Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Performance Evaluation of DTLS with Certificate Authority for Internet of Things

Mukul Panwar, Ajay Kumar
Computer Science Department
DIT University Dehradun, Uttarakhand, India

Abstract— Internet of Things is a revolutionary technology emerging very fast. In IoT (Internet of Things) objects are capable of communicating with each other. Internet of Things got a lot of applications in various areas like in medicine, automobile, production, Information Technology and many more [1].
In IoT devices are very small therefore have limited processing capability. With such constraints, we can not use heavy protocols for communication between these devices. The security of the communication between IoT devices is a matter of great concern which requires additional attention. The obvious security solutions like TLS (Transport Layer Security) can not be deployed with such a constrained environment. Therefore a different version of TLS i.e. DTLS (Datagram Transport Layer Security) which can also be used with unreliable protocols such as UDP may result in better performance. TLS is based on the reliability of underlying protocol that is TCP which can not be usually be applied with constrained devices that are used in IoT communication.
Most of the proposed versions of DTLS is based on the common key i.e. pre shared key. For making the communication more robust we can use a Certificate Authority that can give digital certificates to both client and server and can increase the effectiveness of this communication. This work aims to introduce a CA for the communication and to provide some results that can show its improved performance in contrast to the pre shared key communication with little overhead.

Keywords— Internet of Things (IoT); Datagram Transport Layer Security (DTLS); Certificate Authority (CA)

I. INTRODUCTION

The Internet of Things (IoT) has already been recognized by media and various industrial leaders as the next wave of innovation and spreading in our daily life [9] [10]. IoT is gaining applications almost in every sphere household devices, smart city infrastructure, industrial automation, medical advancement and transportation are some of the key areas showing the enormous possibility for Internet of Things (IoT). Sensors in our day to day life are becoming popular to fulfill the day to day need of a user[1].As these devices are gaining popularity the security of communication can not be ignored. Therefore already a lot of research work has been done for securing IoT communications. The problem while providing security solutions for IoT is its constrained nature therefore usual security solutions can not be applied with such a constrained environment.

The Transport Layer Security protocol (TLS) is a mostly used security protocol for reliable communication protocol [2]. The transport layer protocol that is most commonly used for IoT (Internet of Things) communications is UDP because as it is already been observed that TCP performance is low in wireless networks due to congestion control algorithms. HTTP which is mainly used with TCP is not sufficient in constrained environment. For such a constrained environment IETF has defined a connection less light weight constrained application protocol (CoAP) [11].

Most of Constrained Application protocol uses Datagram Transport Layer security as security protocol. DTLS protocol requires a lot of message exchange between communicating entities. Therefore DTLS protocol can be adopted as per the constrained devices need with some header compression technique. Most of the DTLS versions use per shared key for communication. Nowadays small devices are also getting faster processing due to technological advancements. Therefore the security system can also be extended with some additional concepts. Certificate Authority if used in constrained environment with DTLS can provide better security than pre shared key authentication. So this work aims to introduce Certificate Authority for authentication and elimination of pre shared key.

Certificate Authority is a trusted entity that provides digital certificates for the purpose of authentication. Certificate Authority provides certificates to the clients, servers, databases etc. When an entity wants to communicate with other entity then it sends its certificate to other entity for authentication. Other entity verifies the certificate based on some well known cryptographic techniques. In this paper, our aim is to present a DTLS with certificate Authority and to compare it with DTLS based on pre shared key.

II. RELATED WORK

Earlier security protocols consider only the security of the data on a hop by hop basis is used for the complete network [4]. In most of the security protocols there is not a proper mechanism that how the key will be distributed to the nodes. In many cases the keys are updated on the nodes before communication [5]. So this pre shared mechanism of key

distribution is not so secure and robust, therefore the public key cryptography may result in better performance. Scholars working in the field of IoT have already analysed the challenges in IoT based on IP and also given the solutions for improving the IP based security for the resource constrained devices [6]. Brachmann et al. [7] has also given a TLS-DTLS mapping system for the IoT security but that requires a trusted 6BR. Keoth et al 8] have also told effects of IP connected IoT security with DTLS and have also given an architecture for accessing the network in a secure manner and unicast, multicast key management for DTLS. Daniele Trabalaza [3] provides the confidentiality and integrity for android devices but it is also uses a pre shared key concept.

III. HOW CERTIFICATE AUTHORITY WORKS

Certificate Authority is a trusted entity which provides certificate to the other entities like client, server databases etc. It certifies other entities by means of digital certificates. Digital certificates are signed by means of digital certificates. Digital certificates are signed by the private key of certificate authority with the help of public key cryptography. Certificate Authority publishes its public key while keeps its private key secret. Digital Certificates contains the digital signature which is nothing but encrypted hash code of the subject public key. When other entity wants to verify the digital certificate, it verifies the certificate by comparing the hash code of the subject public key with decrypted digital signature contained in digital certificates. If both are equal, it means the client that is sending the certificates is a genuine user. Following Fig 1 shows the working of Certificate Authority.

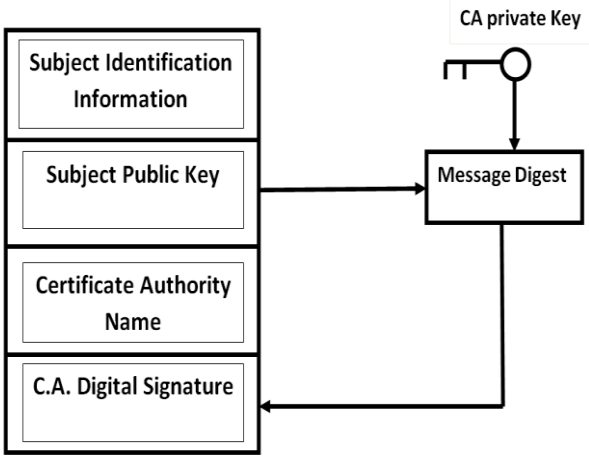

Fig. 1. Working of Certificate Authority

Following are the major working points for the authentication of digital certificate.

1. Certificate Authority calculates the message digest of subject public key with some known algorithm such as MD5.
2. It encrypts the hash code with its private key using public key cryptography and output is called digital signature.
3. Digital certificates also contains some other thing apart from digital signature are subject unique ID, subject public key given by CA that is also unique for each entity, Certificate Authority(CA) name tec.
4. One entity sends this digital certificate to other entity for which it wants to communicate.
5. Other entity verifies this digital certificate for the authentication purpose. Mainly it decrypts the digital signature with the public key of Certificate Authority i.e. already been published. This decrypted value is then compared to the hash code of the subject public key. If both values are same that means sending entity is a valid user that is registered with certificate authority.

IV. PRE SHARED KEY VS PUBLIC KEY CRYPTOGRAPHY

Most of the DTLS work has already been done on DTLS that uses pre shared key for authentication because to make the system less complex. But as the constrained devices are also getting better processing speed due to development in technologies we should also work in the direction of making the communication between these devices more secure. Therefore for making the authentication more robust in this communication another alternative of pre shared key can be deployed. One of the possible alternatives is Public key cryptography in which sender sends the data by encrypting it with his private key and receiver decrypts the data with the help of public key of sender. The receiver is able to decrypt the data with the help of public key of the sender only when it is send by the legitimate sender i.e. it is been decrypted with the private key of the sender. In this manner a better authentication mechanism can be used between the devices used in IoT communication. For this purpose Certificate Authority issues digital certificates to both the client and servers. Digital Certificates contain the digital signature of the Certificate Authority along with the other information such as CA name, Subject public key, version of the certificate, etc.

Digital signature is obtained by the CA by encrypting the hash code of the subject public key with its private key. When an entity wants to communicate with other entity it sends its certificate to other entity. Other entity verifies the certificate by comparing the hash value of subject public key and decrypted value of digital signature which is obtained by the public key of CA. If these values are same it means sender is a verified sender which is recognized by the CA. In this manner a better authentication is achieved between the communicating devices.

V. PERFORMANCE EVALUVATION

While increasing the robustness and effectiveness of DTLS protocol with the help of Certificate Authority, performance is the aspect that can not be ignored. We have to work in such a direction that will lead to us in a better DTLS protocol with slight overhead in its complexity. We have tried to make this overhead as minimum as possible while implementing certificate authority in java language. It is already been discussed that Certificate Authority is trusted entity that will give the certificates to both client and server so here our work is to incorporate the certificate authority in the system. This can be implemented as a separate class in java which generates unique certificates to each client which requests certificate from the Certificate Authority. A digital certificate field may very form version to version but some of the major fields in the digital certificate includes CA name, Subject Public key, Subject ID which is different for different client, and most important is digital certificates.

Digital certificate which is the encrypted hash code of subject public key encrypted with the help of CA private key along with the other mentioned filed is send to the requested entity with the help of socket programming in java. This implementation does not consume much memory space. The memory requirement in terms of heap can be shown with the help of result generated with the help of net beans 7.2.1 profiler functions. Net bean is an IDE for java language as well as for other language also. The following figure 2 depicts the memory requirement for the implementation of CA.

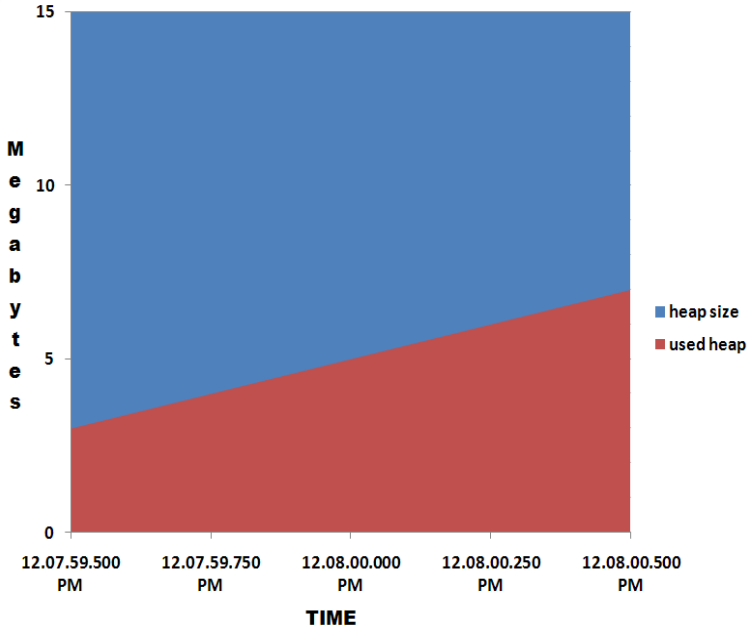

Fig. 2. Memory requirement for Certificate Authority

As it is clear form the above figure that the memory requirement of CA implementation is not a significant overhead when implemented in java. Above figure shows the memory requirement at different time. The memory requirement in terms of used heap varies from 4 MB to 6 MB that can be adjusted at the cost of better authentication that is provided by the CA.

It also requires some multithreading to implement this code where threads maximum range is 4 in our code. We should take care about number of threads limit that can be implemented by our code because if threads increase after certain point it will require more processing resources that is a factor that is concerned significantly in constrained devices. Following figure 3 shows the threads and loaded classes.

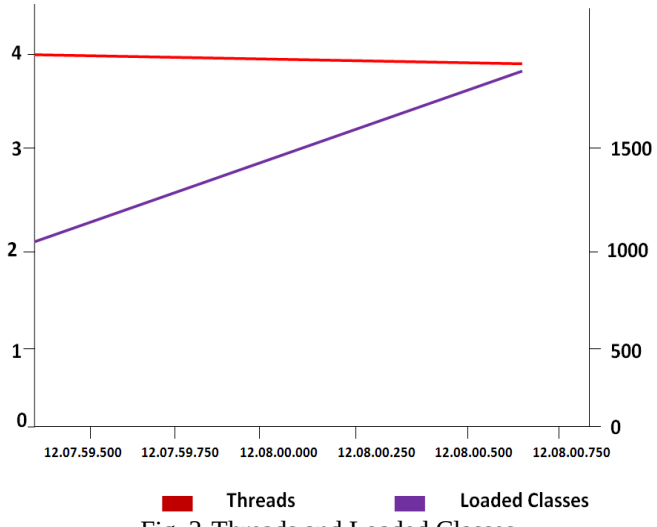

Fig. 3. Threads and Loaded Classes

Above figure shows the threads and loaded classes at different time instances. The values of threads remains somewhat constant form the very beginning while the loaded classes increase along with the time. There are certain classes required while implementing CA such as MD5 for calculating the hash code and RSA for public key cryptography and encrypting the hash code of public key .Therefore above figure shows the different classes comes into picture as the code start running.

Another performance dimension that we have considered while implementing our code is time requirement. Firstly we have tried to analyze the time required for implantation of different packages used in the code. Following figure 4 shows the time required for executing the different major packages in the code in milliseconds.

Entity	Time	Invocation
main	12.7ms	1

Entity	Time	Invocation
Anonymous package	12.7ms	1

Entity	Time	Invocation
Java.lang	6.32 ms	18
Java.sql	5.88 ms	6
Java.net	0.339 ms	36
Java.io	0.105 ms	25

Fig. 4. Execution time of major packages

Above figure clearly shows the time required by the different major packages used in the code. Some of the major packages used in the code are java.lang, java.sql, java.io etc. The above figure also shows the invocation of different packages. It clearly shows the maximum invocation is of java.net package that is invoked 36 times and than java.io with 25 times and so on. We can also see the sequence in which different classes are called with the help of call tree functionality of the net beans profiler. Following figure 5 shows the sequence in which different classes are called.

Entity	Time	Invocation
main	12.7 ms	1

Entity	Time	Invocation
Server1	12.7 ms	1

Entity	Time	Invocation
Java.lang.class	6.7 ms	1
Java.sql.DriverManager	5.88 ms	6
Java.net.ServerSocket	0.339 ms	10
Java.lang.ClassLoader	0.248 ms	10
Java.io.PrintStream	.105 ms	6

Fig. 5. Sequence of major classes execution

Above figure shows the sequence of classes calling along with the execution time. It also shows the number of times a class is called. So by the above analysis it is clear that the overhead which is incurred in the DTLS by adding the Certificate Authority is not much while comparing its robustness. Therefore if we implement CA in any programming language like java in our case and see the execution and memory requirement is not much so it is a better enhancement of DTLS with slight overhead in execution.

VI. CONCLUSION AND FUTURE WORK

In this paper we have presented DTLS with public certificates issued by Certificate Authority. This modified DTLS can be implemented for Internet of Things (IoT). There are lot of work has already been done on DTLS but most of the given DTLS versions uses pre shared key for authentication. Certificate Authority with the help of digital certificates allows the communicating entities to use public key cryptography. By incorporating authentication with the help of digital certificates increases the robustness of the communication. But as for making the system more secure, somewhere speed of the communication is reducing. Therefore in further work DTLS can be made faster by some technique such as header compression, by reducing the message exchange for authentication via some efficient handshake implementation.

DTLS was not primarily designed for IoT but several modifications in DTLS by the working committee made it suitable for Internet of Things. As world of Internet of Things is improving rapidly then its security solution should be improved and the use of public certificate for this purpose can be very beneficial.

REFERENCES

[1] Sye Loong Keoh, Sandeep S. Kumar, and Hannes Tschofenig, "Securing the Internet of Things: A Standardization Prospective", IEEE Internet of Things journal, Vol. 1, No. 3, June 2014.

[2] T. Dierks, E Rescorla, "Transport Layer Security(TLS) protocol", IETF RFC 5246- August 2008.

[3] Daniele Trabalza "Implementation and evaluvation of Datagram Tranpsort Layer Security for the android operating system[2013].

[4] D.R. Raymond and S.F. Midkiff, "Denial of Service in wireless sensor Networks: Attacks and Defenses", Pervasive Computing, vol. 7. No. 1, PP. 74-81, Jan-Mar 2008.

[5] V. Gupta, M. Wurm, Y.Zhu, M. Millard, S. Fung, N. Gura, H. Eberle and S.C. Shantz, "Sizzle: A Standards Based End to End Security Architecture for the Embedded Internet", Pervasive Mobile Computing, Vol 1, PP 425-445, Dec 2005.

[6] Tobias Heer, Oscar Garcia – Morchon, Rene Hummen, Sye Loong Keoh, Sandeep S Kumar, and Klaus Wehrle. Security challenges in the ip- based internet of things. Wireless Personal Communications, 61(3), 527-542, 2011.

[7] M. Brachmann, S.L Keoth, O.G. Morchan and S.S Kumar, "End to End transport security in the IP based internet of things" in Proc. 21st ICCCN, Aug 2012, PP 1-5.

[8] S. Keoth, S. Kumar, and O. Garica-Morchan (2013, Feb) Securing the IP based Internet of Things with DTLS[online]. Available: http://www.ietf.org/lid-abstracts.html.

[9] Cisco (2014, Jan) The Internet of Things[online]. Available: http://share.cisco.com/internet-of-things.html.

[10] Ericsson, "More than 50 billion connected devices", Ericssion white paper 284 23-31 49 Uen. Feb 2011.

[11] Z. Shelby, K. Hartke, C. Bormann, and B. Frank.(2013, May). Constrained Application Protocol(CoAP). Internet- Draft draft-ietf-core-coap-16 [online]. Available: http://datatracker.ietf.org/drafts/current/.

