ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

Secured Transfer of IoT Data Using Attribute Based Encryption

[1] Pradyoth Hegde, [2] Dr. Yuvaraju B N,
[1][2] Department Of Computer Science and Engineering
The National Institute of Engineering, Mysuru

***Abstract*— In the present technologies It is considered as one of the key area of growth. IoT is growing its popularity and market everyday exponentially. Iota is spreading from small to large applications in all fields like Smart Cities, Smart Grids, and Smart Transportation. As the usability of the it is growing, security of the data has become one of the major areas of concern. In this project, we apply the attribute based encryption concept to encrypt the data. Also we make use of cloud technology to store the data. Here since we store the data in encrypted format, it is very difficult to understand the data. The key is stored in the key server and is given only to the authorized users. Thus this project ensures the security of the sensor data.**

***Index Terms*—Attribute based encryption, Internet of things, Cloud Technology.**

I. INTRODUCTION

This project is about securing the data coming from the IoT device. The device sends the data to a server called key server. This server encrypts the sensor data with advanced encryption standard algorithm using the attribute based encryption, i. e. using attribute as a key for encryption. Then the encrypted data is stored in the cloud server. This can be accessed by the user. To access, the user must first register himself. Later he can login and retrieve the data. The retrieved data will be encrypted, so along with the encrypted sensor data, attribute keys are also sent to the user to decrypt and get back the sensor data.

In this project there are majorly four modules:

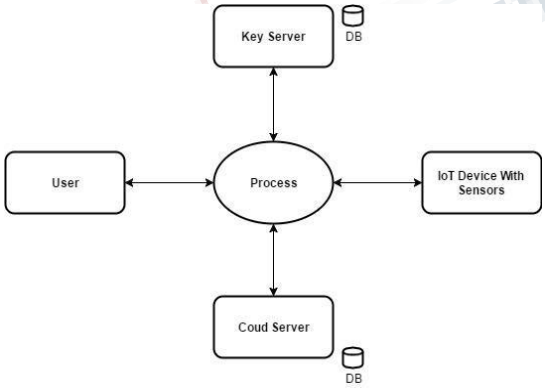

Fig.1: Architecture

- ***IoT device***: IoT devices senses different data from the sensors like light sensors, temperature sensor etc and sends the information to the cloud server. IoT device also interact with the key server for the exchange of the attribute keys for the encryption of the data.

- ***Key server***: Key server which holds the information about the registered IoT devices and the registered end users. This holds the database of the attribute keys and users and used for authentication purpose.

- ***Cloud server:*** Cloud server stores the encrypted data in its database. The data is fed by the IoT device and is accessed by the user.

- ***End user***: End user is a person who wants to access the sensor information from the cloud server.

II.LITERATURE REVIEW

A. Internet of Things

The notion Internet of things (IoT) is currently getting a lot of attention and first real usages are taking place in real world scenarios. Gartner Inc. [1] predicts that by 2020 there will be 26 billion units installed in IoT products. Although to make IoT solutions ready for production-level quality outside of controlled lab environment multiple issues needs to be solved. Security and privacy challenges are considered to be one of the most crucial. The development of the Internet of Things has been

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

primarily driven by needs of large corporations that stand to benefit greatly from the foresight and predictability afforded by the ability to follow all objects through the commodity chains in which they are embedded [2].

B. Cloud Technology

Cloud Computing is a disruptive technology with profound implications for the delivery of Internet services as well as for the IT sector as a whole. The two worlds of Cloud and IoT have seen a rapid and independent evolution. These worlds are very different from each other and, even better, their characteristics are often complementary. Such complementarity is the main reason why many researchers have proposed.

For managing the identity of the devices, the centralized identity store can be used. Each device has an associated account in the store with corresponding roles. For any communication the device retrieves OAuth 2.0 token and uses it to certify itself in every network connection. The proposed framework creates trusted environment and enables rapid response for any security events [3]. In general, IoT can benefit from the virtually unlimited capabilities and resources of Cloud to compensate its technological constraints (e.g., storage, processing and communication).

C. Attribute Based Encryption

Sahai and Waters [4] introduced attribute-based encryption (ABE) as a new means for encrypted access controlThe concept of ABE (Attribute-Based Encryption) in which data owners can insert how they want to distribute data in terms of encryption. That is, only those who match the owner's conditions can successfully decrypt stored data. We can say here that ABE is encryption for privileges, not for users. This makes ABE a very helpful tool for cloud storage services since data sharing is a significant feature for such services.

III.SYSTEM DESIGN

In this chapter, we define the use case diagram, data flow diagram and sequence diagram, of our application. The design of the system is essentially a blue print or a plan for a solution of the system.

A. Use case diagram

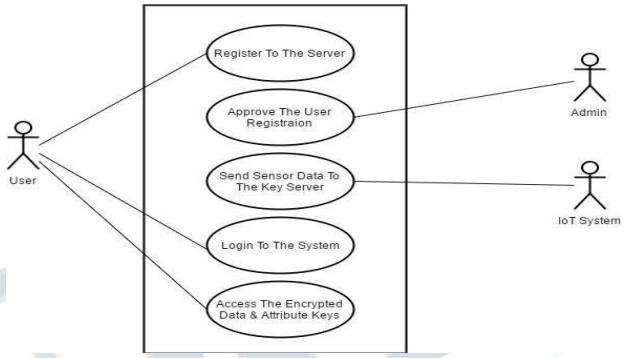

Fig.2: Use case diagram

In the above use case diagram, there are three actors namely user, admin, and IoT system. The user's role here is to register with the key server at first and then he can login to the system to access the data. After the user registers, the registration must be approved by the admin. IoT system doesthe job of sending the sensing the sensor data and sending it to the key server.

B. Data flow diagram

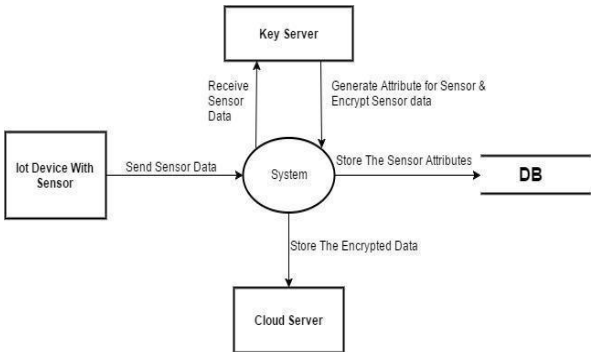

Fig.3: Data flow diagram for IoT device

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

This data flow diagram shows about the IoT device sending the data and finally storing it in the cloud server. Whenever the IoT device sends the data, it is received by the key server. The key server encrypts the data by using the attributes as the key. The key server uses advanced encryption standards to encrypt the data. The attribute keys are stored in the key server's local database. Further, the sensor data receiving from the IoT is sent to the cloud server in encrypted format.

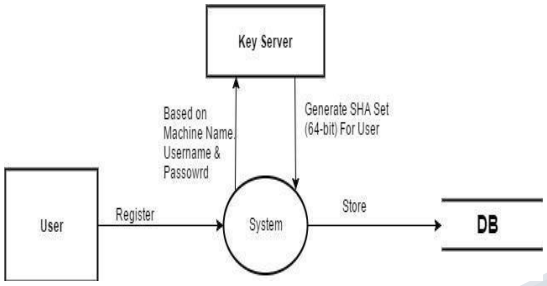

Fig.4: Data flow diagram for user registration

In this data flow diagram user registration process is shown. The user registers with username, password and machine name to the key server. Then the key server generates the 64-bit SHA set for the particular user based on his three inputs. After this, 64-bit SHA set is stored in the key server's database for future use.

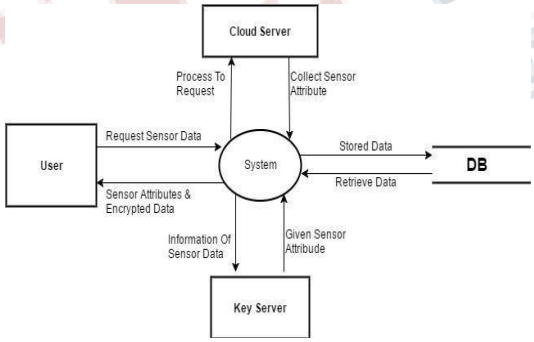

Fig.5: Data flow diagram for accessing the data

This fig. 5 shows how the user gets the data. Whenever the user makes the request for the sensor data, the cloud server processes it by collecting the attributes from the key server. Here the cloud sends the user information along with the request. The key server check for the authenticity of the user in the request by comparing to the 64-bit SHA set. If the user found to be authentic, the key server sends the attribute keys and cloud server sends the encrypted data. By these two the user decrypts the data to get back the original sensor data.

C. Sequence Diagram

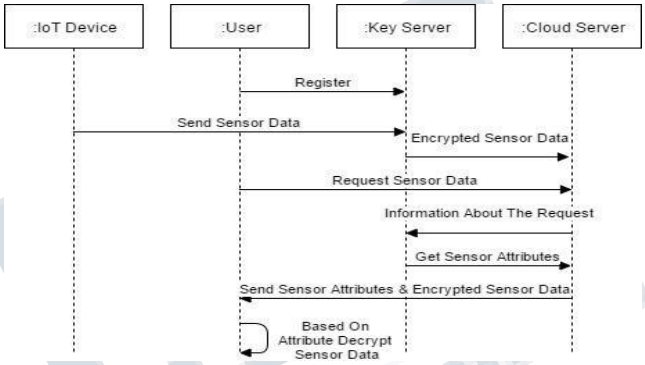

Fig.6: Sequence diagram

The above sequence diagram fig. 6, shows the occurrence of the events in the lifeline. First the user registers. Whenever the IoT device sends the data it is encrypted and forwarded by the key server. The user request is handled by the cloud server with the help of key server, finally to get back the data by decrypting at the user side.

IV.SYSTEM IMPLEMENTATION

In this chapter we briefly explain about the following modules.

A.User registration & login

First the user registers with the key server. This application is built on .NET framework using C# language. This windows form application allows the user to register with the key server. The key server takes 3 parameters, 1. Username 2. Password 3. Machine name. Username and password is entered by the user and the machine name is taken passed by the application. This entire connection is done in TCP because of security purposes.

In the login process user enters the username and password to the windows form application. When the login happens

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 3, March 2017

the machine name is also taken in the background. If the passed parameters are correct, then it receives encrypted sensor data along with the attribute keys. The decryption process is carried out with the keys to get back the sensor data. This whole process is done through establishing UDP connection.

B.Key server

In key server, the user registration request happens with user's username, password and machine name. The server generates the 64-bit SHA set by using SHA algorithm. This 64-bit SHA set is then stored in the key server database which consists of alphanumeric values.

Another job of key server is to process the data coming from the IoT device. The sensor data are encrypted here using Rijndael's algorithm. Rindael's algorithm is advanced encryption standard to which we use the sensor attribute as the key.

During the login process the cloud server communicates with key server to check the user. Even in this time username, password and machine name used to generate the 64-bit SHA set and it is compared with the one which is in the database for the particular user. Second time the same checking is done while asking for the attribute key. But this time attribute key is returned for the authenticated user. This whole process is built using .NET framework and C# language. The communication between entities are by establishing the UDP connection, except for registration which is connection oriented TCP.

C. IoT device

For IoT device we use Raspberry Pi. This board is used because of its simplicity. To this we connect several sensors like temperature sensor, humidity sensor etc. Raspberry Pi gives the option of connecting via USB, Wi-Fi, Bluetooth and GSM/GPRS. The connection established is UDP connection with the key server. Device senses the data from the sensors and sends to the key server in particular interval of time.

D. Cloud server

Cloud is a very reliable technology. For this project we use Google Cloud platform. Google cloud provides durability, availability and much more features with data storage. In google cloud we create a database which is used to store the data sent by the key server. These stored data will be encrypted. Whenever the user makes request with the cloud application for the sensor data, the application communicates with the key server for the attribute key and sends the encrypted data along with the key to the user.

V.APPLICATION OF THE PROJECT

This project mainly focuses on the security concerns. Since here a user can login only from a machine, makes sure that same user cannot login from other system. This provides private networks in order to prevent from the external attacks. The data stored in cloud is encrypted, so even if the data is hacked from the cloud it cannot mean anything unless they have the key. Key for the same is stored in different server making it dead hard to get the key. This architecture checks for user's authenticity twice when asking for sensor data. Such kind of systems can be used in offices and organizations which needs higher level of security of data.

REFERENCES

[1] Gartner Inc. (2013, July) Hype cycle for the internet of things, 2013. [Online]. Available: https://www.gartner.com/doc/2571716/hype-cycleinternet-things.

[2] Lianos, M. and Douglas, M. (2000) Dangerization and the End of Deviance: The Institutional Environment. British Journal of Criminology, 40, 261-278.

[3] Michal Trnka, and Tomas Cerny (2016) Identity management of devices in Internet of Things environment. IEEE – 2016

[4] A. Sahai and B.Waters. Fuzzy Identity Based Encryption. In Advances in Cryptology – Eurocrypt, volume 3494 of LNCS, pages 457–473. Springer, 2005.