applied sciences

MDPI

Article

Improved Attribute-Based Encryption Using Chaos Synchronization and Its Application to MQTT Security

Teh-Lu Liao [1], Hong-Ru Lin [1], Pei-Yen Wan [1] and Jun-Juh Yan [2,*]

[1] Control Engineering Group, Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan; tlliao@mail.ncku.edu.tw (T.-L.L.); z8652076@yahoo.com.tw (H.-R.L.); joey345656@gmail.com (P.-Y.W.)

[2] Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung 41107, Taiwan

* Correspondence: jjyan@ncut.edu.tw; Tel.: +886-4-23924505 (ext. 7379)

Received: 17 August 2019; Accepted: 17 October 2019; Published: 21 October 2019

Abstract: In recent years, Internet of Things (IoT) has developed rapidly and been widely used in industry, agriculture, e-health, smart cities, and families. As the total amount of data transmission will increase dramatically, security will become a very important issue in data communication in the IoT. There are many communication protocols for Device to Device (D2D) or Machine to Machine (M2M) in IoT. One of them is Message Queuing Telemetry Transport (MQTT), which is quite prevalent and easy to use. MQTT is designed for resource-constrained devices, so its security is not as strong as other communication protocols. To enhance MQTT security, it needs an additional function to overcome its weakness. However, considering the limited computational abilities of resource-constrained devices, they cannot use too powerful or complicated cryptographic algorithms. Therefore, this paper proposes novel improved attribute-based encryption (ABE) integrated with chaos synchronization to enhance the MQTT security. Finally, a small size of IoT is implemented to simulate resource-constrained devices equipped with a human–machine interface and monitoring software to show and verify the performance of MQTT communication with this improved ABE algorithm.

Keywords: Message Queuing Telemetry Transport (MQTT); attribute-based encryption (ABE); chaos synchronization

1. Introduction

In recent years, Internet of Things (IoT) technology has flourished, which has led to the development of many peripheral and emerging markets. The IoT is a physical object which contains embedded wireless devices and sensors, and sends its data to the processing platform through internet communication technologies. Therefore, IoT has a wide range of applications in various fields. For resource-constrained devices in IoT, Message Queuing Telemetry Transport (MQTT) is a lightweight publish/subscribe communication protocol, and is quite prevalent and easy to use. For example, the MQTT protocol has been applied in a structural monitoring scenario [1] and a long-term energy monitoring system [2]. In the IoT application, each information processing platform can be connected to another one to form a huge network. Obviously, most IoT data is opened and stored on the Internet or in IoT devices, so many new security and privacy issues have been derived [3]. Taking smart meters, for example, the hacker can know when you will be at home and what devices you have used. Therefore, for these resource-constrained devices, the absence of security or weak security mechanisms will cause an unpredictable crisis, and the use of these devices will then be limited in some aspects. However, considering the limited computational abilities of resource-constrained

devices, they cannot use too powerful or complicated cryptographic algorithms to ensure the security of data. To solve this problem, we have studied and discussed the security mechanism of MQTT protocol commonly used in IoT [4–7]. To solve the data security of resource-constrained devices, cryptography is generally used to protect the data, so that clearly visible information can be transformed into seemingly messy ciphertext to protect them from malicious people. It is usually not a complex process to encrypt data when using the MQTT protocol. In this way, the data will be vulnerable to eavesdroppers. To encrypt the data, the available encryption schemes on the Internet fall into two categories: symmetric encryption and asymmetric encryption. Both of these have to repeatedly encrypt the data by the key and decrypt the data by the same or another key data-by-data. The traditional encryption technique is not suitable for publish/subscribe communication protocol such as MQTT. To solve this problem, Sahai and Waters proposed a new cryptographic technique called attribute-based encryption (ABE) [8]. A data owner (or publisher) can specify access to the data as a Boolean formula related to a set of attributes and the data will be encrypted in relation to this. Then, the data user (or subscriber) can only decrypt the ciphertext if they possess the attributes satisfying the Boolean formula ascribed to the ciphertext. ABE has two types: Ciphertext-Policy ABE (CP-ABE) [9] and Key-Policy ABE (KP-ABE) [10]. However, the disadvantage of these is their complex computation, which is not suitable for resource-constrained devices.

Therefore, in this direction, some approaches have been proposed [6,11,12]. The common way to reduce computation is to use an elliptic curve in a bilinear map [6,11]. A pre-computation method was proposed to store a set of pairs and to simply put and exchange the memory for the encryption time [12]. Since we aimed to design an encryption algorithm suitable for resource-constrained devices, the computing ability and memory requirements had to be reduced enough. A widespread consensus is that the computation overhead of bilinear pairing is excessive in the practical application of ABE, especially for devices with limited computational resources. Therefore, the authors [13] proposed a scheme called Pairing-Free CP-ABE (PF-CP-ABE) [13], replacing bilinear pairing with simple scalar multiplication on elliptic curves, in order to reduce the overall computational overhead. It substantially works like CP-ABE, and the elliptic curve is difficult to compute, so it is still safe enough. However, since data keeps transmitting between resource-constrained devices, and the authority (MQTT broker) needs to identify users and send certification results to them each time, the schemes mentioned above cannot provide effective data communication.

In the last two decades, chaos systems have been widely applied in secure communication. Chaos systems have characteristics such as a butterfly effect sensitive dependence on the initial conditions, so that eavesdroppers cannot easily deal with them. Motivated by the aforesaid discussion, in this paper, we propose a new Chaos Synchronization Attribution-Based Encryption Scheme (CS-ABE) utilizing the random property of the chaos signal and integrating chaos synchronization with PF-CP-ABE. We take the secret number of PF-CP-ABE generated by a chaos random number. Before chaos synchronization, the system we propose, combining PF-CP-ABE with chaos synchronization, works like PF-CP-ABE. After chaos synchronization, the state variables of both chaos systems are synchronized, and the system can just obtain the secret number from itself. Therefore, the performance and security of PF-CP-ABE can be improved.

The remainder of this paper is structured as follows. The algorithms we apply are described in Section 2, such as elliptic curve cryptography (ECC), ABE, and chaos synchronization. Our proposed scheme is proposed in Section 3. The performance and security are discussed in Section 4. Finally, the conclusion is given in Section 5.

2. Algorithms

2.1. Elliptic Curve Cryptography

In 1985, Koblitz and Miller first proposed the new cryptographic algorithm of elliptic curve cryptography, called ECC. The elliptic curve—$E_p(a, b)$—is defined by the equation below:

$$y^2 = x^3 + ax + b \ (mod\ p), \tag{1}$$

where p is the prime value.

Because of the advantage of ECC, which has a shorter key length and faster calculation speed compared with RSA algorithm[14], as solving the elliptic curve discrete logarithm problem (ECDLP) is more difficult than factoring an integer, the application of it has become a very popular research topic in cryptography. To be specific, the elliptic curve must be a cyclic group over a finite field $GF(p)$, given a base point g called a generator with order r, and given $Q = tg$, $t \in Z_r$. It is tough to find an integer t with g and Q. ECC can be generally divided into the four steps below.

(1) Set up

The plain text is first mapped to a point M on the elliptic curve, and A and B then agree on the same elliptic curve $E_p(a, b)$ with a generator g.

(2) Key generation

A selects an integer $t_a \in Z_r$ as its private key and computes $g_a = t_a g$ as the corresponding public key. B also selects an integer $t_b \in Z_r$ as its private key and computes $g_b = t_b g$ as the corresponding public key.

(3) Encryption

A randomly selects an integer $k \in Z_r$ and computes $C_1 = kg$ and $C_2 = M + kg_b$, and then transmits both to B.

(4) Decryption

After B receives C_1 and C_2, B then computes $C_2 - t_b C_1$ to get point M. Finally, M can be mapped back to the original plain text.

$$\begin{aligned} C_2 - t_b C_1 &= (M + kg_b) - t_b kg \\ &= (M + kt_b g) - kt_b g = M \end{aligned} \tag{2}$$

Besides, ECC can define bilinear pairing such as Weil pairing and Tate pairing, which has been widely used in Identity-Based Encryption (IBE) [15] and ABE.

2.2. Attribute-Based Encryption

When it comes to ABE, it has an access structure and Linear Secret Sharing Schemes (LSSS) [16]. The access structure stipulates who the eligible users are, and what the corresponding attributes should be. LSSS is an expressive monotone access structure used to generate the matrix of access structure for mathematical calculation. Because we designed it for resource-constrained devices with MQTT, we made use of the characteristics with topic names like in [6]. Figure 1 shows the access structure. We used MQTT topic names such as User1, WiFiDevice, Led, Temp, and Humidity.

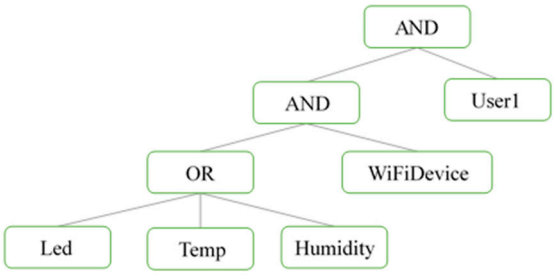

Figure 1. Access structure.

The Boolean formula of it is defined as

$$\text{User1} \wedge \text{WiFiDevice} \wedge (\text{Led} \vee \text{Temp} \vee \text{Humidity}). \tag{3}$$

As long as the access structure is constructed by the data owner, it can be transformed to an LSSS matrix according to the method. Our matrix was generated as below:

$$N = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

Each row of N is associated with the attributes User1, WiFiDevice, Led, Temp, and Humidity, respectively. Given an attribute set S, the LSSS is said to be satisfied by S only if the rows of the N labeled by the attributes in S include the vector (1, 0, 0) in their span.

The research presented in [11] is different from the other ABE algorithm, as its LSSS matrix has no polynomials, which can reduce some computation. Second, replacing the bilinear map with simple scalar multiplication by ECC also reduces some computation.

PF-CP-ABE generally consists of four parts: Set up, Key generation, Encryption, and Decryption [13]. In Set up, the authority selects its main public keys and private key, and everyone should agree on the same elliptic curve. Key generation is employed to generate a key of every attribute for each user and store them in a corresponding user list. In the Encryption part, the plain text is ciphered by an access structure and corresponding attributes. Then, in the Decryption part, the data user should ask the authority to verify the attributes they possess, and requires a result to correctly calculate the secret number and then map the ciphertext back to the plain text. The algorithm will be described in detail in Section 3.

2.3. Chaos Synchronization

In order to reduce the time required for the transmission of certification data and maintain its security of PF-CP-ABE, we propose Chaos Synchronization Attribute-Based Encryption (CS-ABE). As is well-known, the chaos system is a complex nonlinear system and possesses properties such as a broadband noise-like waveform, and is sensitive to initial values like a butterfly effect, etc. The state response does not converge and does not diverge and is limited to strange attractors with a random-like characteristic. These properties offer some advantages for applications in secure communication. Since the concept of the master-slave system emerged and research on the synchronization controller of the chaos system was launched [17], there has been extensive research on it. There are many different design methodologies for synchronization controllers, such as the sliding mode controller [18], adaptive controller [19], etc., but most of them applied in secure communication are analog chaos systems. In this study, since we dealt with digital information, it was necessary to transform it to a discrete

chaos system so that we could discuss and implement a relevant control method, and then apply it to the design of our CS-ABE algorithm.

There are many types of continuous chaos systems, the most well-known of which is the Lorentz system. This paper uses the Lorenz equation and its dynamics can be described as follows:

$$\begin{aligned} \dot{x_1}(t) &= -ax_1(t) + ax_2(t) \\ \dot{x_2}(t) &= cx_1(t) - x_2(t) - x_1(t)x_3(t) \\ \dot{x_3}(t) &= -bx_3(t) + x_1(t)x_2(t). \end{aligned} \tag{5}$$

In consequence, in this paper, we will describe the approach for transforming a continuous system model to a discrete system one. The dynamic equation of the continuous system can be described as follows:

$$\dot{x}(t) = Ax + Bg(x(t)), \tag{6}$$

where $g(x(t)) \in R^n$ is a nonlinear vector. $B \in R^{n \times m}$, A, and B are controllable, so the optimal discrete time system can be matched with System (5) as follows:

$$x_d(k+1)T = Gx_d(kT) + Hg(x_d(kT)), \tag{7}$$

where $G = e^{AT}$, $H = [G - I_n]A^{-1}B$ [20], and T is the sampling time. We can rearrange (5) with a matrix representation satisfying (6):

$$\dot{x}(t) = \begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \\ \dot{x_3}(t) \end{bmatrix} = \begin{bmatrix} -a & a & 0 \\ c & -1 & 0 \\ 0 & 0 & -b \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -x_1x_3 \\ x_1x_2 \end{bmatrix}. \tag{8}$$

According to (7), by selecting the sampling time $T = 0.0001s$, $a = 10$, $b = \frac{8}{3}$, and $c = 28$, we get the discrete system of (5) as

$$\begin{aligned} x_d(k+1)T = & \begin{bmatrix} x_{d1}(k+1)T \\ x_{d2}(k+1)T \\ x_{d3}(k+1)T \end{bmatrix} = \begin{bmatrix} 0.990 & 0.010 & 0 \\ 0.028 & 0.999 & 0 \\ 0 & 0 & 0.997 \end{bmatrix} x_d(kT) + \\ & \begin{bmatrix} 0 & 0 \\ 0.001 & 0 \\ 0 & 0.001 \end{bmatrix} \begin{bmatrix} -x_{d1}(kT)x_{d3}(kT) \\ x_{d1}(kT)x_{d2}(kT) \end{bmatrix}. \end{aligned} \tag{9}$$

The simulation of the discrete chaos system (9) is shown in Figure 2. From this, we can identify some characteristics, such as strange attractors and the unpredictability of its random-like signal.

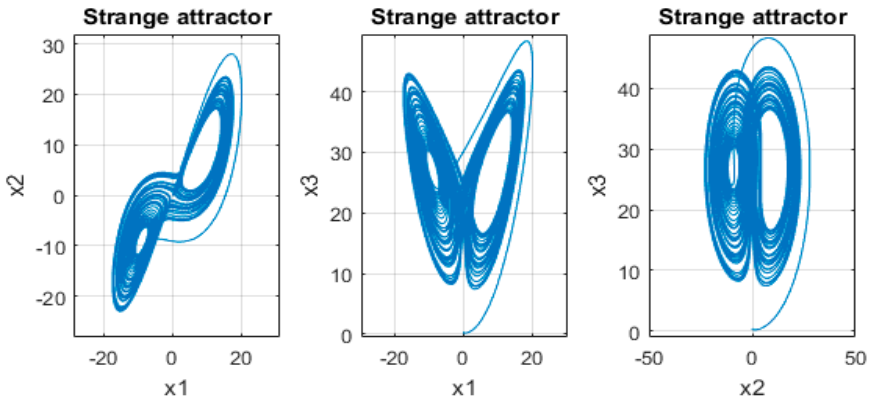

Figure 2. Strange attractors of the discrete Lorenz system (9).

For secure communication application, a synchronization controller is needed to synchronize master and slave chaos systems. When both systems are synchronized with each other, they can be applied to our cryptographic algorithm. We used a sliding mode controller, because it has a better robustness and fewer control parameters, to ensure the synchronization. Similar to [21], master and slave systems were designed as follows:

$$\begin{array}{c} x_{1m}(k+1) = 0.99x_{1m}(k) + 0.01x_{2m}(k) \\ x_{2m}(k+1) = 0.028x_{1m}(k) + 0.999x_{2m}(k) - 0.001x_{1m}(k)x_{3m}(k) \\ x_{3m}(k+1) = 0.997x_{3m}(k) + 0.001x_{1m}(k)x_{2m}(k) \end{array} \tag{10}$$

$$\begin{array}{c} x_{1s}(k+1) = 0.99x_{1s}(k) + 0.01x_{2s}(k) \\ x_{2s}(k+1) = 0.028x_{1s}(k) + 0.999x_{2s}(k) + u(k) \\ x_{3s}(k+1) = 0.997x_{3s}(k) + 0.001x_{1s}(k)x_{2s}(k), \end{array} \tag{11}$$

where x_{1m}, x_{2m}, and x_{3m} are state variables of the master system (10); x_{1s}, x_{2s}, and x_{3s} are state variables of slave system (11); and u is the proposed controller. To synchronize both systems, error functions are defined as $e_i = x_{is} - x_{im}$, $i = 1, 2, 3$. If e_i can converge to zero, it means that the master and slave systems can be synchronized. To achieve synchronization, we used a sliding mode controller to ensure that the system reached the switching function we designed and the switching function in the sliding manifold ensured that e_i could converge to zero, thus achieving synchronization. The switching function for the sliding mode control was selected as follows:

$$s(k) = e_2(k) + ce_1(k). \tag{12}$$

If the system smoothly goes into sliding mode with u, in other words, $s(k) = 0$, then $e_2(k) = -ce_1(k)$. Applied to the error equation of the master–slave system, we have

$$e_1(k+1) = 0.99e_1(k) - 0.01ce_1(k). \tag{13}$$

When the parameter c is selected to satisfy $|0.99 - 0.01c| < 1$, e_1 will converge to zero. Due to system sliding, $s(k) = 0$, so e_2 will also converge to zero. Eventually, e_3 becomes $e_3(k+1) = 0.997e_3(k)$, and will also converge to zero. In this moment, two systems achieve synchronization. In order to let the error function in sliding mode, we used a similar design in [21] for the controller $u(k)$, given as follows:

$$u(k) = -f(\mathrm{k}) - \alpha s(k), \tag{14}$$

where $0 < \alpha < 1$, and

$$\begin{array}{c} f(k) = (0.028 - 0.01c)e_1(k) + (0.01c - 0.001)e_2(k) - 0.001x_{1s}(k)x_{3s}(k) + \\ 0.001x_{1m}(k)x_{3m}(k). \end{array} \tag{15}$$

Computing

$$\begin{array}{c} \triangle s_k = s(k+1) - s(k) = e_2(k+1) + ce_1(k+1) - e_2(k) - ce_1(k) \\ = f(\mathrm{k}) + u(k). \end{array} \tag{16}$$

If $0 < \alpha < 1$, (16) will be transposed to $s(k+1) = (1-\alpha)s(k)$, and $s(k)$ will converge to zero. It is obvious that the parameter α is relative to the convergence speed of the switching function $s(k)$. To test the synchronization control design, the initial condition and parameters were selected as $x_{1m} = 0.1$, $x_{2m} = -0.1$, $x_{3m} = 0.4$, $x_{1s} = 0.2$, $x_{2s} = -0.5$, $x_{3s} = 0.2$, $c = 49$, and $\alpha = 0.5$. The result is presented in Figure 3, showing that e_1 and e_2 converge quickly, and e_3 needs some time to converge.

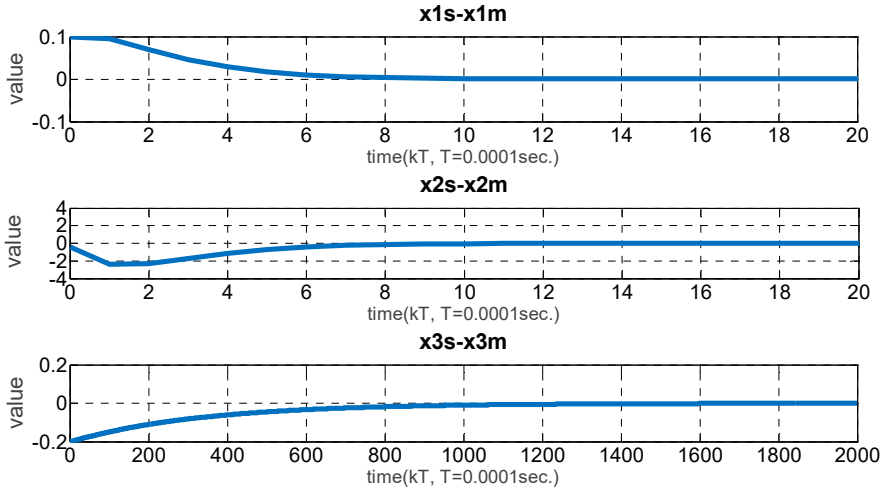

Figure 3. The response of the error function of the master –slave chaos system.

3. Chaos Synchronization Attribution-Based Encryption Scheme

In this section, we give the detailed algorithm of our brand-new system, which combines PF-CP-ABE with chaos synchronization. The synchronization characteristic in PF-CP-ABE is utilized to give another approach to get the secret number. Although PF-CP-ABE greatly reduces the complexity and computation of CP-ABE, if there are a lot of data and they keep being sent, the users should also employ the algorithm every time when receiving the data. This is quite ineffective, so we applied another algorithm to improve it. Due to it combining the concept of ABE with synchronization characteristics of the chaos system, we have given it the name Chaos Synchronization Attribute-Based Encryption (CS-ABE), which can be divided into four parts, as shown below. Set up.

An elliptic curve $E_p(a,b)$ over a finite field $GF(p)$ of order r with generator g is agreed upon. The point g generates a cyclic subgroup in $E_p(a,b)$. In addition, a hash function $\{0,1\}^* \rightarrow Z_r^*$ is defined to map every user's ID to Z_r, and every user's ID is unique, which means $H(ID)_{user1} \neq H(ID)_{user2}$.

The trust authority, which can be a broker, randomly selects an integer $t \in Z_r$ as the authority's main private key and then computes tg to be the main public key. At the same time, the authority randomly selects an integer $k_i \in Z_r$ for every attribute i and computes $PK_i = k_i g$ to be its public key.

The data owner and user hold a chaos master and slave system, respectively, and select α satisfying $|1-\alpha| < 1$; c satisfying $|0.99 - 0.01c| < 1$; and initial conditions such as x_{1m}, x_{2m}, x_{3m}, x_{1s}, x_{2s}, and x_{3s}.

The master data owner hold system is as below:

$$\begin{aligned} x_{1m}(k+1) &= 0.99x_{1m}(k) + 0.01x_{2m}(k) \\ x_{2m}(k+1) &= 0.028x_{1m}(k) + 0.999x_{2m}(k) - 0.001x_{1m}(k)x_{3m}(k) \\ x_{3m}(k+1) &= 0.997x_{3m}(k) + 0.001x_{1m}(k)x_{2m}(k). \end{aligned} \tag{17}$$

The slave data user hold system is as below:

$$\begin{aligned} x_{1s}(k+1) &= 0.99x_{1s}(k) + 0.01x_{2s}(k) \\ x_{2s}(k+1) &= 0.028x_{1s}(k) + 0.999x_{2s}(k) + u(k) \\ x_{3s}(k+1) &= 0.997x_{3s}(k) + 0.001x_{1s}(k)x_{2s}(k). \end{aligned} \tag{18}$$

(1) Key generation

A private key of each attribute (topic name) i is generated for each user ID, and the authority computes the formulas below to get their corresponding private keys and record them.

$$SK_{i,ID} = k_i + H(ID)t \tag{19}$$

(2) Encryption

The plain text is first mapped to a point M on the elliptic curve $E_p(a, b)$. State variables of the chaos system are selected to generate an integer $s \in Z_r$ called a secret number, and then compute

$$C_0 = M + sg. \tag{20}$$

The encryption algorithm is associated with an access structure and does not need any polynomials. The data owner defines the access structure and transforms it to an $n \times l$ LSSS matrix, and then randomly selects a vector $v \in Z_r^l$ with s as its first entry and lets λ_x denote $N_x \cdot v$, where N_x is row x of N. They also randomly select a vector $u \in Z_r^l$ with 0 as its first entry and let ω_x denote $N_x \cdot u$. The ciphertext would be

$$C_{1,x} = \lambda_x g + \omega_x PK_{\rho(x)},\ C_{2,x} = \omega_x g,\ \forall x \tag{21}$$

(3) Decryption

The user transmits their ID and $(C_{2,x}, \rho(x))$ to the authority, and lets the authority verify its identity. If the authority confirms that the user is valid, it secretly sends back a result according to each $(C_{2,x}, \rho(x))$. The result is computed as below:

$$\sum C_{2,x} SK_{\rho(x),ID} = \sum \Big(\omega_x g\Big(k_{\rho(x)} + H(ID)t\Big)\Big) = \sum \Big(\omega_x k_{\rho(x)} g + \omega_x H(ID) t g\Big). \tag{22}$$

With the above result, the user can then compute

$$\begin{aligned} &\textstyle\sum C_{1,x} - \sum C_{2,x} SK_{\rho(x),ID} \\ &= \textstyle\sum \Big(\lambda_x g + \omega_x PK_{\rho(x)}\Big) - \sum \Big(\omega_x k_{\rho(x)} g + \omega_x H(ID) t g\Big) = \sum (\lambda_x g - \omega_x H(ID) t g),\ \forall x. \end{aligned} \tag{23}$$

Then, an integer $c_x \in Z_r$ is selected such that $\sum_x c_x N_x = (1,\ 0,\ 0,\ \ldots,\ 0)$ and computes

$$\sum_x c_x (\lambda_x g - \omega_x H(ID) t g) = sg \tag{24}$$

as $v \cdot (1,\ 0,\ 0,\ \ldots,\ 0) = s$ and $u \cdot (1,\ 0,\ 0,\ \ldots,\ 0) = 0$.

The secret number s is generated from the state variable of the chaos master system, and compared with the state variable of the chaos slave system, such as x_{1s}, x_{2s}, or x_{3s}. If the chaos system does not achieve synchronization, the ciphertext is processed by the above, and the synchronization controller still works. Then, if the chaos system does synchronize, the ciphertext will no longer require the above algorithm, and it can be done by itself, using the same state variable to generate s.

Finally, the user can compute the formula below to get point M and map it back to the plain text on the same elliptic curve.

$$C_0 - sg = M \tag{25}$$

4. Implementation Results

There are four elements in MQTT message transmission, which are the publisher, subscriber, broker, and topic. To simulate resource-constrained devices, we used ESP8266 (ESP-01 and ESP-12F) to act as a publisher and subscriber, respectively, as shown in Figure 4, and raspberry pi 3 to act as the broker, which can also be the authority too. The entire system architecture is shown in Figure 5. ESP-12F and ESP-01 are two wireless IoT devices, and both are equipped with MQTT protocol, a chaos system, and PF-CP-ABE. All the messages are transmitted in an MQTT format, and are passed through the network to everyone. ESP-01 is the data owner (publisher), and it possesses a chaos master system. It needs to transmit all the parameters of chaos synchronization and attributes them to ESP-12F, which possesses a chaos slave system. ESP-01 with a temperature sensor will keep publishing ciphertext of the temperature, and ESP-12F will subscribe, decrypt, and show it on the displayer. There are some

simulation situations where, if the temperature is too high, the alarm led will be lit up, and the switch on ESP-12F will act as a remote controller. All the values and states of devices will be published, and will be analyzed by an open source utility MQTT-SPY, monitoring software on a computer, and the MQTT tool, an IOS application on a cell phone.

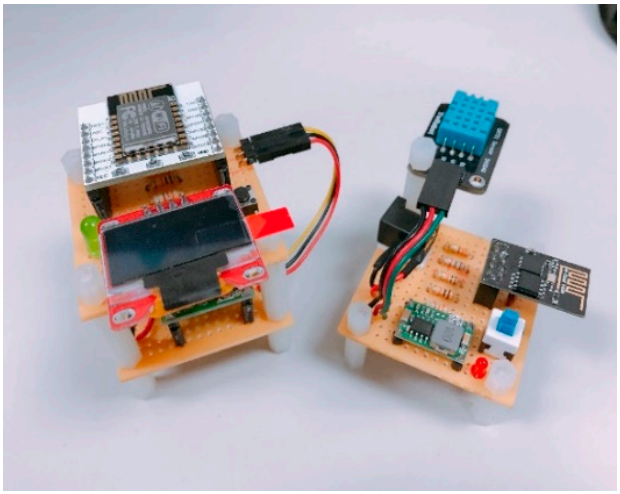

Figure 4. ESP-01 and ESP-12F devices.

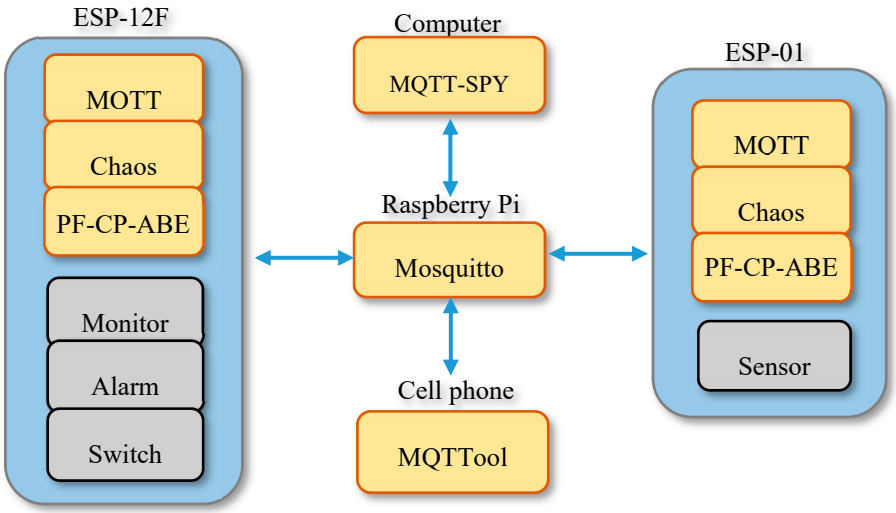

Figure 5. System architecture.

The encryption and decryption of the CS-ABE flow chart are shown in Figures 6 and 7, respectively, where the state variable of the master system also encrypted like M, $x_s = x_{1m} + sg$, and u_m is one of the parameters of the synchronization controller. At first, before chaos synchronization, both ESP-01 and ESP-12F establish the same elliptic curve and chaos system. The data owner, ESP-01, sets an access structure and generates s from its master system, and then publishes all parameters of attributes and synchronization. On the other side, ESP-12F subscribes it and sends its identity to the authority to request a valid result. After the identity is validated and the result is received, ESP-12F can calculate all the values to get the plain text of the temperature, and the synchronization controller still works. After both master and slave systems are synchronized, ESP-01 will stop publishing the parameters of attributes and only publish those of chaos synchronization, as shown in Figure 8. When ESP-12F receives them, it can generate s from its slave system to decrypt the ciphertext.

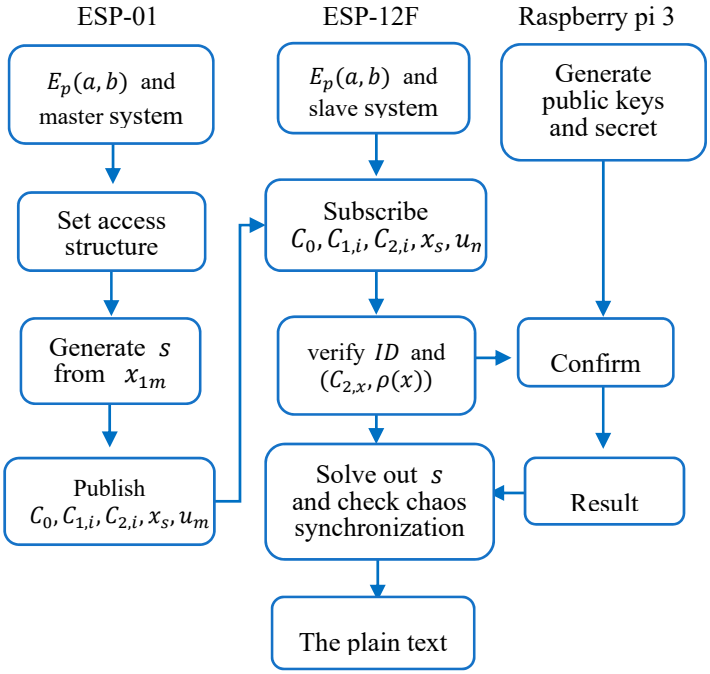

Figure 6. Non-synchronization Chaos Synchronization Attribution-Based Encryption Scheme (CS-ABE) flow chart.

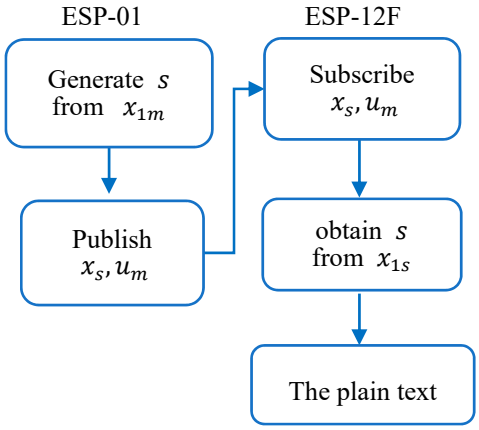

Figure 7. Synchronization CS-ABE flow chart.

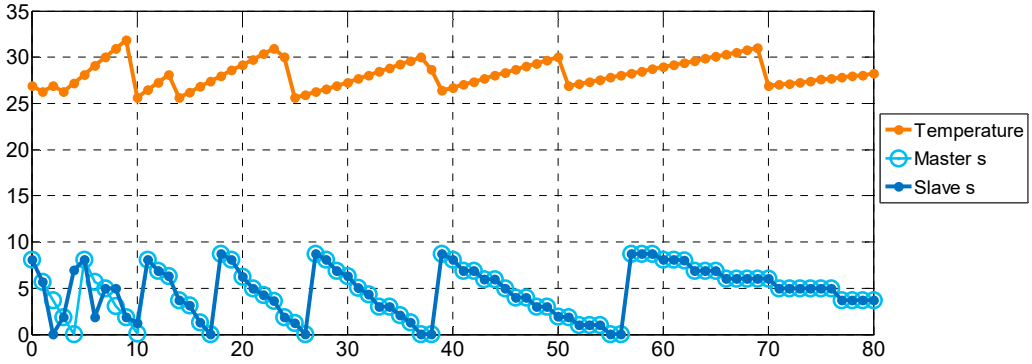

Figure 8. MQTT-SPY analysis.

From the analysis of MQTT-SPY, we recorded 80 data for the original temperature and the secret number *s* in the master and slave system, respectively. Then, we used the plot tool in Matlab to produce a graph, shown in Figure 8. The upper line is the temperature sensed on ESP-01, and the lower two

lines are the secret number s of both the master and slave system used to check that they are the same. From this, we can find out that the secret number is different at the beginning, because chaos systems are not synchronized, and the secret number is solved by the PF-CP-ABE algorithm. After synchronization, the two randomly secret numbers become consistent, which means that the CS-ABE we proposed does work.

5. Performance Analysis

5.1. Data Security

Under the characteristics of ABE, if the user (subscriber) does not meet the attributes set by the data owner (publisher), the user cannot obtain the corresponding private keys. The user meets the conditions and transmits the attribute data to the authority, and the authority then calculates the result related to the data user sent back. After receiving it, the user can correctly resolve the secret number. The plain text is ciphered by the secret number on the elliptic curve, and the secret number is generated by the random chaos signal. It is hard to find out any information in the ciphertext. With LSSS, if you want to get the correct secret number, you must meet the attributes to calculate it. Based on the characteristics of elliptic curves, the difficulty of ECDLP is a well-known problem. Finally, after chaos synchronization, the information about the attributes will not be transmitted and will only be about the synchronization parameters of the chaos system. Due to the chaos system, which is sensitive to initial values, if there is not the same chaos system, the signal will diverge. Then, the eavesdropper cannot get the correct secret number. Therefore, the entire transmitted data is encrypted by it, and it is extremely difficult to identify the plain text.

5.2. Computation

By using chaos synchronization, instead of PF-CP-ABE, to obtain the secret number, the computation can be further reduced. The original encryption means that the user takes their private keys of each attribute, and the secret number can be further calculated. On the contrary, if both Lorenz systems synchronize, the secret number can be generated. Excluding external factors, a comparison of the encryption and decryption times is shown in Figure 9. The horizontal axis is the number of attributes, and the vertical axis is the time (ms). Our system, CS-ABE, could be divided into non-synchronization (Non-syn CS-ABE) and synchronization (Syn CS-ABE). Obviously, the greater the number of attributes, the more time is needed for encryption and decryption. However, for the chaos system, the encryption and decryption times are the same, no matter how great the attributes are. If we use a higher order elliptic curve or a more complex computing technique, both the encryption and decryption times will be even longer. Therefore, the benefit of chaos synchronization determined by the difficulty of elliptic curve calculation.

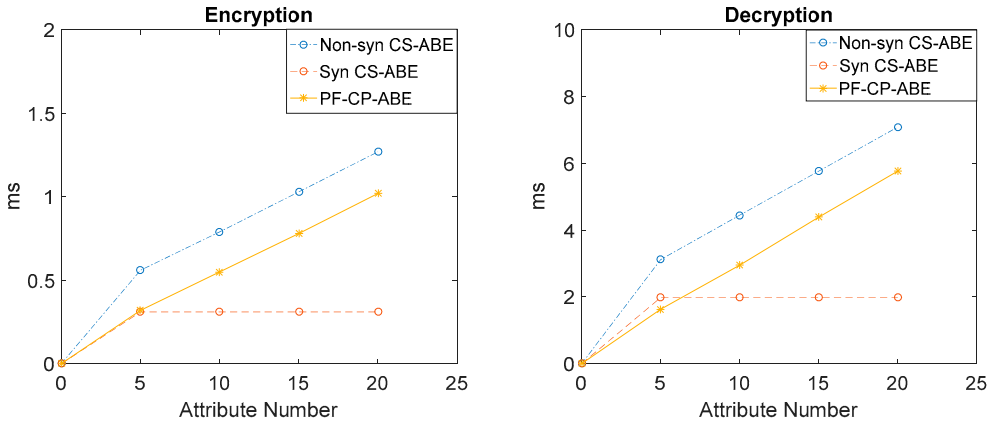

Figure 9. Comparisons of encryption and decryption times.

6. Conclusions

In this paper, we have proposed a new system combining the PF-CP-ABE with chaos synchronization to improve the security of MQTT. Because the original MQTT specification only has TLS/SSL communication encryption, we designed another encryption system especially for resource-constrained devices using MQTT protocol, which offers the devices additional security. Associated MQTT topic names with attributes and the PF-CP-ABE algorithm reduced the computational burden of previous CP-ABE, and the proposed CS-ABE algorithm can be implemented in a resource-constrained device. Combined with chaos synchronization, this gives another way to obtain the secret number. After the chaos systems are synchronized, the secret number is taken from the state variable of the chaos system instead of attribute-based encryption, which skips the mathematical calculation of the elliptic curve. Finally, we made a small size IoT, and designed and simulated a resource-constrained device, equipped with a human–machine interface and monitoring software to show the performance of MQTT communication and the CS-ABE algorithm.

Author Contributions: All authors contributed to the paper. T.-L.L. proposed the research idea, and H.-R.L. and P.-Y.W. wrote the manuscript with supervision from T.-L.L., J.-J.Y. is responsible for the hardware design of the MQTT secure communication system.

Funding: This work was financially supported by the Ministry of Science and Technology, Taiwan, under grant MOST-107-2221-E-167-032-MY2.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Barsocchi, P.; Cassara, P.; Mavilia, F.; Pellegrini, D. Sensing a city's state of health: structural monitoring system by Internet-of-things wireless sensing devices. *IEEE Consum. Electron. Mag.* **2018**, *2*, 22–31. [CrossRef]
2. Barsocchi, P.; Ferro, E.; Fortunati, L.; Mavilia, F.; Palumbo, F. EMS@CNR: An Energy monitoring sensor network infrastructure for in-building location-based services. In Proceedings of the 2014 International Conference on High Performance Computing & Simulation (HPCS), Bologna, Italy, 21–25 July 2014.
3. Bouhdid, B.; Akkari, W.; Belghith, A. A survey on the challenges and opportunities of the internet of things (IoT). In Proceedings of the 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, Tunisia, 30 October–3 November 2017.
4. OASIS. MQTT Version 3.1.1, Plus Errata 01 December 2015. Available online: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html (accessed on 10 December 2015).
5. Bashir, A.; Mir, A.H. Securing Publish-Subscribe Services with Dynamic Security Protocol in MQTT Enabled Internet of Things. *Int. J. Secur. Appl.* **2017**, *11*, 53–66. [CrossRef]
6. Singh, M.; Ma, R.; Vl, S.; Balamuralidhar, P. Secure MQTT for Internet of Things (IoT). In Proceedings of the 2015 Fifth International Conference on Communication Systems and Network Technologies (CSNT), Gwalior, India, 4–6 April 2015.
7. Naik, N. Choice of effective messaging protocols for IoT systems: MQTT CoAP AMQP and HTTP. In Proceedings of the 2017 IEEE International Systems Engineering Symposium, Vienna, Austria, 11–13 October 2017; pp. 1–7.
8. Sahai, A.; Waters, B. Fuzzy Identity Based Encryption. *Adv. Cryptol. Eurocrypt* **2005**, *3494*, 457–473.
9. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 20–23 May 2007; Volume 7, pp. 321–334.
10. Vipul, G.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006.
11. Adiga, B.S.; Rajan, M.A.; Shastry, R.; Shivraj, V.L.; Balamuralidhar, P. Lightweight IBE scheme for Wireless Sensor nodes. In Proceedings of the 2013 IEEE International Conference Advanced Networks and Telecommuncations Systems (ANTS), Kattankulathur, India, 15–18 December 2013; pp. 1–6.

12. Oualha, N.; Nguyen, K.T. Lightweight attribute-based encryption for the Internet of Things. In Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, USA, 1–4 August 2016; pp. 1–6.
13. Ding, S.; Li, C.; Li, H. A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT. *IEEE Access.* **2018**, *6*, 27336–27345. [CrossRef]
14. Bafandehkar, M.; Yasin, S.M.; Mahmod, R.; Hanapi, Z.M. Comparison of ECC and RSA Algorithm in Resource Constrained Devices. In Proceedings of the 2013 International Conference on IT Convergence and Security (ICITCS), Macau, China, 16–18 December 2013; pp. 1–3.
15. Shamir, A. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology*; Springer: Berlin/Heidelberg, Germany, 1985.
16. Beimel, A. *Secure Schemes for Secret Sharing and Key Distribution*; Technion-Israel Institute of Technology, Faculty of Computer Science: Haifa, Israel, 1996.
17. Pecora, L.M.; Carroll, T.L. Synchronization in chaotic systems. *Phys. Rev. Lett.* **1990**, *64*, 821–824. [CrossRef] [PubMed]
18. Chen, Y.Q.; Wang, J.; Cui, S.G.; Deng, B.; Wei, X.L.; Tsang, K.M. Chaos synchronization of coupled neurons via adaptive sliding model control. *Nonlinear Anal.* **2011**, *12*, 3199–3206.
19. Jeong, S.C.; Ji, D.H.; Park, J.H.; Won, S.C. Adaptive synchronization for uncertain chaotic neural networks with mixed time delays using fuzzy disturbance observer. *Appl. Math. Comput.* **2013**, *219*, 5984–5995. [CrossRef]
20. Young, K.D.; Utkin, V.I.; Ozguner, U. A control engineer's guide to sliding mode control. *IEEE Trans. Control. Syst. Technol.* **1999**, *7*, 328–342. [CrossRef]
21. Wan, P.Y.; Liao, T.L.; Yan, J.J.; Tsai, H.H. Discrete sliding mode control for chaos synchronization and its application to an improved El-Gamal cryptosystem. *Symmetry* **2019**, *11*, 843. [CrossRef]