Multikey Based Mutual Authentication Between IoT Device and IoT Server

Akshatha.M[1],Arpitha.A[2], Noushika.B[3] , Prathiksha.C.P[4] , Rampur Srinath[5]

The National Institute Of Engineering

Abstract

IoT is an emerging technology that will help to bridge the gap between the physical and digital spheres.Nowadays everyone show interest towards achieving maximum security for IoT devices but issues like cost consciousness, hardware limitations ,lack of security expertise ,attacks by the third party like side channel and dictionary attacks are common these days .As a result ,to avoid such issues and provide a much secure authentication between IoT device and IoT server we bring in Multi key Two way authentication mechanism .This can be done done using many approaches. Here, we propose an authentication mechanism based on secure vaults which is a collection of equal sized keys.

Keywords - *IoT security, secure vaults, dictionary and side channel attack.*

1.INTRODUCTION

Providing security for IoT devices is at its starting stages . In this paper we provide a conceptual idea to achieve a better security for IoT devices .The scheme used is known as two way authentication .Some of the major security issues faced by IoT devices are authentication , authorization ,data confidentiality ,privacy ,integrity ,vulnerability , availability of devices .

The domain of security attacks on IoT devices is increasing day by day . Some of the attacks can occur at any of the three layers of the IoT device which are : 1) Hardware layer, 2) Network layer and 3)cloud layer.

At the Hardware layer an intruder tries to gets access to the IoT hardware and once he is successful he reclaims the keys or security parameters kept within the IoT device. The intruder can recreate or duplicate the virtual IoT device using the reclaimed parameters. The duplicate IoT device can upload wrong data and retrieve the user's secure information from the server or the network to which the IoT device is connected.

In this paper, we try to provide a secure authentication protocol to achieve better security for the device to authenticate the IoT device and the server. Here we have designed a scheme so that we can eliminate the side -channel attack.This attack is a threat to IoT devices and emerging IoT infrastructure.It makes use of some or all of the information to recover the key that the device is using based on the fact that logical operations have physical charecterstics that depend on the input data . Addition to this attack dictionary attack can also be eliminated in which the attacker breaks the security systems and test all possible keys that have a higher possibility of being used .In order to overcome this attack ,we have designed the system in such a way that the key values in the secure vault will be changed over a specific amount of time as set .

2.SYSTEM MODULES

1). IoT distribute secure vaults session to IoT server:

In this module IoT system will generate set of secure tokens and same will be distributed to the IoT server for further data distribution.

2). User registration and approval module:

User register module auto fetch information from the machine and submit to the cloud and approval module from the admin side will authenticate user and generate hash key used during future request.

3). Request for IoT data and get short SV associated with vault along with hash :

Client request will be converted into secure request by combining machine details , secure vault code distributed during approval to identify machine request. This hash combination used to avoid network attack.

4). Validation of hash by IoT server:

In this module IoT server will compare hash generated and distributed to client is same or not , so that key will be extracted from the vault and submitted as IoT request for IoT data delivery.

5). Distribute IoT data and update vault:

Once the request received IoT server will distribute data and vault key will be
deleted from the list to avoid using same key for future request.

6). Working with IoT model , reading data, cloud sync, vault list tracker:

This is a kind of integration module for reading sensor values , maintaining suitable data structure for secure vault , sending data to cloud using MQTT protocol , Ec2 cloud setup using Amazon server .

3.SYSTEM DESIGN

In this paper,the system consist of three main components such as 1).IoT device ,2). IoT server and 3) User interface .

An IoT server is the connection point between the IoT device and the user interface .It is responsible for allowing the right authenticated user to get access to the required data from the IoT device .An IoT device is a non standard computing device that is used to connect wirelessly to a network and transmit the data which is generated by the sensors to the Iot server when the IoT server sends a request for the data .

Wide area network is used for the IoT server and IoT device to communicate and transfer request and IoT data with each other respectively.An user interface is a means through which the user and the IoT server interact . It is a way to view and understand the data captured by IoT server . The user interface sends the request to the IoT server for gaining an access to the IoT data .User access the data via a web or mobile interface

Characteristics of the components are :

- Equal sized n keys of m bits size are stored within the vault initially and this vault is stored with it's unique code within the IoT server provided by the IoT device . The n and m values are choosen based on memory availability and security requirement.
- The keys within the vault are replaced either after the keys are exhausted after the use or after a specific time that is set.

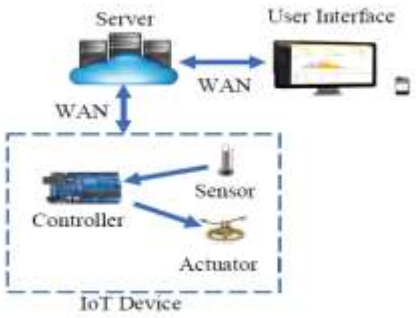

Figure 1 IoT System architecture

4.AUTHENTICATION MECHANISM

During the authentication phase, a shared secret called session key is built by the IoT server and IoT device. Firstly, it is used to encrypt the messages exchanged between server and IoT device. All the messages exchanged between two authentications is treated as a session. The session key remains unchanged throughout the session but different session keys are used for different sessions.

A.Secure vault implementation:

Initial vaults made up of n number of keys are used to request an access to the IoT data. Each secure vault (SV) has a code assigned for unique identification. The keys are distributed to the IoT server and stored in secure vaults for the future access by the user. We represent all the keys as K[0] ,K[1],K[2],....,K[n-1]. The secure vault is shared between IoT device and server during the time of deployment of IoT device. On the IoT device, secure vault is stored in an encrypted format. On the server, vaults are stored in secure database.

Each user has a unique MAC. The user requests for accessing the IoT data to the IoT server. As a result, receives a hash encrypted secure vault of keys. The user now decrypts the hash by the private key. The user sends this decrypted hash along with the request of a data to the server. The server now authenticates a user as valid one as he/she has the private key.

When the user requests for the IoT data his/her MAC is hashed with secure vault's unique code and stored within the server (HMAC = SV +MAC). This hashed value is compared with the hash decrypted by the user. If found equal, then one key from the vault along with request for IoT data is sent to the IoT device. The specific key from the secure vault present at the IoT device is checked for a match with the key sent by the user. If a match is found then the required data is passed on to the end user.

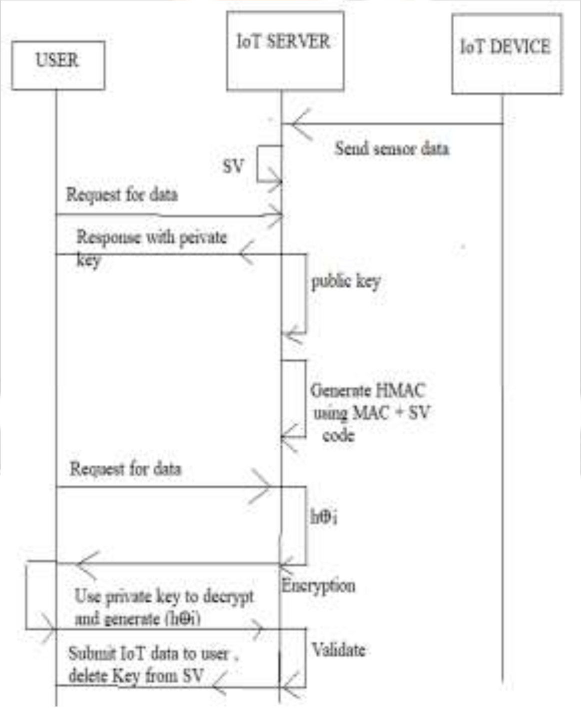
USER
IoT SERVER
IoT DEVICE
Send sensor data
SV
Request for data
Response with peivate key
public key
Generate HMAC using MAC + SV code
Request for data
hΘi
Encryption
Use private key to decrypt and generate (hΘi)
Validate
Submit IoT data to user , delete Key from SV

Figure 2.Sequence diagram

B. Changing the secure vaults

The session duration is determined by the user : Shorter duration gives high security, which results in frequent invocation of the three-way authentication message exchange. Regeneration of the keys can be done in two ways. First, when all keys are used. Second , Time based. IoT environment should generate new set of keys for each new request. The server has a fixed number of keys stored for a fixed number of request.

After every session, the keys within the vault are replaced either after the keys are exhausted as a result of the use or after a specific time that is set.New secure vault with newly generated keys in created by the IoT device and sent to the IoT server.

CONCLUSION

In this paper, we have presented an strategy to provide a secure authentication mechanism between Iot server and Iotdevice . This stratergy is secure is against the side channel attacks which usually violates the security of IoT devices .Since the secure vaults changes each time after a communication session it avoids the dictionary attack . As this stratergy avoids capturing of all the message exchanged between IoT server and IoT device ,this can overcome man in the middle attack .

REFERENCES

[1] Kothmayr, T., Schmitt, C., Hu, W., Brünig, M., & Carle, G. (2012, October). A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication. In Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on (pp. 956-963). IEEE.

[2] Raza, S., Shafagh, H., Hewage, K., Hummen, R., & Voigt, T. (2013). Lithe: Lightweight secure CoAP for the internet of things. IEEE Sensors Journal, 13(10), 3711-3720.

[3] I. Leontiadis, C. Efstratiou, C. Masc olo, and J. Crowcroft, "SenShare: Transforming Sensor Networks into Multi-application Sensing Infrastructures," in Wireless Sensor Networks, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, Jun. 2012, vol. 7158, pp. 65– 81.

[4] ETSI TR 102681, "Machine-to-Machine Communications (M2M); Smart Metering Use Cases," http://www.etsi.org, May 2010.

[5] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained Application Protocol (CoAP)," Internet Engineering Task Force (IETF), Jun. 2012.

[6]T.Kothmayr, C. Schmitt, W. Hu, M. Brnig, and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," Ad Hoc Networks, ELSEVIER, 2013.

[7] M. Brachmann, S. L. Keoh, O. Morchon, and S. Kumar, "End-to-End Transport Security in the IP-Based Internet of Things," in Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN), 2012, pp. 1–5.

[8] Barreto, L., Celesti, A., Villari, M., Fazio, M., &Puliafito, A. (2015, August). An authentication model for IoT clouds. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 (pp. 1032-1035). ACM.

[9] Jan, M. A., Nanda, P., He, X., Tan, Z., & Liu, R. P. (2014, September). A robust authentication scheme for observing resources in the internet of things environment. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on (pp. 205-211). IEEE.

[10] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J. Vasseur, "RPL: IPv6 Routing Protocol for Low Power and Lossy Networks," Internet Engineering Task Force (IETF), Mar. 2011.

[11] N. Modadugu and E. Rescorla, "The Design and Implementation of Datagram TLS," in Proceedings of the Network and Distributed System Security Symposium, 2004