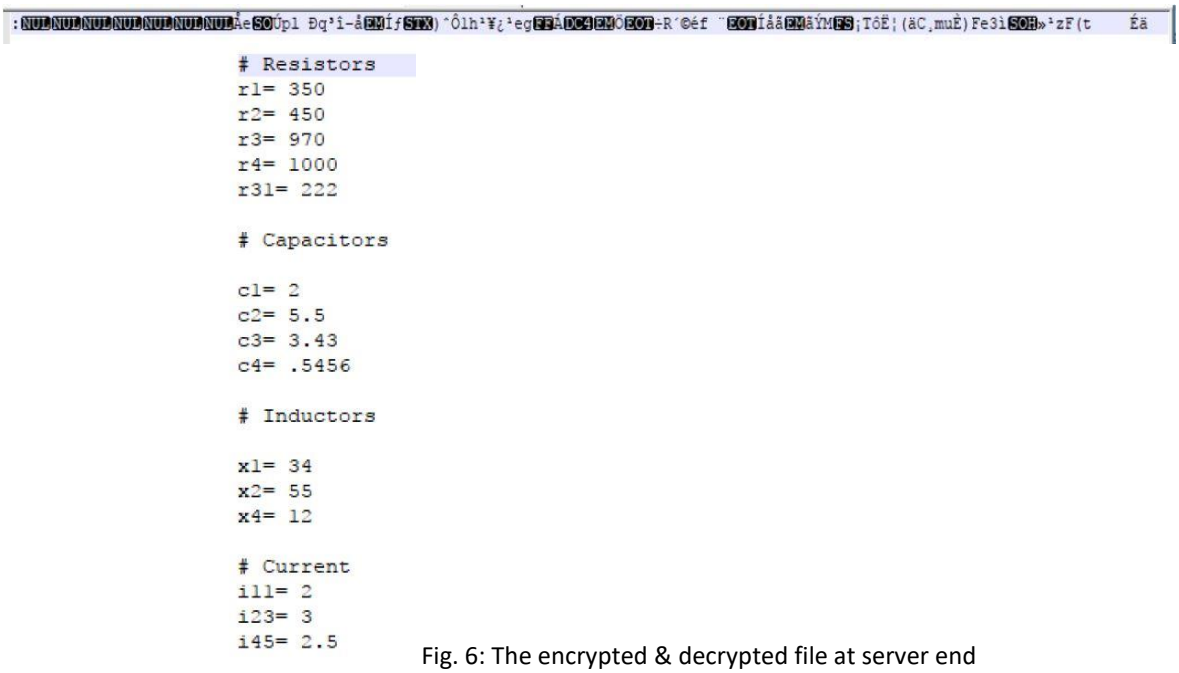


Fig. 6: The encrypted & decrypted file at server end

The final parameters are again sent back to the server by the stations along with acknowledgment whether the job was successful or not that are similarly encrypted at the stations and decrypted at the server. The encrypted message although looks the same as Fig. 6, the decrypted file at the server end is given as follows:

```
Parameter1 = 14.83
Parameter2 = 123.7
Parameter3 = 64.44

Parameter Applied JOB1 completed
```

Fig. 7: The decrypted file at server end

## *RESULTS*

- Eavesdropping
  One of the biggest challenges with Cloud Computing is eavesdropping of communication. Using common cloud services can make the data vulnerable to eavesdroppers listening to entire network data. All communications are encrypted and restrictions like SSL certification is employed on listening channels, the data transfer can be secured.

- Brute Force Attack
  An attacker who has access to all the communication channels can use Brute Force attack to decode the RSA encrypted AES private key. Once obtained the attacker can decrypt parts of the conversation. With the exponential increase in the computational power even encryptions like AES algorithm can be decoded. To make such attacks redundant, private keys used to encrypt the data can be changed regularly and each node should have a different key.

- Man in the middle attack
  In case where the communication pattern and public key is leaked, the attacker can decrypt parts of the conversation. Implementing strict authentication and user access privileges can prevent injection attacks in both cases.

- Replay Attack
  Replay attacks involves the adversary eavesdropping the entire communication of a network and injecting individual messages in a predetermined sequence to emulate a node in the network. The adversary can inject encrypted messages copied from the previous communication to ensure the message is authenticated and crash the system by creating a DDOS type of situation. If the adversary was able to decrypt the messages, he can inject messages aiming to corrupt a sections of the network. To prevent such attacks, we can employ methods like one-time passwords or nonce (once in a lifetime integer) but these solutions will increase the complexity of the system

## *DISCUSSION*

Implementing secure communication over the IoT network has its fair share of challenges. An important feature of IoT networks is speed of data transmission. Data encryption increases the complexity of the system and computation time for the sensor data to be marshalled over the network. Encryption also increases the data overhead of the network linearly to message length. Encryption algorithms like AES requires the length of the plaintext to be encrypted be a multiple of 16. If this condition is not met plaintext needs to be padded with whitespace creating a data overhead. Both these factors result in slowing down the rate of data transmission in an IoT network. Another issue faced while implementing encryption is hardware related. For implementing encryption, extra hardware needs to be added to sensors like a processing unit and memory ultimately increasing the cost and size of the sensors. Also, sensors are subjected to harsh environmental conditions like temperature, pressure etc. Memory units are highly sensitive to electrical, magnetic and other kinds of flux, further complicating the sensor manufacturing process.

## *CONCLUSION*

There are many reasons behind the growing popularity of IOT improvement in internet services and dramatic reduction in the manufacturing cost of sensors being a few of them. For IOT and Cloud Computing to gain ubiquity in Industrial arena, ensuring secure communication between different networks is essential. The challenge lies in choosing the best encryption algorithm and further establishing a secure network for key distribution for decryption. With the exponential increase in the computational power of computers and restrictions over the overheads caused due to encryption algorithms, encrypting communication is not enough to ensure security in our systems. Apart from encryption, authentication is required at each level to prevent masquerading and injection attacks. Authentication is the process of recognizing a user's identity. It is the mechanism of associating an incoming request with a set of identifying credentials. Authentication ensures that even if an adversary is able to decrypt the communication of the proposed system, no unwanted messages can be injected by the adversary. Post authentication the final step to a secure system is authorization, user access privileges is another important factor which needs to be clearly defined. In the fields of physical security and information security, access control is the selective restriction of access to a place or other resource. The act of accessing may mean consuming, entering, or using. In case an active attacker does manage to disguise itself as a node in the system, clearly defined access controls can ensure the data and communication channels at risk are limited.

## *REFERENCES*


[1] Lancen Lachance, “IOT security in the Auto Industry”, https://www.globalsign.com/en/blog/IoT-automobile-industry-sae-congress/

[2] TechTarget, “Prevent Enterprise IoT Security Challenges”, “IOT Agenda”, 2017

[3] WIND, “Security in the Internet of Things”, 2015

[4] “Building Smarter Manufacturing With The Internet of Things” http://cdn.IoTwf.com/resources/6/IoT_in_manufacturing_january.pdf

[5] Liu Tenghong, Yuan Rong, Chang Huating ,“Research on the Internet of Things in the Automotive Industry” , 2012 International Conference on Management of e-Commerce and e-Government

[6] Ahmad-Reza Sadeghi, Christian Wachsmann, Michael Waidner, “Security and Privacy Challenges in Industrial Internet of Things”, DAC ’15, June 07 - 11, 2015, San Francisco, CA, USA

[6] Deutsche Telekom AG, “Security on the industrial internet of things”

[7] Ericsson White paper, “IoT Security”, February 2017

[8] http://www.steves-internet-guide.com/mqtt/

[9] Arya Sahadevan, Deepa Mathew, Jairam Mookatana, Bijoy A. Jose, “An Offline Online Strategy for IoT using MQTT”, 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing, 2017

[10] Meena Singh, Rajan MA, Shivraj VL, Balamuralidhar P, “Secure MQTT for Internet of Things (IoT)”, 2015 Fifth International Conference on Communication Systems and Network Technologies, 2015

[11] X. Wang, J. Zhang, E. Schooler, M. Ion, “Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT,” Communications (ICC), 2014 IEEE International Conference, June 2014

[12] Abdessamad Mektoubi, Hicham Lalaoui Hassani, Hicham Belhadaoui, Mounir Rifi, “New approach for securing communication over MQTT protocol: A comparaison between RSA and Elliptic Curve”, IEEE, 2016

[13] Wei Peng, Song Liu, Kunlun Peng, Jin Wang, Jin Liang, “A Secure Publish/Subscribe Protocol for Internet of Things Using Identity-Based Cryptography”, 5th International Conference on Computer Science and Network Technology (ICCSNT) IEEE, 2016.