International Journal of Security and Its Applications
Vol. 10, No. 11 (2016), pp.79-88
http://dx.doi.org/10.14257/ijsia.2016.10.11.07

An User Authentication Scheme Based on the ECC and OpenID Techniques in the Internet of Things

Jong J. Lee and Ki Young Lee*

Depatment of Infomation and Telecommunication Enginering,
Incheon National University,
Incheon, 22012, Korea
{ljj21089, kylee} @inu.ac.kr

Abstract

Authentication is a communication protocol processing procedure. In the Internet of Things, secure communication should be constructed between one "thing" and another by such a procedure. The identity that the second "thing" or object claims should be consistent with what the first one claims. Claimed identity information becomes a single message. Based on this message, we verify the identity of the "things". The purpose for both communication partners to implement authentication protocol is to have solid communication in the high layer (e.g., application layer). In order to do that, usually the authentication protocol has several sub-tasks such as identification key establishment, or key switching and consultation. In an authentication process, identity of the claimer can be acquired through message identification. In authenticated key establishment protocol, key establishment materials are also important protocol messages, which is part of entity authentication. In this paper, we focus on simple and efficient secure key establishment based on ECC (Elliptic Curve Cryptosystem). And we proposed ECC and OpenID based user authentication scheme. Our analysis shows that our approach can prevent attacks like eavesdropping, the man-in-the middle, key control attack, and replay attacks

Keywords*: Security, Authentication, ECC, Open_ID*

1. Introduction

Nowadays, through the communication between various smart devices including a smart phone, it may be provided to the user after be generated a secondary data. Because a series of information can be gathered, processed, handled and controlled. In these environments, it may be exposed to the attack by sending the information to users which was not justified. Therefore, execution process of authentication for the user is required. However, due to constrained environment such as a low-power, ultra-small objects in the Internet of Things, there are omitted case for necessary authentication phases and process. Accordingly, security damage incidents and accidents are increasing, due to the exposure of the transmitted information to device that it does not authentication and authorization though secure authentication phases. At this time, man-in-the-middle attacks such as an information gathering, imitation, blocking and an invasion of privacy can occur.

In order to solve these security vulnerabilities and problems, various user authentication methods are proposed in earlier studies. In earlier user authentication and identification technologies, there are divided such as ID-based, certification-based and SIM-based methods. And first, ID-based as a traditional authentication method can be lightweight and fast operation, however, there are problems for a relatively low safety and key management [1]. The certification-based method has the problem of the certification management, because it is how to authenticate using by issued certification. Finally, the

* Ki Young Lee is the corresponding author.

ISSN: 1738-9976 IJSIA

SIM-based method is a how to perform the authentication by storing and managing authentication information in file system of a SIM card, thus, it is physically strong. However, it is necessary such as a separate software, a how to manage. Likewise, there is still problem on the aspect of safety and efficiency. In this paper, we analyze the problems and limitations on applying earlier user authentication methods in Internet of Things. And we also propose the method and architecture for user authentication in the Internet of Things. We also propose the hybrid authentication method to apply using both OpenID-based scheme and public key based algorithms.

2. Related Works

In this section, we introduce the basic concepts of ECC and OpenID.

2.1. Elliptic Curve Cryptosystem

Elliptic curve cryptography (ECC) can be categorized as public cryptography with its many advantages over the other public cryptography. After it was first proposed by Koblitz [2] and Miller [3] independently in the nearly same year, it has been extensively studied and implemented by mathematicians, cryptographers and computer scientists over the world. Till now, the best algorithm needs full exponential time to solve the underlying mathematical problem of ECC, which is referred to as the elliptic curve discrete logarithm problem (ECDLP). Contrasted with ECC, there are sub-exponential-time algorithms to tackle the integer factorization and the discrete logarithm problems on which RSA and DSA is relied on, respectively. It is believed that ECC with the key length of 162 bits is at the same secure level as RSA with the key length of 1024 bits.

Table 1. Comparison of Length of Keys for RSA and ECC [4]

Security Level	RSA key length	ECC key length
80	1024	160-223
112	2048	224-225
128	3072	256-283
192	7680	384-511
256	15360	512-571

ECC offers a security level equivalent to RSA while using a far smaller key size; therefore it leads to the better performance in limited environments like cellular phones, PDA, sensor networking, etc. The standard organizations such as IEEE, NIST, IETF and ISO have accepted ECC as an alternative and efficient public key cryptosystem. It can provide various security services such as key exchange, privacy through encryption, and sender authentication and message integrity through digital signature.

Typical elliptic curve can be defined in finite field GF(p) in equation (1), where p is a large prime.

$$y^2 = x^3 + ax + b \tag{1}$$

Figure 1 shows the examples of elliptic curve. As elliptic curve cryptosystem on GF(p) is safer than counterpart in GF(2^m), we only discuss the curve defined on GF(p) where $\Delta = 4a^3 + 27b^2 \neq 0$. $\Delta \neq 0$ means that there is only one tangent line at every point on the elliptic curve. If a pair (x, y) meets the equation (1), it is called one point on the curve.

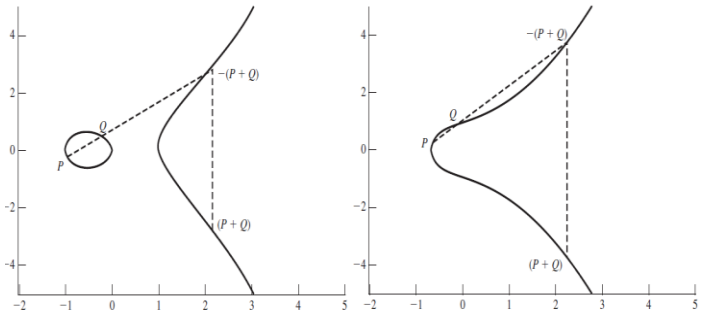

Figure 1. Example of Elliptic Curves ($y^2 = x^3 - x$, $y^2 = x^3 + x + 1$)

The addition of two points on the elliptic curve is very simple. Suppose P and Q are on the elliptic curve, where $P \neq Q$ and $P \neq -Q$, we draw a line which goes through these two points firstly. As the x' s order in the curve is 3, the line will have the third intersection point N with the curve. Then we draw a line which parallels to y axis through Point N. Thus, this line will have the second point R on the curve; and the Point R is the point we want to compute. (R = P+Q)

If P = -Q, the line through P and Q will parallel to y axis; then we consider that the result is infinite point Θ ($\Theta = P + Q$). If P = Q, we will draw a tangent line at P, and this line will have the third point N on the curve because the x's order of the curve is 3. Then we draw a line which parallels to y axis through N point. This line will have the second point R on the curve, which is the point we want to compute. ($R = 2P$)

With these three operations, we can compute kP, where k is a large integer. First of all, k can be represented into the binary form. Then we can use point operations to compute kP. National Institute of Standards and Technology (NIST) recommends four random elliptic curves which can be used in real elliptic curve cryptosystem. To improve the computing efficiency of ECC, it is recommended that the coefficient a is -3.

Now we try to find an integer d so that Q = dP, where P and Q are two points on the elliptic curve. We can compute dP easily if we know d and P, but it is a difficult problem to find out d if we only know P and Q. It is called elliptic curve discrete logarithm problem (ECDLP).

Parameters for elliptic curve cryptosystem include F, a, b, P, n and h. F is the finite field and a and b are the coefficients of the elliptic curve. P is the base point and n is the order of P and a large prime number. H is defined as h = #E(K)/n where #E(K) is the number of the points on the curve.

Suppose that Bob wants to send a message M to Alice. First, Alice chooses an elliptic curve, private key d, and public key Q where Q = dP. Alice distributes her public key and the parameters on an authenticable channel. Bob gets all of these parameters on that channel. The encryption process is summarized as follows;

(1) Bob represents M as an element m in GF(p).
(2) Bob selects a random number $k \in$[1, n-1].
(3) Bob computes $P_1 = (x_1, y_1) = k$P.
(4) Bob computes $P_2 = (x_2, y_2) = k$Q. If x_2 =0, go to step (2).
(5) Bob computes c = m * c = m * x_2.
(6) Bob sends (P_1, c) to Alice.

Then Alice gets the ciphertext from Bob. The decryption process is summarized as follows;

(1) Alice computes the point $P_n = dP_1 = (x_2, y_2)$, then she gets x_2.
(2) Alice computes m = c * x_2^{-1}, then she gets the message M.

We suppose that Alice wants to send a message M to Bob. Bob can verify whether Alice really sends this message with Alice's public key. The signature process is summarized as follows;

(1) Alice represents M as a bit string.
(2) Hash function is used to compute the hash value of m: e = H(M).
(3) A random integer k is randomly selected: $k \in [1, n-1]$.
(4) Alice computes the point $(x_1, y_1) = kP$, and let $r = x_1 \pmod n$. If $r = 0$, go to step (3).
(5) Alice computes $s = k^{-1}(e + rd) \pmod n$. If s = 0, go to step (3).
(6) Alice sends the message M and the signature (r, s) to Bob.

After Bob receives the message M and Alice's signature (r, s), he can verify the signature as follows;

(1) Bob gets Alice's public ECC key.
(2) Bob computes $(x_1, y_1) = sP + rQ$.
(3) Bob computes hash value e = H(M).
(4) Bob computes $r' = x_1 + e$.

The message is really from Alice if $r = r'$, otherwise it is not.

2.2. OpenID

The OpenID mechanism is a decentralized authentication scheme for the SSO mechanism [5]. OpenID users can choose a trustworthy OpenID server to register their OpenID. They are identified by a URL like: *http://yourname.openidserver.com*. In the OpenID mechanism, three parties are involved: the OpenID provider (OP), the service provider which is also called Relying Party (RP) and the user. We assume that the OP and the RP trust each other in advance, OP has a trusted list of RPs. In the OpenID mechanism, users only need to have a pair of identity and password. The typical communication flow of this mechanism is described in Figure 2.

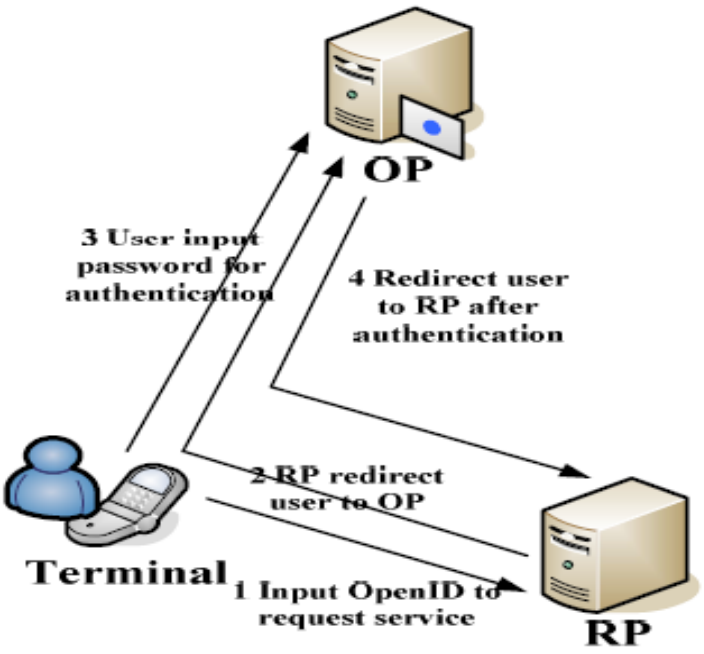

Figure 2. OpenID authentication flow

1) Smart terminal user input OpenID and submit it to the RP;
2) RP normalize the user's OpenID and identify the OP, then RP redirect OP to the smart terminal;
3) User input the corresponding password;
4) OP authenticates the user by OpenID and password pair;
5) If the authentication is successful, the RP page will be redirected to the user

When user submit his/her OpenID like *http://myname.openidserver.com* to the RP, RP will parse the URL and get two things: one is the OP address "*openidserver.com*"; another is the user's identity "*myname*". The RP associate with OP using redirection according to the OP's address and ask OP to authenticate this user's identity. Then OP show user the password login screen and get the password, after authentication, the service page will be redirected to the user. In this way, the RP don't know the user's password, and RP is trusted by OP ahead of this flow. So, users can use one pair of OpenID and password to login onto many service website.

3. Proposed Scheme

In this section, the proposed architecture and user authentication protocol are explained.

3.1. Architecture

Based on what we have learned from current literatures of Internet of Things, we may reasonably draw an abstract architecture for it (as shown in Figure 3). 'Things' or objects become end nodes in the Internet environment. They have unique global addresses (e.g., IPv6 address) and are capable of communicating with each other over the Internet. In order to organize and manage massive resources, every object will pre-register on a nearby trustworthy access point or gateway (denoted as Registration Authority, or RA). This assumption has another advantage that the RA can expend computing and storage capacity of the 'Things' or objects for authentication purpose. Meanwhile, RA is also able to maintain a history record of all access requests for auditing purpose.

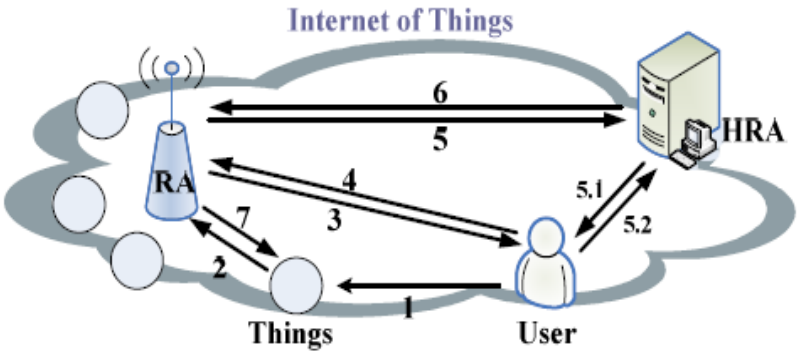

Figure 3. The Architecture of Proposed Scheme

3.2 Authentication Protocol

As shown in Figure 3, a complete request procedure for accessing a 'Thing' involves following seven steps;

- Step 1: User requests to access a 'Thing';

- Step 2: 'Thing' sends an authentication request to its RA for verification purpose;
- Step 3: RA requests User ID;
- Step 4: User responses with HRA information;
- Step 5: RA verifies the user HRA information and sends ID verification request to the HRA;
  - ✓ Step 5.1: HRA challenges the user with a question;
  - ✓ Step 5.2: User responses the challenge with an answer;
- Step 6: HRA responses ID OK or not;
- Step 7: RA responses the 'Thing' about the user ID and issue a session key with the user as we described.

For better description of our protocol, we first introduce some relevant terms here.

Table 2. Notations Used in the Proposed Scheme

Symbol	Description
F_p	a finite field
E	an elliptic curve defined on F_p with a large order
P	a point on E
G	the group of elliptic curve points on E
h()	One-way hash function
s	the RA's private key
ID_u	the identity of the user
ID_t	the identity of the 'Thing'

As we known, key establishments and distribution are the fundamental tasks for entity authentication. We can use either SKC or PKC for their implementations, but we have to know the pros and the cons of each algorithm. SKC based schemes suffer the following problems: they require a large memory to store key materials, provide low scalability due to distribution of the keys, add and revoke keys, and require complicated key pre-distribution. On the other hand, PKC-based schemes suffer from high energy consumption and considerable time delay. PKC provides a more flexible and simple interface compared to SKC, which does not require key pre-distribution, pair-wise key sharing, or complicated one-way key chain schemes. For our situation, it is a wise choice if we adopt a PKC-based solution and at the meantime, we also address the aforementioned constraint problems. Based on current research achievements, we believe ECC-based solution is a solid one to be considered.

To establish a session key for two entities, taking a user and an object as an example, only three steps are required as follows;

- Firstly, the RA who is responsible for the object will produce a random P$\in$G and compute $P_s = sP$ in F_p. Note that, the s is a secret key that is assumed to be assigned before the RA has joined the IoT. For each user with ID_u, RA will generate P_u = h(ID_u) and the private key of the thing $S_u = sP_u$.
- Secondly, the user generate an ephemeral private key a and compute $Q_u = aS_u$ and $Q'_u = aP$. Then the user will send an authentication message $\{ID_u, Q_u, h(ID_u \| ID_t \| Q_u \| Q'_u)\}$ to the RA. Once receive the message, RA will

compute $Q_u'' = s^{-1}Q_u$ and check whether $h(ID_u \| ID_t \| Q_u \| Q_u')$ equal to h($ID_u || ID_t || Q_u || Q_u''$) or not. If not, authentication fails. Otherwise go to step 3.

- The third step is session key establishment. Similarly, the RA will choose a random ephemeral key *b* and compute $Q_t = bP$ for the desired 'Thing'. The session key will be h(abP) based on ECC algorithm.

The next question is how to authenticate a legitimate user in the IoT. 'Things' and users are in different domains. They could locate in different hierarchy level of the network. Central authentication method is only valid if a wide accepted KDC (key distribution center) is available. In industry, OpenID technology solves this problem. OpenID enables users to have a single account that allows them to log on to many different sites by authenticating a single identity provider [6]. One approach to identity management is federated identity management, in which participating sites form a circle of trust. Therefore, if the user is authenticated to one site, the other sites will automatically log the user in if the user visits them [6]. This lightweight idea should be adopted into our design. As such, user authentication is performed in the user domain or registered OpenID service provider. We denote it as home registration authority (HRA). Note that, peer-to-peer authentication method is another solution that can be utilized for further research. However, without solving the mutual-trust problem between two entities, this approach cannot be success.

The IoT needs to authenticate entities that are accessing the pervasive network in order to provide service to only registered members. The entity may be an IoT user or a device. The IoT is able to support a wide range of ages of users and reflect their own characteristics and needs. As a result, we can selectively use our favorite authentication method among existing authentication methods. The authentication mechanisms are safe and reliable. Our proposed authentication mechanism satisfies these requirements. The RA verifies the certificate contents and the identity of the 'Thing'. Two RA models exist in general PKI. In the first model, the RA collects and verifies the necessary information for the requesting entity before a request for a certificate is submitted to the HRA. The HRA trusts the information in the request because the RA already verified it. In the second model, the HRA provides the RA with information regarding a certificate request that it has already received. The RA reviews the contents and determines if the information accurately describes the user. The RA provides the HRA with a "yes" or "no" answer. It is a device of the kind that has the same or more computing power, memory, and data protection module. Therefore, the RA generates key pairs and requests and receives certificates for all 'Thing'.

4. Analysis of the Proposed Scheme

In this section, we analyze the proposed user authentication protocol against the various kind of possible attacks.

4.1 Eavesdropping Attack

For each authentication process a different session key is produces. Therefore, knowledge of past session keys does not allow deduction of future session keys. In our scheme, the session key is decided by one-way hash function, so the parameter abP value cannot be calculated by eavesdropping the session key. Only the user and RA know the abP, which is computed from the random ephemeral key. That is, even if the previous session secrets are revealed, the other secrets will remain unknown to the adversary.

4.2 Man-in-the-middle Attack

Compromising of a long term secret key, such as SA' at some point in the future, does not lead to compromise of communications in the past. Note that in our scheme, even if the adversary compromises the RA's secret key, it cannot compromise the previous session key because the adversary cannot know the ephemeral key a or b such that it cannot compute the session key. Also, our protocols satisfy both partial forward secrecy and perfect forward secrecy since it is hard to compute the session key without knowing the ephemeral key a or b.

4.3 Key Control Attack

Both communication entities select a random number to generate the session key, which would be discarded after the session expired. Neither one can control the outcome of the session by, for example, restricting it to lie in some predetermined small set. In other words, neither entity can force the session key to a pre-selected value. Hence, our proposed protocol can resist any key control attack.

4.4 Replay Attack

In case a malicious one gained a valid session key or captured network traffic in the IoT, the protocol should resist replay attack by introducing a nonce in every transmitted message. However, it is an optional choice that could vary on different applications. Besides, the session key could be used for identification. Therefore, replayed message from unidentified person will be discarded.

5. Conclusion

With the rapid development of IoT, it has penetrated into every aspect of our lives and works, but information security has become the bottle neck of its further development. In this paper, we have proposed a user authentication protocol in Internet of Things based on the ECC and OpenID. The OpenID system enables compact and simple authentication scheme, and the ECC provides better security strength for lightweight authentication system. Analysis results show that our approach can prevent attacks like eavesdropping, the man-in-the middle, key control attack, and replay attacks.

The future work includes implementation of this system and doing performance analysis. Also, we need to study more about analysis of various threat to user authentication which are main hindrance to success of IoT.

Acknowledgments

This work was supported by the Incheon National University Research Grant in 2015.

References

[1] Toan-Thinh Truong, Minh-Triet Tran and Anh-Duc Duong, “Robust Mobile Device Integration of a Fingerprint Biometric Remote Authentication Scheme”, IEEE Conference on Advanced Information Networking and Applications (AINA), pp.678-685, **(2012)**.
[2] Neal Koblitz, “Elliptic Curve Cryptosystems”, Mathematics of Computation, vol.48, no.177, January **(1987)**, pp.203-209.
[3] M. Rosing, “Implementing Elliptic Curve Cryptography”, Greenwich, CT: Manning Publication, **(1999)**.
[4] M. Ohkubo, K. Suzuki and S.Kinoshita, “Efficient Hash-chain Based RFID Privacy Protection Scheme”, International Workshop on Ubiquitous Computing (IWUC), **(2004)**, September.
[5] V. Gayoso Martinez and L. Hernandez Encinas, “Implementing ECC with Java Standard Edition 7”, International Journal of Computer Science and Artificial Intelligence, Vol.3, Iss.4, **(2013)**, pp.134-142.

[6] R. Watanabe and T. Tanaka, "Federated Authentication Mechanism using Cellular Phone-Collaboration with OpenID", Conference on Information Technology: New Generations, Las Vegas, USA **(2009),** pp.435-442.
[7] L. Xiong, X. Zhou and W. Liu, "Research on the Architecture of Trusted Security System Based on the Internet of Things", Proceedings of the 4th International Conference on Intelligent Computation Technology and Automation, **(2011)**, pp.1172-1175.
[8] A. C. Sarma and J. Girao, "Identities in the Future Internet of things", Wireless Personal Communications, vol.49, issue 3, May **(2009)**, pp. 353-363.
[9] A. Vapen, D. Byers and N. Shahmehri, "2-clickAuth – Optical Challenge-Response Authentication", Proceedings of 2010 International Conference on Availability, Reliability and Security, **(2010)**, pp.79-86.
[10] K. Wong, Y. Zheng, J. Cao, and S. Wang, "A Dynamic User Authentication Scheme for Wireless Sensor Networks", IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, **(2006)**.
[11] H. Tseng, R. Jan and W. Yang, "An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks", IEEE Global Communications Conference, **(2007)**.
[12] H. Wang and Q. Li, "Distributed User Access Control in Sensor Networks", Distributed Computing in Sensor Systems (LNCS4026), **(2006)**, pp. 305-320.
[13] J. Zheng, J. Li, M. J. Lee and M. Anshel, "A lightweight encryption and authentication scheme for wireless sensor networks", International Journal of Security and Networks, vol.1, no.3/4, **(2006)**, pp.138-146.

Authors

Jong Jin Lee, Mr. Lee received the B.S. and M.S. Degrees in Information and Telecommunication Engineering from Incheon National University, Incheon, Korea in 2014 and 2016, His research interests include Network Security, Cloud Computing, User Authentication, Cryptographic Algorithm, and Iot Security & Applications.

Ki Young Lee, Dr. Lee received the B.S. and M.Eng. Degrees in Electrical Engineering from Yonsei University, Seoul, Korea in 1982 and 1984, respectively. And he received M.S. (1987) from the University of Colorado, Boulder and the Ph.D. (1993) from the University of Alabama in Electrical & Computer Engineering. Since 1994, he has been a professor in the Department of Information and Telecommunication Engineering at Incheon National University. His research interests include Internet traffic control and protocols, user authentication protocols, and security mechanism of IoT environment.