ACIT'2017
The International Arab Conference on Information Technology
Yassmine Hammamet, Tunisia
December 22-24, 2017

Secure and Energy-effictive CoAP Application Layer Protocol for the Internet of Things

FirasAlbalas
Department of Computer Science
Jordan University of Science and Technology
Irbid, Jordan
Faalbalas@just.edu.jo

Majd Al-Soud
Department of Computer Science
Jordan University of Science and Technology
Irbid, Jordan
majdsoud5@gmail.com

Omar Almomani,
Network Computer and Information Systems Department
Faculty of Information Technology
The World Islamic Sciences & Education University
Amman, Jordan
Omar.almomani@wise.edu.jo

Ammar Almomani
Department of Information Technology,
Al-Huson University College,
Al-Balqa Applied University, Jordan
ammarnav6@bau.edu.jo

***Abstract*—Nowadays, the concept of the Internet of Things (IoT) has become more noticeable and realistic where it is being used in all aspects of life, such as smart cities, agriculture, military surveillance, home automation, security, healthcare, etc. The idea behind IoT is making everyday objects, especially those with constrained resources, to communicate effectively with each other and with the internet; making it possible for anyone to communicate with both digital and physical worlds at any time. To enable the communication of constrained devices the Internet Engineering Task Force (IETF) came up with a Constrained Application Protocol (CoAP) to be used in the IoT application layer as a replacement for the HTTP protocol. However, the heterogeneity of the constrained devices and the complexity of the internet bring up the need for a security system to secure all the communications, data and participating things. In this paper, we proposed a lightweight secure CoAP using Elliptic Curve Cryptography (ECC) to transport security between IoT objects and the resource directory (RD). The advantage of using ECC is its compact key size enabling it to utilize a smaller key size compared to the other identical methods such as RSA. This, in turn, increases the computational speed and at the same time saves more energy. This paper mainly concentrates on comparing CoAP using the ECC method to CoAP using the *Rivest–Shamir–Adleman* (RSA) Cryptography algorithm in terms of the energy consumption.**

Keywords— ***Internet of Things;CoAP; ECC; Energy saving, security, IoT.***

I. INTRODUCTION

In 1999, Internet of things (IoT) was at first originated by Kevin Ashton in the supply chain industrial management context [2]. Nevertheless, throughout the past years, this definition has been extended to cover a wider range of fields and applications such as transport applications, healthcare applications, utility applications, etc. In spite of "Things" definition has completely changed as new technologies evolved the trends of creating and making computer sense information without any type of human assistance or intervention remain the same. The growth and evolution of the Internet to a network of interconnected things which can not only collect information from the around world through sensing as well as interacting with the physical environment by controlling, actuating, sending or receiving commands, but it can also use the available internet standards to supply services for many applications such as data transfer, information analytics, and communications. With the large spread of the devices that were enabled by the wireless networks, actuator nodes, and embedded sensors, IoT has emerged from its cradle as well as is on the brink of transforming the traditional Internet to a complete integrated interconnected future Internet [3].

The recent revolution of the Internet has resulted in new connections between groundbreaking scale and people. The next Internet revolution will guarantee connections between objects and things to establish smart environments.In 2011, the number of the interconnected objects and devices in the world started to exceed the people number [1].

Currently, there exist 9 billion connected devices to the internet and by 2020 it is expected to reach up to 24 billion devices. This neutralizes to $1.3 trillion total revenue opportunities for the operators of the mobile networks that transferring essential segments such as utilities, automotive, health and consumer electronics according to GSMA association [4].

IoT's has added a new dimension to the world of Information and communication technologies, by creating a new form of communication between things and people, between things themselves, connecting everyday devices such as smart-phones, internet TV's, sensors and actuators to the internet. To enable these low-power devices with limited processing capabilities to participate in the IoT, standardization organizations and the research community have defined several architectures and protocols. Therefore, the concept of 6LoWPAN originated, which stands for IPv6 over Low Power Wireless Personal Area Network, to enable IPv6 connectivity even to the smallest objects. IPv6 and IPv4 are used to deliver data for local area, metropolitan area and wide area networks.

The 6LoWPAN protocol enables IPv6 packets to be sent to and received from over IEEE 802.15.4 based networks, which provides the devices with sensing communication-ability in the wireless domain. However, to cope with constrained resources and the size limitations of IEEE 802.15.4 based networks, the 6LoWPAN group has defined the header compression mechanisms, although the standard 6LoWPAN already defines the header compression format for the IP header, IP extension headers, and the UDP header. For example, the header compression mechanisms standardized in RFC6282 can be used, to provide header compression of IPv6 packets over IEEE 802.15.4 based networks.

The reason for making the IoT's IP connected is the need of the applications for wireless internet connectivity at lower data rates for devices with limited constraints. Examples of such applications are: smart cities, smart grids, home automation, e-healthcare, and others.

Since we have a lot of devices that are unable to communicate in an efficient way with constrained resources, the Internet Engineering Task Force (IETF) came up with a lightweight protocol: Constrained Application Protocol (CoAP). CoAP is considered as a replacement of HTTP protocol, to be used as the IoT application layer protocol. This protocol was designed specifically to meet the requirements of the constrained devices such as low overhead, simplicity, and multicast support. Furthermore, as there exist many constrained devices in buildings and vehicles, IPv6 is used for the IoT implementation, since it provides a larger address space to let more devices get connectedto the internet.CoAP was also developed as a candidateprotocol to connect energy–constrained devices to the internet.Table I[1] presents a comparison of the resource consumption between HTTP and CoAP.

TABLE I. RESOURCE CONSUMPTION COMPARISON BETWEEN HTTP AND COAP [1]

Parameters	HTTP	CoAP
Bytes per transmission	1451	154
Power (mw)	1333	151
Lifetime (days)	0.744	84

Although IP networking brings new opportunities and improvements in our daily life, security remains a concern that has to be tackled. Past experiences have proven that designing a right security protocol is a difficult and error-prone process.Thus, when researchers were concerned about security in IoT, they were designing lightweight variants and porting them to constrained devices leading to a situation where security doesn't keep up, although standardized Constrained Application Protocol (CoAP) completely supports the requirements of the application.

Many research works have been done focusing on the security methods for IoT, which can face different security attacks that affects the functionalities and services provided by the IoT network. In this paper, we have analyzed the different available security solutions for the communication between devices. We also proposed a secure CoAP usingElliptic Curve Cryptograph (ECC).Then we compared it with CoAP using Rivest–Shamir–Adleman (RSA) algorithm.

Thearrangement of this paper is as the following: Section II, views the literature review including background about CoAP, ECC algorithm, and RSA. Section III introducesthe reader to the proposed algorithmic design and equations for the proposed secure CoAP. Section IV shows the results that compare the CoAP using RSA and the proposed secure CoAP using ECC. Section V, concludes this paperand shows the future work.

II. LITERATURE RIVEW

Designing a secure and trustworthy network, and providing end-to-end security during communication is probably the most challenging task in IoT. And in IoT, security becomes more important, since most of the exchanged information are in general sensitive and private. A lot of problems can occur by adding security to the network; the most important is the consumed power, since it is directly related to the life time of the network. Recently, a lot of research has been done to investigate this issue, by trying to adapt the traditional security methods and techniques into IoT. In this section, we introduce some background information about CoAP protocol and discuss some of the approaches and solutions provided to secure the constrained environments.

A. Background

Integrating security into CoAP is a very difficult thing to do, since IoT is a heterogeneous network[]. Here we provide brief introduction ofCoAP, its operations and methods, message format and transaction model.

1. CoAP

Recently, the Constrained RESTful Environments (CoRE) which is an IETF working group aims to contribute the

standardization and development of the REST architecture in LLNs. They founded a new web application transfer protocol called Constrained Application Protocol (CoAP) [12]. CoAP applies the same application transfer model of HTTP to LLNs, whilst it maintains a soft design as well as less overhead. Hence, CoAP implements some features of HTTP. As mentioned in [13], one of CoAP goals is designing a special requirements web protocol to be employed in 6LowPAN, considering the constructing automation, energyand otherM2M applications that are usually utilized in Smart Cities.

These applications can be managed using CoAP web protocol, this increases the importance of web services in people's daily lives. However, CoAP is bound to UDP by default which allows it to be best tailored for the IoT applications because the client and server communicate through connectionless datagram [14]. The next are CoAPfeatures which distinguish it from another application layer protocols. These features makeCoAP moreconvenientand interoperableprotocol for constrained networksand mostlyforindustry-field networks:

- **Web protocol that fulfills M2M constrained requirements.** It is a compact and simple protocol specified forconstrained nodes, since it is applicable on 8-16 bit microcontroller nodes, a RAM of 8-12K and a flash of64-256K.
- **Lowsize header overhead.** Since it is optimized for the limited throughput and restricted bandwidth (i.e. maximum of 80 bytes for the payload of the application-layer and order of tens of kbits/s throughput).It also hashighpacket lossratio.
- **Simple caching and proxy abilities.**CoAP supports caching and when theproxy is used, the latest responses can be cached to replay later as the agent of the sleeping nodes.
- **The variety of message exchanging.** It depends on the REST architecture; hence it allows creating, deleting, reading and updating a resource on a node to a device.Furthermore, the common single transaction of CoAP consists of a request as well as a response exchange.
- **UDP binding that supports reliability usingthe multicast and unicast requests.** The packet loss high rate is a requirement in constrained networks. Hence, CoAP has the option of sending a larger amount of data packets using UDP as it supports UDP with a back-off approach for reliability. It also supports multicast requests but without reliability.
- **Content-type and URI support.**Sincethe Internet contains a huge media types, CoAP supports these types.

2. CoAP Methods and Operations

As mentioned before CoAP implements some of the HTTP features. Hence, the standard methods of HTTP are fully supported by CoAP, this includes DELETE, GET, PUT and POST methods. These methods are akin to their HTTP equivalents. These methods have the ability to control and influence the resources. All of them have the same safe (in term of retrieval) as well as idempotent properties since they have a similar effect although if they executed many times. Moreover, the response mentions any unsupported method using the response code of "method not allowed". These methods can be briefly illustrated as follows:

- **GET:** to obtain any resource information that is defined using a URI, this method can be used.
- **POST:** This method is mainly utilized when a novel subordinate resource is needed to be created under an exists resource URI.
- **PUT:** this method is utilized in creating or updating a resource fixed with a predefined URI.
- **DELETE:** is employed in deleting a URI specified resource. On success, a deleted code of 2.02 is returned. On a field, an error message is returned. DELETE is idempotent method but unsafe.

3. CoAP Messages Format

A compact and simple binary header is used by CoAP as mentioned before. This header is immediately followed by options in Type-Length-Value (TLV) format. Any byte comes after both the options and the header is examined as payload. This payload is calculated using the length of the datagrams. Fig.1illustrates the CoAP header format and its fields where Ver is the version of CoAP, T is the type of the transaction, OC is the options count within the message, and Code is the response or the method code. After the header, the options always take place. At the end, the payload is attached.

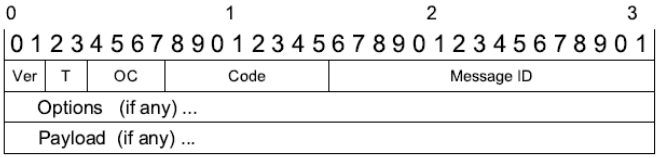


Figure 1. The format of CoAPMessage [15]

CoAP header fields are explained in the following way:

- **Verfor Version:**2-bits with the type of unsigned integer. Specifies the version of CoAP. This field has to be set to 1 by all the implementations of this indication because other values are preserved for the future versions.
- **T for Type:** 2-bits with the type of unsigned integer. Specifies the type of CoAP message. Whereas CON message has the 0 value, Non-CON message has the 1 value, ACK message has the 2 value, and RST message has the 3 value.

- **OC for Option Count:** 4-bits with the type of unsignedinteger. Specifies the options number that comes after the header. The value of 0 means no option is used and payload can directly come after the header. Options format is illustrated in the following figure:

4. CoAP Transaction Model

Similar to HTTP, CoAP has the client/server interactions model as shown in *Fig.2*. Whenever the client is sending a request for anaction on aresource (that is defined as a URI), the server replays with a response message that contains the resource representation and the response code.Nevertheless, the implementation of CoAP mainly plays the roles of the client (end-point) and the server.CoAPfulfills M2Mrequirements by supportingasynchronous transactions over the UDP. This is done using fourinterchangesmessages (CON, Non-Con, ACKand REST).

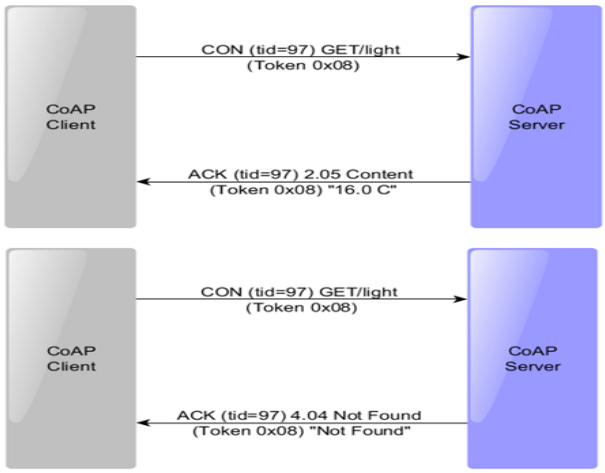


Figure 2. CoAP Transaction Model

B. *Related work*

Many researchers are currently focusing on the IoT security issue;therefore, a lot of security solutions have been provided. In [16], the authors proposed to use the DTLS (Datagram Transport Layer Security) protocol with CoAP to provide end-to-end security in IoT. Since DTLS was designed for the internet not for constrained IoT devices, it is a heavyweight protocol with too long headers to fit in a single IEEE 802.15.4 MTU (Maximum Transmission Unit). Therefore, 6LoWPAN is used in IoT to compress the long IP layer headers, and in [16], they proposed mechanisms to exploit the compression capabilities of 6LoWPAN to compress the DTLS headers and messages. The authors also defined the Record header, Handshake header, ClientHello message, and the ServerHello message and their compression techniques. One of the results showed that for the DTLS record header the number of additional bits can be reduced by 62%. However, their study did not go further to ensure that compression would not affect the security. To continue with this approach, the authors in [23, 24] proposed a security solution based on RSA, the most widely used public cryptography algorithm. Their goal was to achieve high interoperability and low overhead. However, because of the overhead of DTLS handshake process, RSA consumes a large amount of energy.

Therefore, the authors in [17] presented another solution by using DTLS compression as well. The authors proposed Lithe – an integration of DTLS and CoAP for the IoT. Lithe consists of four components: DTLS, CoAP, DTLS header compression (using 6LowPAN), and a CoAP-DTLS integration module which was developed to allow the application to access CoAP automatically. They evaluated Lithe by running it on Contiki OS using real sensor nodes. The evaluation results show significant gains in the processing time, network response time and energy consumption by reducing the packet size. With this work, they were able to avoid fragmentation or decrease the number of fragments by using compression when the payload was slightly above the fragmentation threshold.

In [18], a short outline to ensure a secure IP-based Internet of Things was presented. The authors specifically discussed two issues that need to be solved to secure the communication links: end-to-end security and secure group communication. End-to-end security goal is to achieve a complete secure communication between an HTTP andCoAP entity using the DTLS-PSK protocol, and the 6LoWPAN Border Router, which acts as a proxy. As for group communication security, the goal is to establish a secure connection using DTLS for a group of devices with a single session key. Different available approaches have been identified to achieve this goal, which needs to be analyzed according to the application requirements.

In [19], the authors analyzed two of the known security protocols that can be used to secure CoAP networks: DTLS and IPSec. Their analyses was based on the X.805 standard which with its architecture can provide a complete top-down systematic method to correct, predict, and detect the security vulnerabilities of the network. They highlighted the advantages of the two security protocols, and mentioned their drawbacks as well. And since these protocols were not designed to deal with constrained environments the authors mentioned the reasons why they were not the most optimized solutions for CoAP security, and argued for the need of a new lightweight and secure version of CoAP, which addresses the issues of the above mentioned protocols.

In [20], the authors proposed a lightweight security approach using AES (Advanced Encryption Standard) 128-bit symmetric key algorithm. They came up with Auth-Lite approach which enables the authentication mechanism, and by modifying the CoAP header, they came up with CoAP-Lite which enables lightweight security for CoAP. But this approach can only be used in vehicle tracking systems, and it depends on the application, so for other applications it may or may not be efficient.

In Another work [21], the authors proposed a different solution based on the idea of session security. They used the

LESS (Lightweight Establishment of Secure Session) algorithm a cross layer approach using CoAP and DTLS-PSK channel encryption. In the path of communication, LESS protects the session key during message exchange, enables integrity during session establishment using AES-CCM rather than AES-CBC encryption, and enables separation in key used in reverse paths of communication. The algorithm consists of six steps: pre-sharing secret, session initiation, server challenge, client response and challenge, client authentication and server authentication. The results show that LESS outperforms DTLS-PSK in all aspects. First, the full session was performed in two request/response steps in LESS compared to six in DTLS. And since each step may consist of more than one message in DTLS, physically fragmented datagrams may be transmitted. On the other hand, LESS was implemented as small CoAP messages which guarantee no or little fragmentation. But this solution works only with unicast security, therefore multicast security is still an open challenge [22].

In this work, we propose another security solution using ECC,which is an Asymmetric key based security mechanism based on some algebraic structure.The next section shows the details of the proposed solution.

III. Proposed Algorithmic Design

This section mainly describes the algorithmic design that is proposed in order to secure communications between the objects in IoT networks and from the other hand to reduce consumed energy while these objects are communicating with each other. As mentioned in the literature, there are many methods that secure communications inn IoT networks but each has drawbacks. In order to overcome these limitations, our approach uses Elliptic Curve Cryptography (ECC) method that depends on elliptic curves' algebraic structure over finite fields. The main usage of elliptic curves in this approach is generating private and public keys. Also, they are used in encrypting and decrypting the messages. This algorithm was used in this proposed solution to reduce the needed computational power since the key size in this algorithm has the smallest size comparing with other cryptographic algorithms. In this paper, we used the same Adoptive energy model that was proposed.

- The proposed secure CoAP using ECC:

Algorithm 1

begin

in CoAP protocol if a service agent (SA) node want to send a message to the resource directory (RD)

1) SA sends request for public key (PU) of the RD
   a) SA $\rightarrow RD : PU_{RD}$
2) RD generates public key PU and private key PR with the help of ECC
3) RD sends the public key PU to the SA
   a) RD $\rightarrow SA : PU_{RD}$
4) SA encrypts the message with the received RD public key PU_{RD}
5) SA sends the encrypted message to RD
   a) SA $\rightarrow RD : CT$
6) RD decrypts the message with its private key PR_{RD}

A) ECC algorithm Key generation:

In ECC, two keys are generated; the private key and the public key. Each Service Agent (SA) sender node will encrypt its message with the Resource Directory (RD) public key. Then the RD, in its turn, will decrypt the message with its private key.

Then a random number (R) is selected within a range (n). After that, the next Equation (1) is used to generate the public key:

$$PU = R * P \quad (1)$$

Where R: is a random number between 1 to n-1 as well as P refers to the point on the curve that is generated using the next Equation (2). And PU refers to the public key and n refers to the private key.

$$y^2 = x^3 + ax + b \quad (2)$$

1) Encrypting messages:

Suppose "s" is the message that is sending from the SA. Let "s" has the point S on the curve "E". K is a random point is selected from the range [1-(n-1)] then the cipher text that will be send is C_1 in equation (3) and C_2 in equation (4):

$$C_1 = K * P \quad (3)$$

$$C_2 = S + K * PU \quad (4)$$

2) Decrypting messages:

$$S = C_2 - d * C_1$$

S is the original message that was sent.

- **Logical prof- to get the original message back:**

$$S = C_2 - d * C_1$$
$$C_2 - d * C_1 = (\boldsymbol{S + K * PU}) - \boldsymbol{d} * (\boldsymbol{K * P})$$

Where

$$C_1 = K * P \; and \; C_2 = S + K * PU \;\; from \; 3 \; and \; 4$$
$$= S + K * d * P - d * K * P$$

Now, canceling out d* k* P

$$= S$$

That is the original message.

- CoAP using RSA

B) RSA algorithm key generation

As discussed before, RSA is the most known and popular cryptography algorithm that uses the public key. Three main authors came up with this algorithm in 1976 namely: Ron(R)ivest, Adi (S)Hamir as well as Leonard (A)dleman. Generally, the next Table shows the properties that are employed in RSA.

TABLE II. PROPERTIES THAT ARE EMPLOYED IN RSA.

P and q, prime numbers	Private
r = p.q	Public
Φ(r) = (p-1)(q-1)	Private
PK (encryption key)	Public
SK (decryption key)	Private
X (plaintext)	Private
Y(ciphertext)	public

In RSA, key pair generation can be done through next steps:

- P and q prime numbers must be defined
- r = p.q where q and p can't have the same value because that would make it easy to obtain p from the square root of r.
- Φ(r) = (p-1)(q-1)
- PK, the public key is chosen to be relatively prime with Φ(r) .
- Private key generation using the next equation:

SK.PK = 1 (mod Φ(r)). Hence, SK can be obtained from the next equation

$$SK = \frac{1+m\ \Phi(r)}{PK} \qquad (5)$$

The integer m, through which SK can be obtained.

1) Encrypting messages:

In RSA encryption, Plaintext is structured into block $x_1, x_2, \ldots$ such that each block represents a value in the range from 0 to r-1. After that, each block xi is encrypted to block yi with this equation:

$$yi = x_i^{PK} \, mod \, r \qquad (6)$$

2) Decrypting messages:

Following the next equation, each cipher text block yi is decrypted into block xi.

$$xi = y_i^{SK} \, mod \, r \qquad (7)$$

The next section shows the results obtained from implementing these algorithms with CoAP along with the explanation.

IV. EXPERIMENTAL RESULTS AND EVALUATION

In order to evaluate the proposed algorithm, CONTIKI operating system [25] was employed in this paper experiments for the devices. Where it is one of the most suitable OS for low-power Internet communication. It provides a good environment to run Cooja that is a novel simulator that can be used with ContikiOS. It was used because it provides small as well as large networks of Contiki motes. It also allows them to be simulated. The proposed algorithm and CoAP-RSA algorithm were implemented and tested using the same parameters and the same topology in order to compare their performance using the same conditions. The used simulation parameters to test the two algorithms are listed in the next Table.

A) Simulation Parameters

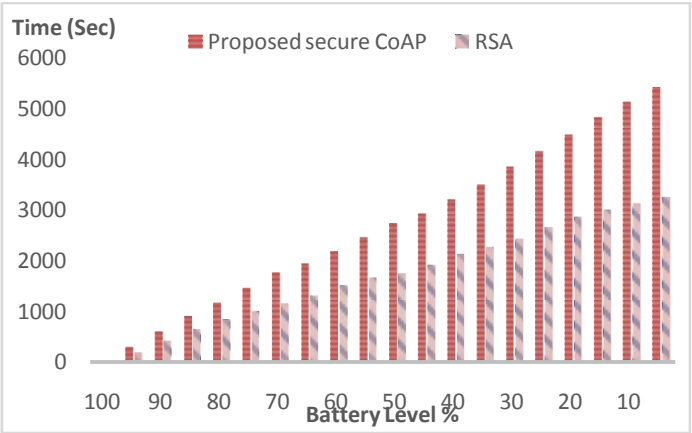
TABLE III. COOJA SIMULATION PARAMETERS

Parameters	Values
Operating System	CONTIKI 3.0
Simulator	COOJA
Nodes Type	Tmote Sky
Physical topologies	1,2,3,4,5 (see section 4.3)
MAC/adaptation layer	ContikiMAC/ 6LowPAN
Routing Protocol	RPL

Radio Environment	Unit Disk Graph Medium (UDGM)
Nodes count	5-320 + RD node
Simulation Duration	Variable
Full Battery	7000 mJ
Transmission Range	50

B) RESULTS

The next figure shows the relation between the battery level and the time in seconds. As shown, the proposed secure CoAP overcome the CoAP using RSA by 47% in terms of saving energy. The lifetime of the network was calculated based on the average of all nodes to sleep where each node lifetime was calculated from the start of sending update messages to the RD until the battery level reaches 5% because as we mentioned in this is the lowest battery level with which the Tmote Sky can convert to the sleeping mode. Furthermore, when using RSA, the energy of the battery consumed faster than the proposed secure CoAP.



C) Explanation

As mentioned before, the main advantage of ECC is its compact key size because the Elliptic curve often utilize a smaller key size rather than the other classical equivalent algorithms such as RSA. The variation in identical key sizes can increase whenever the key sizes increase. The next table IV shows a comparison between the ECC, symmetric algorithms and asymmetric algorithms in terms of security strength similarity.

TABLE IV. KEY SIZES COMPARISON [1]

Symmetric Key Length	Standard asymmetric Key Lenght	ECC Key Lenght
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

As illustrated in the previous table, ECC's key has the smallest sizes that are much less than the other algorithms. Thus, reducing the key size can in turn increase the computational speed and at the same time save more energy. Moreover, fast key generation method is a core advantage in ECC. Hence, the processing speed will be increased. As a result, the residual energy of the nodes will be saved and the network lifetime will increase. Using ECC we can provide authorization, authentication, Integrity and Confidentiality as security services to the IoT networks as follows:

- **Confidentiality:** using this security service any unauthorized node is denied from accessing the data.
- **Authorization**: this security service gives each node a unique key pair (public and private) in order to make encryption and decryption.
- **Integrity**: messages are secured by interfering because of encrypt with the receiver public key.
- **Authentication:** this service is achieved by employing public key, if any malicious/ anonymous node needs to communicate with network nodes then it needs the public key pair of authorized node.

V. CONCLUSION

In this paper a proposed secure and energy efficient method is proposed to be used with resource constrained devices over IoT networks. The proposed method elected the Elliptic Curve Cryptograph (ECC) technique to be the underlying security protocol for authenticated. The reason of choosing this protocol is that this protocol uses smaller key sizes for encryption and decryption which is expected to reduce the power consumption in this type of networks. The proposed method used the CoAP protocol as an application layer protocol and is planning to be implemented using COOJA simulator.

.

REFERENCES

[1] Colitti, W., Steenhaut, K., & De Caro, N. (2011). Integrating wireless sensor networks with the web. Extending the Internet to Low power and Lossy Networks (IP+ SN 2011).

[2] Jiye Park; Namhi Kang, "Lightweight Secure Communication For CoAP-Enabled Internet Of Things Using Delegated DTLS Handshake," Information and Communication Technology Convergence (ICTC),

2014 International Conference on, pp. 28, 33, 22-24 Oct. 2014. doi: 10.1109/ICTC.2014.6983078

[3] Curtis, Bill, "Delivering Security By Design In The Internet Of Things," Test Conference (ITC), 2014 IEEE International, pp. 1, 1, 20-23 Oct. 2014. doi: 10.1109/TEST.2014.7035283

[4] Ludena R, D.A.; Ahrary, A.; Horibe, N.; Won Seok Yang, "IoT-security Approach Analysis for the Novel Nutrition-Based Vegetable Production and Distribution System," Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on, pp. 185, 189, Aug. 31 2014-Sept. 4 2014. doi: 10.1109/IIAI-AAI.2014.47

[5] Porambage, P.; Schmitt, C.; Kumar, P.; Gurtov, A.; Ylianttila, M., "Two-Phase Authentication Protocol For Wireless Sensor Networks In Distributed IoT Applications," Wireless Communications and Networking Conference (WCNC), 2014 IEEE , vol., no., pp.2728,2733, 6-9 April 2014. doi: 10.1109/WCNC.2014.6952860

[6] Addo, I.D.; Ahamed, S.I.; Yau, S.S.; Buduru, A., "A Reference Architecture for Improving Security and Privacy in Internet of Things Applications," Mobile Services (MS), 2014 IEEE International Conference on, pp. 108, 115, June 27 2014-July 2 2014. doi: 10.1109/MobServ.2014.24

[7] Ray, B.; Chowdhury, M.; Abawaiy, J., "PUF-based Secure Checker Protocol for Networked RFID Systems," Open Systems (ICOS), 2014 IEEE Conference on, pp.78,83, 26-28 Oct. 2014. doi: 10.1109/ICOS.2014.7042633

[8] Priller, P.; Aldrian, A.; Ebner, T., "Case Study: From Legacy To Connectivity Migrating Industrial Devices Into The World Of Smart Services," Emerging Technology and Factory Automation (ETFA), 2014 IEEE, pp. 1, 8, 16-19 Sept. 2014. doi: 10.1109/ETFA.2014.7005136

[9] Schukat, M.; Flood, P., "Zero-knowledge Proofs in M2M Communication," Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014). 25th IET, pp. 269, 273, 26-27 June 2013. doi: 10.1049/cp.2014.0697

[10] Van den Abeele, F.; Hoebeke, J.; Moerman, I.; Demeester, P., "Fine-Grained Management Of COAP Interactions With Constrained IoT Devices," Network Operations and Management Symposium (NOMS), 2014 IEEE, pp. 1, 5, 5-9 May 2014. doi: 10.1109/NOMS.2014.6838368

[11] Chaoliang Li; Qin Li; Guojun Wang, "Survey of Integrity Detection Methods in Internet of Things," Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on, pp.906,913, 24-26 Sept. 2014. doi: 10.1109/TrustCom.2014.120

[12] IEEE 802.15.4 [Internet]. Ieee802.org. 2016 [cited 30 December 2016]. Available from: http://www.ieee802.org/15/pub/TG4.html

[13] Kushalnagar N, Montenegro G, Schumacher C. IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals. 2007.

[14] Kerasiotis F, Prayati A, Antonopoulos C, Koulamas C, Papadopoulos G. Battery lifetime prediction model for a wsn platform. InSensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on 2010. 525-530.

[15] Cao Z, Gomez C, Kovatsch M, Tian H, He X. Energy Efficient Implementation of IETF Constrained Protocol Suite. IETF Internet Draft, draft-ietf-lwig-energy-efficient-00; 2014.

[16] Raza, S., Trabalza, D., & Voigt, T. (2012, May). 6LoWPAN compressed DTLS for CoAP. In *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on* (pp. 287-289). IEEE.

[17] Raza, S., Shafagh, H., Hewage, K., Hummen, R., & Voigt, T. (2013). Lithe: Lightweight secure CoAP for the internet of things. *IEEE Sensors Journal*, *13*(10), 3711-3720.

[18] Brachmann, M., Garcia-Morchon, O., & Kirsche, M. (2011). Security for practical coap applications: Issues and solution approaches. *GI/ITG KuVS Fachgesprch Sensornetze (FGSN). Universitt Stuttgart.*

[19] Alghamdi, T. A., Lasebae, A., & Aiash, M. (2013, November). Security analysis of the constrained application protocol in the Internet of Things. In Future Generation Communication Technology (FGCT), 2013 second international conference on (pp. 163-168). IEEE.

[20] Ukil, A., Bandyopadhyay, S., Bhattacharyya, A., Pal, A., & Bose, T. (2014). Lightweight security scheme for IoT applications using CoAP. *International Journal of Pervasive Computing and Communications*, *10*(4), 372-392.

[21] Bhattacharyya, A., Bose, T., Bandyopadhyay, S., Ukil, A., & Pal, A. (2015, March). LESS: Lightweight Establishment of Secure Session: A Cross-Layer Approach Using CoAP and DTLS-PSK Channel Encryption. In *Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on* (pp. 682-687). IEEE.

[22] Rahman, A., & Dijk, E. (2012). Group communication for coap. *draft-ietf-core-groupcomm-00 (work in progress)*.

[23] Kothmayr, T. (2011). A security architecture for wireless sensor networks based on DTLS. Master's Thesis in the Software Engineering Elite Graduate Program at the University of Augsburg.

[24] Kothmayr, T., Schmitt, C., Hu, W., Brünig, M., & Carle, G. (2013). DTLS based security and two-way authentication for the Internet of Things. Ad Hoc Networks, 11(8), 2710-2723.

[25] A. Dunkels, "The contiki operating system "،*Web page. Visited 10 augest*.2017 ،