International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 6, December 2021, pp. 5379~5389
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp5379-5389

Developed security and privacy algorithms for cyber physical system

Dhuha Dheyaa Khudhur[1], Muayad Sadik Croock[2]
[1]Computer Engineering Department, University of Technology, Baghdad, Iraq
[2]Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

Article Info

Article history:

Received Feb 15, 2021
Revised May 25, 2021
Accepted Jun 11, 2021

Keywords:

CPS
DoS attack
HTTP post request
MQTT Broker
TLS

ABSTRACT

Cyber-physical system (CPS) is a modern technology in the cyber world, and it integrates with wireless sensor network (WSN). This system is widely used in many applications such as a smart city, greenhouse, healthcare, and power grid. Therefore, the data security and integrity are necessary to ensure the highest level of protection and performance for such systems. In this paper, two sides security system for cyber-physical level is proposed to obtain security, privacy, and integrity. The first side is applied the secure sockets layer (SSL)/transport layer security (TLS) encryption protocol with the internet of things (IoT) based message queuing telemetry transport (MQTT) protocol to secure the connection and encrypt the data exchange between the system's parties. The second side proposes an algorithm to detect and prevent a denial of service (DoS) attack (hypertext transfer protocol (HTTP) post request) on a Web server. The experiment results show the superior performance of the proposed method to secure the CPS by detecting and preventing the cyber-attacks, which infect the Web servers. They also prove the implementation of security, privacy and integrity aspects on the CPS.

Corresponding Author:

Dhuha Dheyaa Khudhur
Department of Computer Engineering
University of Technology
Alsinaa Street, Baghdad, Iraq
Email: ce.19.07@grad.uotechnology.edu.iq

1. INTRODUCTION

Today, Cyber-physical system (CPS) is one of the modern technology in the cyber world [1] and it is an integration for the wireless sensor network (WSN) [2]-[4]. This system consists of a physical and cyber component. A CPS aims to control and monitor the physical element behaviors, as well as it has an integration of computing, communication, and physical components [5]-[7]. The cyber component in CPS is used to process and analyze the received data from the physical components in real-time and then to take the right decision for better work, high performance of these components [8]. Due to the increased dependence on CPS, it is widely used in many applications such as power [9], transportation, factories, healthcare [10], and greenhouse. Therefore, data protection and integrity are significant to guarantee the security level of cyber-physical systems [11] to avoid many cyber-attacks [12]. Thus, we need to obtain an in-depth knowledge of all these weaknesses, risks, and attacks by analyzing CPS security and privacy controls [13], [14]. WSNs are the physical components of the CPS. Besides, these components use a lightweight message queuing telemetry transport (MQTT) protocol to organize the process of sending and receiving data across the network [15], [16]. Therefore, preserving these data has become extremely important.

In this paper, we present two aspects in CPS to cover the security and privacy. Firstly, the transport layer security (TLS) protocol is considered, which provides a secure network connection [17], privacy, and data integrity. TLS is evaluated over the MQTT network protocol by using an authentication algorithm based on certificate generation between the system devices. Secondly, we test our algorithm that has been proposed to detect and prevent DoS attack, which is (hypertext transfer protocol (HTTP) post request) on the website (cyber component). To cover the parts of the proposed CPS, more information is provided as:

a) TLS and the cipher suite

The TLS protocol is a commonly used safe connection protocol used to safe communication and protect the exchanged data between devices and ensure the authenticity and integrity of these data. There are two main protocols for the TLS: The first is the TLS handshake protocol, and the second is the TLS record protocol [18]. The TLS handshake protocol is responsible for the authentication and key exchange to create a secure session between the system's devices. Besides, this protocol manages cipher suites negotiation, session key information exchange. But the TLS record protocol is responsible for encrypting the application data and verifying its integrity and originality, using the keys created by the handshake protocol [19]. Also, this protocol manages the following:

- Splitting outgoing letters into controllable blocks and regrouping incoming letters.
- Encryption of outgoing messages and decryption of incoming messages by (AES, CHACHA20, Camellia, ARIA) encryption algorithms.
- Employing the message authentication code (MAC) to outgoing letters and verifying incoming letters using the same (MAC).
- After the TLS record protocol is finished, the sending encrypted data is passed to the transmission control protocol (TCP) layer to transfer.

b) Elliptic curve diffie-hellman ephemeral (ECDHE) Key exchange algorithm

ECDHE is an algorithm used to key exchange between two parties to establish a secret session key (premaster secret) based on temporary secrets (various secrets for each session). This algorithm is the adaption of the diffie-hellman (DH) that uses elliptic curve cryptography to minimize the key length and enhance the performance. The algorithm also provides an RSA-like protection standard but with smaller key sizes, which results in quick calculations and low power consumption for devices used. This algorithm allows each party to create a pair of keys (private key and public key). The two parties exchange their public keys. Also, they compute the master secret key by using their private key and the public key of another party. The ECDHE is one of the key exchange algorithms used in cipher suites of TLS. The TLS_ECDHE is quick than TLS_RSA. Also, TLS_ECDHE has less power consumption and better performance than other algorithms DHE, RSA, and ECDH [18].

c) Rivest-shamir-adleman (RSA) authentication algorithm

RSA is a widely used asymmetric encryption algorithm and considered one of the most secure encryption algorithms because it uses two keys, the public key for messages encryption and the private key for decryption. Besides, this algorithm is also used for the signature. To create the signing, we apply the private key while we use the public key to verify the authenticity of the signing to ensure the message's integrity [20].

d) Advanced encryption standard (AES) with secure hash function (SHA)

AES is a block cipher standard published by the National Institute of Standard and Technology (NIST) and has become one of the most popular symmetric encryption algorithms [21]. Besides, AES is considered a secure encryption method due to the length of the Key (128, 192, 256 bit) used for encryption and decryption. Also, this algorithm used with the TLS protocol to encrypt the data to ensure its confidentiality. Besides, the hashing functions as SHA that is unidirectional encryption and cannot be decrypted using decryption keys are applying for encrypted data to ensure its integrity and originality from any eavesdropping process [22].

e) Denial of service attack (HTTP post request)

In the denial of service (DoS) attack, the attacker does not steal the user's confidential data but rather consumes all the server's resources. Thus, server or communication networks become unable to deliver the required services [23]. One of the most important DoS attack is a hypertext transfer protocol (HTTP Flood) attack on the application layer or webserver. This attack includes two types of HTTP flooding: HTTP POST and HTTP GET. These attacks exploit web server vulnerabilities to send the highest amount of malicious HTTP requests in a short time to the victim's server [24].

Due to the importance of maintaining the cyber-physical systems free from any issues that cause them to fail and disrupt their functioning. So finding optimal solutions for the security of these systems are the most important topic for many researchers. We address some of the researchers' works that focused on the approaches used to secure and private the data of the CPS. The related work is structured as:

a) Security and privacy of CPS

Dikii [25] proposed an authentication algorithm between two devices via the IoT based MQTT protocol. This algorithm protected the IoT from unauthorized access and provided the privacy of data during the generation of a session key. The encryption power of the proposed algorithm was adopted on the discrete logarithm problem. Thite *et al.* [26] proposed a lightweight key generation algorithm, which is used to establish new keys for each communication session between the cyber-physical system hardware. As every session produced a new generation of keys, it was hard for cyber-attacks to crack these systems' protection. Pérez-Jiménez *et al.* [27] proposed a lightweight encryption method that uses physical unclonable functions (PUF) to create secret keys. This method indicated any attempt to obtain the secret key will change the original data flow because the manufacturer did not create two identical keys from the PUF. While the Sachian *et al.* [28] proposed cyber-physical system security for healthcare based on the IoT based MQTT protocol. This security is done by SSL/TLS encryption protocol to encrypt all data that are sent in real-time to the web from the prototype different sensors. Junejo *et al.* [29] proposed a lightweight security scheme Focused on an encryption algorithm, which was elliptic curve cryptography (ECC) for the fog-cyber physical systems (Fog-CPS). They also presented a new side of the proposed scheme to secure connections among Fog-CPS entities. Each entity has two keys, partial and final, to create the public key final for all the connecting CPS devices without the need to send and create signature certificates. Chung *et al.* [30] proposed a method to encrypt the payload of the IoT devices and provide the necessary power for them. This was done using the SSL/TLS encryption protocol with the MQTT protocol. Diro *et al.* [31] proposed a lightweight scheme that uses elliptic curve encryption to secure the IoT connection model. Besides, they analyzed and simulated the encryption as mentioned above. Amnalou *et al.* [18] proposed a lightweight authentication algorithm and encryption method for IoT restricted devices. This method utilized ECDHE-PSK, the perfect forward secrecy authentication algorithm for TLS via MQTT protocol. They tested this algorithm with a default MQTT security mechanism and an authentication algorithm based on certificate generation as ECDHE-ECDSA. Ahamed *et al.* [32] proposed the AES 256 symmetric key algorithm and Secure Hashing Algorithm-256 to encrypt and decrypt data sent between the devices via MQTT protocol. In Sadio *et al.* [33], the authors suggested an encryption method to secure the MQTT protocol using multiple layers that used authenticated encryption with associated data (AEAD). Chen *et al.* [34] proposed a framework for new distributed authentication for the M2M multi-domain environment. This work included a hybrid cryptography method under which IBC is implemented and AES.

b) Detection and prevention the cyber-attack on CPS

Hirakawa *et al.* [35] proposed a method to defend against a slow HTTP DOS attack. This attack was carried out by a single attacker that was saturating the intended with requests for processing. The proposed method prevented the saturation of processing requests by separating clients who are made connections longer than the specified threshold. Kshirsagar *et al.* [36] suggested an ontology to detect flood attacks on a web server, such as the HTTP flooding attack. This ontology, which they proposed, was useful for developing semantic rules to detect such an attack. Besides, they expanded this ontology to allow a higher detection rate for various other types of HTTP flooding attacks such as HTTP GET, and HTTP POST. While the Tripathi *et al.* [37] proposed a system to detect a slow HTTP DoS attack for evaluating the vulnerability of the web servers and other sites. The proposed system was based on anomaly measurement where measures Hollinger's distance between two probability distributions during the training and testing phase. Besides, they also assessed the detection system's effectiveness by collecting the real HTTP traffic from public web servers over the LAN. The obtained results showed high accuracy and efficiency of the proposed system in detecting slow attacks of the head and half of the message during the attack period. Brynielsson *et al.* [23] suggested developing a low-rate DoS attack emulator that exploits vulnerabilities in the Apache HTTP server program 2.2. They performed a spectral analysis to verify the detection quality of this attack. While the Mualfah *et al.* [24] suggested a method to monitor the network traffic and detect flood attacks on a web server. This was performed applying an intrusion detection system (IDS) like snort, an open-source tool that used a file to record all network traffic activities. Soliman *et al.* [38] suggested a new technique for denial of service attacks, which is a blind DoS attack. This technique combined two types of attack features, which exploited all web application weaknesses to exhaust and make it unavailable to legitimate users. Mohamed *et al.* [39], suggested a method to mitigate the HTTP application layer DDOS attack on a web server by applying a smart load balancing mechanism on the webserver. This mechanism depended on the low version of the database technique that helps absorb the impact of this attack on the webserver and increase its resistance. While the Sreeram *et al.* [40] proposed a Bio-Inspired Anomaly-based BAT algorithm based on HTTP flooding (DDOS attack) detection on the application layer. This algorithm was characterized by its speed and high accuracy to detect this attack. Chandak *et al.* [41] suggested a system divided into two modules, active and passive, to detect and deter many cyber-attacks such as SQL Injection IP spoofing, MAC attack, and DDoS web applications.

In this paper, we present two aspects in CPS to cover the security and privacy. Firstly, the transport layer security (TLS) protocol is considered, which provides a secure network connection, privacy, and data integrity. TLS is evaluated over the MQTT network protocol by using an authentication algorithm based on certificate generation between the system devices. Secondly, we test our algorithm that has been proposed to detect and prevent DoS attack, which is (HTTP post request) on the website (cyber component).

2. PROPOSED SECURE CPS

The proposed CPS security system includes SSL/TLS technique, to secure and private the proposed prototype cyber-physical system as well as detection and prevention of the DoS attack (HTTP post request) on the webserver. The cyber-physical system that is integrated with the wireless sensor network deals mainly with three nodes (NodeMCU 12E-module), Raspberry Pi3 as the base station, two sensors (DS18B20, Soil moisture), and two actuators (LED1, LED2). To send the data captured from the sensors by NodeMCU to the Raspberry Pi3, a secure Wi-Fi connection is applied through the application of the SSL/TLS encryption protocol with the IoT based MQTT protocol to secure connection and encrypt the exchanged data of the CPS. The managing procedure is done through testing the received data in real mode, under certain conditions to the environmental parameters measured by the sensors mentioned above. The base station decides to send a message to the NodeMCU to turn on/off the corresponding actuator (LED1, LED2). Figure 1 shows the block diagram of the proposed system structure that can be explained in the following sub-sections.

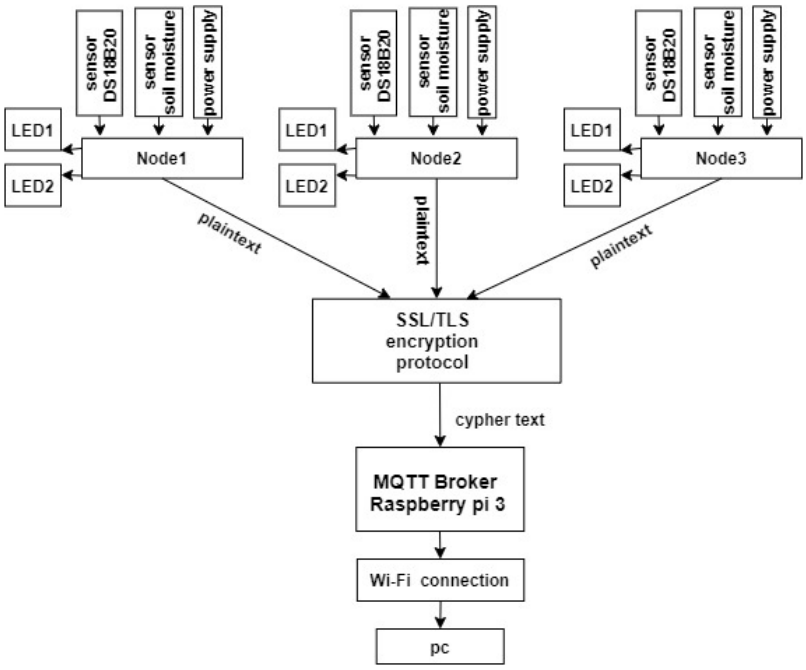

Figure 1. The block diagram of the proposed system

2.1. TLS algorithm for CPS

Since the MQTT protocol is used to send and receive the cyber system's data, securing data for these systems is extremely important. Therefore, the TLS 1.2 protocol is applied that uses safe connection protocol with IoT based MQTT protocol for end-to-end safe communication, data protection and appliances authenticity. The MQTT port (8883) is used for security instead of the default port (1883) unsecured communication. The hardware authentication with MQTT over TLS protocol is performed by certification instead of the username and password. There are two main protocols for the TLS: TLS handshake protocol, and TLS record protocol. The handshake protocol enables MQTT clients (NodeMCU ESP8266) and MQTT server (Raspberry Pi3) to verify each other and work together. It also builds a cryptosystem based on encryption algorithms and session keys for data cryptography in TLS. After the TLS handshake protocol is created, the TLS record protocol is used to secure the connection by using symmetric encryption algorithms such as AES and hash functions like SHA.

To perform the secure connection between the MQTT client and the MQTT server. First, establishing the TSP session and then creating the handshake process steps through the TLS handshake protocol to complete the authentication process. This requires the following:

2.1.1. Establishing the connection

Figure 2 shows establishing TCP session between client and server.

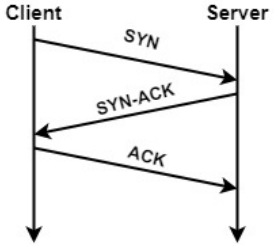

Figure 2. The TCP session

2.1.2. Negotiation

It includes two functions; Client Hello and Server Hello. The client sends a "client hello message" to the server. This message includes, TLS version, random byte number, session ID, compression_ method, and cipher suites. The cipher suites include the client supported cipher algorithms that use in the session to generate a secure connection with the server. At the server Hello, the server sends a "server hello message" to the client. This message includes, TLS version, random byte number, session ID, compression_ method, and cipher suite. The server selects the cipher suite with the highest choice given by the server and client. After the agreement is complete, the selected cipher suite, which is TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, is used during the entire session, and the secure connection is started.

2.1.3. Server authentication and key exchange

The server sends its signed certificate to the client to prove its identity. This certificate contains a server's public key. After this, the authentication takes place between the client and server. The client begins to verify the X509 server certificate in more than one way, by following:

- The client verifies the server's digital certificate by checking its expiration date and the fingerprint of the certificate. This fingerprint, a hash SHA_1 of the actual certificate and which previously exists in the client. This verification is done manually.
- The client verifies the server's public key by comparing it to the server's public key that is previously existing. This verification is done with less computation than SHA_1fingerprint.

2.1.4. Server key exchange

The server sends this message that contains on server’s DH public key when the certificate has not sufficient information for the client to generate the master secret key.

2.1.5. Client certificate request

The server sends this message to request and verify client certificate for authenticate. After this, the server sends “server hello done” to the client to inform that the “server hello message” is ended.

2.1.6. Client authentication and session key generation

The client sends its signed certificate to the server to prove its identity after receiving its "client certificate request" from the server. This certificate contains a client's public key. The server verifies from the client's digital certificate with the CA's public key (certificate authority). The client sends this message that contains on client's DH public key to the server to generate (master secret key). The client sends to the server a message that contains a signature generated by its private key to prove that it is the owner of the certificate, which was previously sent. This signature is verified by the server using the public key of the client certificate.

After completing the TLS handshake processes, "server key exchange" and "client key exchange" that are sent previously. Both parties, the client and server in the ECDHE key exchange algorithm, must

generate the master secret key. That later provides the required data to the server to generate symmetric keys encryption. The ECDHE key exchange algorithm generates a pair of keys (private key and public key) of both client and server. It also must contain each other the domain parameters (a, d, p, h, G, n) that determine the elliptic curve to generate the master secret key. So, the private key Kc (a random integer) and the public key Qc (Q=Kc G) were generated for the client, and the private key Ds and the public key Qs=Ds G for the server. The server and the client exchange their public keys. Also, they compute the master secret key by using their private key and the public key of another party. The client calculates the secret key S=Kc Ds G, and the server calculates the secret key S=Ds Kc G. So, we notice that the same master secret key for both parties was generated. The ECDHE algorithm does not authenticate on its own since each time the key is various, and any party may not be sure that the key is from the intended party. To employ the authentication, a RSA authentication algorithm is used along ECDHE for both parties. The RSA algorithm's digital signature is performed by the signer's private key, while the recipient uses the signer's public key to verify the signature.

2.1.7. End the TLS handshake

The client sends this message to the server and vice versa. This message includes that both parties are ready to switch to an encrypted environment. Thus, any data sent and received between them will be encrypted by the symmetric encryption algorithm. At the other side, the client sends this message to the server and vice versa. This message indicates to end the handshake process between them and encrypts it by a master secret key.

2.1.8. Encrypted application data

In this message, the TLS record protocol is used to encrypt the application data using negotiated encryption algorithms at the beginning of the handshake process. This is done by applying an (AES-GCM) symmetric-key algorithm, which used a secret key size (256 bit) to encrypt and decrypt. Besides, GCM is an authenticated encryption with associated data (AEAD) mode used to operate the block ciphers with 256-bit blocks and eliminate the message's separate hash function. But the hash function (SHA384) is applied to be used as part of the pseudorandom function (PRF) for key derivation from the master secret key in the TLS protocol and for authentication of the finished message. Both parties, client and server, are doing this to ensure the exchanged data's integrity and confidentiality. Figure 3 explains the flowchart of the proposed TLS. Note/the TLS encryption protocol session steps between the client and the server are updated every hour to regenerate the random keys used in cipher suites to ensure that the hacker does not eavesdrop and sniff out of such keys.

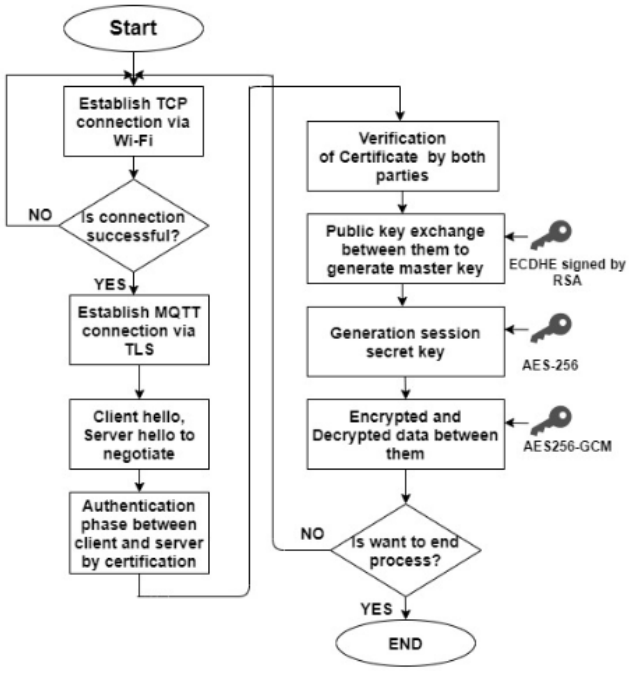

Figure 3. The flowchart for the proposed TLS

2.2. Cyber-attack detection algorithm

We proposed an algorithm to detect and prevent the effect of DoS attack that is (HTTP POST request) on the webserver. This algorithm tests the received HTTP requests by the web server by fetching the source's IP address that sent the requests and examining the time difference of the received requests from the same source within specified thresholds, which is 10 seconds. These requests increase the probability of being such are a DoS attack (HTTP post request). So, these requests are entered into the prediction stage to decided later to be DoS (HTTP post) or not. The decision is based on the time difference for the received requests from the same IP address. Figure 4 explains the flowchart of the proposed algorithm that can be summarized as:

- A request is received from any source by the webserver.
- The incoming requests are tested from the same source's IP address by examining the time difference and compared to the threshold for the sequential requests, respectively.
- If a difference is found between sequential requests, then we put the IP address under (HTTP DoS) suspicion and increase the suspicion counter by one.
- If the suspicion counter of the source's IP address is greater than or equal to the total tolerance for the incorrect requests, we block the suspicion source's IP address and consider it an (HTTP DoS) attack.

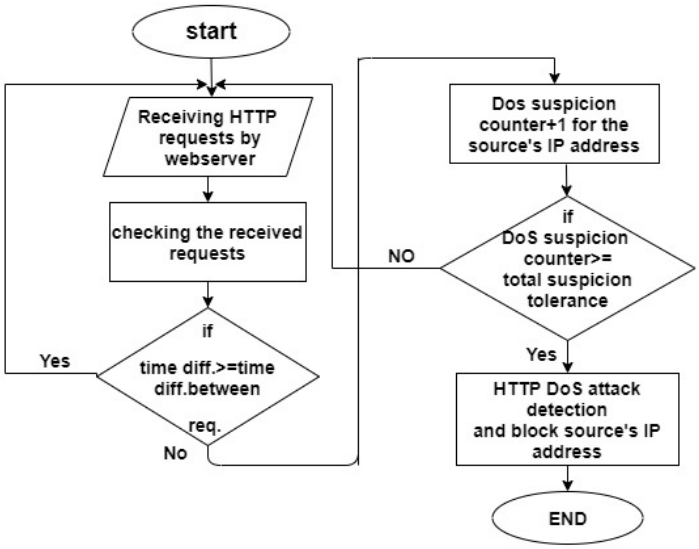

Figure 4. The flow chart of the proposed algorithm

3. RESULTS AND DISCUSSION

As mentioned above, the cyber system consists of three sensor nodes (NodeMCU ESP-12) that are communicated with the base station (Raspberry pi3) inside the local area network. To send the captured data from the sensors with more security, we applied the TLS encryption protocol with the IoT based MQTT protocol to secure the network connection and ensure the data's integrity and confidentiality. To monitor the network traffic and know whether the TLS protocol has been implemented for the exchanged packets, Wireshark platform is applied to monitor the network traffic. This platform is used to sniff and analyze the exchanged packet on the local area network. Figures 5 and 6 represent the Wireshark window that shows all the packages (publish/subscribe) topics that contains exposed and non-encrypted messages between the MQTT client (NodeMCU ESP- 12) and MQTT server (Raspberry pi3) before applying the TLS protocol.

Figure 7 represents the Wireshark window that shows all the messages that have been encrypted by TLS protocol, which is used with MQTT protocol to establish a secure connection. On the other hand, the algorithm that has been proposed to detect and prevent the effect of a DoS attack (HTTP POST request) on a web server. This algorithm tests the correctness of the received requests and prevents any malicious HTTP POST requests that lead to web server failure. Therefore, we conduct a series of tests to examine the reliability of the proposed algorithm and its impact on the inclusive performance of the webserver to send a large number of requests based on different sources in a short time to the victim's server. These tests are applied for three case studies based on the impact of a DoS attack (HTTP POST request) facing a Web server through any source.

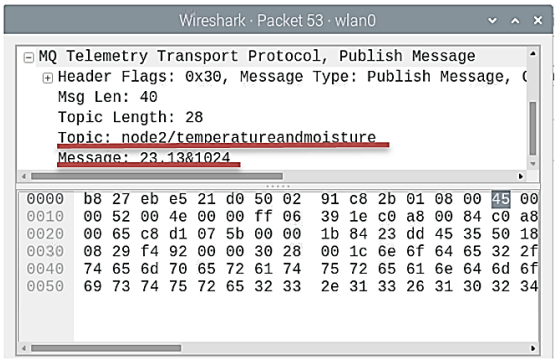

Figure 5. Wireshark window, sensors

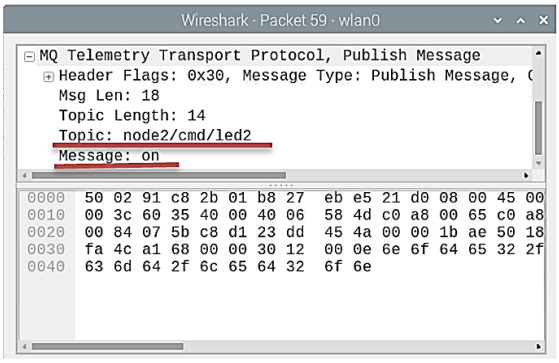

Figure 6. Wireshark window, LEDs

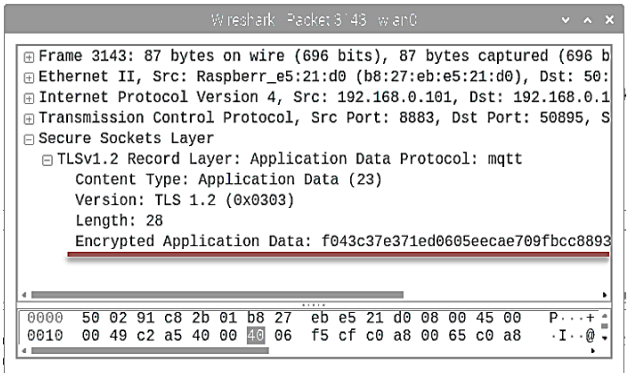

Figure 7. Encrypted messages by TLS protocol

3.1. Case study one

This study involves checking the received requests from any source through the algorithm mentioned above. If there is no difference found in the time between two HTTP requests to the same source, this case study is not regarded as (DOS HTTP post request) attack but (Normal HTTP post request) as shown in Figure 8.

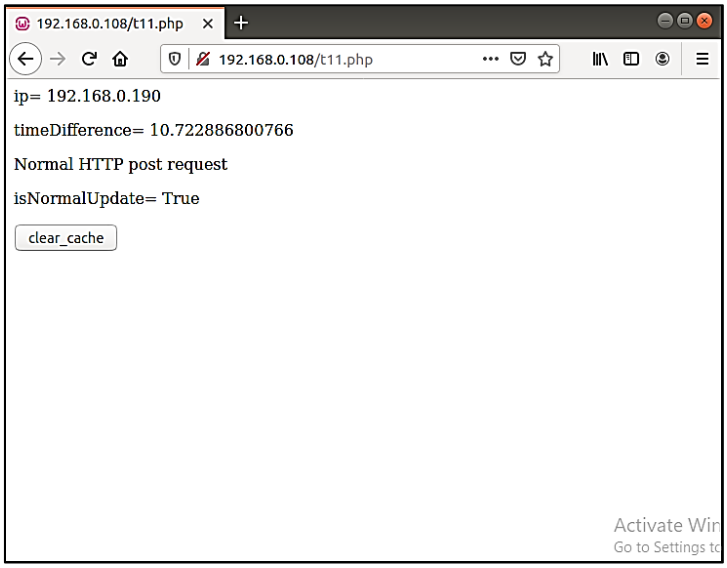

Figure 8. Case study one, normal HTTP post request

3.2. Case study two

This study involves checking the received requests from any source through the algorithm mentioned above. If a difference is found in the time between two HTTP requests to the same source, respectively, this source is considered as a DoS suspicion (HTTP post request). DoS suspicion counter is increased by one, and this source is placed under surveillance. It is necessary to note that the nature time difference threshold is 10 seconds, as shown Figure 9.

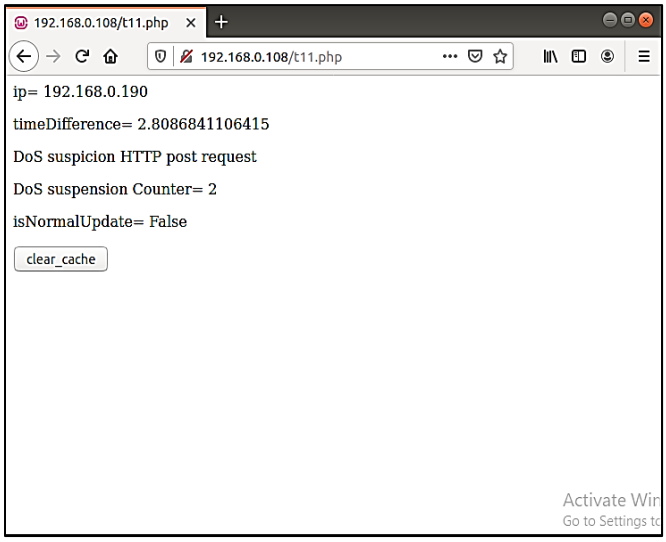

Figure 9. Case study two, DoS suspicion (HTTP post request)

3.3. Case study three

This study takes the impact of DoS HTTP post request attack on the webserver. The proposed algorithm detects this form of attack by surveillance the time difference between two HTTP requests to the same source, respectively, and compared to the threshold. If there a difference is found, this source is DOS suspicion to be monitored for five received HTTP post requests. If the thresholds are kept passed, the algorithm decides this source is a DoS attacker. As a result, the source's IP address is blocked to avoid receiving other requests from it, as shown in Figure 10. Finally, this source cannot send HTTP requests anymore.

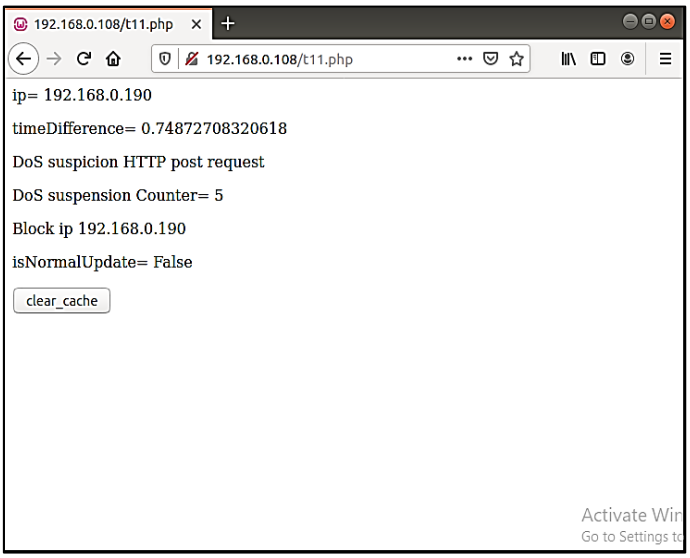

Figure 10. DoS attack detections

4. CONCLUSION

This paper aimed to present a secured CPS cyber-physical system using the TLS encryption protocol with the IoT based MQTT protocol to cover whole system. The proposed system included two sides: privacy and cyber-attack detection. The privacy was performed using TLS protocol, while the DoS attack was detected and prevented using the proposed cyber-attack detection. The cipher suite TLS_ECDHE_RSA_AES256_GCM_SHA384 was chosen to complete the authentication and encryption process between the sender and receiver through TLS. In addition, The ECDHE_RSA key exchange algorithm was signed by the RSA authentication algorithm to guarantee that the message's content does not tamper by the hacker. This was done through the AES256 algorithm with hashing function SHA384 to encrypt data for both parties. The proposed system performed in high level of satisfaction in terms of integrity for security and privacy.

REFERENCES

[1] J. Jamaludin and J. M. Rohani, "Cyber-physical system (CPS): State of the art," in *2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube),* 2018, doi: 10.1109/icecube.2018.8610996.

[2] N. Y. Kim, S. Rathore, J. H. Ryu, J. H. Park, and J. H. Park, "A survey on cyber physical system security for IoT: Issues, challenges, threats, solutions," *Journal of Information Processing Systems,* vol. 14, no. 6, pp. 1361-1384, 2018.

[3] S. Davidson, "Cyber-physical system design with sensor networking technologies," *IEEE Des. Test,* vol. 34, no. 3, pp. 105–107, 2017.

[4] A. Singh and A. Jain, "Study of cyber-attacks on cyber-physical system," *SSRN Electron. J.,* 2018, pp. 26-27, doi: 10.2139/ssrn.3170288.

[5] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial internet of things: A cyber-physical systems perspective," *IEEE Access,* vol. 6, pp. 78238–78259, 2018, doi: 10.1109/access.2018.2884906.

[6] S. Ali, T. Al Balushi, Z. Nadir, and O. K. Hussain, "Cyber security for cyber physical systems," 1st ed. Cham, Switzerland: Springer International Publishing, 2018.

[7] H. Kure, S. Islam, and M. Razzaque, "An integrated cyber security risk management approach for a cyber-physical system," *Appl. Sci. (Basel),* vol. 8, no. 6, pp. 1-29, 2018, doi: 10.3390/app8060898.

[8] J. Wurm *et al*., "Introduction to cyber-physical system security: A cross-layer perspective," *IEEE trans. multi-scale comput. syst.,* vol. 3, no. 3, pp. 215–227, 2017, doi: 10.1109/tmscs.2016.2569446.

[9] D. Baumann, F. Mager, H. Singh, M. Zimmerling, and S. Trimpe, "Evaluating low-power wireless cyber-physical systems," in *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench),* 2018, pp. 13-15, doi: 10.1109/cpsbench.2018.00009.

[10] H. Qiu, M. Qiu, M. Liu and G. Memmi, "Secure Health Data Sharing for Medical Cyber-Physical Systems for the Healthcare 4.0," in *IEEE Journal of Biomedical and Health Informatics,* vol. 24, no. 9, pp. 2499-2505, Sep. 2020, doi: 10.1109/JBHI.2020.2973467.

[11] N. Chaudhry, M. M. Yousaf, and M. T. Khan, "Security assessment of data management systems for cyber physical system applications," *Journal of Software: Evolution and Process,* vol. 32, no. 2, 2019, doi: 10.1002/smr.2241.

[12] M. N. Yasir, and M. S. Croock, "Cyber DoS attack-based security simulator for VANET," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 10, no. 6, pp. 5832-5843, Dec. 2020, doi: 10.11591/ijece.v10i6.pp5832-5843.

[13] J. Giraldo, E. Sarkar, A. A. Cardenas, M. Maniatakos, and M. Kantarcioglu, "Security and privacy in cyber-physical systems: A survey of surveys," *IEEE Des. Test,* vol. 34, no. 4, pp. 7–17, 2017, doi: 10.1109/mdat.2017.2709310.

[14] A. A. Nazarenko and G. A. Safdar, "Survey on security and privacy issues in cyber physical systems," *AIMS electron. Electr. eng.,* vol. 3, no. 2, pp. 111–143, 2019.

[15] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," *IEEE International Systems Engineering Symposium (ISSE),* 2017, pp. 1-7, doi: 10.1109/syseng.2017.8088251.

[16] H. Sardeshmukh and D. Ambawade, "Internet of Things: Existing protocols and technological challenges in security," *International Conference on Intelligent Computing and Control (I2C2),* 2017, pp. 1-7, doi: 10.1109/i2c2.2017.8321835.

[17] D. Zelle, C. Krauß, H. Strauß, and K. Schmidt, "On using TLS to secure in-vehicle networks," in *Proceedings of the 12th International Conference on Availability, Reliability and Security-ARES '17,* 2017, pp. 1-7, doi: 10.1145/3098954.3105824.

[18] S. Amnalou and K. Azmi, "Lightweight security mechanism over MQTT protocol for IoT devices," *Int. J. Adv. Comput. Sci. Appl.,* vol. 11, no. 7, 2020, doi: 10.14569/ijacsa.2020.0110726.

[19] S. Kamil, M. Ayob, S. N. H. Sheikh Abdullah, and Z. Ahmad, "Challenges in Multi-Layer Data Security for Video Steganography Revisited," *Asia-Pacific Journal of Information Technology and Multimedia,* vol. 07, no. 02, pp. 53–62, Dec. 2018, doi: 10.17576/apjitm-2018-0702(02)-05.

[20] F. J. Aufa, Endroyono, and A. Affandi, "Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm," *2018 4th International Conference on Science and Technology (ICST),* Aug. 2018, pp. 1-5, doi: 10.1109/icstc.2018.8528584.

[21] P. Kumar and S. B. Rana, "Development of modified AES algorithm for data security," *Optik,* vol. 127, no. 4, pp. 2341–2345, Feb. 2016, doi: 10.1016/j.ijleo.2015.11.188.
[22] Y. Yuan, Y. Yang, L. Wu, and X. Zhang, "A High Performance Encryption System Based on AES Algorithm with Novel Hardware Implementation," *2018 IEEE International Conference on Electron Devices and Solid State Circuits (EDSSC),* Jun. 2018, pp. 1-2, doi: 10.1109/edssc.2018.8487056.
[23] J. Brynielsson and R. Sharma, "Detectability of Low-Rate HTTP Server DoS Attacks using Spectral Analysis," *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015,* Aug. 2015, pp. 954-961, doi: 10.1145/2808797.2808810.
[24] D. Mualfah and I. Riadi, "Network Forensics For Detecting Flooding Attack On Web Server," *International Journal of Computer Science and Information Security (IJCSIS),* vol. 15, no. 2, pp. 326-331, 2017.
[25] D. Dikii, "Authentication algorithm for internet of things networks based on MQTT protocol," *Serbian J. Electr. Eng.,* vol. 17, no. 3, pp. 389–403, 2020, doi: 10.2298/sjee2003389d.
[26] S. Thite and D. S. Thakore "Key establishment algorithm for secure Cyber Physical system to prevent cyber attacks," *Regular Issue,* vol. 9, no. 2, pp. 3589–3594, 2019, doi: 10.35940/ijitee.b7683.129219.
[27] M. Pérez-Jiménez, B. Sánchez, A. Migliorini, and R. Alcarria, "Protecting private communications in cyber-physical systems through physical unclonable functions," *Electronics (Basel),* vol. 8, no. 4, pp. 1-22, 2019, doi: 10.3390/electronics8040390.
[28] M.-A. Sachian, G. Suciu, F. Osiac, R. Roșcăneanu, and R. Streche, "Cyber-physical healthcare security system based on a Raspberry Pi," *Advanced Topics in Optoelectronics, Microelectronics and Nanotechnologies X,* 2020, doi: 10.1117/12.2571307.
[29] A. K. Junejo and N. Komninos, "A Lightweight Attribute-Based Security Scheme for Fog-Enabled Cyber Physical Systems," *Wireless Communications and Mobile Computing,* vol. 2020, pp. 1–18, Sep. 2020, doi: 10.1155/2020/2145829.
[30] J. Hee Chung and T. Ho Cho "An adaptive energy-efficient SSL/TLS method for the internet of things using MQTT on wireless networks," in *Proceedings of 2016 the 6th International Workshop on Computer Science and Engineering,* 2016, doi: 10.18178/wcse.2016.06.053.
[31] A. A. Diro, N. Chilamkurti, and P. Veeraraghavan, "Elliptic curve based cybersecurity schemes for publish-subscribe internet of things," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Cham: Springer International Publishing,* 2017, pp. 258–268, doi: 10.1007/978-3-319-60717-7_26.
[32] J. Ahamed, M. Zahid, M. Omar, and K. Ahmad, "AES and MQTT based security system in the internet of things," *J. Discrete Math. Sci. Cryptogr.,* vol. 22, no. 8, pp. 1589–1598, 2019, doi: 10.1080/09720529.2019.1696553.
[33] O. Sadio, I. Ngom, and C. Lishou, "Lightweight Security Scheme for MQTT/MQTT-SN Protocol," *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS),* Oct. 2019, doi: 10.1109/iotsms48152.2019.8939177.
[34] S. Chen, M. Ma, and Z. Luo, "An authentication framework for multi-domain machine-to-machine communication in cyber-physical systems," in *2015 IEEE Globecom Workshops (GC Wkshps),* 2015, pp. 1-6, doi: 10.1109/glocomw.2015.7414062.
[35] T. Hirakawa, K. Ogura, B. B. Bista and T. Takata, "An Analysis of a Defence Method against Slow HTTP DoS Attack," *2018 International Symposium on Information Theory and Its Applications (ISITA),* Singapore, 2018, pp. 316-320, doi: 10.23919/ISITA.2018.8664272.
[36] D. Kshirsagar and S. Kumar, "HTTP Flood Attack Detection using Ontology," *Proceedings of the International Conference on Advances in Information Communication Technology & Computing - AICTC '16,* 2016, pp. 1-4, doi: 10.1145/2979779.2979794.
[37] N. Tripathi, N. Hubballi, and Y. Singh, "How Secure are Web Servers? An Empirical Study of Slow HTTP DoS Attacks and Detection," *2016 11th International Conference on Availability, Reliability and Security (ARES),* Aug. 2016, pp. 454-463, doi: 10.1109/ares.2016.20.
[38] M. Soliman and M. A. Azer, "Web application API blind denial of service attacks," in *2018 14th International Computer Engineering Conference (ICENCO),* 2018, pp. 249-253, doi: 10.1109/icenco.2018.8636115.
[39] M. A. Mohamed and N. Abdelbaki, "HTTP application layer DDoS attack mitigation using resources monitor," in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017, Cham: Springer International Publishing,* 2018, pp. 213–221, doi: 10.1007/978-3-319-64861-3_20.
[40] I. Sreeram and V. P. K. Vuppala, "HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm," *Appl. Comput. Inform.,* vol. 15, no. 1, pp. 59–66, 2019, doi: 10.1016/j.aci.2017.10.003.
[41] G. Chandak *et al*., "Attack Detection and Prevention Techniques in Web-based Applications," *International Journal of Computer Science Engineering (IJCSE),* vol. 6, no. 04, 2017.