ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 12, December 2017

Secure Group Communication in CoAP

[1] Nithin Krishna N, [2] Navin K
[1] M.Tech, Information Security and Cyber Forensics, SRM University, Kattankulathur, Chennai, India
[2] Assistant Professor, IT Department,SRM University, Kattankulathur, Chennai, India

***Abstract*: The interest in Internet of Things (IoT) has been increased. The IoT can be used to exchange information between people and physical devices or between devices and devices. For constrained IoT network environments, the lightweight application protocols have recently been proposed, is Constrained Application Protocol (CoAP) However, this protocols still have the scalability problem in the network with a large number of sensors. DTLS does not support multi-cast messages in group communications. In this case, the proxy can translate from HTTP to CoAP, but the proxy has to decide if it is a multi-cast or uni-cast message. A robust key management process provides a solution for all protocols including CoAP. However, in the case, key management can help in solving the multi-cast message issue within the DTLS and to ensure security In this paper, we propose a simple extension of CoAP using a clustering approach, in which a set of CoAP sensors are grouped into a cluster, and a cluster head is used for message aggregation and transmission for the sensors associated with the cluster. And also, key management is a basic security mechanism for IoT, but is still a challenging issue in IoT. While several key management schemes have been proposed for IoT networks, it is still a challenging issue in these networks. Secure key management is one of the issues attracting researchers' attention in terms of energy consumption and processing load on sensor nodes, as well as security problems. Due to storage limit in these networks, scalability is one of the most important components discussed in key management. In this paper, we also propose a scalable key management scheme based on clustering for IoT networks in CoAP.**

***Key Words:* CoAP, group communication, key management, clustering protocol**

I. INTRODUCTION

Today the Internet has become ubiquitous, has touched almost every corner of the globe, and is affecting human life in unimaginable ways. However, the journey is far from over. [1] We are now entering an era of even more pervasive connectivity where a very wide variety of appliances will be connected to the web. We are entering an era of the "Internet of Things" (abbreviated as IoT). the Internet of Things is a paradigm in which computing and networking capabilities are embedded in any kind of conceivable object. refers to a new kind of world where almost all the devices and appliances that we use are connected to a network. We can use them collaboratively to achieve complex tasks that require a high degree of intelligence.

All IoT applications need to have one or more sensors to collect data from the environment. Sensors are essential components of smart objects. [2] One of the most important aspects of the Internet of Things is context awareness, which is not possible without sensor technology. IoT sensors are mostly small, have low cost, and consume less power. They are constrained by factors such as battery capacity and ease of deployment.

With the growth of IoT, the number of sensor nodes are increasing steadily. It is important to receive data efficiently from the server in constrained environment. In MQTT, each subscriber must communicate with the broker even though a lot of subscribers have the same interests of topics. It may cause the traffic overhead problem and the energy wasting of constrained devices. In the meantime, CoAP is a request response model that is also difficult to manage many devices. However, the CoAP has also the similar scalability problem with MQTT. [7-3-1]

There are four security modes defined for CoAP which are:

NoSec: this alternative assumes that security is not provided in this mode or in the CoAP transmitted message.

• PreshardKey:this mode is enabled by sensing devices preprogramed with symmetric cryptographic keys. This mode is suitable for applications that support devices that are unable to employ the public key cryptography. Also, applications can use one key per device or one key for a group of devices.

• RawPublicKey: the mandatory mode for devices that require authentication based on public key. The devices are programmed with pre-provisioned list of keys so that devices can initiate a DTLS session without certificate. [10-7-4-1]

In the end-to-end communication as in some scenarios a HTTP client needs to access resource from CoAP back-end server. In that case, the proxy must translate the packet without scanning to check if there is a malicious code. However, the challenge is that DTLS does not

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 12, December 2017

support multi-cast messages in group communications. [10-7-5-1] In this case, the proxy can translate from HTTP to CoAP, but the proxy has to decide if it is a multi-cast or uni-cast message. A robust key management process provides a solution for all protocols including CoAP. However, in the case, key management can help in solving the multi-cast message issue within the DTLS and to ensure security

To overcome these problems, in this paper, we propose a simple extension of CoAP using a clustering approach, in which a set of CoAP sensors are grouped into a cluster, and a cluster head is used for message aggregation and transmission for the sensors associated with the cluster. Sensors in the proposed scheme are managed by a cluster head. The cluster head will manage many sensors. By using this clustering approach, it is easy to transfer messages containing topics. In the proposed scheme, the cluster head is used to reduce the amount of traffics concentrated on the server. The proposed scheme can also be used for CoAP-based many-to-many communication. [6-4]

The sensors and users don't need to communicate directly with each other. When the cluster head receives messages on a topic, it sends messages to the sensor nodes in the cluster by using multicast. The cluster head also manages the members of cluster by using CoAP POST, PUT, and DELETE methods. The server collects and stores temporarily the messages from the sender until these messages are transmitted to the cluster head.

A. CoAP SECURITY:

Because CoAP is built on top of UDP not TCP, SSL/TLS are not available to provide security. DTLS, Datagram Transport Layer Security provides the same assurances as TLS but for transfers of data over UDP. [3]
Datagram Transport Layer Security (DTLS)is a communications protocol that provides security for datagram-based applications by allowing them to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. But DTLS (Datagram Transport Layer Security) protocol only secure unicast communication (client-server communication). DTLS done not provide security in multicast (one to many communication) because CoAP support multicast communication but DTLS does not support multicast communication.

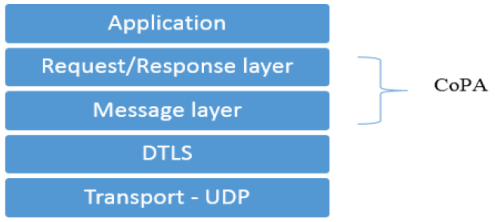


Fig:1 CoAP abstract layer with DTLS

B. Key Management Schemes

Key management protocols [6] are responsible for key pre-distribution and key updating in case of changes in the cluster group. Group keys are shared among all the members of cluster and the contents to be shared are encrypted with this key and broadcasted to all group members. Such groups are supposed to be flexible enough to allow new hosts to join and present members to leave. Joining and leaving of hosts require change in the group key, so that the privacy and secrecy of the group members and their communication can be preserved.To maintain group keys, secure key management protocols are devised and employed. These protocols provide the authentication services, along with changing of the group keys with each user joining and leaving. The process of changing keys on every user join or leave is called key updating or rekeying.

II. RELATED WORK

CoAP Architecture

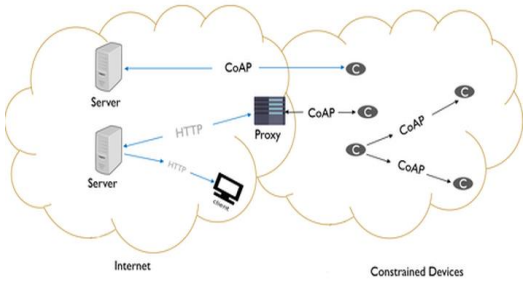


Fig:2 CoAP Architecture.

CoAP architecture split into two layers, message layer and request/response layer. The first layer is responsible for controlling the message exchange over UDP between two end points. The format of the message will be

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 12, December 2017

outlined next. While the second layer carries the request and response which hold the method code and response code in order to avoid issues such as the arrival of messages that are out of order, lost or duplicated. Thus, CoAP is a reliable mechanism with rich features such as, simple stop-and wait re-transmissions, duplicate detection and multicast support. CoAP uses a short fixed-length binary header and components, and messages are encoded in binary simple format of each component and the message format.

III. RESEARCH CHALLENGE

Although a lot of work has been proposed, there are still several challenges of the key establishment problems in IoT.

To provide security and privacy, IoT devices should participate in well-established security mechanisms. The well-established security mechanisms such as Diffie-Hellman key exchange, SSL etc. require performing numerous computationally intensive cryptographic operations. However, the devices in IoT are resource-constrained and thus lightweight key management systems are needed. The constraints could be very little computation and storage capabilities, short battery life and limited network bandwidth. The solution to this problem is easier for symmetric algorithms sharing the same key for a group of devices. However, it is intuitive to see that it may be more vulnerable to attacks. Once a key is revealed, communication of the whole group can be compromised.

IV. PROPOSED SYSTEM

In this paper, we propose a lightweight protocol extension to CoAP, named Cluster-based CoAP (Cluster CoAP). The proposed scheme uses the CoAP observe function which is suitable for less periodic messages. We use a cluster head to manage a cluster of constrained devices and interact with the server. In the proposed scheme, the cluster head uses the topics representing the attributes of sensors. For this reason, user or server can communicate with sensors through the cluster head. Each node will register to the cluster head with an interested topic. After configuration of a cluster, the cluster head receives messages from the server and sends messages to sensors on a particular topic by using multicast.

To maintain connectivity between the cluster head and the server, server sends appropriate control messages to the cluster head.

Also, we propose a system with an efficient key management scheme for IoT network. The system model shows a network which is divided into number of clusters. This cluster is used to improve the scalability and energy efficiency of system. Each cluster consists of cluster head which aggregates information from all sensor nodes in the cluster and transfers the aggregated information to the other cluster head or the base station. Node to node communication is an intra-cluster communication, done by node to node link. Transfer between cluster head and cluster head or cluster head and base station is an inter cluster communication which is done by cluster head to cluster head link. The key management algorithm considers the cluster head as key manager. This is based on the assumption that a node may move from one location to another one but the cluster head and the base station are fixed in a location.

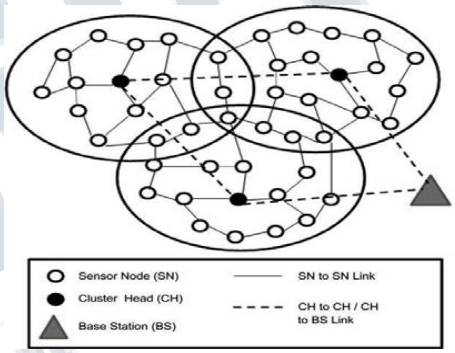


Fig:3 clustering based key managment

V. METHODOLOGY

In this section we will discuss which methodology we are going use for cluster based key management in coap.

A. Phases of clustering based key management

Mainly there are five phases in clustering based key management namely; registration, initial join, group key distribution, multicast data transmission, and run time join/leave to a multicast group.

Registration Phase: In this phase, each Receiver (R) interested in listening a multicast traffic contacts to Registration Authority by establishing Group Security Association.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 12, December 2017

Initial Join to the Multicast Group: When a user wants to receive a multicast traffic from a multicast group, he sends a JOIN message to its designated router along with a group address.

Group Key Distribution: Now each member has security parameters and common group key (CGK).

Multicast Data Transmission: Sender which is a member of multicast group in CCSMS encrypts the multicast traffic by using CSK and forwards it to multicast group. This group distributes the traffic to the last hop routers. Finally, the traffic is multicast to the end members of a group by designated routers.

Run Time JOIN/LEAVE to a Multicast Group: When a new user sends a JOIN request to Group Manager. Group Manager generates a unique UID and Shared Key as a Group Controller and distributes it along with multicast group address (multicast IP) to a user by applying secure Group Security Association. Now for maintaining the forward and backward secrecy, new security parameters and Common Group Key are distributed to each user using GSA in unicast message. Each user calculates a new group key 'Common shared Key' and then handles the further multicast group communication through this Common Shared Key. Similarly, when a user LEAVES the group the same process is performed for computing a Common Shared Key for backward secrecy. On every leave or join a new Common Shared Key is calculated and used.

VI. RESULT

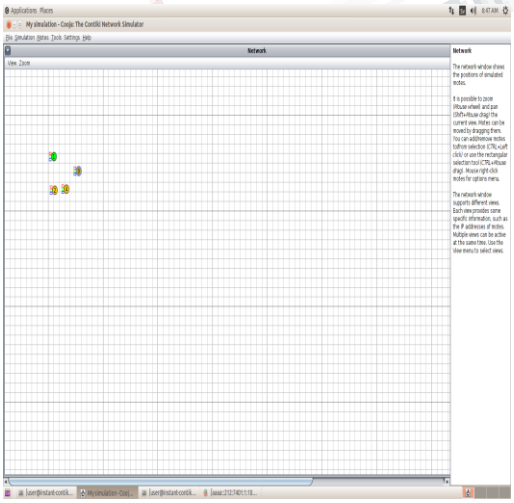


Fig:4 Network diagram

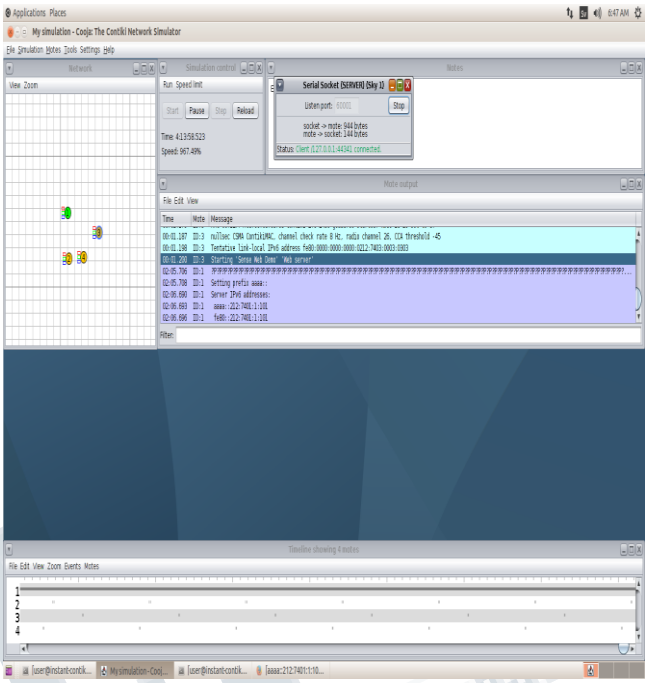


Fig 5 Connecting bridge

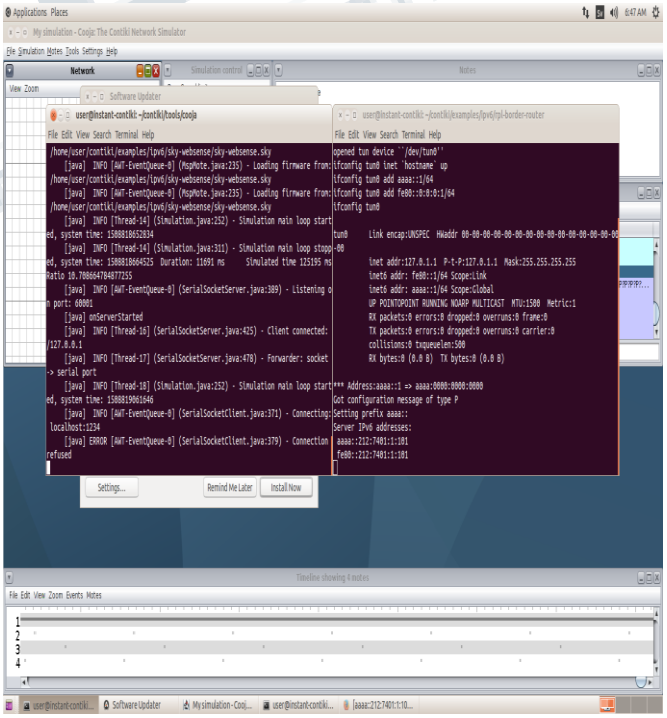
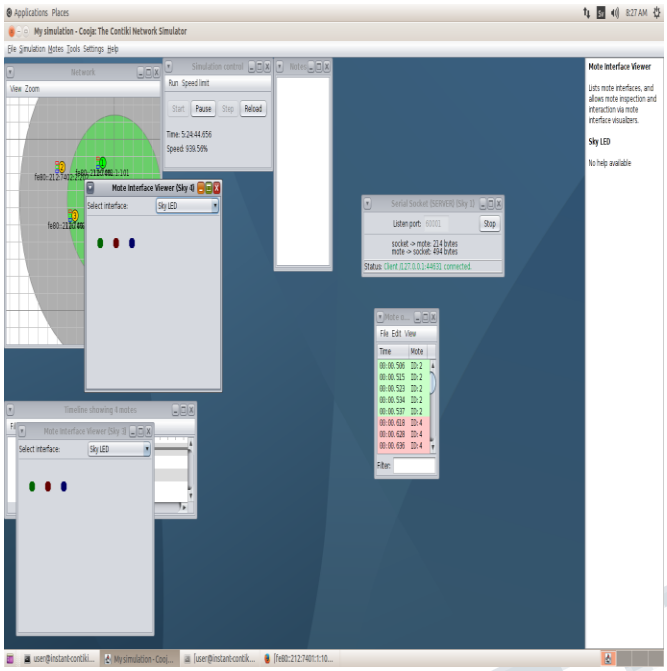


Fig:6 after bridge connection

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 12, December 2017



Fg:7 Request and response

VII. CONCLUSION

Due to the sensitivity of IoT networks applications and lack of support of constant infrastructure and also storage restrictions such as processing speed, storage size and energy in wireless sensor nodes, performing security mechanisms such as key management are mentioned as challenging issues in this network. One of the main concerns in designing a key management scheme in IoT network is scalability.

The paper proposes a novel cluster based key management scheme that will improve scalability, security and mobility requirements by reducing the computational complexity of the algorithm. The scheme runs in two phases, first phase will setup the cluster and assign the home and foreign keys to each node. The second phase maintains the key during the node and cluster head mobility.

REFERENCES

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M., 2015. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), pp.2347-2376.

2. Yang, Y., Wu, L., Yin, G., Li, L. and Zhao, H., 2017. A Survey on Security and Privacy Issues in Internet-of-Things. IEEE Internet of Things Journal.

3. Keoh, S.L., Kumar, S.S. and Tschofenig, H., 2014. Securing the internet of things: A standardization perspective. IEEE Internet of Things Journal, 1(3), pp.265-275.

4. Rahman, R.A. and Shah, B., 2016, March. Security analysis of IoT protocols: A focus in CoAP. In Big Data and Smart City (ICBDSC), 2016 3rd MEC International Conference on (pp. 1-7). IEEE.

5. Raza, S., Seitz, L., Sitenkov, D. and Selander, G., 2016. S3K: Scalable Security with Symmetric Keys—DTLS Key Establishment for the Internet of Things. IEEE Transactions on Automation Science and Engineering, 13(3), pp.1270-1280.

6. Zhang, Y., Shen, Y. and Lee, S., 2010, April. A cluster-based group key management scheme for wireless sensor networks. In Web Conference (APWEB), 2010 12th International Asia-Pacific (pp. 386-388). IEEE.

7. Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F. and Alonso-Zarate, J., 2015. A survey on application layer protocols for the internet of things. Transaction on IoT and Cloud Computing, 3(1), pp.11-17.

8. Tsai, I.C., Yu, C.M., Yokota, H. and Kuo, S.Y., 2017, January. Key Management in Internet of Things via Kronecker Product. In Dependable Computing (PRDC), 2017 IEEE 22nd Pacific Rim International Symposium on (pp. 118-124). IEEE.

9. Ouakasse, F. and Rakrak, S., 2017. An Adaptive Solution for Congestion Control in CoAP-based Group Communications. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 8(6), pp.234-239.

10. Shaheen, S.H. and Yousaf, M., 2014, December. Security Analysis of DTLS Structure and Its Application to Secure Multicast Communication. In Frontiers of Information Technology (FIT), 2014 12th International Conference on (pp. 165-169). IEEE.