Mitigation of Spoofing Attacks on IOT Home Networks

Archana K C, Harini N

Abstract - internet of things is now everywhere and even if people are aware of it or not, it is part of our everyday life. For something that is so much in pace with our life, iot collects a lot of information about our day today life, which in case of a data leak or hijacking could lead to catastrophic effects in the society. Still iot devices are not manufactured keeping in mind the security factor. This paper dives into the problem of spoofing attacks dealt by iot devices and comes up with an authentication mechanism, which uses variants of elliptic curve cryptography to protect against such said attacks without exhausting the devices in case of computational power and storage area. The experimentation clearly revealed the strength of the scheme to mitigate spoofing attacks on the iot home networks.

Index terms— MQTT, IoT, ECC, PKG, ECDH, CoAP, REST, ECDSA

I. INTRODUCTION

Internet of things is a buzzword and the number of devices that is becoming a part of the world of internet of things is a gigantic number. People have started to rely on it for their day-to-day activities. This means more connected devices and thus more communication. This communication needs to be protected from getting into the wrong hands. The IoT network has insufficient of security services as it presently relies on security schemes developed for internet services. Taking the example of the 2016 Dyn Cyberattack, the botnet used IoT devices as bots, which had guessable usernames and weak passwords. Even though spreading the awareness of unguessable usernames and strong passwords are going on, but it has not reached most of the audience using IoT devices. IoT is a new area for everyone, but still is out in the world used by everyone even though not all the dimensions are tested and successful. Many attacks have been reported till date. Not all the issues have been addressed completely. Most of the solutions that are used hinders with the characteristics of IoT device leading to massive amount of power consumption and storage. IoT is a field where there is a lot of exploration left out. Therefore, there is a huge need for the tech teams to come up with alternate ways to keep IoT devices secure.

This work aims at addressing the security of MQTT protocol used by IoT devices for message transfer. MQTT, which is a lightweight, publish/subscribe protocol, which runs over TCP/IP. MQTT is a great pick for resource-constrained devices. Facebook Messenger uses MQTT protocol so that it does not drain out the battery life of the devices using it. MQTT was not created with security in mind; therefore, it suffers from lack of security and is prone to many security attacks like hijacking and man in the middle attacks.

The work proposes a scheme to mitigate spoofing attacks in IoT networks.

The organization of the paper is as follows. Section II presents the need for the proposed solution. Section III summarizes the literature survey related to work completed in the security in IoT. Section IV discusses the underlying architecture used for the experimental work to illustrate the vulnerability in the protocol. Section V explains the cryptographic schemes and the modifications included in them to enhance security. Section VI discusses the results obtained from the test bed setup. Section VII concludes with the future work on how this technique could be taken ahead to solving more problems in IoT.

II. SCENARIO

IoT devices are mostly the ones, which lack on memory, power and resources. On top of this exist communication protocols to facilitate information exchange between devices.

MQTT is a lightweight protocol, which addresses the constraints of an IoT device. MQTT at the same time lacks a proper authentication mechanism. Username and passwords in an MQTT packet can be easily sniffed using a sniffing tool. MQTT being built on top of TCP can use SSL/TLS for more secure communication, but this will be an overhead on the resource constrained IoT devices. Figure 1, shows the entities subscribers, publishers and brokers in the environment with an attacker using a sniffing tool to listen to traffic and perform malicious activities on the network.

Subscriber and publisher can be termed as clients. The client can be any devices that runs TCP/IP stack running MQTT over it. In MQTT protocol, the subscriber and publisher has no knowledge about each other. It is the MQTT broker who decides where to send the packets to which it received.

Revised Manuscript Received on October 12, 2019.
Ms. Archana K C, Quality Assurance Engineer did her M.Tech and B.Tech Amrita School of Engineering, Coimbatore Campus.
Dr. Harini N, Assistant Proffesor, Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore Campus.

Imagine two IoT devices communicating with each other. Let us say device A(publisher) is publishing data and device B(subscriber) is a subscribed device which are working under the same network. Device B sends CONNECT packet to device A that contains username and password. Person Y uses a sniffing tool to sniff the CONNECT packet being sent from device A to device B. After sniffing, person Y retrieves the username and password of the original subscriber (device B). Now person Y can make device X to pretend as device B and subscribe to device A. Device A as usual will send an CONNACK packet back to where it got the CONNECT packet. After successful spoofing, device X can easily receive the data being published by the publisher.

Once the connection is created, it is kept open until the client sends a command to disconnect. Given person Y knows the topic to which it has to subscribe, it can send a SUBSCRIBE packet to the publisher with the topic name and publisher would acknowledge it with a SUBACK packet. Publisher would keep sending data until it receives a DISCONNECT packet. In Figure 2, it can be seen that messages are transmitted in plain text and this applies to even the secret messages (keys, passwords etc.) that are transmitted using this protocol. This observation clearly brings out a need for integrating security schemes on data transmissions handled by MQTT protocol

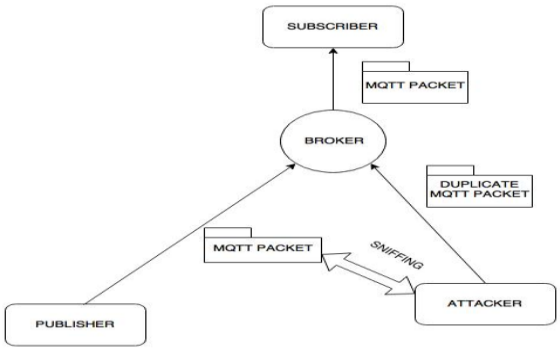

Fig. 1. Attack Scenario

Such scenarios are very easy to replicate in case of IoT devices and thus could cause severe cases of data insecurity leading to major leakage of data to unauthenticated resources. This paper tries to bring up an authentication mechanism, which would reduce the chances of such spoofing attacks efficiently without tampering with the computational and storage capacity of IoT devices.

III. RELATED WORK

In [8], discusses the challenges faced by IoT as a whole. The paper discusses the problems, which increases with the increasing number of IoT devices and does a survey on the security challenges faced by IoT. In the year 2016, Dyn cyberattack happened, which alarmed all the IoT industry and its customers. The IoT devices that exists are so vulnerable to spoofing that a mirai botnet army was created with the help of these vulnerable IoT devices, which kept sending data packets to Dyn servers, which became the victim of DDoS attack. This was caused because there are no proper authentication systems or security factors considered while manufacturing IoT devices. In [9], the paper explains various kinds of attacks that could happen. For example, there is eavesdropping that could happen when the channel in which the data exchange is happening is not secure. If the data channel is not secure, then it will be easier for an attacker to listen to the packets and read all the data being exchanged. This could lead to more attacks. An insecure channel would make it easier for an attacker to read the contents of the packets and understand the metadata of the IoT devices, which makes the attacker a step ahead on gaining control over the IoT devices. When the attacker is being able to read the contents of the packet, it gets to know about the nodes, which in turn gives up the IP address and such important information about the device which could be further used for spoofing the device. This in turn leads to missing packets which means that an unauthenticated device is receiving the packets instead of the authenticated device.

[9] discusses about MQTT and compares it with CoAP which is also a protocol used for data transfer in IoT devices which are mostly constrained in the cases of resources. While MQTT is a protocol which works on a publisher-subscriber basis, on the other hand, CoAP works on REST. MQTT rely on 3 QoS level. CoAP can also work on publish-subscribe architecture but that would need an extended GET message. MQTT runs over TCP and CoAP does on UDP. According to the paper, when both the protocols were subjected to a common middleware, MQTT performed better and has lower message delays in case of low packet losses compared to CoAP. [11] uses AVL particle counter device which uses MQTT for message transfer to do threat analysis on MQTT routing information and tries to identify the requirements for a proper authentication. The paper validates the use of TLS as an authentication step by incorporating another hardware security controller. It discusses on the needs and points to keep up with before while using TLS. TLS had a history of vulnerabilities identified like the POODLE bug because of which TLS is now discussed a lot in the research field to make it more secure. OpenSSL makes use of TLS, but the earlier versions of OpenSSL has dealt with problems like the Heartbleed bug which led to a huge realization in the need of security in this field.

[13] gives an example of an IoT based product called “New Barbie” meant for children which had a potential privacy threat of having attackers spying on the people having the product, which means spying on the whole family. The attack could be on the camera or the voice interaction feature of the product which could be used to spy on everyone who interacts with the owner of the product.

In [1], the feasibility of enabling security using CP/KP-ABE for MQTT protocol is checked. Publisher and subscriber is register to a public key generator first by providing some unique value of the component. The public key generator generates a key set and parameters according to CP/KP-ABE scheme and these are published alongside certain attributes. The devices who owns the attributes will receive the key set. Publisher sends the data to the broker and broker sends it to all the subscribers and deletes the data later. Subscriber decrypts it if it with the help of private keys and if it satisfies the access policy. Only end to end security is satisfied in here. But in case the attributes gets leaked to an unauthenticated resource they will be able to hijack all the data.
In [4], the technique used is for the communication channel between IoT devices. Unlike others, it uses a combination of symmetric and asymmetric cryptography for the security of the gateway between the IoT devices.

It uses a modified version of vigenere cipher and RSA algorithm. After obtaining the current timestamp, it generates a random key and encrypts the data with vigenere cipher. To withdraw from the vulnerability of using a public key it encrypts the public key using RSA and the message is passed which would be a concatenated version of cipher text encryption using symmetric cryptography and the public key encrypted using asymmetric cryptography. Even though the security of the message and the chances of retrieving the key decreased rapidly with this technique, the key sizes are too large to be handled by an IoT device and thus exhausts the IoT device by the computation this technique would need.

In [5], the algorithm is divided in such a way that it addresses both authentication and access control. It is divided into key generation using ECDH which another cryptographic scheme in ECC like ECDSA, identity establishment and access control.

IV. PROPOSED SYSTEM

The proposed work is to create a subscriber and publisher who are able to communicate without having a third person steal the messages which were supposed to reach the authenticated subscriber without any problems.

Let us first discuss MQTT in detail which needs to be known before diving into the details of the algorithm. MQTT is the protocol that is being used for message transfer. MQTT client which is the subscriber or publisher sends a CONNECT packet to the broker to initiate the connection. CONNECT packet will have username and password where password is sent in plain text format inside the packet. It also contains the client ID which is unique for each subscriber. After getting the CONNECT packet from the subscriber, the broker sends back an acknowledgement packet to the subscriber telling that the CONNECT packet has reached the broker and the connection has been established successfully. Now each of the clients would have established the connection with the MQTT broker. Now is the turn to send publisher to publish the messages on a certain topic and send it to subscribers of the topic via broker. The PUBLISH packet will contain the packet ID, topic name and the payload which is the message under the topic. The message could be in any format. People usually send it in plain text format which means that anyone can see if they try to sniff the packet. For a subscriber to receive these published messages it needs to send a SUBSCRIBE packet to the publisher. The subscriber packet will contain the topic name it is subscribing to. The subscription of the topic will be confirmed by the broker by sending back the SUBACK packet.

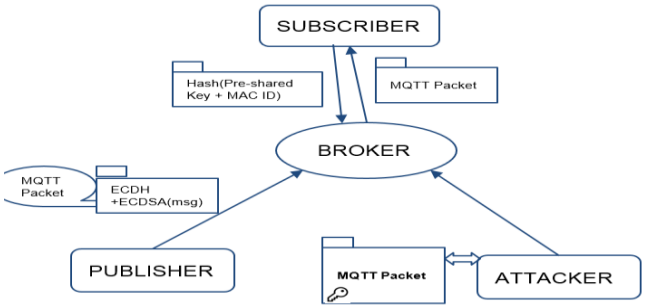

Fig. 2. Proposed Model

Now we can see the problems that are being faced by the MQTT protocol. The packets send are mostly in plain text format which is easily sniffable and read by unauthenticated devices which could read and misuse the data which should have been private to only the authenticated subscribers. Another problem that we are seeing here is with the authentication mechanism being used. The only security provided by MQTT is the username and password where the password is sent in plain text and therefore does not serve the purpose of security well. If some other unauthenticated wanted to get the password, it will be easy for them and thus the unauthenticated device can spoof the authenticated device and login as the authenticated device thus cheating the publisher to be the authenticated device and reading the messages being sent under the topic which was meant to be read only by the authenticated devices.

This calls on for additional security features in MQTT which will help in avoiding spoofing attacks on these IoT devices and also securing the messages. For this we make use of two asymmetric encryption techniques called ECDH and ECDSA. Let us first discuss about how these encryption methods work.

ECDH is a variant of Diffie-Hellman algorithm for Elliptic curves [7]. It is an algorithm for secure key exchange. The purpose of this algorithm is to exchange keys between two devices, which cannot be decoded by another device. In this algorithm, our publisher and subscriber are trying to exchange keys with each other.

- Publisher generates one public and private key, which we will call, Pb and Pr. Publisher and Subscriber will be using the same base point B which means that Pr = B*Pb. In the same way, subscriber generates a public key Sb and private key Sr where Sr = B*Sb.
- Pb and Sb will be exchanged with

each other across a channel. The unauthorized device that is watching this exchange will not be able to solve it the public key since it does not know the public key.

- Publisher and Subscriber will compute Vp = Pr*Pb and Vs = Sr*Sb. Both Vp and Vs will be the same.

ECDSA on the other hand is to send a message which would only be read by the authenticated person [7].

- A random number would R would be generated and let B be the base point.
- Calculate a new point N which is N = R*B.
- Calculate K = XbaseNmodO where O is the subgroup order. If K = 0, then choose another random number R. XbaseN is the x coordinate of N.
- Let publishers private key be called Pr and calculate R inverse. Inverse of R would be multiplicative inverse of R mod O.
- Calculate E = (inverse of R)*(G+K*Pr)modO. If E = 0, then choose another K again and repeat.

(K, E) is the signature.

ECDH helped the algorithm in secure key exchange. Since the double encryption only exists in the beginning, it will not cause any serious lag in data exchange after ECDH is done.

- Registration phase:
  - Publisher/Subscriber will request for registration with the broker
  - Publisher/Subscriber will provide a password along with the request packet
  - Domain parameters: TimeOfCon+device Address+password provided by the requester
  - Random number generated
  - Secure key exchange using ECDH
- Authentication Phase:
  - ECDSA to send data through channel
  - Subscriber will try to get data from the publisher under the same topic
  - Messages will be sent each time with an ECDSA encryption
- Occasional Handshake:
  - Publisher asks for a cookie(Hash(Pre-shared Key+IP Address))
  - Attacker won't be able to provide the right key and thus will be pushed out

If the CONNECT packet is not in the format it is supposed to be or if it takes more time to reach its destination then broker closes the connection.

V. EXPERIMENTAL SETUP

We implemented ECC with our proposed scheme on a python platform. Publisher and subscriber packets were created using python 2.7 library scapy and used opensource hivemq as the broker. We applied the variants of ECC with different key sizes and different parameters. Another random number generator was used to provide the input needed for ECDH. Different curves were tried on to check the security parameters strength.

- **Sniffing tool**: Wireshark
- **Broker**: HiveMQ
- **CPU**: 2.6GHz Intel Core i5-3320M
- **Operating System**: MS Windows 7 Professional (64-bit)
- **RAM**: 4 GB
- **Hard Drive Size**: 500 GB
- **System**: Lenovo ThinkPad T430
- **Programming language**: Python
- **Packet generation tool**: Scapy
-

VI. ANALYSIS

Using a cryptographic algorithm like ECC, which is an asymmetric key cryptography, has its own advantages and disadvantages. Use of symmetric key cryptography would have provided us with less computation overhead and thus satisfying one need of the proposed objective. However, the channels in which IoT devices would work on are mostly insecure channels leading to its increased vulnerability to attacks. Another disadvantage of symmetric key cryptography, which led to selecting asymmetric key cryptography, is the difficulty in sending key across a channel without having the risk of the key to be discovered by someone who is not supposed to be viewing it and using the same to decrypt the messages. To reduce such attacks, we go ahead with asymmetric key cryptography which although might seem expensive while looking from one side, but in a broader perspective would benefit our purpose.

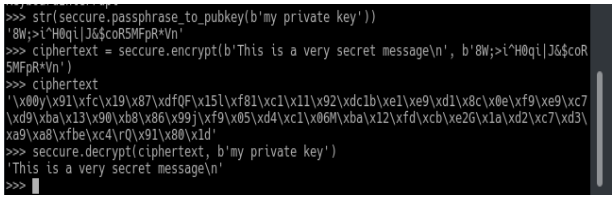

Fig. 3. Encryption using ECC

Choosing between RSA and ECC was a much more easy process since the difference between the key sizes was a significant one. RSA clearly has a larger key size, which leads to increased use of the computational power of the device, thus leading to more power consumption, which will not prove beneficial for the goal.

The key size for RSA algorithm is much bigger than ECC. Even symmetric key algorithms used by TLS uses 128 bit keys, so we go ahead with ECC with 256 bit keys which could be bearable. With the size of keys used by ECC it is easier for the IoT devices to deal with its lack of storage and

processing power, at the same time, boosts its security. ECC has the added advantage of faster key generation than compared to RSA. There were options of using HMAC or Digital Signatures. In case of HMAC we need to be working on a shared secret. When shared secret kept true for a long time could lead to catastrophic changes in the privacy, because HMACs are based on a key that is shared between the two parties, if either party's key is compromised, it will be possible for an attacker to spoof them. This could be overcome with the usage of digital signature.

VII. RESULTS AND DISCUSSION

An effort was made to evaluate the performance of the cryptographic primitives with varied key sizes to secure MQTT protocol in terms of the time taken to generate keys, encrypt, decrypt, sign and verify the signature. Extensive simulation was done in a controlled environment setup with three laptops connected in 802.3 network. When the algorithm is applied, the packet contents are no more in plain text form and will be difficult for anyone to decode and view the contents to spoof it.

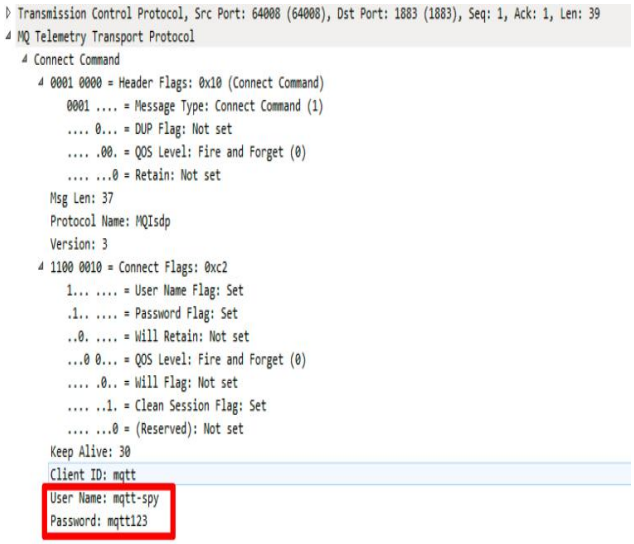

Fig. 4. MQTT CONNECT Packet when sniffed using Wireshark (Source: https://dzone.com/articles/dissecting-mqtt-using-wireshark)

However, it cannot be denied that in cases of a high-end brute force attack, the algorithm may give up and the attacker may be able to decrypt the contents. Even though theoretically brute force attack is always possible, it would be almost impossible to do it in case of such encryption. The experimentation results clearly brought out the suitability of the proposed scheme in terms of strengthening the security as per claims mentioned in the paper. As a future work, we are planning to encrypt the entire stream instead of payloads. following items when proofreading spelling and grammar:

VIII. CONCLUSION AND FUTURE WORK

Distributed, lightweight and attack resistant solutions, being the most favorable choices for IoT, it puts resilient challenges for authentication and access control of devices. Paper presented is efficient and scalable and is an ECC based authentication and access control protocol. Power of ECC is extended to achieve mutual authentication of devices with novel capability based approach for access control. The paper efficiently deals with the problem of authentication and mitigating spoofing attacks with the help of asymmetric key cryptography and blends in well with the characteristics of IoT keeping in mind the additional computational overhead that has come up with the usage of asymmetric key cryptography. The use of ECC helped us in not identifying the IP address of the subscribers which was before found using protocol analyzers. The paper comes up with solution which works around asymmetric key cryptography, but could be in a better version, in terms computation, if it uses symmetric key cryptography instead. The future work could include a method, which incorporates symmetric key cryptography instead of asymmetric key cryptography. In addition, usage of hashing techniques could increase the integrity of the messages leading to more secure channels. Proper choice of parameters would lie in the future work of this paper were the random number and the base point can be chosen based on these parameters which would make it more difficult to decrypt for a non-authenticated device.

REFERENCES

1. M. Singh, M. A. Rajan, V. L. Shivraj and P. Balamuralidhar, "Secure MQTT for Internet of Things (IoT)," 2015 Fifth International Conference on Communication Systems and Network Technologies, Gwalior, 2015, pp. 746-751
2. Singh, S., Sharma, P.K., Moon, S.Y. et al. J Ambient Intell Human Comput (2017). https://doi.org/10.1007/s12652-017-0494-4.
3. Isha and Ashish Kr. Luhach, "Analysis of Lightweight Cryptographic Solutions for Internet of Things.
4. M. S. Henriques and N. K. Vernekar, "Using symmetric and asymmetric cryptography to secure communication between devices in IoT," 2017 International Conference on IoT and Application (ICIOT), Nagapattinam, 2017, pp. 1-4
5. P. N. Mahalle, B. Ang gorojati, N. R. Prasad and R. Prasad, "Identity establishment and capability based access control (IECAC) scheme for Internet of Things," The 15th International Symposium on Wireless Personal Multimedia Communications, Taipei, 2012, pp. 187-191
6. Borgohain, T., Kumar, U. and Sanyal, S. (2018). Survey of Security and Privacy Issues of Internet of Things.
7. Diego Mendez, Ioannis Papapanagiotou, Baijian, Internet of Things: Survey on Security and Privacy, Purdue University
8. Rodrigo Roman, Jianying Zhou, Javier Lopez, On the features and challenges of security and privacy in distributed internet of things, Computer Networks, Volume 57, Issue 10, 2013
9. C. Lesjak et al., "Securing smart maintenance services: Hardware-security and TLS for MQTT," 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Cambridge, 2015, pp. 1243-1250
10. Md Tanvir Arafin, Dhananjay Anand, and Gang Qu. 2017. A Low-Cost GPS Spoofing Detector Design for Internet of Things (IoT) Applications. In Proceedings of the on Great Lakes Symposium on VLSI 2017(GLSVLSI '17). ACM, New York, NY, USA, 161-166. DOI: https://doi.org/10.1145/3060403.3060455
11. Dr T.R Padmanabhan and Dr.C.K.Shyamala, N.Harini, "Cryptography and security", Wiley India, First Edition, 2011
12. Kamakshi Devisetty R N, Aruna D, Harini.N, "Secure Proxy Blind ECDS Algorithm for IoT", International Journal of Pure and Applied Mathematics, Volume 118, No. 7 2018, 437-445

13. D. Locke, ”MQ Telemetry Transport (MQTT) V3.1 Protocol Specification,” IBM DeveloperWorks Technical Library, August 2010
14. T. Dierks and E. Rescorla, ”The Transport Layer Security (TLS) Protocol Version 1.2,” IETF RFC 5246, Standards Track, August 2008
15. R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-based Encryption with Non-monotonic Access Structures,” in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS ’07, 2007, pp. 195–203
16. M. Ion, G. Russello, and B. Crispo, “Supporting Publication and Subscription Confidentiality in Pub/Sub Networks,” in Security and Privacy in Communication Networks, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 50, 2010, pp. 272–289
17. P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” ACM Comput. Surv., vol. 35, no. 2, pp. 114–131, Jun. 2003
18. D. Thangavel, X. Ma, A. Valera, H. X. Tan and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 2014, pp. 1-6

AUTHORS PROFILE

Ms. Archana K C, currently working as a Quality Assurance Engineer did her M.Tech and B.Tech from Amrita School of Engineering. Her research interests include Internet of Things and Cloud Computing.

Dr. Harini N, currently serves as Assistant Proffesor in the Department of Computer Science and Engineering at Amrita School of Engineering, Coimbatore Campus. Her primary area of research is Security.