An Energy Efficient ECC based Light-weight Authentication Mechanism for IoT

Shamini Emerson*, Ki-Hyung Kim*, Kangseok Kim*

Abstract

With the development of IoT (Internet of Things), the use of various devices connected to the Internet and Internet services related to IoT have grown rapidly, but the security for IoT is still vulnerable. Therefore, it is essential to prevent things from being compromised. As a solution for the challenges, we propose a security enhanced integrated approach of a lightweight mutual authentication and access control mechanism. The mutual authentication is based on ECC (Elliptic Curve Cryptography) algorithm because the authentication scheme must meet the requirements of constrained IoT devices. Also, for the access control mechanism, we used a capability based access control mechanism which has fewer burdens for the device to store user credentials. The proposed work is analyzed against various attacks and is compared with other works in terms of computational time for performance evaluation.

*This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2015R1D1A1A01060236).

* First Author : Ajou University Department of Computer Engineering, shamini.sv@gmail.com

* Second Author : Ajou University Department of Cyber Security, kkim86@ajou.ac.kr

* Corresponding Author : Ajou University Department of Cyber Security, kangskim@ajou.ac.kr
I. INTRODUCTION

Internet of things (IoT) is a collection of many interconnected objects, services, humans, and devices that can communicate with each other and share data and information to achieve a common goal in different areas and applications [1]. Recently, IoT has been enabled by the latest technologies in smart sensors, communication technologies, and Internet protocols.

The main purpose is to collaborate IoT devices without the intervention of humans. Moreover, IoT has been implemented in a variety of areas like transportation, agriculture, healthcare, energy production, and so on. Gartner report [2] describes the recent emerging technologies and its Hype Cycle. A variety of IoT devices and their technologies equally increase a wide variety of security issues. Due to the inherent vulnerabilities of Internet before the IoT is widely deployed, security and privacy issues should be considered and addressed [3]. Fremantle and Scott [4] describe the main challenges in IoT world under three aspects such as hardware/device, network and cloud/server. Devices in constrained environments need more interoperable authentication and access control mechanisms to ensure the security of IoT devices. An authentication is a method to verify the identity of devices mutually. An access control is to allow an authenticated user to be able to access the right data which the user has privileges to access. By this way, the devices can be secured from unauthenticated and unauthorized users.

Elliptic Curve Cryptography (ECC) [5] is a public key technique based on elliptic curves over finite fields [6] which can reduce the issues of symmetric key based schemes such as key distribution, key storage and scalability. Moreover, ECC based schemes show better performance as compared to traditional public key based schemes because ECC schemes have the advantages of low computation overhead and small key size. Traditional systems such as RSA yield a level of security with 1024 bit key while ECC achieves the equal level of security with 164 bit key [7]. Therefore, ECC is becoming widely used for mobile applications because it helps to establish equivalent security with lower computing power, battery resource usage, memory requirement and processing.

In this paper, we have proposed a mutual authentication mechanism between user and device (or node) based on ECC. Even though users and devices are mutually authenticated and aware of each other’s identity, an access control mechanism is necessary to allow the authenticated user to access the data which the user has privileges to access. Access control also enables the management of traffic and provides secure access to and from the network. Access control is mostly based on access control lists or capabilities. In this paper, we have proposed the access control based on capability which is a permission (token) to access devices.

The rest of the paper is structured as follows. Section 2 describes the related works regarding authentication and access control in IoT field with security as main key point. Section 3 presents our proposed mutual authentication and access control mechanism. Section 4 gives the security analysis of our proposed scheme. Section 5 gives the performance evaluation in terms of comparison with other existing solutions. Finally we conclude with a brief discussion of future work in Section 6.

II. RELATED WORK

Recently various works have been done with regard to authentication and access control mechanisms in order to secure the constrained devices meanwhile to meet up the requirements of IoT.

Ramos, et al. [8] proposed a distributed capability-based access control for Internet of things. It provides a distributed approach using a capability token, in which an end-device carries out the authorization process. It decides whether to allow the user or not. But authentication between the user and end-device is not provided. Instead of authentication, the author says that an issuer (a trusted third party on behalf of the sensor node) has to be available
for issuing the capability token to the user before she can access resources from the device. The issuer issues a capability token when the user is identified as a legitimate one. But how this authentication process takes place between the issuer and the user is not provided. Thus, secure process between the communication of user and issuer is not provided. 

Mahalle, et al. [9] provided an efficient and scalable ECC based authentication and access control protocol between two devices. ECDH (Elliptic Curve Diffie-Hellman) is used for the key generation phase among devices. An identity establishment phase shows one-way and mutual authentication by performing hash function, encryption and generation of MAC value. In order to perform all these functions, the computational time taken by the mentioned protocol increases. Ye, et al. [10] proposed authentication mechanism by establishing session key based on ECC. The paper provides an authentication method between the devices (user and sensor nodes) but does not provide the relation between the authentication mechanism and such access control policy as mentioned in our paper. Moreover, how the proposed authentication mechanism gives an effect towards the access control to the IoT devices are not provided. It describes attributes based access control. In case of attribute based access control, it has to depend on other entities to make decisions instead of the end-device.

Ndhianje, et al. [11] suggested the use of home registration authority where all the users are registered through the mentioned registration phase. Only authenticated entities among IoT can access to get the service requested. But this system needs third party or an availability of a central entity is needed to authenticate and authorize the user to access the network. In the paper, a gateway as Registration Authority verifies the certificate contents and the identity of things or devices and reviews the contents in order to determine whether the information accurately describes the user. Authentication processes go through two phases: registration phase and authentication phase. Mainly in the work, the user does not interact with devices, instead user interacts only with the gateway which shows the lack of end-to-end security. Also, the access control mechanism is not mentioned in the paper.

Most of recent proposals have addressed the problem of access control using centralized approaches where a central entity or gateway is responsible for managing the corresponding authorization mechanisms, allowing or denying requests from external entities. Although in these approaches, traditional security mechanisms and access control models such as Role Based Access Control (RBAC) [12] or Attribute Based Access Control (ABAC) [13][14] can be used, but the end-to-end security between devices and any Internet hosts cannot be achieved. Works [15-19] have addressed RBAC and ABAC mechanism using a central entity or gateway which carries out authorization mechanisms and decides whether or not the requester is allowed to access system resources. In this process, an end-to-end security cannot be achieved. Also, the traditional access control methods such as RBAC and ABAC do not fulfill the IoT requirements because they lack flexibility, usability and scalability in environments with several million or billions of devices. We proposed a solution of mutual authentication between device and user without the requirement of a third- party issuer. In case of access control mechanism, devices can make the authorization decision by themselves. Therefore, we provide an approach which integrates mutual authentication and capability based access control method. Mutual authentication done in our proposed approach uses the ECC scheme which fits the requirements of IoT such as scalability, flexibility and interoperability.

III. PROPOSED METHOD

Our research focuses on providing an efficient mutual authentication and access control mechanism based on capabilities for IoTs. The algorithm presented in this paper addresses both mutual authentication and access control. The proposed method is composed of two phases; first phase is a
key establishment and mutual authentication procedures, and then an access control phase follows. The key establishment and mutual authentication flow are shown in Fig. 1 and Fig. 2 respectively. The notations and descriptions are given in Table 1.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_u, K_n$</td>
<td>private keys of user and node</td>
</tr>
<tr>
<td>$Q_u, Q_n$</td>
<td>public keys of user and node</td>
</tr>
<tr>
<td>SSK</td>
<td>shared secret key between user and node</td>
</tr>
<tr>
<td>$P$</td>
<td>base point</td>
</tr>
<tr>
<td>$Q$</td>
<td>order of elliptic curve group</td>
</tr>
<tr>
<td>$h()$</td>
<td>one-way hash function</td>
</tr>
<tr>
<td>$</td>
<td></td>
</tr>
<tr>
<td>ID$_u$, ID$_n$</td>
<td>IDs of user and node</td>
</tr>
<tr>
<td>ACR</td>
<td>Access Control Rights</td>
</tr>
</tbody>
</table>

1. Key Establishment Phase

The key establishment is mainly based on the Elliptic Curve Cryptography (ECC). Initially, Key Distribution Center (KDC), a trusted third party, performs the basic operations like the generation of base point $P$. The KDC generates public keys and private keys for user and device node based on $P$. Also, a random number generation and an ECC multiplication are done by KDC. KDC generates a random number which is the private keys for user and device. KDC does an ECC multiplication to obtain public keys. We assume that the public keys are exchanged between user and device through a secure channel such as SSL or TLS beforehand. A transport encryption method such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS) secures the data while a communication takes place between two entities.

KDC generates random number $K_u \in GF(P)$, and $K_n \in GF(P)$ as private keys for user and device respectively, corresponding to that, it generates public keys $(Q_u, Q_n)$ using the private keys where $GF(P)$ is Galois Fields which is one of finite fields.

$$K_u \in GF(P) \rightarrow Q_u = K_u \times P \quad (1)$$

$$K_n \in GF(P) \rightarrow Q_n = K_n \times P \quad (2)$$

![Fig. 1. Key Establishment Phase](image)

This paper considers sensor node as a device. Once the private keys and public keys are generated for users and devices, public keys get exchanged in a secure communication channel. User and device compute the shared secret key (SSK) $SSK_u = K_u \times Q_n$ for user’s shared secret key and $SSK_n = K_n \times Q_u$ for node’s (or device’s) shared secret key. At this stage of establishing the shared secret key, no other parameters are disclosed except the public keys. Then, any third party, who is not aware of the private credentials of each device, will not be able to calculate the shared secret key from public information available.

![Fig. 2. Mutual Authentication Phase](image)
2. Mutual Authentication Phase
After the initialization phase of generating the keys, both user and device have the shared secret key (SSKu for user and SSKn for device or node) where SSKu is equal to SSKn. A user sends a request message to access resources from the device. The request message is represented as

\[ M_i = \{X, ID_u, T_u \parallel ACR\} \quad \text{--- (3)} \]

Where \( X = h(ID_u \parallel ID_d \parallel T_u \parallel SSK_n) \) is the hashed value of user’s credentials, device credentials and shared secret key. The \( T_u \) is the timestamp generated by the user. The ACR is the access control rights. It will be well explained in capability based access control method in the next section.

When a node \( n \) receives a request message, it generates its timestamp \( T_n \) and checks the validity of \( T_u \) using \( (T_n - T_u) \leq \Delta T \), where \( \Delta T \) is the expected time interval for transmission delay. Then the node calculates a hash value and verifies that the hash value is equal to \( X = h(ID_u \parallel ID_n \parallel T_n \parallel SSK_n) \) and also checks whether the shared secret key is the same or not as its own shared secret key. If all the conditions regarding the user match, we can say that the user is authenticated with the node. The node sends a response message to the user as \( M_2 = \{Y, ID_n, T_n\} \), where \( Y \) is a hashed value of device’s credentials, user credentials and shared secret key. That is, \( Y = h(ID_u \parallel ID_n \parallel T_n \parallel SSK_n) \). Once the user receives the message, it verifies the timestamp values, \( (T_n - T_u) \leq \Delta T \) and also checks the hash value obtained. Finally, it verifies that the shared secret key is equal. If a match is found, the device is also authenticated to the user. After the user and device are mutually authenticated, the access to device will be granted according to the ACR (Access Control Rights).

3. Capability based Access Control Method

Access control models can be classified into two classes, based on capabilities and access control lists. In this paper the access control mechanism is based on the capability based access control. In the capability based access control, capability is a token which gives a permission to access devices or not. The capability based access control has the following advantage such as the Principle of Least Privilege which supports a more fine grained access control. In [20], if a user wishes to access a resource from a device, the user sends this token together with the request. Thus, the entity or device that receives the capability already knows the right level (i.e. permissions) that the requester has been granted to process. This simplifies the authorization mechanism with resources-constrained devices since complex access control policies are not required.

The information carried by the token includes user credentials, device credentials and operations to be performed. The capability token can carry more parameters for more security but we prefer to use only few important parameters. More number of parameters may decrease the efficiency of the process. The capability token is represented as follows:

\[ \text{ACR} = \{\text{"fr"}, \text{"to"}, \text{"op"}\} \quad \text{where,} \]

"fr" : ID of a user or requester
"to" : address of a target resource
"op" : operations (CREATE, RETRIEVE, UPDATE, DELETE) to be performed

Devices also have their security attributes such as access control rights and other information. Once a device receives a request for accessing resources, it compares user’s credentials with the capability stored in the device. Therefore, if the capability carried by the user matches with the information stored in the device, then the device grants the access to the resources. This method reduces the number of capabilities stored in the device to offer more scalability.
IV. SECURITY ANALYSIS

In this section, we analyse the security of our proposed method.

1. Mutual Authentication
The proposed scheme provides mutual authentication between user and node. A secret key is securely shared between user and node. When the node verifies a message \( M_1 = \{ X, ID_u, T_u \| ACR \} \) from user and the user verifies a response message \( M_2 = \{ Y, ID_u, T_n \} \) from the node, a secure mutual authentication is achieved. Then both user and node can make sure that they are legitimate ones. Through this mutual authentication, a trusted relationship between user and node can be established.

2. Replay Attack
The proposed scheme is resistant to replay attacks since the messages \( M_1 = \{ X, ID_u, T_u \| ACR \} \) and \( M_2 = \{ Y, ID_u, T_n \} \) are time-stamped. Suppose an attacker intercepts the message and tries to send the request again. Then the device will verify the timestamp value \( (T_n - T_u) \leq \Delta T \). If the timestamp sent by attackers is older than the predefined threshold then the authentication process will stop because the timestamp was used for the previous authentication.

3. Man-in-the-Middle Attack
After an attacker modifies a request message \( M_1 = \{ X, ID_u, T_u \| ACR \} \) if she sends the modified message \( M_1 = \{ X', ID_u', T_u' \| ACR' \} \) to the device for authentication, then the device checks its shared secret key. If it is not same as the key shared with device, then the device will not authenticate the attacker. By this way, Man-in-the-Middle attack can be avoided in the proposed mechanism.

4. DoS (Denial of Service) Attack
When a device receives a request message \( M_1 \) from user, it first validates the timestamp \( T_u \). Therefore the device will stop the authentication process. Even if the attacker tries to access a single resource simultaneously using different IDs, it is easy to control accesses using one ID because the system is able to maintain the session. Thus, the access of the same ID to the same resource can be restricted to only one session at a time. Therefore, DoS attack can be prevented.

V. PERFORMANCE EVALUATION

In this section, we show that the proposed scheme is light-weight and fits for IoT constrained devices. Then we show that the proposed authentication scheme is energy efficient. The test setup for the performance evaluation consists of mica2 motes, user device, and KDC. Assuming that devices with Mica2 motes act as a node. Kerberos Key Distribution Center (KDC) is a network service that accepts requests for tickets from clients, validates their identity, and grants tickets to them. When a user tries to access the nodes, it has to come across three phases such as key establishment phase, mutual authentication phase and capability based access control phase as mentioned in the proposed work. Then the performance evaluation mainly focuses on mutual authentication because it is one of the most important processes during the authentication phase. First, we show the performance analysis on computation time for authentication phase. Then, we show the proposed authentication scheme is energy-efficient. Performance analysis is done by analyzing computational time using different parameters such as one way hash function \( h() \), random number generation \( R \), ECC multiplication, encryption, XOR function, and message authentication code (MAC) function in comparison with other related works. According to Mahalle, et al. [9], which shows time taken by Mica2 motes to process functions mentioned above, Table 2 shows the computational time taken by Mica2 motes to perform each function. SHA-1 as one way hash function takes 3.63 milliseconds on Mica2 motes. The time taken by Mica2 motes to
generate random number is 0.44 milliseconds. For generating MAC value and ECC by performing point multiplication, they take 3.12 milliseconds and 800 milliseconds respectively. The time taken by Mica2 motes to encrypt and decrypt by RC5 is 0.26 milliseconds. Table 3 shows the comparison among protocols mentioned above. Security level of the proposed work depends mainly on hash function and shared secret key. We have calculated the total time of our proposed mechanism using the time taken by Mica2 motes given in [9]. We have not taken ECC point multiplication into account because it takes place in KDC. KDC is responsible for generating keys and distributing them to users and devices. Moreover, KDC is a powerful device, so computational overhead is trivial as compared to nodes. However, Mahalle, et al. [9] performs two times hash function and encryption respectively, and sends messages as a MAC value between the two devices, thus it takes around 14.02 milliseconds. Therefore, the total time taken by our scheme is 7.26 milliseconds which is less as compared to other protocols. This shows that our scheme is light-weight and fits for IoT constrained devices. The performance delay of the compared protocols with our proposed scheme is shown in Fig. 3. The protocol described in [10] shows the highest delay to perform the authentication phase, and our proposed scheme gives much less delay performance as compared to other protocols.

Table 2. Computational Time Taken on Mica2 Motes

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Time in milliseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH</td>
<td>Time to perform one-way hash function</td>
<td>3.63</td>
</tr>
<tr>
<td>TR</td>
<td>Time to generate a random number</td>
<td>0.44</td>
</tr>
<tr>
<td>TMAC</td>
<td>Time to generate MAC value</td>
<td>3.12</td>
</tr>
<tr>
<td>TMUL</td>
<td>Time to perform ECC point multiplication</td>
<td>800</td>
</tr>
<tr>
<td>TRC5</td>
<td>Time to encrypt or decrypt by RC5</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Table 3. Comparison of Computational Time for Authentication Protocols

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash function (H)</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Random number (R)</td>
<td>N/A</td>
<td>2</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>ECC multiplication (MUL)</td>
<td>N/A</td>
<td>2</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>XOR</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Encryption (RC5)</td>
<td>2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MAC</td>
<td>2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Total time in milliseconds</td>
<td>14.02</td>
<td>1615.4</td>
<td>8.7</td>
<td>7.26</td>
</tr>
</tbody>
</table>

Fig. 3. Performance Delay for Authentication Protocols

Fig. 4. Comparison of Energy Consumption for Authentication Protocols

Next, we compute the energy consumption of security computations as mentioned in [21]. For
Mica2 mote, when the processor is in active mode I = 8 mA [22], it is assumed that two AA batteries are used whose voltage \( v = 3.0V \) [22]. Therefore, the total energy consumption of our protocol can be calculated according to the eq. (1) mentioned below.

\[
E = v \times q
\]

where \( q = i \times t \), \( \therefore E = v \times i \times t \)

In the above eq. (4) and eq. (5), the “\( E \)” represents the energy consumption, the “\( q \)” denotes the charge, the “\( v \)” represents the voltage, the “\( i \)” is the current and the “\( t \)” is the elapsed time. Therefore, the total energy consumption by our scheme according to the eq. (4) is 0.17mJ which is less as compared to the other protocols mentioned in which IECAC consumes energy around 0.34mJ [11] takes around 0.20mJ and [10] takes around 38.76mJ where more amount of energy consumption takes place. Thus, we can say that our authentication scheme is more efficient. The energy consumption between the protocols is shown in Fig. 4.

VI. CONCLUSION

To overcome challenges for security and privacy issues in Internet of Things environment, we need lightweight, distributed and attack resistant authentication and access control methods. As a solution for the aforementioned challenges, we have provided a lightweight mutual authentication and access control mechanism. The mutual authentication is based on ECC algorithm because the authentication scheme has to meet the requirements of IoT devices. Also, we have proposed a capability based access control mechanism which has fewer burdens for the device to store more credentials regarding user. Moreover, the device itself involves both authentication and authorization without interception of any other entities. We have also analysed our work against a variety of attacks and have shown that our proposed mechanism is attack resistant. Furthermore, a performance evaluation is provided by comparing the proposed work with other related works in terms of computational time.

Finally, as a future work, we plan to develop more efficient access control mechanism based on capability and to implement the mechanism in real time applications like smart home control systems. Also we will consider to apply our proposed mechanism to IoT integrated collaboration and access system which we are currently developing. The development focuses on integrating data from a variety of sources and reflecting them to collaboration service. The data integration (mashup) is needed to integrate collaboration data and activities, large-scale web contents and available sensing information from multiple IoT sources to create a new service. Therefore, we are doing research on the issues like consistency, data security, and personal privacy preservation in developing universal mashup of data generated from IoT integrated collaboration. In future work, we will consider more security issues raised in IoT network with our developing collaboration system.

References


샤 미 니 (Shamini Emerson)

2007년 5월 : Department of Electronics and Communication Engineering, Jeeppiaar Engineering College, India
2016년 8월 : 아주대학교 컴퓨터 공학과 석사
2016년 11월~현재 : Embedded QA Engineer, Aprecomm Pvt.Ltd, Bengaluru, India

<관심분야> 사물인터넷, 사물인터넷 보안

김 기 형 (Ki-Hyung Kim)

1996년 2월 : 한국과학기술원 전자 공학과 박사
1997년~2005년 : 영남대학교 컴퓨터공학과 부교수
2005년 3월~현재 : 아주대학교 사이버보안학과 교수

<관심분야> 사물인터넷, 네트워크 보안, 임베디드 소프트웨어

김 강 석 (Kangseok Kim)

2007년 11월 : 미국 인디애나대학교 컴퓨터공학과 박사
2010년 9월~2016년 2월 : 아주대학교 대학원 지식정보 공학과 연구교수
2016년 3월~현재 : 아주대학교 사이버보안학과 부교수

<관심분야> 클라우드 컴퓨팅, 사물인터넷, 빅데이터, 정보보안