**Security for Internet of Things: Industrial Internet of Things**

Prabhat Bhatt1, Ritika Ray1, Snehith Sharath1

1Department of Electrical and Computer Engineering

University of Massachusetts Amherst

Spire Id- 30994754, 30294657, 30331982

**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 advisory may incur huge losses to the company. Also, if test gate or security features of say a car are the stations in the network being tampered with, the results can be catastrophic as it may result in a car with a design flaw being sold to customer and may cause death. 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 main components of an IIoT network are small embedded devices that have long lifespans, thus making them very efficient. The main objective of security in an IIoT 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/IIoT 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 directly enabling multiple systems to obtain real-time sensor data directly from the hub. 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 IIoT 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 one other authorized individuals can monitor or modify the parameters. The requirement of the project is the transfer of large files between server and the subunit 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. ~~To ensure the validity of the file MD5 hash of the file will be calculate and appended to the file which will be sent finally~~. By using this method, the system ensures that there can be no way in which company data can be 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.*

**A. Specific Aims**

By creating the value chain industrial internet of things gives access to authorized individuals in any part of the world the details of the functions and data of the stations he/she is associated with. Thus, they are able to solve problems remotely rather than having to be connected to the company intranet or be on the site.

If every stakeholder involved can be ensured and convinced about the confidentiality and integrity provided by the security systems in place, it is possible to continue tracking the product once it has been manufactured. In this case, a car can be continuously monitored if it is allowed by the customer, it can, at regular intervals of time, send out information about critical parameters - say something like brake fluid level or tyre pressure or as in case of modern cars the functioning of the ECU (Electronic Control Unit) which basically controls most of the functions of the cars. These can be monitored on a cloud server which contains its standard values and if the cars fall below these values an alert can be sent to the customer. Also for any requests that come in a company, it can provide authorized repair shops with access to repair plans for specific cases as per company standards and also help repair certain parts if they can be done remotely through the server. Thus, it ensures that the system in existence can at all times monitor the car and prevent any change in the car standards other than the specified values, increasing safety of cars on the road in the process.

For our system we are securing the communications between the central server and individual stations in a manufacturing industry by finding a way to secure the files so that no adversary will be able to intercept the data sent between server and station and even if an adversary is able to access it will be an encrypted file and will require cryptoanalysis and by this time the process would have gone through thus the adversary will not be able to cause any damage with the information.

**B. Background and significance**

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”.

The proposed project deals with the security in Industrial Internet of things for an automobile manufacturing company. An automobile manufacturing process must be very precise taking in all the minute details into consideration. The safety of everyone, from the driver to all on road, depends on the quality of the vehicles. The automobile manufacturing industry is almost fully automated, and all the facilities and equipment connected to each other thanks to the Industrial Internet of things (IIoT). The equipment shares and analyses important data that deals with the various parameters required for manufacturing. The basic elements in an automobile manufacturing company will be the main server that hosts all the information and parameters for manufacturing each part of the automobile and the individual stations that make use of these parameters to build these parts. The central server and the connection network need to be made secure from any adversary intervention. There are different attacks that can compromise the entire process. Denial of service attack is a kind of disruptive cyberattack that have new detrimental consequences for the process. If the parameters are not available for any of the stations, then that part will not be able to be manufactured and the entire system will come to a standstill. Loss of confidentiality and integrity may lead to simple modifications in the parameters that hamper the entire process.

The most notable cyberattack is the Stuxnet worm that targeted SCADA systems. Stuxnet specifically targets PLCs. It compromised the PLCs by collecting information on industrial systems in Iran’s nuclear program and caused the fast spinning centrifuges to tear themselves apart. It is said to have infected 200,000 computers and caused 1000 machines to degrade. Stuxnet has three modules: a worm that executes all routines related to the main payload of the attack; a link file that automatically executes the propagated copies of the worm; and a rootkit component responsible for hiding all malicious files and processes, preventing detection of the presence of Stuxnet. It is typically introduced to the target environment via an infected USB flash drive. The worm then propagates across the network, scanning for Siemens Step7 software on computers controlling a PLC. In the absence of either criterion, Stuxnet becomes dormant inside the computer. If both the conditions are fulfilled, it introduces the infected rootkit onto the PLC and Step7 software, modifying the codes and giving unexpected commands to the PLC while returning a loop of normal operations system values feedback to the users.

In a manufacturing industry PLCs are used in almost every machine and this kind of attack needs to be avoided. For this security mechanisms must be developed so as to nurture a trust that this kind of attack cannot happen again, especially if the end goal stated for an automobile manufacturing company is to be able to update or support remote repair of the vehicles and have all of them transmitting and receiving from a central server.

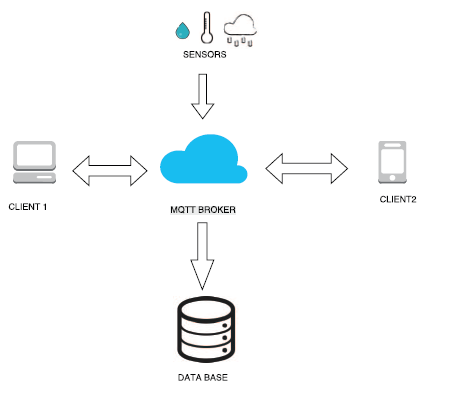
Safety authorized user credentials from being stolen or hacked into is of utmost importance. In any IOT system if the user credentials are hacked or stolen there is no telling the level of damage that can be caused. An example of this is when hackers were able to gain access to Ukraine’s power grid recently using stolen credentials and were able to shut off power to 30 substations and around 225,000 customers, they also installed new firmware into the system and deleted many master record files.

Security measures have been taken up since a long time for issues similar to these. Evolution of security controls have been seen from the first packet-filtering firewalls in 1980s to intrusion detection and prevention systems (IDS/IPS), and security incident and event management (SIEM) solutions. They prevented the admission of any malware into the network and if any managed to pass through the tight sieve of firewalls, the antivirus in the system remedied the issue. As the number of devices started to grow in an IOT network, malwares started to evolve such that they were able to get away without being detected. More advanced security protocols have been implemented for ensuring the security of data in motion namely, virtual private network (VPN), physical media encryption like 802.11i (WPA2) or 802.1AE (MACsec). However, implementation of these in the IOT platform required a lot of changes to address the device constraints.

The suitable platform of cloud computing achieves secure data storage and processing. Equipment in a smart factory maybe connected with cloud via networks and programming interfaces (APIs). A secure basis for all traffic to and from the cloud is the Transport Layer Security (TLS) encryption protocol. End-to-end (E2E) encryption is considered to be the most suitable security precaution for all devices on the IoT.

The deployment of IoT systems is a challenge as the devices involved have no user interface. To enable secure and automated bootstrapping, each device should have device credentials provided during manufacturing. The use of trusted execution environments has substantial cost benefits and facilitates the use of 3GPP technologies in various industries. Authorization and protection of data in transfer are best carried out using over-the-top security on the application layer.

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 [9]**

The only security in the MQTT environment to the communication among the users is provided by SSL and TLS protocols. But, this is not at all sufficient to provide optimal security. There is no security in the broker level. As a result, all the data published by the broker can be accessed by all the users irrespective of their authorization.

Hence, additional security measures need to be employed over the MQTT protocol for optimal secure data communication between the broker and the subscribers and also among the individual subscribers.

In general, IoT security and privacy are the most important aspect for the general public, media, enterprises, governments, and companies. Security is a major concern among enterprises adopting IoT and cloud solutions.

**C. Preliminary Studies**

Security is and has always been the most important aspect in any type of system. Security in IoT is one of the most researched topic since its implementation.

The main aspect of IIoT is Device to Device communications (D2D). This is obtained via several protocols such as Constrained Access Protocol (CoAP), Message Queue Telemetry Transport (MQTT), etc. The most important facet of IoT deployment is to ensure security of the participating devices. Now, the above existing protocols are devoid of any security features. Hence, additional security measures are required in every IoT operations.

[9] points out that with a surge in the production of several low power and low-cost sensors for building more and more IoT applications, a simple and effective communication scheme between the devices is required. The main requirement for this is the robustness of the infrastructure. The paper provides an online offline technique for asynchronous communication done using MQTT protocol. The strategy is primarily developed for the E-voting system that requires both online and offline management. The infrastructure dependency problems are addressed using MQTT in Intel Galileo. The broker used is the Mosquitto Broker that is most suited for a constrained device such as the Intel Galileo board.

For online communication cloud computing is employed to provide Amazon web services as the cloud platform to the IoT devices. AWS provides the storage of data obtained from the Hub via the MQTT bridge. The Hub connects to all other IoT devices.

There may exist network issues that might render unpredictable network availability. Hence, the offline data storage and communication strategy is most essential. As a precautionary measure a local copy of the data is saved in a portable machine.

While MQTT was implemented for communication in the server side for online communication, offline communication was achieved using intranet.

Our project employs the MQTT protocol for IoT for file and message transfer among a few individual workstations and the main. All the strategies mentioned in the online communication structure is employed barring the use of a cloud platform. Even though the offline communication strategy is not explicitly modelled, the main server acts as a localized, centralized database for all the parameter and result comparison files.

In [10] a novel security protocol has been proposed. This has the base as the existing IoT communication protocol of MQTT coupled with advanced security features and has been coined as SMQTT – Secure MQTT.

The security has been provided using Cipher text policy based Attribute based Encryption (CP-ABE) or Key policy based Attribute based Encryption (KP-ABE). Here, a new publish service named ‘Spublish’ is provided that uses the message type of ‘0000’. This message is encrypted using ABE. This ABE scheme is lightly based on Elliptic Curve Cryptography (ECC).

Our project has a similar premise differing in the use of a mix of AES and RSA encryption-decryption algorithm developing a secure MQTT protocol for IoT. Our project includes security over MQTT in an industrial manufacturing setting.

In [11] the entire payload transmitted and received in the IoT operation over the MQTT protocol has been encrypted using Advance Encryption System (AES) cryptography and the key used for AES Encryption and decryption has been encrypted using the ABE scheme. The main function of ABE other than secure key management is to make sure that the lengths of the ciphertext and payload are same.

Our project takes up the use of AES cryptography algorithm for encrypting the files and messages transferred among the main server and clients and the public key used for this encryption is then encrypted using RSA algorithm instead of ABE. This ensures the security over the MQTT protocol in the IoT operation.

In [12] another secure MQTT protocol has been developed using RSA and AES encryption decryption algorithm. Limited usability of the Elliptic curve algorithm has also been used. Here it is stated that the only security that is provided to the communication between devices in the MQTT protocol is via SSL and TLS protocols. How secure they are in normal communication, in MQTT based IoT operation they are not deemed very secure. The main reason being, the security provided is not extended to the broker level as the entire data in the broker is published to all the subscribing candidates. The subscribers have authorized access to all the broker data irrespective of the fact that they require them or not.

The authentication of the subjects is done using certificates obtained from a Certificate Authority (CA). If a new subject wishes to join the relationship a request is sent to the CA and the certificate is obtained after due processing by the CA. This transmission of requests, certificates and acknowledgements are clubbed together as “Messages”.

These back and forth transferred messages are encrypted using the hybrid of symmetric AES and asymmetric RSA algorithms. The main object/message is encrypted using the symmetric public key generated for AES encryption whereas, the symmetric public key is encapsulated using the asymmetric public key generated using RSA algorithm.

This paper though is the main source for our project is different in a few aspects such as, it is more of a comparison study between encryption using RSA and Elliptic Curve (EC) and not the actual implementation. Moreover, the messages that are encrypted in this paper are the communication between the broker, subjects, and the CA whereas, our project deals with the encryption decryption of parameter files, acknowledgment messages, service requests, result files, etc. among the main server and individual stations.

[13] points out three major problems in the existing protocols for IoT operation.

Firstly, and most importantly, the Public key infrastructure (PKI) employed for security in IoT has a significant overhead.

Secondly, the Wireless Sensor Network (WSN) receives the full-frontal blow of the attack via the Internet.

Thirdly, as IoT uses a centralized publish/subscribe feature protocol such as MQTT there exists a serious performance bottleneck.

As a solution, a secure MQTT-like protocol has been provided in this paper that eliminates the use of PKI and instead employs identity-based cryptography. This provides authentication and is successful in isolating the WSN from the internet by creating a trust zone. This provides enhanced security in the IoT communication and at the same time diminishing the additional overhead of the public key infrastructure.

The model protocol is deployed in a local gateway instead of a centralized server as is generally the case in MQTT protocol. This is accomplished in a distributed method so as to increase the speed of operation by decreasing the local access delay and overcoming single point failure. The introduced publish/subscribe protocol is termed AIPS or Another Identity-based Publish/Subscribe protocol. In order to develop this protocol, the security operations were simplified using identity-based cryptography (IBC). As a result, the protocol became optimized for the resource-constrained wireless sensors. Here comes the concept of the trust zone mentioned previously which basically describes the scope of the network which is controlled by IBC.

There may be an existing third party through which the data between the broker and the subscribers flow, namely, another server inside an organization among others. This scheme gets rid of the third party and ensures a trusted dataflow directly to and from the broker and the subscribers. This helped in relieving the IoT server performance bottleneck.

Although this paper points out the issues with MQTT and symmetric and asymmetric key based encryption-decryption algorithms to ensure secured data communication among the server and the clients all of which are done in our project, it also points out that MQTT is more simple and efficient publish/subscribe protocol that has been designed for Machine-to-Machine (M2M) communications than their proposed AIPS irrespective of the performance bottleneck. Moreover, the paper also lightly points out that the used identity-based encryption schemes techniques have high computational cost and are not very efficient as AES or RSA. As a result, the authors developed a new IBC-based key negotiation technique. Our project acknowledges the raised concerns and provides a modified structure for security infrastructure while implementing MQTT as the IoT protocol and a hybrid of AES and RSA algorithm providing increased security from the attacks from the internet not only to the server and data but also to the wireless sensor networks in the individual devices.

**D. Research Design and Methods**

A screenshot of a map

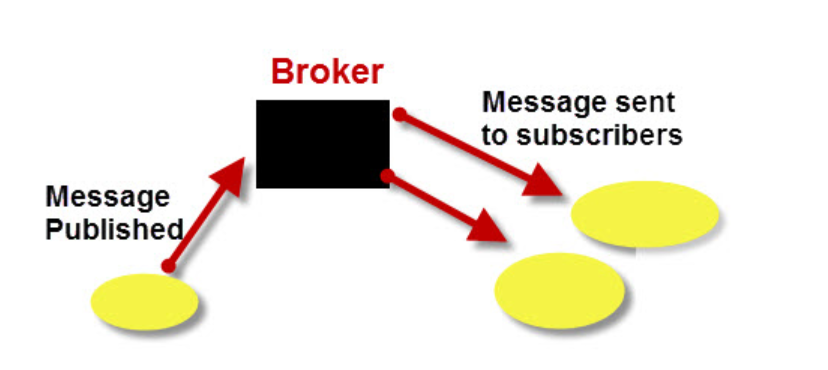
Description generated with very high confidence

**Fig. 2. 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 to send the file, the station then decrypts the files and performs its operation, next 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.

**MQTT:**



**Fig. 3. Basic block diagram of MQTT protocol**

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](http://searchstorage.techtarget.com/definition/byte)), 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.

Main Server

Requested File, Fail or Pass

Requested File, Fail or Pass

Request File, Result Value, Ack Done

Request file, Result Value, Ack Done

Station 1

Station 2

Ack Done

**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 also to the main server. Now once station 2 receives this Acknowledgement it starts with its operations like station 1.

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.

**E. Implementation details or Test Bed**

Along with MQTT our system uses AES and RSA encryption. The process at both the Main server end and the Individual station end is as described below.

**Encryption:**

Plaintext

AES Encryption

RSA Key Encryption

Client Public Key

Encryption Key

Ciphertext

**Fig. 5. The encryption process**

**Decryption:**

Ciphertext

RSA Key decryption

Client Private Key

Key

AES Decryption

Plaintext

**Fig. 6. The decryption process**

The encryption and decryption process are as shown above.

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](http://en.wikipedia.org/wiki/Advanced_Encryption_Standard)(Advanced Encryption Standard) 256 bit Encryption for file encryption and [RSA](http://en.wikipedia.org/wiki/RSA) public key encryption to encrypt the Symmetric [AES](http://en.wikipedia.org/wiki/Advanced_Encryption_Standard)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.

|  |  |  |
| --- | --- | --- |
| **RSA Encrypted AES symmetric key** | **IV** | **AES Encrypted File** |

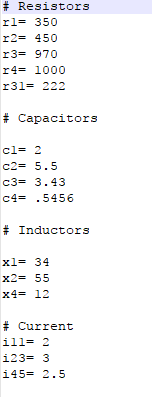
**Fig. 7. Ciphertext as obtained**

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.

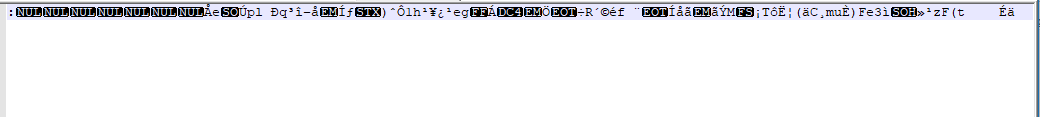
In our system, the server sends the respective encrypted parameter files to the broker, the individual stations subscribe to the respective parameter files, decrypt these and perform their respective operations and sends the encrypted result back to the server, the server now decrypts and checks this 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.

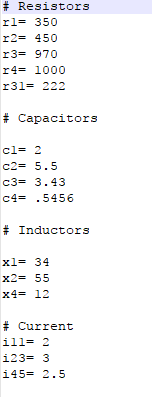
The robustness of our algorithm and the system has been tested by a display of encryption and decryption of the parameter files by the main server and the subsequent stations respectively. The respective parameter files are encrypted during the transfer to the client side which are then decrypted at the client end. The encrypted and decrypted files for the said “Station 1” are provided as follows:



**Fig. 8. The original parameter file**

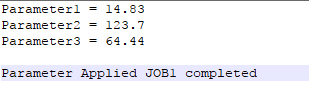


**Fig. 9. The encrypted file**



**Fig. 10. The decrypted file at station 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. 9, the decrypted file at the server end is given as follows:



**Fig. 11. The decrypted file at server end**

**Attack:**

To simulate an attack on our system we decided to use a case where an attacker is able to subscribe to all topics of the broker. In this case even if the attacker is able to get all the files of the topics the attacker gets only the encrypted versions of the files and will not be able to decrypt the file as the attacker does not have the private key of the stations associated with the respective topics. Also since the broker sends out the files to all subscribers subscribed to the topic the authentic stations also receive the files and thus will not be denied access to the required files.

**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