ATTRIBUTE BASED ENCRYPTION FOR IOT

V Anushka[1] , V Sandhya[2], Sourabha R[3], Vaishali Kasnoor[4]

1 Student, Information Science and Engineering, National Institute of Engineering, Karnataka, India

2 Student, Information Science and Engineering, National Institute of Engineering, Karnataka, India

3 Student, Information Science and Engineering, National Institute of Engineering, Karnataka, India

4 Student, Information Science and Engineering, National Institute of Engineering, Karnataka, India

ABSTRACT

IoT has enormous amount of data generated and the need for security of this data is also increasing. Attribute based encryption is more efficient when security needs to be provided to the data which will be accessed by a large number of users. In this system, there is no need of maintaining the public keys of all the users. Data can only be decrypted if and only if the access policy associated with the ciphertext matches.

Keyword : *ABE, IoT, CP-ABE, access tree, access policy, Key server*

1. INTRODUCTION

In the recent years IoT is considered pivotal for smart technology as it can be used for several applications such as smart cart, smart healthcare. Large quantities of data are produced through deployment of a large number of IoT devices thus leaving new challenges to application providers with respect to security. Cryptographic encryption technique can be used to achieve Data Privacy. Hence Fine-grained access control mechanism called attribute based encryption was introduced that could be the solution to manage a large number of devices generating trickles of data. Attribute based encryption is a type of algorithm of public key cryptography in which the private key used to decrpyt data is dependent on certain user attributes such as position, place of residence, type of account. It defines policies that provides access control based on the attributes that are associated with it. Here, policy is bound to the key and set of attributes are associated with encrypted data. Each user is given a (key,policy) pair. At least k attributes of the ciphertext and user's keys need to overlap in order to decrypt the ciphertext

The types of attribute based encryption are: Key Policy Attribute Based Encryption and Ciphertext paper policy Attribute Based Encryption. In KP-ABE, data is encrypted with attribute sets and an access tree is used to generate the secret key of the user and the access key specifies the concerned user's scope. In CP-ABE, data is encrypted using the access tree and secret key of the user is generated over a set of attributes.

2. ACCESS TREE

Policy to access the data is defined using access tree. AND,OR operators are used for defining policy. For example, if a file must be accessible only to head of the department of computer science or it can only be accessed by dean, the policy is defined in the access tree as follows:

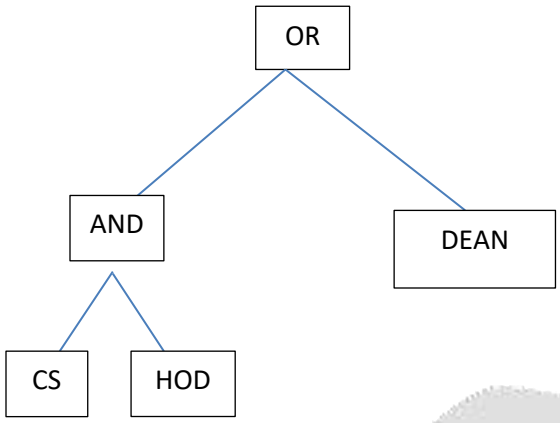

Fig 2.1:Access Tree

.

3. DESIGN

In this paper, we propose an authorization system which uses attribute based encryption for applications which involve multiple users who need to access only specific data and the users not authorized to use a particular data shouldn't be allowed access. Here, we use a case where different users need to access data being generated at an IOT device and only specific users can access the data being generated at specific sensors. The major components of the system are a key server, an IoT device and a cloud server. The Iot device is connected to sensors and only specific users can access the data being generated at specific sensors.

Each user is provided with a portal to access the data. This portal can be accessed by the users by registering with the administrator. The user then is authenticated by the administrator and is granted access. Once the administrator provides the password to the user, an SHA key is generated using the username, password and the machine name of the machine designated to that user and is stored in a database and this database is stored in the key server. Whenever the user tries to access his portal using the username and password, an SHA key is generated using the username and password entered by the user and the machine name of the machine through which he is accessing and this key is compared to the key stored in the database. If the keys match, the user is granted access.

Every sensor in the system is uniquely identified by a sensor ID.A set of attributes is generated based on this ID. These attributes are used to encrypt the data. The encrypted data, along with the associated sensor ID is stored in the cloud server through the key server. The key server contains a list of attributes associated with the sensor ID. For each sensor, the users who can access the data being generated at that sensor are sent a particular number of attributes. The attributes sent are such that, using these attributes alone, the user must be able to decrypt the data. Different users accessing data from the same sensor need not be given the same set of attributes. When a user requests the data being generated at a particular sensor, the cloud server requests the key server to send the attributes associated with this sensor. The key server then sends the attributes to the cloud server. The cloud server then along with the encrypted data sends these attributes to the user. At the user site, the attributes received are compared with the attributes available with the user for this specific sensor. If the attributes match, then the data can be encrypted and this data is now available to the user to view.

In our system, we have considered a temperature sensor and an LDR(Light Dependent Resistor) sensor. A temperature sensor measures the temperature through an electrical signal. An LDR sensor is a device whose resistivity is a function of the incident electromagnetic radiation. The measurements from these sensors are read through Arduino and Raspberry Pi3 device. Arduino uno is a single board microcontroller. Code can be compiled on it and it will execute according to this code. Even though arduino supports multiple accessories, it has less RAM and flash memory. Hence multiple lines of code cannot be uploaded to it. An open-source hardware board designed based on an 8-bit Atmel AVR microcontroller, or 32-bit Atmel ARM makes the hardware of Arduino.

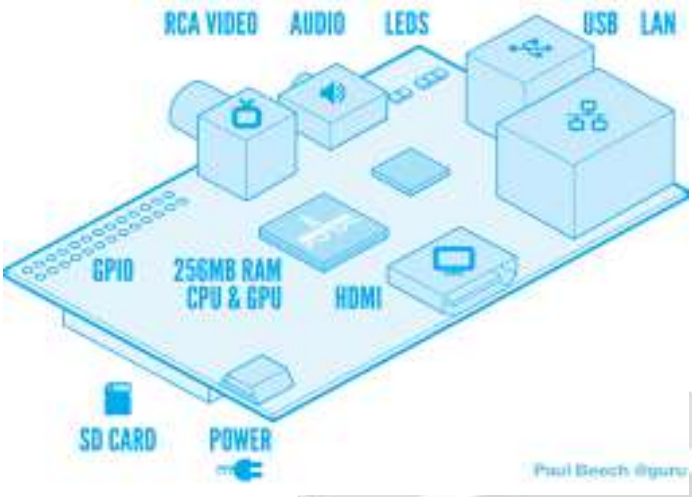

Fig 3.1: Raspberry pi board

A boot loader is preprogrammed into the microcontroller. This allows the programmer to upload programs into the microcontroller memory. without the necessity of a chip programmer. Here, we connect the temperature sensor and the LDR to Arduino. This Arduino is then connected to a Raspberry Pi 3. This is because Arduino has 'real-time' and 'analog' capability and Raspberry Pi does not. This extent of flexibility allows it to work with almost any kind of sensors. But for programming Arduino, there is no API and there is no operating system, hence there is no interactivity if we don't create it. It is running code on bare metal.

When programming a Raspberry Pi, on the other hand, it is like computer programming. Multiple programs can be run together, Linux API can be used, and you can interact with the program with a keyboard and mouse, and the process can be viewed on the monitor. Raspberry Pi is a small single board computer. Even though the Raspberry Pi Foundation recommends Python for learners, a language which can compile for ARMv6 (Pi 1)/ARMv7 (Pi 2) can be used with the Raspberry Pi. C++, Java, Scratch, and Ruby all come installed by default on the Raspberry Pi. In our system, we use Java to connect the raspberry Pi to the Arduino device and also to an FTP server. The data being collected from the sensors, through Arduino and Raspberry Pi and stored in a file on this FTP cloud server. This data is encrypted using the AES encryption system. Whenever the administrator requires this data, this data can be pushed from this FTP server to the cloud server, where it is decrypted and stored.

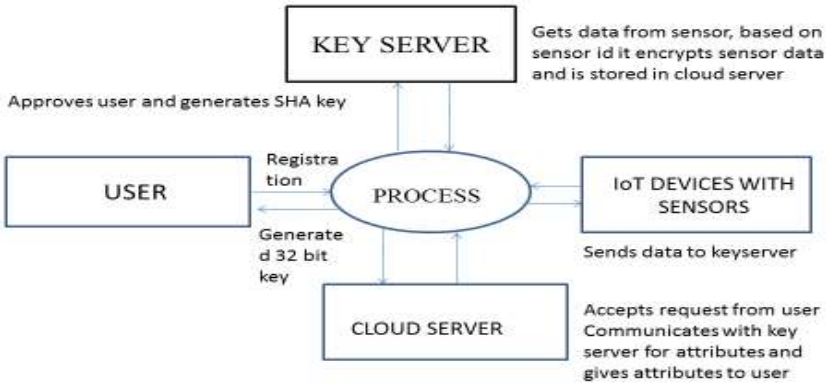

Fig 3.1: Flow Diagram

4. CONCLUSIONS

Attribute based encryption provides fine grained access control mechanism. Since data is decrypted based on the attributes associated with the ciphertext, there is no overhead of management of keys and hence provides more scalable approach.

6. REFERENCES

[1]A. Sahai and B. Waters," Fuzzy Identity Based Encryption".In Carmer R.(eds) Advances in Cryptology-EUROCRYPT 2005.Lecture Notes in Computer Science,vol3494.Springer,Berlin,Heidelberg

[2] Goyal,Pandey,Sahai and Waters,"Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data" in13th ACM conference 2006

[3] John Bethencourt,Amit Sahai,Brent Waters "Ciphertext-Policy Attribute-Based Encryption" inUniversity of california

[4]S. SM. Chow,"A Framework of Multi-Authority Attribute-Based Encryption with Outsourcing and Revocation" in SACMAT'16 Proceedings of the 21st ACM on Symposium in Access Control Models and Technologies.

[5] L. Ming et al., "Data security and privacy in wireless body area networks", in IEEE Wireless Communication, 2010.

[6] M. Ambrosin et al.,"On the feasibility of attribute-based encryption on smart phone devices" in IoT-Sys'15 Proceedings of 2015vWorkshop on IoT challenges in Mobile and Industrial applications.

[7]https://www.ijarcsse.com/docs/papers/Special_Issue/ITSD2015/9.pdf

[8] http://ieeexplore.ieee.org/document/6997578/