Design and Evaluation of Identity-based Cryptography Algorithm for The Smart-home Solution*

Hung Quang Nguyen

VSB-TUO, Ostrava, Czech Republic
hung.quang.nguyen.st@vsb.cz

Abstract. Smart homes meet all the needs of people with their "intelligence." However, to reduce costs, providers currently only use existing encryption techniques of wireless networks. In fact, these encryption techniques are beneficial for data transmission in a wireless environment. But Smart-home is built on wireless sensor networks (WSN) with the energy-saving requirements. Encryption based on the Elliptic curve uses fewer strings than any other encryption technique with the same level of safety, and using Elliptic curve cryptography (ECC) is optimal for energy for the Smart-home deployment. Currently, Smart-homes only use one server (Home server) to manage and implement the user's requests. Therefore, if it uses ECC for communication between the user and the Home server, the first thing required is the agreement between two sides to select the Elliptic curve.

To alleviate this stage, the proposed solution is to use another server (Key server). Its main task is sending keys and parameters of the Elliptic curve for the users and the Home Server. The key servers can be shared among multiple Smart-homes, which helps to increase the links between suppliers and customers. Using the Elliptic curve for encryption and decryption, information should be converted into points on the curve, and ASCII is the current solution. But with a higher security level, using conversion tables to change the values for each curve is a better solution. Moreover, this solution also offers certification as one more security layer for device authentication. Finally, it makes Smart-home safer as it prevents forgery cases.

Keywords: Smart-home, security, ECC.

1 Introduction

1.1 Identity-based encryption (IBE)

IBE is an important primitive of ID-based cryptography, a type of public-key cryptography, in which a publicly known string representing an individual or organization is used as a public key. The public string could include an email address, domain name, or a physical IP address. As such it is a type of public-key encryption in which the public key of a user is some unique information about the identity of the user. Agrawal [1] proposed a secure IBE model on the basis of hard problems. IBE is an efficient public-key encryption mechanism for secure communication between any pair of entities without identity ID disclosure.

The steps involved are described in the below figure:

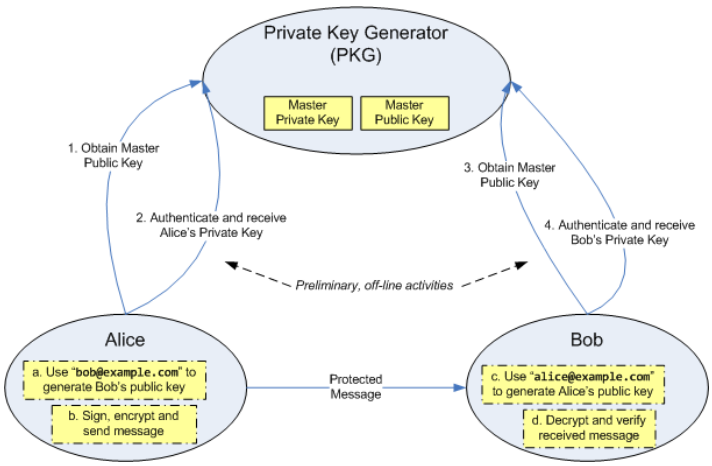

Fig. 1. ID Based Encryption: offline and online Steps [2]

IBE was first referred to in 1984 by Adi Shamir [3], when he described the profile of the properties and how to use such a system. Although he failed to establish a safe technology with operational feasibility, but he described the workflow of IBE as follows: While the traditional public key contains all the necessary parameters, in the IBE system, users need to get a set of parameters from a third party. Along with these parameters, users can also calculate IBE public key of any other user to encrypt the information sent to them. The recipients of the encrypted information then validate it in with a private key generator (PKG), the IBE private key corresponding to a specific IBE public key generated by a trusted third party calculate. The master private key is generated by PKG from a trusted third party using confidential information and the identity of the user. The private key is then delivered securely to the authorized users.

On the other hand, an IBE scheme has four algorithms used to create and use a pair of private key - public key: setup algorithms, extraction algorithms, encryption algorithm and decryption algorithm. "Setup" is the algorithm to initialize the parameters

needed for calculating of IBE, including the master private key that PKG used to create the IBE private keys. "Extraction" is the algorithm for calculating an IBE private key from the parameters created using the identity of the user and the master private key of PKG. "Encryption" is performed by IBE public key calculated. "Decryption" was performed by IBE private key.

1.2 Elliptic Curve Cryptography

With the increasing popularity of cloud services and social networks, personal information in the Internet faces the risk of leakage. Thus, requirements for data security receive increasing concerns. An effective and most widely used method is data encryption. Among many existing ciphers, Elliptic Curve Cryptography (ECC) is one of the strongest encryption algorithms, and also the most complex. Modern encryption is established based on the idea that the key used to encrypt the data can be released while the key for decrypting the data must be kept confidential. These systems are therefore known as public key encryption system. In 1985, public key encryption algorithm proposed new declaration based on an elliptic curve. Elliptic curves are a set of points correspond to a particular mathematical equation. The equation for an elliptic curve looks like:

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

It indicates that if there are any two points, provided a performance of "present itself n times", then finding out n when only the beginning and the end points are known is very difficult. Applied to the example of billiard games, a player is in the room alone for a period of time, hitting the ball to follow the described rules. If someone else came into the room then and saw where marbles are, even if they know all the rules of the game and the start position, they cannot determine how many times balls are hit without play through the entire game again. Easy to implement, difficult to reverse action, this is a good TF. An ECC system can be defined by selecting a limited number of elements, making a curve equation and a point on that curve. A private key is a privy number and a public key as a result of the plus first point with itself privy times. Calculating the private key from the public key encryption system is called Elliptic Curve Discrete Logarithm Problem (ECDLP). It is the TF that researchers are looking for.

In the current era of information and communications technology, the need to ensure information security is indispensable. With the increasing length of encryption key, ECC is a suitable candidate to replace RSA in creating the shorter code lock while safety is still ensured. It can be deployed on multiple platform devices from simple electronic circuits to the mainframe to create a reliable network to serve society better. The addition of points on elliptic curves over the real numbers is a good approach to see the underlying steps in performing the operation. However, calculations prove to be slow and inaccurate due to rounding errors, and the implementation of these calculations into cryptographic schemes requires fast and precise arithmetic. Therefore elliptic curve groups over finite fields such as Z_P, when $p > 3$ is prime, are used in practice. An elliptic curve with Z_P as its underlying field can be formed by choosing a and

b within the field Z_P. Similar to the real case, the curve includes all points (x, y) in Z_P * Z_P that satisfy the elliptic curve equation.

$$y^2 \equiv x^3 + ax + b \bmod p \quad (2)$$

where x and y are numbers in Z_P. Note that there are only finitely many points on this type of curve. As in the real case, if $4a^3 + 27b^2 \neq 0 \bmod p$, then the corresponding elliptic curve forms a group [4]. This group consists of the points on the curve, along with ∞, the point at infinity. Again, we define the negative of the point at infinity to be $-\infty = \infty$ and the negative of a point $P = (x_p , y_p)$ to be $-P = (x_p, -y_p \bmod p)$. The arithmetic in an elliptic curve group over Z_P is very similar to that done algebraically with elliptic curve groups over the real numbers, the only difference is that all calculations are performed modulo p [5].

Suppose $P = (x_p,y_p)$ and $Q = (x_q,y_q)$ and that $P \neq -Q$. Let s be given by

$s \equiv (y_p - y_q) / (x_p - x_q) \bmod p$. Then $P + Q = R$, where $x_r \equiv (s^2 - x_p - x_q) \bmod p$

$y_r \equiv -y_p + s * (x_p - x_r) \bmod p$.

As before, we define $P + (-P) = \infty$. If the y-coordinate of P is 0, modulo p, then $P = -P$, to double the point $P = (x_p,y_p)$ with $y_p \neq 0 \bmod p$, let s be given by:

$s \equiv (3x^2_p + a) * (2y_p)^{-1} \bmod p$.

We define $2P = P + P = R$ where:

$x_r \equiv s^2 - 2x_p \bmod p$, $y_r \equiv -y_p + s * (x_p - x_r) \bmod p$.

2 System design

As mentioned above, IBE and ECC are very strong for encryption, thus combining them is a great solution to integrate the advantages of both method. Additionally, this combination also adds authentication by ID-device to increase the safety for the system. For maximum security, Smart-homes can only be controlled by the devices which were registered with the Home server. Key server is placed at the service provider to create and allocate private keys, public keys for users and Home server.

The processes in this model include the below steps: ID authentication, create and send keys and information of Elliptic, encrypt and send the request, decrypt and execute the request, decrypt and execute the feedback.

Step 1/ User use a registered device (smartphone, tablet, laptop...) to connect to Home server

Step 2/ If the device is genuine, Home server will notify Key server

Step 3/ Key server requires User login

Step 4/ User inputs username and password

Step 5/ If the information is true, Key server will:

Choose a curve Elliptic and 36 points in the curve with a selected base point P_E. Using the agreed code table, 36 points are mapped to 26 characters of Alphabet (A,B,C ... X,Y,Z) and ten numbers from 0 to 9. Each character or number is corresponding to each point of Elliptic.

Select a random number, it is a master private key (M_{PK})

Create the private key (PR), the public key (PU) for User (U) and Home server (H) by calculate:

H (ID_U): hash value to the identity of User (ID_U)

$PR_U = M_{PK} * H(ID_U)$: a private key of User

$PU_U = PR_U * P_E$, PU_U: a public key of User

Choose a random number, it is the private key of Home server (PR_H)

$PU_H = PR_H * P_E$: the public key of Home server

Send keys and information of curve to User and Home server

User uses PU_H to encrypt and send the request (RQ) to the Home server by:

Choose a random number K_1

$RQ_1 = K_1 * P_E$

$RQ_2 = RQ + K_1 * PU_H$

Send $\{RQ_1, RQ_2\}$ to Home server

Step 7/ Home server uses PR_H to decrypt request and execute it using the equation:

$RQ_2 - PR_H * RQ_1 = (RQ + K_1 * PU_H) - PR_H * (K_1 * P_E)$

$= RQ + (K_1 * PR_H * P_E) - (PR_H * K_1 * P_E) = RQ$

Step 8/ Home server uses PU_U to encrypt and send the feedback (FB) to User by:

Choose a random number K_2

$FB_1 = K_2 * P_E$

$FB_2 = FB + K_2 * PU_U$

Send $\{FB_1, FB_2\}$ to User

Step 9/ User uses PR_U to decrypt feedback using the equation:

$FB_2 - PR_U * FB_1 = (FB + K_2 * PU_U) - PR_U * (K_2 * P_E)$

$= FB + (K_2 * PR_U * P_E) - (PR_U * K_2 * P_E) = FB$

The devices used to control Smart-homes must have a specific license, without with the attempt to access or control the system will be denied. The users are created for all members of Smart-homes, and they will be able to log in from any device provided the license. According to the theory, if each user can use only on one device, it will be more secure. But in reality, it can cause some troubles. For example, one device is lost or its battery is dead, the owner of that device must use other people's device, along with the information to control Smart-home. Such situations are not feasible. Also, the majority of users are using many devices (smartphones, tablets, laptops...), so we have to create several users only for a person, which is wasteful and difficult to remember all credentials. Elliptic is calculated based on the numbers, so related information must be converted to numbers. That is the reason the identities of users need to use hash function to get corresponding numbers in the database. Also, when the encryption and decryption requests are made by users, they must be converted into points on the curve. If a user wants to open the door and send a request message “DOOR”, all the points on the elliptic curve can be directly mapped to an ASCII value. Select a curve and we will get a minimum of 128 points so that we fix each point on this curve to an ASCII value. For example, “DOOR” can be written as a sequence of ASCII as “68-79-79-82”, we can map these values to fixed points on the curve. The steps for encoding and decoding are given in the flowchart. Each character is mapped to a corresponding number in ASCII

table, and this number is converted to a point on the curve. Because all requests for Smart-homes are not the custom chain of commands and they are based on available scenarios. Every request can be assigned to a particular number. Hash function is used to make a number compatible with a request. The two mentioned methods have some disadvantages: if a hacker knows how to get the numbers, they can calculate from the given numbers and find the corresponding command, the system will not be safe anymore. For example, if they know the periods when the door open, they will penetrate before when nobody is home. Another method using the change in the process of creating Elliptic curve to build many rules to convert numbers. We will choose 36 points in each curve and calculate as follows:

Choose x = 0, calculate y give us 2 points in the curve.

Plus x with 1 and continue calculating, until we find 36 points in the curve.

Assign 36 points with 26 characters of the alphabet and 10 numbers from 0 to 9. This method has different mapping rules for each command, so hackers cannot rely on them to find out the requests of the user.

3 Simulation scenario

We use Sagemath 7.0 to simulator the scenario, this tool runs in CentOS system. The hardware of the used computer: Intel Core i5-4200M CPU, 16GB RAM and 1TB HDD.

Create the private key (PR), the public key (PU) and Elliptic curve

Choose a Elliptic curve and a base point P in curve: $y^2 = x^3 + ax + b \bmod p$

If a = 2, b = 9 and p = 37; we have: $E_{37}(2,9) = (y^2 = x^3 + 2x + 9) \bmod 37$

Selects a random number, it is a master private key (M_{PK}): we choose number 4

$H(ID_U)$: hash value to the identity of the user (ID_U), assumption it is 5

$PR_U = M_{PK} * H(ID_U) = 4 * 5 = 20$

And $PU_U = PR_U * P_E = 20 * (10,20) = (26,32)$

Choose a random number, it is a private key of Home server (PR_H), assumption it is 6

$PU_H = PR_H * P_E = 6 * (10,20) = (15,26)$

Assign 36 points with 26 characters of the alphabet and 10 numbers from 0 to 9, we have:

A	B	C	D	E	F	G	H	I
(0,3)	(0,34)	(1,7)	(1,30)	(2,13)	(2,24)	(4,9)	(4,28)	(5,12)
J	K	L	M	N	O	P	Q	R
(5,25)	(7,12)	(7,25)	(9,4)	(9,33)	(10,17)	(10,20)	(11,17)	(11,20)
S	T	U	V	W	X	Y	Z	0
(13,7)	(13,30)	(15,11)	(15,26)	(16,17)	(16,20)	(21,5)	(21,32)	(23,7)
1	2	3	4	5	6	7	8	9
(23,30)	(25,12)	(25,25)	(26,5)	(26,32)	(27,5)	(27,32)	(29,6)	(29,31)

Fig. 2. Mapping of $E_{37}(2,9)$

Suppose a user wants to open the system 1 in Smart-home and send the request: OPEN1, corresponding to points on the E_{37}:

O = (10,17) and we select number 7 is K_1, Encryption:{RQ_1, RQ_2} = {N, Y}

P = (10,20) and we select number 8 is K_1, Encryption:{RQ_1, RQ_2} = {F, V}

E = (2,13) and we select number 9 is K_1, Encryption:{RQ_1, RQ_2} = {X, S}

N = (9,33) and we select number 10 is K_1, Encryption:{RQ_1, RQ_2} = {Q, A}

1 = (23,30) and we select number 11 is K_1, Encryption:{RQ_1, RQ_2} = {3, G

The request was sent: {N,Y; F,V; X,S; Q,A; 3,G}

The request was received: {O,P,E,N,1}

```
(10 : 20 : 1) 20 (26 : 32 : 1) 6 (15 : 26 : 1)
14
N
25
Y
(9 : 33 : 1) (21 : 5 : 1) (10 : 17 : 1)
0
6
F
22
V
(2 : 24 : 1) (15 : 26 : 1) (10 : 20 : 1)
P
24
X
19
S
(16 : 20 : 1) (13 : 7 : 1) (2 : 13 : 1)
E
17
Q
1
A
(11 : 17 : 1) (0 : 3 : 1) (9 : 33 : 1)
N
30
3
7
G
(25 : 25 : 1) (4 : 9 : 1) (23 : 30 : 1)
1
['N', 'Y', 'F', 'V', 'X', 'S', 'Q', 'A', '3', 'G']
['O', 'P', 'E', 'N', '1']

26.018044
```

Fig. 3. Encryption {N,Y;F,V;X,S;Q,A;3,} and Decryption {O,P,E,N,1}

4 Discussion

Compared with using IBE method, this method has two steps less. They agree on points of the Elliptic curve and send the keys together between User and Home server. Thus, information of Elliptic curve and the keys will be safer. Compared with using ECC method, this method converts 36 points on Elliptic curve to the alphabet and the 10 digits to make the attackers impossible to guess the encoding rule because 36 points are not fixed, they change depending on the parameters of Elliptic curve.

In fact, the existing encryption techniques still meet demands of coding and decoding information as trust of the suppliers. But it is not the most optimal solution, and encryption based on Elliptic curve has proven to be especially excellent for wireless sensor networks in general and Smart-home in particular, given the energy savings benefit. However, it depends on the users' understandings to choose whether they should use the Home server to manage keys and parameters of Elliptic curve or not. Many users think having a complex password is secure enough. They can set a complicated password without paying attention to the password management system, which only consist of defaut login parameters. The majority of users are not thinking of having additional securities.

Also, there are still problems in information channels between Home servers and Smart-home devices (lights, gates, cameras, etc.), that will need to be solved in the future. If hackers succeed in attacking this information channel, they are capable of occupying the devices. In case there is no person in the house, the device will serve the attackers with malicious intent. One possible solution for this situation is to statistically consider and compare between control commands from users and the number of times the device operate. If the information does not match, there are possibilities to believe the system was damaged or hacked. However, this solution needs further research and thorough assessment.

References

1. Slamanig, D.: More privacy for cloud users: Privacy-preserving resource usage in the cloud. In: 4th Hot Topics in Privacy Enhancing Technologies, HotPETs (2011)
2. https://en.wikipedia.org/wiki/IDbasedencryption
3. Stallings, W.: Cryptography and network security. Prentice Hall, Boston (2011)
4. Koblitz, N.: Algebraic Aspects of Cryptography. Springer-Verlag, Location (1998)
5. https://www.certicom.com/content/certicom/en/ecc-tutorial.html