Reliable and Secure Traffic Exchange Approach for Internet of Things (IoT) Devices

Mustafa Abdullah Azzawi
Faculty of Information Science and Technology,
The National University of Malaysia,
Bangi, Selangor, Malaysia
Mustafaa@siswa.ukm.edu.my

Rosilah Hassan
Faculty of Information Science and Technology,
The National University of Malaysia,
Bangi, Selangor, Malaysia
rosilah@ukm.edu.my

Khairul Azmi Abu Bakar
Faculty of Information Science and Technology,
The National University of Malaysia,
Bangi, Selangor, Malaysia
khairul.azmi@ukm.edu.my

***Abstract*— Recently, Internet of Things (IoT) has become one of the most developing network environments. Enhancement in electronic and microcontrollers has resulted into increasing number of electric devices which are connected to the internet. Therefore, secure data exchange between IoT devices and other systems is required to prevent any attempts of attacks which can lead to dangerous consequences. Authentication is a critical stage for an efficient security level. IoT device has constrained neither resources which require a lightweight mechanism that neither consumes power nor resources. Thus, in this paper, a novel authentication mechanism for IoT networks has been proposed. This mechanism mainly depends on Elliptic Curve Cryptography (ECC) algorithm over the Constrained Application Protocol (CoAP). The proposed authentication approach provides an efficient authentication mechanism with a high level of security. Implementing ECC algorithm of small key of the light weight web protocol CoAP can minimize the computation and energy resource with powerful security features.**

***Index Terms*— IoT, Authentication, CoAP, ECC.**

I. INTRODUCTION

IoT is one of the most common research topics. Advances in electronics, IPv6 and wireless networks deployment have led to a growth of the IoT technology. With the speedy development of IoT devices and technologies, IoT has spread widely and has been used in different environments, including homes, health care [1] institutes, aerospace and various transportations. Controlling systems and IoT combination represent one of the main concerns of researchers. Different approaches have been proposed to control IoT devices. IoT security has been the highest priority concerns and has become the first topic for research in the field of IoT [2]. Multiple devices including embedded sensor, wireless communication, processor, can connect to each other to form the network of IoT. Based on these contents, the IoT network has two main characteristics: the first one is that IoT is considered as a network extension on the basis of the Internet, where it integrates mobile communication networks, sensors networks and the Internet. The second characteristic is that the IoT clients can be extended to anything and are able to exchange data directly [3]. The IoT environment requires a protocol architecture which has the ability to provide efficient performance in dealing with a big amount of information computation, and queries in providing new paradigms for data processing, stream processing, aggregation, and data mining that may be sustained using communication standards like HTTP [4] and Internet Protocol (IP). However, to address the requirement of the IoT device resources, a low power consumption mechanism is required where IoT devices that have Internet access are powered using small batteries or limited resources of energy. Energy can be wasted in any unneeded data transmission which can be resulted from the protocol overhead, or any miss-optimized communication patterns [5]. The remaining part of this paper is divided as follows: Section 2 reviews and discusses previous related work. Then, in section 3, the proposed authentication mechanism is illustrated and described. Section 4 presents the experimental results and compares the performance of the proposed approach with that of the state of art authentication mechanism. In the last section, conclusion and future work implications are provided.

II. LITERATURE REVIEW

Different authentication mechanisms have been proposed for different fields using different protocols and encryption mechanisms. Certificates have been also widely deployed for authenticating Wireless sensor networks and IoT devices. In addition, a two-phase authentication protocol for wireless sensor networks in distributed IoT applications has been proposed in [6]. It supports resource limitation of sensor nodes and takes into consideration network scalability and heterogeneity. This protocol also encompasses an end to end application layer authentication approach, and it depends on other lower layer security features. A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways has been proposed in [7]. It mainly depends on the certificate-based DTLS handshake protocol to authenticate medical sensors. Moreover, DTLS handshakes depend on ECC, RSA and X.509 certificates. Another certificate-based authentication mechanism with the TPM is provided in [8]. In this authentication mechanism, both direct user's request access to IoT devices and user's request access to IoT devices are performed through the IoT Cloud, where certificates are issued by a trusted third party, such as a Certification Authority (CA). However, the main drawbacks of the certificate-based approach using a third-party certificate authority are the high communication and computation overhead resulted from certificates and the vulnerability of the

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)

authentication mechanism to attacks of certificates. The ID-based approach utilizes a sensor identity to complete authentication, and its identity can be hardware or software based. The enhanced mutual authentication model of IoT using RFID [9] improves the algorithm of authentication of the challenge-response-based RFID authentication protocol for a distributed database environment, thus making it more suitable to an IoT control system environment. An ID-based multiple authentication scheme against attacks in wireless sensor networks [10] proposes that each sensor node should be authenticated by its neighbor's ID. The authentication process is performed at the sink that maintains a binding list of nodes in authentication neighbors' ID**.** A dynamic ID-based authentication scheme for M2M communication of healthcare systems [11] uses IoT device ID as a secret key in the system to mutually authenticate each other where the probabilistic key management framework is being applied to assign key information to each node. Yet, the main drawbacks of the ID-based authentication are the restrictions of the RFID support device and most of ID-based authentication depends on hashing for encrypting data which is vulnerable to different types of attacks. The group-based authentication approach mainly depends on a group communication model, Threshold Cryptography-based Group Authentication (TCGA) scheme for the Internet of Things (IoT) [12].

Other authentication mechanisms mainly depend on application layer protocols and encryption protocols to provide a reliable authentication mechanism. Examples of these are ECC with HTTP used in [13], and AES with CoAP been used in [14], where 128-bit is used. OAuth 2.0 protocol was used in [15], where Security manager using OAuth 2.0 protocol was also built as a third party. Thus, the HTTP protocol adds extra communication overhead than CoAP, however AES require long keys to provides reliable and secure communication.

III. RELIABLE AND SECURE AUTHENTICATION MECHANISM

Preventing resource exhaustion in IoT environments has been the highest concerns for researchers in developing approaches. The resource restricted nature of the IoT environment devices requires an authentication mechanism that fits the limited memory, processing and energy of IoT devices.

In this section, we describe in details the phases of an ECC based authentication mechanism for embedded devices which are CoAP clients. Various authentication protocols for IoT devices have been proposed in previous research. However, the idea of using a password-based ECC encryption for IoT device communication authentication is novel.

Proposed authentication architecture implements the ECC authentication mechanism over the CoAP connection. By combining these two approaches, optimized overhead can be added to the IoT network, which leads to minimizing the communication and processing required to authenticate IoT devices and achieving a powerful security level. Our proposed protocol consists of four main phases as shown in Figure 1 the initialization phase, the registration phase, the authentication phase and finally the encryption phase.

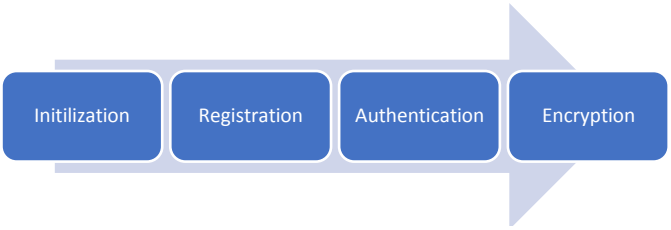

Fig. 1. Proposed Mechanism Phases

The IoT devices are configured with TCP/IP protocol stack to act as a network node and communicate with a control machine to deliver data. Using ECC on the top of the CoAP protocol for IoT device authentications with the server is a novel authentication approach which is based on the CoAP IoT application protocol. It is designed for IoT devices which have limited resources and computation capabilities. The notation which describes different mechanism process is shown in Table I.

TABLE I. Notations used in proposed protocol

Symbols	Definition
I_n	IoT Device *n*
ID_n	The ID of the IoT Device *n*
K_v	The private key of the control machine
K_p	The public key of the control machine
R_n	Random number for device *n*
PW_n	Generated Password for device *n*
H()	One way has function
G	ECC Generator point
P_1	ECC Point 1
P_2	ECC Point 2
|	Concatenation operation
$\oplus$	XOR Operation

A) Initialization Phase

This phase includes the process of public and private key generation for the control machine. The control machine starts the process by implementing the elliptic curve equation where $Y^2 = x^3 + ax + b$ *over* Z_p *where* Z_p $(p > 2^{16})$ is the finite field group. *a*, *b* belongs to Z_p is selected by the control machine to meet the equation of $4a^3 + 27b^2 \neq 0$. Suppose that the Generation Point *G* is the elliptic curve base point with a prime order of *n* where $n > 2^{160}$ and *O* be the point at infinity such that $n \times G = O$, then the control machine will select a random number K_v as its private key. K_p Is the public key of the control machine where $K_p = K_v$ x *G*. A 256 bit ECC encryption key has been used which can provide encryption level up to 3072 bit of RSA encryption.

1. Control Machine uses ECC: $Y^2 = x^3 + ax +$ b over Z_p where $4a^3 + 27b^2 \neq 0$.
2. Control Machine selects the generation point *G*.
3. Control Machine selects K_v.
4. Control Machine generates $K_p = K_v$ x *G*.

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)

B) Registration Phase

This phase includes the IoT device registration process of the control machine. To perform this operation, IoT device (I_n) sends a unique identification number (ID_n) to the control machine. When the control machine receives the IoT device unique ID, it generates a random number (R_n) and uses it to generate a unique password (PW_n) for this IoT device. Where:

$$PW_n = H\ (K_v \oplus R_n \oplus ID_n\ ) \quad (1)$$

The generated password (PW_n) is then stored as an ECC point($PW_n{}^c$) using the ECC generation point G where:

$$PW_n{}^c = PW_n\ x\ G \quad (2)$$

The generated password ($PW_n{}^c$) and the password expiration time (*EXP_TIME*) are sent to the IoT device.

1. IoT Device sends its unique ID to the control machine.
2. Control machine generates a random number for the corresponding IoT device.
3. Control machine use generated random number to calculate a password for that IoT device.
4. Password is then stored as an ECC point.
5. ECC Stored password and Expiration Time is sent back to IoT device.

C) Authentication Phase

When IoT device receives, the password sent by the control machine then, it generates a random number *N* and uses it to calculate the two parameters P_1 and P_2 using the following equations:

$$P_1\ = N\ x\ G \quad (3)$$

$$P_2\ = H(N\ x\ PW_n{}^c) \quad (4)$$

After calculating these two parameters, the IoT device sends it with a device unique ID to the control machine for authentication. As the control machine receives P_1, P_2 and ID_n from the IoT device which asks for authentication, it recalculates the password for this IoT device PW_n using the previously generated random number (R_n) and the device ID (ID_n) where:

$$PW_n = H\ (K_v \oplus R_n \oplus ID_n\ ) \quad (5)$$

Then, the password is authenticated using the following equation where the value of P_2 is retrieved using both P_1 and PW_n.

$$P_2 = H\ (P_1\ x\ PW_n) \quad (6)$$

If the calculated value of P_2 equals the sent value of P_2 then the IoT device is authenticated.

1. IoT device generate a random number.
2. IoT device calculates the two authentication parameters P1, P2 using generated number and received password.
3. IoT device send the two-authentication parameter and device ID to the control machine.
4. Control machine recalculate the IoT device password.
5. Control machine calculate the value of P2 using stored password.
6. Resulted P2 is compared against received P2, if both are equal, then IoT device is authenticated, otherwise request is rejected.

D) Encryption Phase

When the IoT device is authenticated, a session key is agreed between the IoT device and the control machine to encrypt the data exchange between the IoT device and the control machine. Thus, the session key can be calculated based on the following equation:

$$Session_{key} = H\ (PW_n{}^c |\ P_2\ |\ ID_n\ ) \quad (7)$$

All symbols are shown in Table I and all operations are self-defined. Figure 2 shows the exchange of the proposed authentication mechanism, where the four phases for both IoT devices and control machine are illustrated.

1. Control Machine generate and use Session key.
2. IoT device generate and use session key.

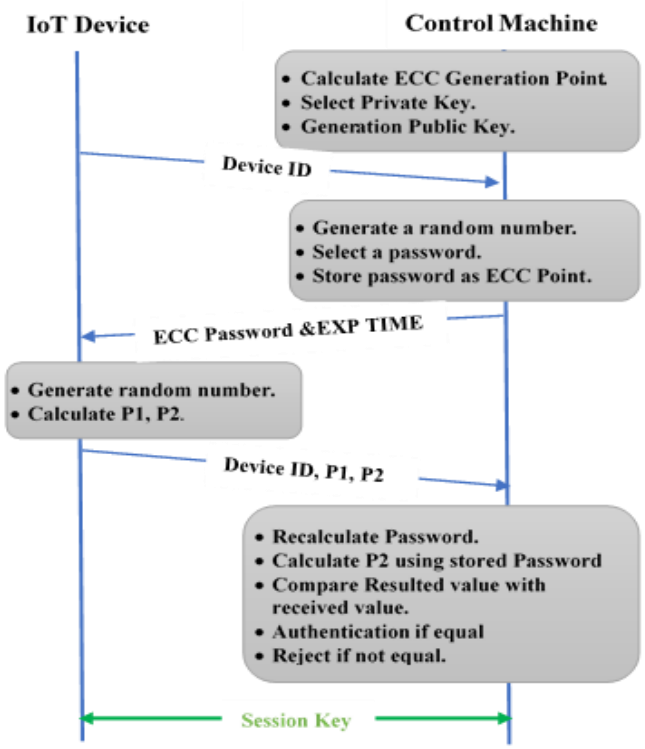

Fig. 2. Exchange the Proposed Authentication Mechanism

IV. EXPERIMENTAL RESULTS AND PERFORMANCE COMPARISION

In order to measure the performance of the proposed authentication method, a suitable simulation environment should be selected. Measuring the performance of the proposed method can be measured using both hardware and software environments. However, implementing software simulation can be less expensive and more flexible. To implement the proposed authentication mechanism, Cooja simulator [16] and Contiki OS[17] are used.

A) Simulation Topology

The network topology that is used to estimate the performance of the proposed authentication mechanism is shown in Figure 3. It mainly contains eight sensors of two main mote types Z1

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)

and sky mote. This network monitors a patient room in a hospital. As shown, sky motes are placed to monitor the room temperature and light. However, Z1 motes are used to detect the status of the patient and its position. It can be used to detect its status if the patient is standing, walking, running or falling. In addition, the patient's temperature will also be monitored. Data will be collected using the monitoring system which the CoAP server is using the border router where all data will be forwarded from sensors with router range using wireless communication. Monitoring the system and sensors communication is implemented using the CoAP application.

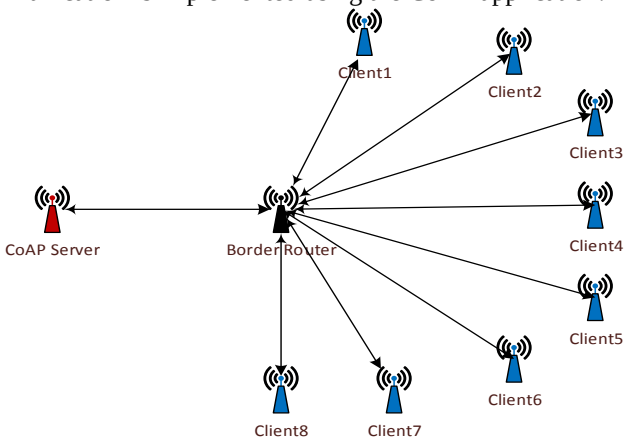

Fig. 3. Simulation Topology

B) Simulation Parameters

The following table illustrates the working protocol at different implementation layers. ECC and DTLS are used for encryption. The network topology mainly depends on the REST API which is an approach to communications that is often used in the development of web services. Table II and III illustrate the simulation parameters used in evaluating the performance of our proposed approach.

TABLE II. Simulation layer parameters

Stack Layer	Working Protocol
Radio	IEEE 802.15.4
Link	Ethernet
Encryption	ECC, DTLS
MAC	CSMA
Network	IPv6/6LoWPAN
Mesh	RPL
Transport	UDP
Application	CoAP
API	REST

The number of IoT devices is set to 8 nodes and the default CoAP packet size is used. IoT devices send a regular 1 packet at each second. For power consumption, the values used are based on the datasheets of both Z1 and sky motes, and power consumption for each status is set depending on the average values specified in these datasheets[18, 19].

TABLE III. Simulation Parameters

Parameter	Value
Simulation Time	100 seconds
CoAP Clients Number	8 clients
COAP message size	69 bytes
Receive Consumption	60 milliwatt
Transmit Consumption	53.1 milliwatt
Active Consumption	5.4 milliwatt
Idle Consumption	0.163 milliwatt
Transmission Rate	1 packet /second

Three main scenarios were simulated to evaluate the performance of the proposed authentication mechanism. These three scenarios are:

- Traditional CoAP without authentication
- CoAP with DTLS authentication
- CoAP with ECC authentication

C) Results and Discussion

In this section, the communication overhead and cost are compared against the three authentication mechanisms. The network performance metrics is compared to investigate the consequence of using different approaches in term of consumed bandwidth, delay and consumed power. Furthermore, the steadfastness in the face of multiple common attacks is investigated and discussed in this paper. The vulnerabilities of compared scenarios are illustrated as well.

1) Communication Cost and overhead

The communication cost includes consumed bandwidth, end to end delay, and consumed energy. The overhead for the traditional CoAP data transfer which does not include any extra communication packets and the size of the packet is equal to 69 bytes. The Datagram Transport Layer Security (DTLS) adds high communication overhead when it is used for constrained devices. Different encryption key size can add 91 up to 156 bytes based on the encryption key size [20]. Once the DTLS session is established, DTLS with AES CCM 8 cipher adds 29 bytes to each datagram (including an 8-byte nonce and 8-byte authentication tag) [21]. In order to simulate the CoAP with DTLS, the CoAP packet size increased to 100 bytes where 29 bytes were added to the packet and about 200 bytes were added to the whole traffic, thus resulting into two extra bytes for each packet. Therefore, 31 bytes were added as overhead related to DTLS handshaking and packet overhead. The proposed mechanism communication overhead involves adding security parameters in the authentication mechanism which is transmitted between the IoT devices and the control machine. The proposed mechanism also uses the ECC key of a length of 256 bits. The communication parameters involve $P1$, $P2$, $PW_n{}^c$, $Session_{key}$ which has a length of 256 bits and the device Identity ID_i which has a length of 128 bits and which is sent twice. Based on the transferred parameters, the communication overhead is = (4 * 256) + (2 * 128) = 1280 bits ≈ 2 bytes. Therefore, it adds two extra bytes to the trinational CoAP.

The consumed bandwidth is the first metric. The consumed bandwidth of the link is measured to represent the added communicating overhead for each authentication mechanism using the following equation mathematics:

$$\text{Consumed Bandwidth} = \frac{\text{No of received packets} * \text{Packet Size} * 8}{\text{Total Simulation Time}} \quad (8)$$

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)

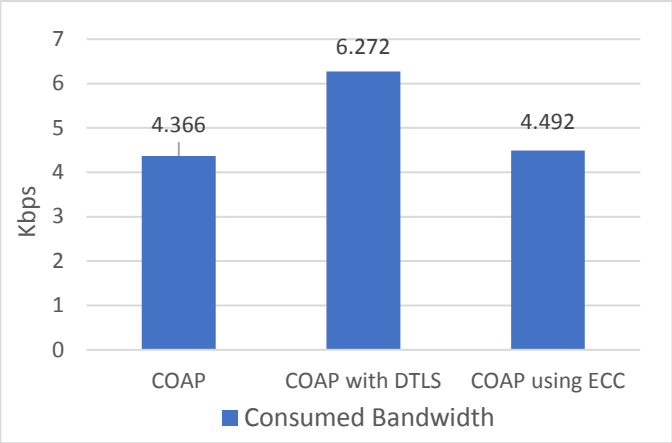

Fig. 4. Consumed bandwidth for CoAP, CoAP with DTLS and CoAP using ECC

As illustrated in Figure 4, using authentication and encryption mechanism can result into increasing the consumed bandwidth, where the extra packets are added for negotiation and such packets may include extra bytes for handling the encryption parameters. The proposed authentication mechanism using the ECC produces less bandwidth overhead when it is compared with the CoAP over DTLS mechanism which adds higher bandwidth overhead. Therefore, the proposed mechanism adds only 3% bandwidth overhead over pure unsecure CoAP however, it decreased the bandwidth overhead to 28% than CoAP with DTLS.

The second network metrics is the end to end delay, which was, measured by recording the packet send time and the ACK received time and calculating the difference between them. This value is divided into 2 to get the value of one direction. The value is calculated for each packet. The average value is also calculated by dividing the sum of all time difference by the number of packet. Therefore, the overall end to end delay is calculated based on the following equation:

$$\text{E2E Delay}(ms) = \frac{\sum(\text{ACK arrive time} - \text{PKT send time})}{\sum \text{No of packets}} /2 \qquad (9)$$

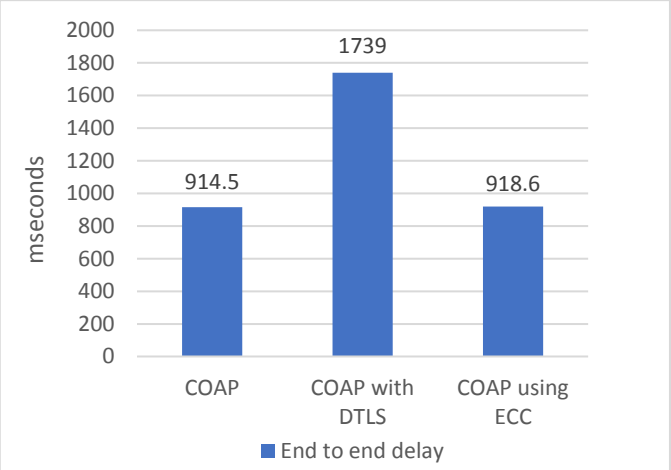

Fig. 5. End to end Delay in milliseconds for CoAP, CoAP with DTLS and proposed mechanism

As illustrated in Figure 5, the delay of the proposed mechanism is 47% lower than CoAP with DTLS. However, it adds only 0.4% to the traditional CoAP. The DTLS handshaking process and high packet overhead result into higher processing time. The proposed mechanism includes a simple handshaking process with a small number of packets.

Energy consumption is the third metric, and increasing the traffic volume results into higher processing. As illustrate in Figure 6, in terms of the consumed energy, it only 1.6% to the traditional CoAP traffic energy consumption. However, the DTLS consumes more energy, which is around 25%. The energy was also calculated during the simulation period where the consuming power for each node status is defined in CoojaTrace [22].

These statuses include sending, receiving, active and idle status as well as the initial power. For each node status in the simulation time, CoojaTrace calculates the node energy based on predefined values and logs the final remaining value. The consumed energy for CoAP Server and Router was calculated based on the following equation:

$$E_c = E_s + E_r + E_a + E_i \qquad (10)$$

Where E_c represents the total consumed power of the node and E_s stands for the data sending consumed power, while E_r is the data receiving consumed power, and E_a is the active state consumed power. Finally, E_i refers to the idle state consumed power.

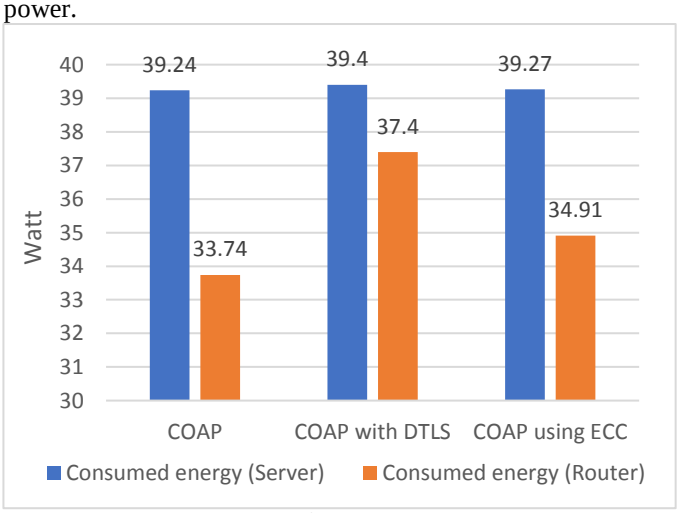

Fig. 6. Consumed energy in watt for CoAP, CoAP with DTLS and proposed mechanism

2) *Attacks Resistance*

Attacks resistance can be defined as the ability to overcome different types of authentication attacks including:

- **Eavesdropping attack**: Attackers can retrieve the IoT device's secret information by eavesdropping that enables the attacker to launch another attack by using the eavesdropped message[23].
- **Traffic analysis**: Attackers can analyze the traffic during communication between the IoT device and the control machine. By analyzing such information, the attacker can retrieve information needed to authenticate the device.
- **Replay attack**: Popular attacks occur in cases when intruders transmit a message acquired by eavesdropping on a regular communication between control machine and IoT device during the process of authentication.

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)

- **Man-in-the-middle attack**: This is similar to the replay attack, but intruders impersonate a legal device by sending back a reply message that is needed from the IoT device by impersonating a legal control machine.
- **Cookie theft attack**: In this attack, the cookie used for authentication is acquired by intruders and it is used to connect to the control machine.
- **Offline dictionary attack**: Intruders capture a transmitted packet and try to obtained security parameters from the previously transmitted packet.
- **Certificate and RSA-Related Attacks**: This type of attacks is related to certificates by utilizing exploitable timing issues and certificate attacks.

The proposed mechanism using the CoAP with ECC provides confidentiality where all data exchanged between the CoAP client and the server are protected using the ECC points and hash functions. It mainly confirms that only authenticated IoT clients can get access to the corrected Server. In addition, the CoAP with ECC is not vulnerable to traffic phishing and eavesdropping attempts. Therefore, it guarantees a high level of confidentiality. It also ensures a high complexity against any brute force attack attempts.

Based on the CoAP with ECC, it can be confirmed that no malicious client is able to retrieve IoT device information where a new and fresh password is generated with any new authentication with the CoAP server which can avoid any replay attack attempts.

Anonymity is also ensured using the CoAP with ECC mechanism where no attackers can act as the CoAP server and this malicious CoAP Server cannot retrieve a previously generated password for a potential CoAP client. Therefore, authentication parameters cannot be verified and thus cannot be authenticated. Based on this mechanism, anonymity is confirmed for all CoAP clients and any CoAP server attacker's sessions will not be authenticated.

Therefore, the results have been analyzed where the difference in bandwidth was related to the packet size overhead added by ECC over CoAP mechanism is much lower than to the overhead of DTLS and very close to traditional CoAP with two bytes overhead. Therefore, the delay is lower than DTLS where much less computation is required. For energy consumption as show in the results, ECC over CoAP consumes less energy where less communication and computation overhead is required. As shown in Table IV is security and status comparison for each scenario.

TABLE IV. Security and Status Comparison

Items	CoAP	CoAP with DTLS	CoAP with ECC
Packet overhead	No overhead	High	Low
Power consumption	Vulnerable	High	Low
E2E Delay	Low	High	Low
Eavesdropping	Vulnerable	Secure	Secure
Traffic analysis	Vulnerable	Secure	Secure
Replay attack	Vulnerable	Secure	Secure
Man-in-the-middle attack	Vulnerable	Secure	Secure
Cookie theft attack	Vulnerable	Secure	Secure
Offline dictionary attack	Vulnerable	Secure	Secure
Certificate and RSA-Related Attacks	Vulnerable	Vulnerable	Secure

V. CONCLUSION

The main goal of this paper is to provide a reliable and secure authentication mechanism for IoT networks. IoT concepts, architecture and protocols have been investigated and reviewed. The restrictions or limitations of the IoT device make the traditional network mechanism authentication unreliable where high communication and computation overhead is required. Therefore, we proposed a mechanism that utilizes shorter keys and higher security of ECC for the authentication mechanism over CoAP protocol which is a light weight application protocol.

One of the interesting aspects for future work is the group-based authentication, where a group of IoT devices can authenticate itself as a single unit with control machines. Inter communication between IoT devices can be done to minimize the communication overload with the control machine. This enhanced authentication mechanism can fit the requirement of high density IoT device networks where reduction of authentication cost is required.

ACKNOWLEDGMENT

The authors would like to acknowledge the assistance provided by the Network and Communication Technology Research Group, FTSM, UKM in providing facilities throughout the research. This project is partially supported under the Fundamental Research Grant Scheme FRGS/1/2015/ICT03/UKM/02/2.

REFERENCES

[1] M. A. Azzawi, R. Hassan, and K. A. A. Bakar, "A Review on Internet of Things (IoT) in Healthcare," International Journal of Applied Engineering Research, vol. 11, pp. 10216-10221, 2016.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, vol. 54, pp. 2787-2805, 2010.

[3] Y. Liu and G. Zhou, "Key technologies and applications of internet of things," *in Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on*, 2012, pp. 197-200.

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, *et al.*, "Hypertext transfer protocol--HTTP/1.1," ed: RFC 2616, June, 1999.

[5] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, *et al.*, "Standardized protocol stack for the internet of (important) things," *Communications Surveys & Tutorials, IEEE*, vol. 15, pp. 1389-1406, 2013.

[6] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, 2014, pp. 2728-2733.

[7] S. R. Moosavi, T. N. Gia, A.-M. Rahmani, E. Nigussie, S. Virtanen, J. Isoaho, *et al.*, "SEA: a secure and efficient authentication and authorization architecture for IoT-based

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online) © 2016 MID Publications in association with Vidya Publications.

healthcare using smart gateways," *Procedia Computer Science*, vol. 52, pp. 452-459, 2015.

[8] L. Barreto, A. Celesti, M. Villari, M. Fazio, and A. Puliafito, "An Authentication Model for IoT Clouds," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, 2015, pp. 1032-1035.

9] J.-c. YANG, P. Hao, and X. ZHANG, "Enhanced mutual authentication model of IoT," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, pp. 69-74, 2013.

[10] S. Peng, "An Id-based Multiple Authentication scheme against attacks in wireless sensor networks," in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, 2012, pp. 1042-1045.

[11] T. D. Nguyen and E.-N. Huh, "A Dynamic ID-Based Authentication Scheme for M2M Communication of Healthcare Systems," *Int. Arab J. Inf. Technol.*, vol. 9, pp. 511-519, 2012.

[12] P. N. Mahalle, N. R. Prasad, and R. Prasad, "Threshold Cryptography-based Group Authentication (TCGA) scheme for the Internet of Things (IoT)," in *Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014 4th International Conference on*, 2014, pp. 1-5.

[13] S. Kalra and S. K. Sood, "Secure authentication scheme for IoT and cloud servers," *Pervasive and Mobile Computing*, vol. 24, pp. 210-223, 2015.

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

[15] S. Emerson, Y.-K. Choi, D.-Y. Hwang, K.-S. Kim, and K.-H. Kim, "An OAuth based authentication mechanism for IoT networks," in *Information and Communication Technology Convergence (ICTC), 2015 International Conference on*, 2015, pp. 1072-1074.

[16] A. Sehgal, "Using the Contiki Cooja Simulator," *Computer Science, Jacobs University Bremen Campus Ring*, vol. 1, p. 28759, 2013.

[17] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, 2004, pp. 455-462.

[18] W. Zolertia, "platform, Z1 Datasheet," ed.

[19] M. Corporaton, "Tmote Sky: Datasheet," ed, 2006.

[20] Hartke, K., & Bergmann, O. (2012). Datagram Transport Layer Security in Constrained Environments. draft-hartke-core-codtls-01 (work in progress).

[21] Vucinic, M., Tourancheau, B., Watteyne, T., Rousseau, F., Duda, A., Guizzetti, R., & Damon, L. (2015). DTLS Performance in Duty-Cycled Networks. arXiv preprint arXiv:1507.05810.

[22] Strübe, M., Lukas, F., & Kapitza, R. (2012). Demo Abstract: CoojaTrace, Extensive Profiling for WSNs. Paper presented at the Poster and Demo Proc. of the 9th European Conf. on Wireless Sensor Networks (EWSN 2012).

[22] A. S. Ahmed, R. Hassan, and N. E. Othman, "Security threats for IPv6 transition strategies: A review," in Engineering Technology and Technopreneuship (ICE2T), 2014 4th International Conference on, 2014, pp. 83-88.

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)