INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

IoT security with MQ telemetry transport

Punam Lokhande
punamtawade@gmail.com
Alamuri Ratnamala Institute of Engineering and Technology, Thane, Maharashtra

ABSTRACT

With the ever increasing number of connected devices and the overabundance of data generated by these devices, data privacy has become a critical concern in the Internet of Things (IoT). One promising privacy-preservation approach is Attribute-Based Encryption (ABE), a public key encryption scheme that enables fine-grained access control, scalable key management and flexible data distribution. We evaluate two major types of ABE,B Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE ABE is best suited for use in the IoT.

The main aim of the project is to applying security measures to the pub-sub architecture using the encryption scheme of KP-ABE. KP-ABE mechanism emerges as one of the most suitable security scheme for asymmetric encryption. It has been widely used to implement access control solutions. We are implementing AES cryptography on the pay load. We intend use cipher key to generate dynamic S-Box that is changed with every changing of cipher key. That cause increasing the cryptographic strength of AES algorithm.

For the purposes of privacy measures on IOT based generic pub-sub architecture, we implement Key-Policy Attribute Based Encryption(KP-ABE) scheme, With the emergence of sharing confidential corporate data on cloud servers, it is imperative adopt and efficient encryption system with a fine grain access control to encrypt outsourced data. Key Policy Attribute Based Encryption (KP-ABE) scheme is designed for one to many communications. KP-ABE scheme can be achieve fine grain access control and more flexibility to control users. In this paper we are going to enhanced KP-ABE Access Control to Encryptor can decide who can Decrypt data.

Keywords: *Advanced Encryption Standard, AES, Dynamic S-Box Generation, S-Box, Plain text, cipher text, Block cipher, Inverse S-Box.*

1. INTRODUCTION

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol that provides resource-constrained network clients with a simple way to distribute telemetry information. The protocol, which uses a publish/subscribe communication pattern, is used for machine-to-machine (M2M) communication and plays an important role in the Internet of Things (IoT).MQTT allows devices to send (publish) information about a given topic to a server that functions as an MQTT message broker. The broker then pushes the information out to those clients that have previously subscribed to the client's topic. To a human, a topic looks like a hierarchial file path. Clients can subscribe to a specific level of a topic's hierarchy or use a wildcard character to subscribe to multiple levels. MQTT is a good choice for wireless networks that experience varying levels of latency due to occasional bandwidth constraints or unreliable connections.

MQTT is a Client Server publish/subscribe messaging transport protocol. It is light weight, open, simple, and designed so as to be easy to implement. These characteristics make it ideal for use in many situations, including constrained environments such as for communication in Machine to Machine (M2M) and Internet of Things (IoT) contexts where a small code footprint is required and/or network bandwidth is at a premium. It is very light weight and binary protocol, which excels when transferring data over the wire in comparison to protocols like HTTP, because it has only a minimal packet overhead. Another important aspect is that MQTT is extremely easy to implement on the client side. This fits perfectly for constrained devices with limited resources.

Traditional Client/Server communication model employs RPC, message queue, shared memory etc. Synchronous, tightly-coupled request invocations. Very restrictive for distributed applications, especially for WAN and mobile environments. When nodes/links fail, system is affected. Fault Tolerance must be built in to support this. Require a more flexible and decoupled communication style that offers anonymous and asynchronous mechanisms.

The publish/subscribe pattern (pub/sub) is an alternative to the traditional client-server model, where a client communicates directly with an endpoint. However, Pub/Sub decouples a client, who is sending a particular message (called publisher) from another client (or more clients), who is receiving the message (called subscriber). This means that the publisher and subscriber don't know about the existence of one another. There is a third component, called broker, which is known by both the publisher and subscriber, which filters all incoming messages and distributes them accordingly. So let's dive into a little bit more details about the just mentioned aspects.

2. METHODOLOGY

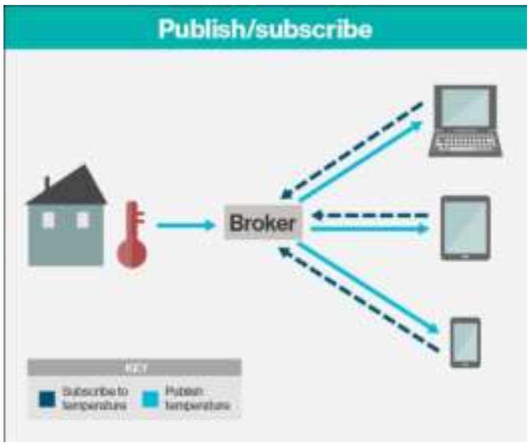

Fig.1. Block Diagram: PUB/SUB SYSTEM

Pub/Sub System:

Distributed Pub/Sub System is a communication paradigm that allows freedom in the distributed system by the decoupling of communication entities in terms of time, space and synchronization.

An event service system that is asynchronous, anonymous and loosely-coupled.

Ability to quickly adapt in a dynamic environment.

3. COMPONENTS

Publishers: Publishers generate event data and publishes them.
Subscribers: Subscribers submit their subscriptions and process the events received P/S service: It's the mediator/broker that filters and routes events from publishers to interested subscribers.

Various approaches have been proposed for providing security in publish-subscribe system: Mudhakar Srivatsa, et.al have proposed Event Guard, a reliable design that protects the publisher subscriber work from various types of attack. It provides security characteristics that are critical to publisher and subscriber overlay network benefits, like integrity, confidentiality, authenticity, providing flexibility to continuous flood based denial of service attack. Jean Bacon David, et.al tells the needs of numerous application areas where the eventbased design is suitable but security is matter of concern. They have discussed about how securitycan be provided to publish subscribe system; first they tell about the access control policy at service API and then apply it, and they then apply the various security form of these policies in the service network.

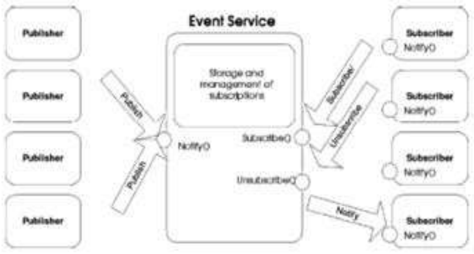

Fig.2. Block Diagram

Publishers publish events, and subscribers subscribe to and receive the events they are interested in. The main characterization of pub/sub is in the way notifications flow from senders to receivers. Receivers are not directly targeted from publisher but indirectly addressed according to the content of notifications. Subscriber expresses its interest by issuing subscriptions for specific notifications, independently from the publishers that produces them, and then it is asynchronously notified for all notifications, submitted by any publisher, that match their subscription.

One of the main challenges that pub/sub systems are still facing is protecting the confidentiality of the exchanged information without limiting the decoupling of the paradigm. Publishers and subscribers do not establish contact so they cannot exchange keying material. Moreover, protecting the confidentiality from malicious brokers is very difficult. Brokers should be able to route events by matching them against filters expressed by the subscribers without having access to the actual content of events and filters.
The payload received form the pub-sub system is being taken as an input and been encrypted using encryption standards of AES using KP-ABE schema.

4. AES ENCRYPTION

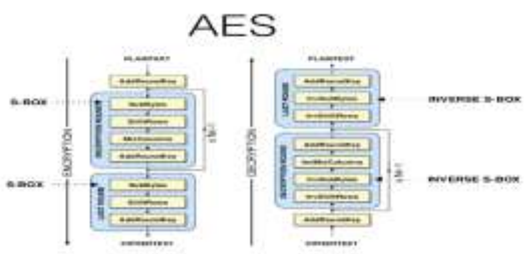

Fig.3. AES ENCRYPTION SCHEME

This standard specifies the Rijndael algorithm, asymmetric clock cipher can process data blocks of 128 bits, using cipher key with lengths of 128, 192, and 256 bits. Rijndael was designed to handle additional block sizes and key length, however they are not adopted in this standard.

DEFINITIONS:

Cipher: Series of transformations that converts plaintext to cipher text using the Cipher Key.

- **Cipher Key:** Secret, cryptographic key that is used by the Key Expansion routine to generate a set o f Round Keys; can be pictured as a rectangular array of bytes, having four rows and Nk columns.
- **Ciphertext**: Data output from the Cipher or input to the Inverse cipher.
- **Plaintext:** Data input to the Cipher or output from the Inverse Cipher.
- **S-Box**: Non-linear substitution table used in several byte substitution transformations and in the Key Expansion routine to perform a one-for-one substitution of a byte value.

5. ALGORITHM PARAMETERS, SYMBOLS AND FUNCTIONS

The following algorithm parameters, symbols, and functions are used throughout this standard:

- AddRoundKey(): Transformation in the Cipher and Inverse Cipher in which a Round Key is added to the State using an XOR operation. The length of a Round Key equals the size of the State.
- MixCloumns(): Transformation in the Cipher that takes all of the columns of the State and mixes their data (independently of one other) to produce new columns.
- Nb: Number of columns (32-bit words) comprising the State. - Nk: Number of 32-bit words comprising the Cipher Key.
- Nr: Number of rounds, which is a function of Nk and Nb (which is fixed).
- RotWord(): Function used in a Key Expansion routine that takes a four-byte word and performs a cyclic permutation.
- ShiftRows (): Function is the Cipher that processes the state by cyclically shifting the last three rows of the State by different offsets.
- SubBytes(): Transformation in the Cipher that processes the State using a non-linear byte substitution table (S-Box) that operates on each of State bytes independently.
- SubWord(): Function used in the Key Expansion routine that takes a four-byte input word and applies an S-Box to each of the four bytes to produce an output word.

Cipher Algorithm From the beginning of the Cipher, the input is copied to State array. After an initial Round Key addition, the State array is transformed by implementing a round function 10, 12, or 14 times(depending of key length), with the final round differing slightly from the first Nr-1 rounds. The final State is then copied to the output. The Cipher is described in the pseudo code in algorithm.

```
public word[] Cipher(byte[] plainText, byte[] cipherKey)
 {
state = new word[4];
sBox = new newSbox(cipherKey);
ks = new KeySchedule(cipherKey);
for (int i = 0; i < 4; i++)
```

```
 {
for (int j = 0; j < 4; j++)
 {
if (state[j] == null) state[j] = new word();
 state[j].w[i] = plainText[i * 4 + j];
 }
 }
AddRoundKey(0);
 for (int i = 1; i < Nr; i++)
 {
 SubBytes();
ShiftRows();
 MixColumn();
 AddRoundKey(i);
 }
SubBytes();
 ShiftRows();
 AddRoundKey(Nr);
 return state;
 }
```

6. THE SUBSTITUTION BOX (S-BOX)

Substitution is a nonlinear transformation which performs confusion of bits. A nonlinear transformation is essential for every modern encryption algorithm and is proved to be a strong cryptographic primitive against linear and differencial cryptanalysis. Nonlinear transformations are implemented as lookup tables (S-Boxes). An S-Box with p input bits and q output bits is denoted p * q. The DES uses eight 6 * 4 S_boxes. S-Boxes are designed for software implementation on 8-bit processors. The block ciphers with 8 * 8 S-Boxes are SAFER, SHARK, and AES. For processors with 32-bit or 64-bit words, S-Boxes with more output bits provide high efficiency. The Snefru, Blowfish, CAST, and SQUARE use 8 * 32 S-Boxes. The S-Boxes can be selected at random as in Snefru, can be computed using a chaotic map, or have some mathematical structure over a finite Galois field. Examples of the last approach are SAFER, SHARK, and AES. S-Boxes that depend on key values are slower but more secure than key independent ones (Schneier,1996). Use of key independent chaotic S-Boxes are analyzed in which the S-Box is constructed with a transformation F ((X + K) modM), where K is the key

7. DYNAMIC S-BOX GENERATION FROM CIPHER KEY ALGORITHM

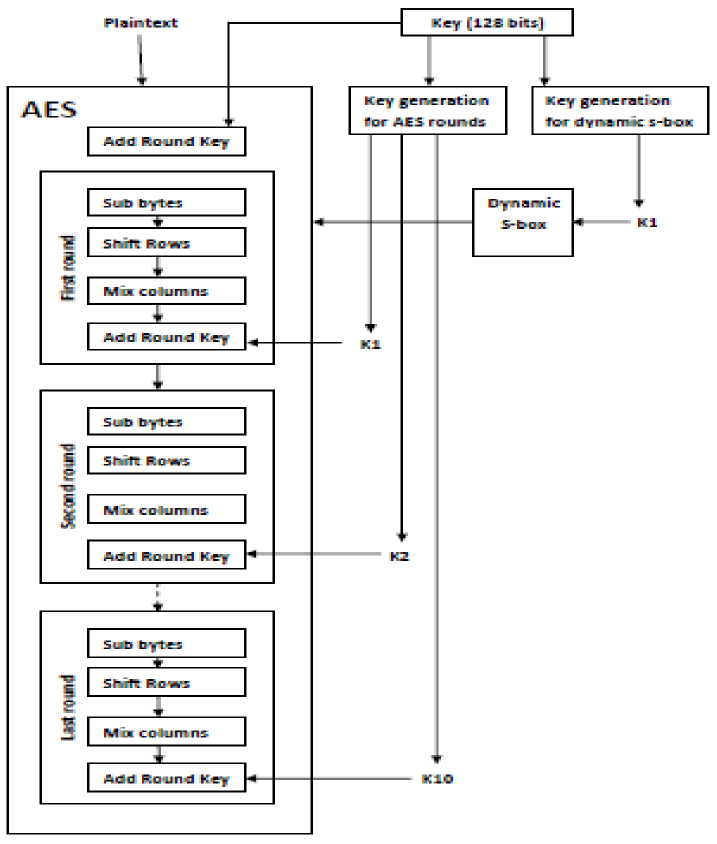

Fig.4. DYNAMIC S-BOX GENERATION

1. First Step We need primary S-Box to generate dynamic S-Box,
We use S-Box generation algorithm that introduced in AES, to create primary S-Box as follows. Take the mulltiplicative inverse in the finite feild GF(28); the element {00} is mapped itself.

2. Second Step In this step, rows swapped with columns of primary S-Box in GenerateDynamicSbox(cipherKey) function. This function guarantees new S-Box remain onefor-one. This routine get cipher key as input and generate dynamic S-Box from cipher key. Note that in this paper if cipher key has 192 or 256 bits size, we use only first 128 bits of cipher key.

GenerateDynamicSbox Algorithm:

```
void GenerateDynamicSbox(byte[16] key)
 {
 byte rowIndex, columnIndex;
 byte shiftCount = GetShiftCount(key);
 byte[,] sBox = GeneratePrimarySbox();
for(int i = 0; i < 16;i++)
 {
 GetProperIndex(key[i], out rowIndex, out columnIndex);
ShiftRow(rowIndex, shiftCount, sBox);
ShiftColumn(columnIndex, shiftCount, sBox);
 Swap(rowIndex, columnIndex, sBox);
 }
}
```

We used an algorithm to generate dynamic S-Box from cipher key. The quality of this algorithm tested by changing only two bits of cipher key to generate new S-Boxes.
For that purpose we are testing difference of S-Box element by many intervals. This algorithm will lead to generate more secure block ciphers, solve the problem of the fixed structure S-Boxes and will increase the security level of the AES block cipher system. The main advantage of this algorithm is that many S-Boxes can be generated by changing Cipher key.

8. ADVANTAGES

Highly suited for mobile applications, ubiquitous computing and distributed embedded systems

Security-Making the pub-sub architecture more secure
Scalability- Suited to build distributed applications consisting a large number of entities
Adaptability- can be varied to suit different environments (mobile, internet game, embedded systems etc…)

9. CONCLUSION

By implementing KP-ABE using dynamic s box AES encryption standards the project is helping making a security structure on the public-subscribe architecture used in the MQTT.

10. FUTURE S COPE

Implementing strong guarantee on broker to deliver content to subscriber.

After a publisher publishes the event, it assumes that all corresponding subscribers would receive it.

Solving Potential bottleneck in brokers when subscribers and publishers overload them   by load balancing techniques

Encryption   hard to implement when the brokers has to filter out th events according to context.

11. REFERENC ES

[1] J. Zhang, Q. Li, and E. M. Schooler, “iHEMS: An Information-Centric Approach to Secure Home Energy Management,” in *Proc. of IEEE SmartGridComm*, 2012.
[2] M. Ion, J. Zhang, and E. M. Schooler, “Toward content-centric privacy in icn: Attribute-based encryption and routing,” in *Proc. 3rd ACM SIGCOMM Workshop on ICN*, ICN ’13, pp. 39–40, 2013.
[3] T. Hardjono and B. Weis, “The Multicast Group Security Architecture.” RFC 3740 (Informational), Mar. 2004.
[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proc. of ACM CCS*, pp. 89–98, 2006.
[5] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attributebased encryption,” in *Proc. of IEEE SP*, pp. 321–334, 2007.
[6] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in *Proc. of IEEE INFOCOM*, pp. 1–9, 2010.
[7] G. Wang, Q. Liu, and J. Wu, “Hierarchical attribute-based encryption for fine-grained access control in cloud storage services,” in *Proc. Of ACM CCS*, pp. 735–737, 2010.
[8] M. Li, S. Yu, K. Ren, and W. Lou, “Securing personal health records in cloud computing: Patient-centric and fine-grained data

access control in multi-owner settings," in *Security and Privacy in Communication Networks*, pp. 89–106, Springer, 2010.
[9] Z. Wan, J. Liu, and R. H. Deng, "Hasbe: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, 2012.
[10] N. Koblitz and A. Menezes, *Pairing-based cryptography at high security levels*. Springer, 2005.
[11] B. Lynn, *On the implementation of pairing-based cryptosystems*. PhD thesis, Stanford University, 2007.
[12] M. Ion, *Security of Publish/Subscribe Systems*. PhD thesis, University of Trento, 2013. http://eprints-phd.biblio.unitn.it/993/.
[13] Performance Evaluation of Attribute-Based Encryption: Toward Data Privacy in the IoT by Xinlei Wang, Jianqing Zhang, Eve M. Schooler, Mihaela Ion at IEEE ICC 2014 - Communication and Information Systems Security Symposium