

Received September 21, 2019, accepted September 30, 2019, date of current version October 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946266

Group-Oriented Cryptosystem for Personal Health Records Exchange and Sharing

ZHEN-YU WU 

Department of Information Management, National Penghu University of Science and Technology, Magong 880, Taiwan

e-mail: zywu@gms.npu.edu.tw


ABSTRACT Personal health records (PHRs) have been developed into a type of patient-centered health information exchange model in recent years. It provides users powerful saving, reading, and sharing of medical data. Considering the fullness of current Cloud construction, complicated combination of hospital staff, differences of prioritization between hospital staff and patients, and varied levels of privacy regulation of people in groups or individuals, the difficulty of security exchange and information sharing will increase. Therefore, there is necessity of existence for one flexible and efficient group-oriented cryptosystem. We proposed a bilinear pairing-based group-oriented cryptosystem to overcome above situations. This proposal owns the following advantages: (I) The cryptosystem can simultaneously realize four decryption strategies, enabling receivers to designate appropriate decryptors according to the content of plaintext. (II) All group members need only one private key, which can be used for decryption regardless of the decryption modes. Therefore, errors resulting from the misuse of keys can be avoided, and the difficulty of key management can be reduced. (III) The system is required to disclose only six parameters, thus decreasing spatial complexity. (IV) Regardless of the encryption and decryption modes, receivers must perform encryption only one time, and the length of the ciphertext comprises only four parameters. Thus, the proposed cryptosystem computing (including environment setting and the processes of encryption and decryption) is highly efficient, with easy key management, low spatial complexity, and small amount of ciphertext being transmitted.

INDEX TERMS PHRs, security exchange, group-oriented cryptosystem, key management.

I. INTRODUCTION

With the emergence of cloud computing, most healthcare information technology providers and healthcare service providers have begun to transfer the PHR service to cloud systems. Cloud systems provide storage space and software as a service (SaaS), enabling software service providers to use nearly unlimited and flexible storage space and computing resources [1], [2]. To reduce their operating costs, an increasing number of PHR providers are transferring their PHR applications and data storage services to clouds, instead of establishing a specific data center. For example, Google and Microsoft, the two major cloud platform providers, both provide PHR services on their clouds, namely Google Health and Microsoft HealthVault [3]. PHR investment generally is based on the interest and efficiency-oriented goals of increasing patients' power or improving disease management. However, patients are most concerned about the security and confidentiality of PHR and other healthcare systems. Health Insurance Portability and Accountability

Act (HIPAA) formulated in 1996 outlines the legal protection of PHR privacy and security. However, HIPAA does not address all relevant issues, especially because HIPAA only applies to covered entities including health plans, healthcare clearinghouses, and healthcare providers. Emerging cloud-based PHR service providers like Dossia, Microsoft, and Google are not covered entities [4], [5]. Healthcare Organizations (HCOs) and eHealth services covered by HIPAA confront the problem of implementing powerful and cost-efficient security and privacy policies, while constantly manifesting compliance with HIPAA regulations [6], [7]. Healthcare Organizations (HCOs) and e-health services also face the same problems as HIPAA; they must implement effective and cost-effective security and privacy policies while manifesting their compliance with HIPAA. Thus, related security and privacy policies [8] are also applicable to PHR, that is, patients' information should be protected in accordance with the HIPAA regulations. HCOs must implement comprehensive policies, standards, guidelines, and procedures to protect their healthcare information, including Electronic Health Records (EHR) and Electronic Medical Records (EMR). Although business third groups providing

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai .

PHR solutions are not issue to HIPAA regulations, nonetheless security and privacy for PHRs are significant issues, both for the patients using the PHR and for the providers themselves [9], [10].

Therefore, when introducing PHRs into cloud services, PHR privacy and system security must be carefully evaluated. Although additional security functions such as passwords and record tracking are available for PHRs, in comparison with traditional paper records, storing PHRs in cloud servers causes patients to lose their control over their personal healthcare data. Moreover, cloud systems involve many threats to information privacy, such as the lack of strict and careful verification of user identity, insecure user interface for verification and authorization, abuse of cloud computing for illegal activities, malicious internal employees of cloud service providers, problems related to shared environments, data theft, and service theft [11]. However, HIPAA has not provided favorable legal restrictions on these threats. Therefore, this study asserted that additional procedures must be adopted to provide strong evidence demonstrating that storing these sensitive data on cloud servers is sufficiently safe.

Considering the security of cloud computing environment, the ability of an information system' security mechanism to effectively ensure data confidentiality and adequate data access should be enhanced [12]. In response to the potential risk of privacy disclosure, PHR service providers should be able to encrypt patients' healthcare data, and more importantly, patients (i.e., PHR owners) should be allowed to fully control the medical records they want to share with others. Therefore, in addition to the encryption of medical records by service providers, PHRs should offer the function of group-oriented encryption for individual users in such an environment, which is called group-oriented cryptosystem [13].

A group consists of more than two people. Groups can be categorized as homogeneous, heterogeneous, and hybrid. A group-oriented cryptosystem typically determines the decryption weights for group members according to the importance of the members. That is, a member who plays a crucial role in the group is allocated a large decryption weight. When all members have the same weight, the situation is called homogeneous membership, and the group is called a homogeneous group. If the weights for all members are different, it is heterogeneous membership, and the group is called a heterogeneous group [14].

However, in practice, some groups contain large homogeneous subsets and a few heterogeneous members. For example, in a hospital, members may have dissimilar decryption weights based on their positions. However, nurses who constitute the majority of the members have the same weight. Only physicians or attending physicians, who are the minority in the group, have relatively large weights. Such a group is called the hybrid group, neither homogeneous nor heterogeneous.

Studies on group-oriented cryptosystems have rarely explored the hybrid group environment [15], [16]. In fact,

a hybrid group-oriented cryptosystem with a single decryption strategy cannot meet multiple needs in reality. Apart from the strategy that all legitimate subgroups cooperate to decrypt, the strategy that only a certain access subgroup can decrypt should also be adopted. Take hospital as an example. An access subgroup is a medical team for a patient; the medical team and the patient establish a private communication channel for them to send messages to each other. Other legitimate subgroups cannot read the message regarding medical records. Therefore, complete cooperative decryption should adopt two decryption strategies: any legitimate subgroups can cooperate to decrypt and only designated access subgroups can cooperate to decrypt. Apart from cooperative decryption, hybrid groups also have the need for independent decryption. One approach is that senders send ciphertext that has high confidentiality (e.g., unmentionable disease or personal information) to certain members, and only designated members can decrypt the ciphertext independently. The other approach is that senders send general ciphertext (e.g., appointment information or physicians' outpatient schedules) to all members, and each member can decrypt the ciphertext independently. In response to the need for independent decryption, two decryption strategies are therefore developed: any group member can decrypt ciphertext independently and only designated members can decrypt ciphertext independently. Thus, in the hybrid group environment, a flexible cryptosystem should consider four decryption strategies to realize complete cooperative decryption while taking the need for independent decryption into consideration.

An intuitive approach to satisfy the four decryption strategies in a hybrid group environment is simultaneously adopting multiple independent encryption systems, including group-oriented encryption, broadcast encryption, and public key encryption. However, this approach requires system administrators to maintain three systems simultaneously and group members to store keys from three systems, thus increasing the difficulty of key management. In addition, legitimate receivers must retrieve the corresponding keys to decrypt the ciphertext, which easily leads to errors. Thus, highly complicated systems and encryption and decryption operations are not a favorable solution.

This study proposed a flexible and efficient group-oriented system based on bilinear pairing that enables receivers to flexibly assign a specific member, all members, an access subgroup within a hybrid group, or all legitimate subgroups of a hybrid group as the decryptor according to the content of plaintext. Although the proposed bilinear-pairing-based group-oriented cryptosystem involves four decryption strategies, it requires the system administrator to manage only one system, and all members are only required to store one key. Compared with the aforementioned complex approach, the proposed system can not only reduce system maintenance cost but also enable easy key management and encryption and decryption.

II. PRELIMINARIES

A. ELLIPTIC CURVE CRYPTOSYSTEM

Koblitz [17] and Miller [18] separately proposed elliptic curve cryptosystem (ECC), the security of which is based on the elliptic curve discrete logarithm problem (ECDLP). ECDLP is defined as follows: within a finite field F_p , two points P and Q on the elliptic curve E are given. It is infeasible to determine the integer k to let $Q = kP$. In other words, if k and P are known, calculating Q will be easy. ECDLP has been verified to be an NP-complete problem.

In the real number system, an elliptic curve equation can be defined as a set consisting of points (x, y) in $E : y^2 = x^3 + ax + b$. If $4a^3 + 27b^2 \neq 0$, the set can be a cyclic group. Therefore, in mod P , the elliptic curve is defined as $E : y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \neq 0$. In mod F_{2^n} , the elliptic curve is defined as $E : y^2 + xy = x^3 + ax^2 + b$, where $b \neq 0$. Moreover, any elliptic curve has two characteristics:

- Element O exists, called infinity point.
- $-P = (x, -y)$ is the symmetry point of point $P = (x, y)$ in relation to X -axis; $-P$ is the negative point of point P .
- If $nP = O$ and n is the smallest positive integer, n is the order of point P on the elliptic curve E .

(1) Geometric property of addition:

- O is an addition unit element; therefore, for all $P \in E$, $P + O = O + P = P$, and $P + (-P) = P - P = O$.
- $P + P = 2P$.
- Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, and $P \neq -Q$, then $P + Q = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$,

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q \\ \frac{3x_1^2 + a}{2y_1} & \text{if } P = Q. \end{cases}$$

and

- If $s, t \in F_p$, for any point $P \in E$, $(s + t)P = sP + tP$.

(2) Geometric property of multiplication:

- If $k \in F_p$, for any point $P \in E$, $kP = \overbrace{P + P + \dots + P}^k$.
- If $s, t \in F_p$, for any point $P \in E$, $s(tP) = (st)P$.

ECC is regarded as a cryptosystem with high security because of the following features:

- No effective algorithm can solve the discrete logarithm problem within polynomial time; therefore, the security of ECC is higher than the RSA cryptosystem.
- The private key length in the RSA cryptosystem must be 2048 bits, whereas that in the ECC is only 160 bits to achieve the same level of security. Thus, ECC requires relatively small amount of computation and small space for key storage.

Numerous methods are available for practicing elliptic curves; one is the supersingular elliptic curve method. However, because the system created using a supersingular elliptic curve has characteristics different from those of the

original ECC, in this study, a cryptosystem based on a bilinear pairing function is developed.

B. BILINEAR PAIRING AND RELATED HYPOTHESES

Among the bilinear pairing functions, Weil pairing [19] and Tate pairing [20] are relatively well-known; both are defined as linear mappings between two cyclic groups. Assume G_1 is an addition cyclic group; point P is its generator, and its rank is a prime number q . G_2 is a multiplication cyclic group and its rank is the same with that of G_1 . Therefore, a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ exists, which can map the points of G_1 onto an element of G_2 .

This bilinear map possesses the following features:

- Bilinear: Let points P, Q , and R belong to G_1 , then
 - a. $e(P + Q, R) = e(P, R) \cdot e(Q, R)$.
 - b. $e(P, Q + R) = e(P, Q) \cdot e(P, R)$.
 - c. $e(aP, bP) = e(bP, aP) = e(P, P)^{ab}$, where a and b belong to Z_q^* .
- Non-degeneracy: If P is the generator of G_1 , then $e(P, P)$ is also the generator of G_2 , where $\forall P, Q \in G_1$ and $e(P, Q) \neq 1$.
- Computable: Between any two points $\forall P, Q \in G_1$ an efficient algorithm must exist, which can calculate $e(P, Q)$ within the polynomial time.

In the field of cryptography, to prove system security, some generally acknowledged computation problems must be assumed. This indicates that the probability of solving these problems within the polynomial time is negligible. Hence, system security can be indirectly analyzed through these problems. In particular, the decisional bilinear Diffie-Hellman (DBDH) problem is defined as follows: A generator $P \in G_1$ is randomly selected, and (P, aP, bP, cP, T) is given, where $a, b, c \in Z_q^*$ and $T \in G_2$. It is difficult to determine whether the equality $T = e(P, P)^{abc}$ is accepted.

C. SECRET SHARING

In some environments or systems, the master key owner is not willing to entrust the master key to another person (administrator), not only because the key may be lost but also because this may cause the person to have excessive power; imbalanced power can induce many problems. The simplest method is dividing the original master key into n secondary keys that are kept by n administrators. Each administrator has only one secondary key; only when all administrators gather together can the master key be obtained. Take the bank coffer password as an example. The bank divides the password into 10 segments and gives them to 10 managers; each manager must individually take care of one segment. When the client intends to open the coffer, all the 10 managers have to gather together to open the coffer cooperatively. This method is to increase the security of the master key and, particularly in a group environment, to prevent the private key of a group from being used by malicious group members for private purpose. However, this method is inefficient. Shamir [21] and Blakley [22] proposed a threshold secret sharing method, which is

more efficient than the aforementioned method. Shamir's method is based on the Lagrange interpolating polynomial, whereas Blakley's method is based on linear geometric projection. In addition to the two methods, numerous methods are also available [23], [24], among which interpolating polynomial has been most widely discussed.

The threshold and generalized secret sharing systems both adopt the Lagrange interpolating polynomial. The generalized secret sharing system is more flexible and can clearly determine the legitimate subgroup, instead of being limited by the threshold value; however, it requires additional calculation and disclosure of the open value corresponding to each legitimate subgroup. Because the system calculation is highly complex, it requires relatively large parameter space.

Threshold value is the most crucial parameter in the threshold method. Generally, (t, n) is used, where t denotes the threshold value and n denotes the total number of people. This method involves two roles: master key holder and shadow holder. The master key holder divides the master key s into n secondary keys and secretly sends them to the shadow holders; each shadow holder possesses only one secondary key. Two situations may occur when the master key is reconstructed:

- (1) The master key can be derived when the number of secondary keys is larger than or equal to the threshold value t .
- (2) The master key cannot be derived when the number of secondary keys is smaller than the threshold value t .

A threshold secret sharing system involves two stages: key sharing and key recovery.

(1) Key sharing stage:

- The system administrator selected the master key s . Assume group members to be $\{U_1, U_2, \dots, U_n\}$, and the threshold value is represented by t .
- Let all members possess a public and sole identity ID_i , and incorporate ID_i into the polynomial to obtain $s_i = f(ID_i)$, which serves as each member's secondary key. (ID_i, s_i) is the coordinates of point of the polynomial on a two-dimensional space. Because $f(x)$ is a polynomial of degree 2 , it can be solely determined by any t or more coordinates of points. If the number of points is less than t , the polynomial cannot be solely determined.

(2) Key recovery stage:

- Assume that the number of people the system administrator intends to cooperate to reconstruct the master key reaches the threshold value, namely $\{U_1, U_2, \dots, U_t\}$. The group members separately provide secondary keys to $t-1$ participants.
- When $U_i \in \{U_1, U_2, \dots, U_t\}$ receives all secondary keys, the Lagrange interpolating polynomial is used to reconstruct the polynomial as follows:

$$f(x) = \sum_{i=1}^t f(ID_i) \prod_{j=1, j \neq i}^t \frac{x - ID_j}{ID_i - ID_j} \pmod{p}$$

The master key can also be reconstructed as follows:

$$s = f(0) = \sum_{i=1}^t f(ID_i) \prod_{j=1, j \neq i}^t \frac{ID_i}{ID_i - ID_j} \pmod{p}$$

The threshold method can achieve the goal of secret sharing, but it is a special method. Each member owns only one secondary key. The master key can be recovered when a certain number of secondary keys are collected, that is, the recovery of master key depends on the number of people who cooperate, instead of the identity of the people. Therefore, Ito *et al.* [25] in 1987 proposed as generalized secret sharing method. Specifically, logistic equation Γ is used to explicitly share the master key s . For example, if group members are $\{A, B, C, D\}$, and $\Gamma = (AB) \cup (BC) \cup (ACD)$, this indicates that there are three cooperative subgroups $\{AB\}$, $\{BC\}$, and $\{ACD\}$, where the master secret (i.e., master key) can be recovered through the cooperation between only members A and B .

A generalized secret sharing system involves key sharing and recovery stages.

1) KEY SHARING STAGE

- The system administrator selects the master key s . Assume group members to be $\{A, B, C, D\}$ and logistic equation to be $\Gamma = (AB) \cup (BC) \cup (ACD)$.
- Let the identities of group members be $\{ID_1, ID_2, ID_3, ID_4\}$. The system selects four random numbers k_1, k_2, k_3 , and k_4 from Z_p^* to be the members' secondary keys.
- The public values of the legitimate subgroups of Γ are calculated. Take $\{AB\}$ as an example. Two coordinates of points on the two-dimensional space are (ID_1, k_1) and (ID_2, k_2) . Incorporating (ID_1, k_1) and (ID_2, k_2) into the Lagrange interpolating polynomial can obtain the only second degree polynomial $f_1(x)$. $V_{AB} = s - f_1(0) \pmod{p}$ is calculated, and the obtained value is the public value corresponding to $\{AB\}$.

2) KEY RECOVERY STAGE

- Assume $\{AB\}$ to be the master key that requires cooperative reconstruction. $\{AB\}$ separately provides k_1 and k_2 to each other.
- For members A and B , two sets of coordinates for points can be obtained, namely (ID_1, k_1) and (ID_2, k_2) , and the Lagrange interpolating polynomial is used to obtain $f_1(0)$.
- The master key is successfully recovered as follows: $s = V_{AB} + f_1(0) \pmod{p}$.

III. GROUP-ORIENTED CRYPTOSYSTEM BASED ON BILINEAR PAIRING

A. PROPOSED SCHEME

In a group environment, different encryption and decryption demands exist according to the level of importance of documents, which lead to the development of four encryption and decryption models. Examples are as follows:

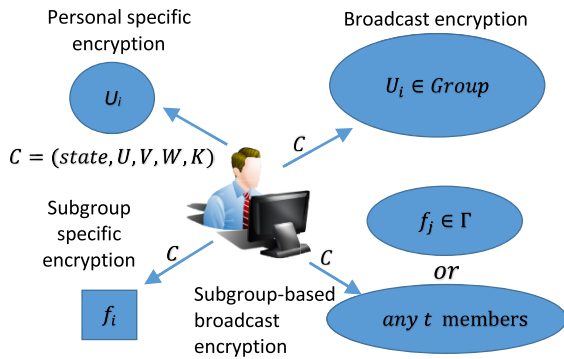


FIGURE 1. Bilinear pairing-based group-oriented cryptosystem.

- (1) If the plaintext comprises letters sent between individual people, the appropriate model is that the letters are encrypted for a specific person, which is called personal specific encryption.
- (2) If the plaintext is company annual plans, the suitable model is broadcast encryption, which allows the encrypted content to be broadcasted to all members.
- (3) If the plaintext consists of letters sent by a person to a specific subgroup, the suitable model is subgroup specific encryption, where the letters are encrypted and sent to the specific access subgroup.
- (4) If the plaintext comprises important orders, the suitable model is subgroup-based broadcast encryption, which allows the encrypted content to be broadcasted to all legitimate subgroups. In this model, the legitimate subgroup includes all access subgroups or any subgroup consisting of t people.

Models 1 and 2 adopt independent decryption, whereas Models 3 and 4 adopt cooperative decryption. Models 1 and 2 differ in whether the receivers are designated, and Models 3 and 4 differ in whether the access subgroups are designated. In particular, Model 4 is the hybrid group-oriented cryptosystem.

The group-oriented cryptosystem based on bilinear pairing proposed in this section is capable of effectively meeting the demands in the aforementioned environments. The system architecture is shown in Fig. 1.

B. FUNCTION DEFINITION

Three algorithms were proposed in this study: system setup algorithm (Setup), encryption algorithm (Enc), and decryption algorithm (Dec).

- Setup(k, n, t): Security parameter (k), total number of members (n), and threshold value (t) are input in the algorithm, and assume all members in the group to be $\{u_1, u_2, \dots, u_n\}$. The algorithm will output (PK, VK, SK) . Specifically, PK denotes all the parameters in the system; $VK = (Y_1, Y_2, \dots, Y_n)$, denoting the set of all members' public keys; $SK = (x_1, x_2, \dots, x_n)$, representing the set of all members' private keys. Member u_i will secretly obtain private key x_i . This algorithm

also outputs the decryption strategy used by the group, including threshold value and access structure.

- Enc($state, PK, M$): State value ($state \in \{1, 2, 3, 4\}$), public key set (PK), and plaintext to be encrypted (M) are input in the algorithm. This encryption algorithm will output ciphertext C . The state value represents the encryption and decryption model to be adopted, explained as follows:
 - a. If $state = 1$, the receiver is a certain member.
 - b. If $state = 2$, the receiver is all members.
 - c. If $state = 3$, the receiver is a certain access subgroup.
 - d. If $state = 4$, the receiver is all legitimate subgroup.
- Dec(SK, C): The private key set (SK) and ciphertext to be decrypted (C) are input in the algorithm. This decryption algorithm will output plaintext M .

C. SYSTEM PROCEDURE

The environment parameters set by the system are as follows:

- Three random numbers x, y , and z are selected from Z_q^* , where x denotes the private key of the group.
- System parameters are calculated: $Q_1 = xP, Q_2 = yP$, and $Q_3 = zP$, where Q_1 denotes the public key of the group.

1) KEY GENERATION

The pairs of keys for all group members and all access subgroups are generated.

- Random numbers a_1, \dots, a_{t-1} are selected from Z_q^* , and a polynomial of degree t is defined as follows: $f(x) = x + \sum_{i=1}^{t-1} a_i x^i$.
- $x_i = xz + yf(i)$ is calculated to be private key of u_i , and $Y_i = f(i)P$ is its public key. Through the security channel, the private key is sent to u_i , and its public key is disclosed. $(x_i, Y_i) \in (Z_q^*, G_1)$ is the key pair of u_i , where $i = 1, 2, \dots, n$. An equality relationship exists between group members' key pairs and system parameters, namely $e(x_i P, P) = e(Q_1, Q_3)e(Y_i, Q_2)$. Thus, the public $VK = (Y_1, Y_2, \dots, Y_n)$ and private $SK = (x_1, x_2, \dots, x_n)$ are obtained.
- Random number r_j is selected from Z_q^* . $Z_{f_j} = xQ_2 + zr_j \sum_{u_i \in f_j} x_i P$ and $Z'_{f_j} = r_j P$ are calculated to be the two public values of f_j , where $j = 1, 2, \dots, s$. An equality relationship exists between the two public values and the system parameters, namely $e(Z_{f_j}, P) = e(Q_1, Q_2) \prod_{u_i \in f_j} e(x_i Q_3, Z'_{f_j})$. Finally, a public system parameter is obtained as follows: $PK = (P, Q_1, Q_2, Q_3, \Gamma, \{(Z_{f_j}, Z'_{f_j}) | f_j \in \Gamma\})$.

2) ENCRYPTION

One of the following encryption processes is adopted according to the state values.

- If $state = 1$ and the receiver is u_i , the implementation steps are as follows:
 - a. Select random number s from Z_q^* .
 - b. Calculate:
 $U = H(e(Q_1, Q_3)^s) \oplus M, V = sY_i, W = sP$ and $K = sG(U, V, W)$. That is, $Enc(1, PK, M) = C = (1, U, V, W, K)$.
- If $state = 2$, the implementation steps are as follows:
 - a) Select random number s from Z_q^* .
 - b) Calculate
 $U = H(e(Q_1, Q_3)^s) \oplus M, V = sQ_2, W = sP$ and $K = sG(U, V, W)$. That is, $Enc(2, PK, M) = C = (2, U, V, W, K)$.
- If $state = 3$ and the receiver is access group f_j , the implementation steps are as follows:
 - a) Select random number s from Z_q^* .
 - b) Calculate
 $U = H(e(Q_1, Q_2)^s) \oplus M, V = sZ_{f_j}, W = sQ_3$ and $K = sG(U, V, W)$. That is, $Enc(3, PK, M) = C = (3, U, V, W, K)$.
- If $state = 4$, the implementation steps are as follows:
 - a. Select random number s from Z_q^* .
 - b. Calculate
 $U = H(e(Q_1, Q_2)^s) \oplus M, V = sP, W = sQ_3$ and $K = sG(U, V, W)$. That is, $Enc(4, PK, M) = C = (4, U, V, W, K)$.

3) DECRYPTION

Once the receiver receives the ciphertext C , he or she adopts one of the following decryption processes according to the state value in C .

- If $state = 1$, only the designated receiver u_i is allowed to decrypt the ciphertext. The decryption steps are as follows:
 - a. Whether the equation $e(K, P) \stackrel{?}{=} e(G(U, V, W), W)$ is true is first verified to determine whether the ciphertext format is legitimate. If the equation is false, the decryption process stops.
 - b. $M = U \oplus H(\frac{e(x_i P, W)}{e(V, Q_2)})$ is calculated to successfully recover M .

The correctness of the accuracy of the decryption process is as follows:

$$\begin{aligned}
 & U \oplus H\left(\frac{e(x_i P, W)}{e(V, Q_2)}\right) \\
 &= U \oplus H\left(\frac{e(xz + yf(i)P, P)^s}{e(Y_i, Q_2)^s}\right) \\
 &= U \oplus H\left(\frac{e(xP, zP)^s e(f(i)P, yP)^s}{e(Y_i, Q_2)^s}\right) \\
 &= U \oplus H(e(Q_1, Q_3)^s) = M
 \end{aligned}$$

- If $state = 2$, any member $u_i \in Group$ is allowed to decrypt the ciphertext. The steps are as follows:
 - a. Whether $e(K, P) \stackrel{?}{=} e(G(U, V, W), W)$ is true is verified to determine whether the ciphertext

format is legitimate. If the equation is false, the decryption process stops.

- b. $M = U \oplus H(\frac{e(x_i P, W)}{e(Y_i, V)})$ is calculated to successfully recover M .

The correctness of the decryption process is verified as follows:

$$\begin{aligned}
 U \oplus H\left(\frac{e(x_i P, W)}{e(Y_i, V)}\right) &= U \oplus H\left(\frac{e(xz + yf(i)P, P)^s}{e(Y_i, Q_2)^s}\right) \\
 &= U \oplus H\left(\frac{e(xP, zP)^s e(f(i)P, yP)^s}{e(Y_i, Q_2)^s}\right) = M
 \end{aligned}$$

- If $state = 3$, only the designated f_j can cooperate to decrypt. The steps are as follows:
 - a. Whether $e(K, Q_3) \stackrel{?}{=} e(G(U, V, W), W)$ is true is first verified to determine whether the ciphertext format is legitimate. If the equation is false, the decryption process stops.
 - b. $u_i \in f_j$ calculates the decryption shadow $c_i = e(x_i W, Z'_{f_j})$ and secretly sends the decryption shadow to members intending to decrypt the ciphertext.
 - c. After receiving all correct decryption shadows, $M = U \oplus H(\frac{e(V, P)}{\prod_{u_i \in f_j} c_i})$ is calculated to successfully recover M .

The correctness of the decryption process is verified as follows:

$$\begin{aligned}
 & U \oplus H\left(\frac{e(V, P)}{\prod_{u_i \in f_j} c_i}\right) \\
 &= U \oplus H\left(\frac{e(Z'_{f_j}, P)^s}{\prod_{u_i \in f_j} e(x_i W, Z'_{f_j})}\right) \\
 &= U \oplus H\left(\frac{e(xQ_2 + zr_j \sum_{u_i \in f_j} x_i P, P)^s}{\prod_{u_i \in f_j} e(x_i Q_3, Z'_{f_j})^s}\right) \\
 &= U \oplus H\left(\frac{e(Q_1, Q_2)^s \prod_{u_i \in f_j} e(x_i Q_3, Z'_{f_j})^s}{\prod_{u_i \in f_j} e(x_i Q_3, Z'_{f_j})^s}\right) \\
 &= U \oplus H(e(Q_1, Q_2)^s) = M
 \end{aligned}$$

- If $state = 4$, any legitimate subgroup can cooperate to decrypt, as in the hybrid group-oriented cryptosystem. Two situations may occur:

Situation 1: Assume all members of a legitimate f_j intend to decrypt the ciphertext, the process is as follows:

- a. Whether the equation $e(K, Q_3) \stackrel{?}{=} e(G(U, V, W), W)$ is true is verified to determine whether the ciphertext format is legitimate. If the equation is false, the decryption process stops.
- b. $u_i \in f_j$ calculates the decryption shadow $c_i = e(x_i W, Z'_{f_j})$ and secretly sends the decryption shadow to other members.

- c. Once all correct decryption shadows are received, the ciphertext is decrypted to recover $M = U \oplus H\left(\frac{e(Z_{f_j}, V)}{\prod_{u_i \in f_j} c_i}\right)$.

The correctness of the decryption process is verified as follows:

$$\begin{aligned} U \oplus H\left(\frac{e(Z_{f_j}, V)}{\prod_{u_i \in f_j} c_i}\right) &= U \oplus H\left(\frac{e(Z_{f_j}, P)^s}{\prod_{u_i \in f_j} e(x_i W, Z'_{f_j})}\right) \\ &= U \oplus H\left(\frac{e(xQ_2 + zr_j \sum_{u_i \in f_j} x_i P, P)^s}{\prod_{u_i \in f_j} e(x_i Q_3, Z'_{f_j})^s}\right) \\ &= U \oplus H\left(\frac{e(Q_1, Q_2)^s \prod_{u_i \in f_j} e(x_i Q_3, Z'_{f_j})^s}{\prod_{u_i \in f_j} e(x_i Q_3, Z'_{f_j})^s}\right) \\ &= U \oplus H(e(Q_1, Q_2)^s) = M \end{aligned}$$

Situation 2: Assume member u_i in the set $A = \{u_1, u_2, \dots, u_t\}$ intends to decrypt the cipher text, the process is as follows:

- Whether $e(K, Q_3) \stackrel{?}{=} e(G(U, V, W), W)$ is true is verified to determine whether the ciphertext format is legitimate. If the equation is false, the decryption process stops.
- $u_i \in A$ calculates decryption shadow $d_i = \left(\frac{e(x_i P, V)}{e(Q_1, W)}\right)^{\lambda_i}$, where λ_i denotes the Lagrange coefficient, and secretly sends it to other members who also send decryption shadows.
- Once all correct decryption shadows have been received, $M = U \oplus H\left(\prod_{u_i \in A} d_i\right)$ can be recovered by decrypting the ciphertext.

The correctness of the decryption process is verified as follows:

$$\begin{aligned} U \oplus H\left(\prod_{i=1}^t d_i\right) &= U \oplus H\left(\prod_{i=1}^t \left(\frac{e(x_i P, V)}{e(Q_1, W)}\right)^{\lambda_i}\right) \\ &= U \oplus H\left(\prod_{i=1}^t e(Y_i, Q_2)^{s\lambda_i}\right) \\ &= U \oplus H\left(e\left(\sum_{i=1}^t \lambda_i f(i) P, Q_2\right)^s\right) \\ &= U \oplus H(e(Q_1, Q_2)^s) = M \end{aligned}$$

IV. SECURITY PROOF

The concept of problem reduction can be applied to transfer the difficulty of cracking the bilinear pairing-based generalized threshold cryptosystem and hybrid group-oriented cryptosystem to cracking the bilinear pairing-based group-oriented cryptosystem. To reduce the space used for the security proof, in this paper, only the security of the bilinear pairing-based group-oriented cryptosystem is demonstrated (assume DBDH to be a problem). This chapter first describes the basic concepts related to security proof and then verifies the security of the bilinear pairing-based group-oriented cryptosystem.

A. BASIC CONCEPTS

Before the system security analysis, four cryptanalyses are introduced as follows:

- 1) Ciphertext-only attack (COA): The adversary intercepts certain ciphertexts and intends to decrypt them to obtain the plaintexts.
- 2) Known-plaintext attack: The adversary can intercept some plaintext-ciphertext pairs $(m_1, c_1), (m_2, c_2), \dots, (m_n, c_n)$ and intends to decrypt ciphertext c_{n+1} to obtain the corresponding plaintext m_{n+1} .
- 3) Chosen-plaintext attack: The adversary actively chooses the plaintexts to be encrypted m_1, m_2, \dots, m_n , and the system sends back the corresponding ciphertexts c_1, c_2, \dots, c_n , based on which attacks can be performed.
- 4) Chosen-ciphertext attack (CCA): The adversary actively chooses the ciphertexts c_1, c_2, \dots, c_n , and the system sends back the corresponding plaintexts m_1, m_2, \dots, m_n , based on which attacks can be performed.

The adversary's ability in the COA is most limited; on the contrary, the adversary is able to exercise the strongest ability in the CCA. Therefore, if system A claims to be able to resist COA and system B claims to be able to resist CCA, this indicates system B has higher security level than system A. As the computing capability of computers is improving exponentially, the developed cryptosystems are expected to be able to resist CCA.

Blum and Micali [26] proposed a definition of cryptosystem security in 1984, where the roles of adversary and simulator exist. The interaction between adversary and simulator is as follows:

- 1) Challenge: The adversary chooses a pair of two plaintexts of the same length (m_0, m_1) to the simulator. When receiving the plaintexts, the simulator tosses a fair coin $b \in \{0, 1\}$ and encrypts m_b to obtain the ciphertext. The ciphertext is then sent back to the adversary.
- 2) Guess: The adversary must guess the value of $b' \in \{0, 1\}$. If $b' = b$, the adversary can successfully crack the system.

Subsequently, the CCA-secure (semantically secure) attack scenario was defined. The scenario also involves an adversary and a simulator, with five phases from the simulator setting up the system environment to the adversary making responses.

- 1) Setup: The simulator sets up system parameters.
- 2) Phase 1: The adversary can choose the ciphertext c_i to be decrypted in a limited number of times, and the simulator sends back the corresponding plaintext m_i .
- 3) Challenge: The adversary chooses a pair of plaintexts of the same length (m_0, m_1) and sends them to the simulator, and the corresponding ciphertext should not appear in Phase 1. After receiving the plaintexts and tossing a fair coin $b \in \{0, 1\}$, the simulator encrypts m_b and sends the ciphertext back to the adversary.

- 4) Phase 2: Phase 2 is similar to Phase 1, but the ciphertext to be decrypted should not be the ciphertext sent back by the simulator at the Challenge phase.
- 5) Response: The adversary guesses the value of $b' \in \{0, 1\}$. If $b' = b$, the adversary can successfully crack the cryptosystem. The probability that the adversary can crack the system was rigorously defined as follows:

$$AdvCCA_{n,t,k}(A) = \left| \Pr[b' = b] - \frac{1}{2} \right|$$

Random oracle model was proposed by Bellare and Rogaway [27] in 1993 and is often used to prove system security. In the random oracle model, a “perfect” hash function is assumed to be similar to the random number generator. A random oracle can be regarded as a perfect hash function; therefore, the random oracle model is only applicable to cryptosystems that adopt hash functions. The model is based on an abstract idealization of hash functions; a perfect hash function cannot be actually designed in practice.

The output values of a random oracle are defined as follows: An input value (query) x is given. If the query x has been submitted, the random oracle responds with the same output value as last time. If the random oracle has not received x before, it randomly responds an output value, which should conform to a uniform distribution.

All roles (e.g., decryptor and signer) in the cryptosystem should be ensured to be able to use the random oracle, which is regarded as an open, available resource. In the scenario of simulated attack, the simulator simulates the operation of a random oracle. The simulator interacts with the adversary without knowing the secrets in the cryptosystem and prevents the adversary from noticing that who he or she interacts with is the simulator instead of the cryptosystem. Finally, the strategy of problem reduction is applied to explain that the difficulty of cracking this cryptosystem is similar to that of solving the problems recognized in the number theory. In other words, if the adversary has high probability to crack the system, the simulator can use the adversary’s ability to solve the recognized problem. However, the relevant mathematical problems cannot be solved, thus leading to contradiction. Therefore, the original hypothesis is rejected, that is, the adversary is unable to crack the system.

B. SECURITY PROOF

This study demonstrated that the DBDH problem [9], [10] can be reduced to the problem of the “group-oriented cryptosystem based on bilinear pairing.” The two problems have the same level of difficulty. Because DBDH is a known hypothesis problem in the number theory, it indirectly demonstrates that the bilinear pairing-based group-oriented cryptosystem is semantically secure. In addition, the attack scenario must be limited to the model of selective strategy, which is similar to the selective-ID model, meaning that the adversary must

decide and output the model he or she intends to attack ($state^* \in \{1, 2, 3, 4\}$) before issuing an attack.

For the purpose of convenience, BPGOC was subsequently used to represent the bilinear pairing-based group-oriented cryptosystem.

Theorem 1: Assume an adversary (A) submits query to the random oracle H q_H times, to the random oracle G q_G times, and to the decryption share oracles q_D times within the time period of t_{CCA} . The probability of cracking the BPGOC system is $AdvCCA_{n,t,k}(A)$ and is negligible. The BPGOC can be regarded as (t_{CCA}, q_H, q_G, q_D) CCA-secure.

Proof: Assume that the BPGOC A intends to crack is claimed to be (t_{CCA}, q_H, q_G, q_D) CCA-secure. The researcher constructs another adversary B and assumes the probability of B solving the DBDH problem to be $\epsilon(k)$, then the probability of A cracking BPGOC is $AdvCCA_{n,t,k}(A) > \epsilon(k)$. Let P be the generator of DBDH, and B receives a random parameter (P, aP, bP, cP, T) sent from DBDH. After the interaction between A and B , the output value is determined by using the ability of A . The interaction process comprises five phases as follows:

Init: A outputs the target strategy, represented by $state^*$. If $state^* = 1$, then A must output the target user’s ID, which is represented by u_i^* . If $state^* = 3$, then A must output the ID of the access subgroup he or she intends to attack, which is represented by f_j^* .

Setup: The process in which B generates parameters (PK, VK, SK) is as follows:

- a. If $state^* = 1$, the parameter was set to be $(Q_1, Q_2, Q_3) = (aP, bP, cP)$. If $state^* = 2$, a random number γ is selected; let $(Q_1, Q_2, Q_3) = (aP, \gamma P, cP)$, otherwise a random number γ is selected; let $(Q_1, Q_2, Q_3) = (aP, bP, \gamma P)$.
- b. Assume $S = \{u_1, u_2, \dots, u_{t-1}\}$, and if $state^* = 1$, the set S must contain u_i^* . A random number $\{f(i)|u_i \in S\}$ is selected to be their private key, the corresponding public key is calculated as $Y_i = f(i)P$. Subsequently, let $\lambda_0, \lambda_1, \dots, \lambda_{t-1}$ be the Lagrange coefficient. Other members’ public key is calculated as $Y_j = \lambda_0 Q_1 + \sum_{u_i \in S} \lambda_i Y_i$. Finally, all group members’ public key $VK = \{Y_i|u_i \in S\} \cup \{Y_j|u_j \notin S\}$ is sent to A .
- c. Two random numbers γ_j and γ'_j are selected, and the two public values of f_j are calculated as $Z_{f_j} = \gamma_j P$ and $Z'_{f_j} = \gamma'_j P$, where $1 \leq j \leq s$. All system parameters: $PK = (P, Q_1, Q_2, Q_3, \Gamma, \{(Z_{f_j}, Z'_{f_j})|f_j \in \Gamma\})$ are sent to A .

Phase 1: H and G are random oracles. When a query is sent to H regarding the output value of $Q \in G_2$, a character string $H \in \{0, 1\}^l$ is randomly selected as the response to be sent back to A . HList is used to establish a random oracle H that contains two fields $\langle query, answer \rangle$; in other words, if $H = H(Q)$, then $\langle Q, H \rangle$ can be found in one row of HList. Similarly, if a query is sent to G regarding the output

value of $(U, V, W) \in (\{0, 1\}^l, G_1, G_1)$, a random number t is selected, and $G = tQ_1$ is sent back to A as a response. GList is used to construct a random oracle G , and GList also contains two fields $\langle \text{query}, \text{answer} \rangle$. In other words, if $G = G(U, V, W)$, then one row of GList is presented as $\langle (U, V, W), G \rangle$.

Subsequently, the operation of decryption share oracle is introduced as follows. A sends a decryption query $C = (\text{state}, U, V, W, K)$ to B , where $\text{state} \in \{1, 2, 3, 4\}$ and $\text{state} \neq \text{state}^*$. When receiving the decryption query, B first determines whether $\langle (U, V, W), G \rangle$ exists in GList. If not, B rejects the query; if yes, B implements the following steps:

- Retrieve $\langle (U, V, W), G \rangle$ from GList to obtain the G point that corresponds to the ciphertext.
- When $\text{state}^* = 1$ or $\text{state}^* = 2$, determine whether $e(K, P) = e(G, W)$ is true, otherwise determine whether $e(K, Q_3) = e(G, W)$ is true.
If the equation is true, then $R = \frac{1}{t}K$ (note: $\frac{1}{t}K = sQ_1$);
If the equation is false, then reject the query.
- If $\text{state}^* = 1$ or $\text{state}^* = 2$, calculate $Q = e(R, Q_3)$, otherwise calculate $Q = e(R, Q_2)$.
- Retrieve $\langle Q, H \rangle$ from HList and respond $M = U \oplus H$ to A , thereby completing the decryption.

Challenge: A randomly selects two plaintexts of the same length (M_0, M_1) and sends them to B ; B generates a ciphertext and sends it back to A . The process is as follows: First, $b \in \{0, 1\}$ is obtained by randomly generating a number, based on which a plaintext M_b is selected. Subsequently, the target strategy is used to encrypt the plaintext to obtain the target ciphertext $C^* = (\text{state}^*, U^*, V^*, W^*, K^*)$ that corresponds to M_b . The target ciphertext is then sent back to A .

- If $\text{state}^* = 1$ and the target user is u_i^* , then let $U^* = H(T) \oplus M_b$, $V^* = f(i)Q_2$, and $W^* = Q_2$. Subsequently, a random number $k \in Z_q^*$ is selected, and let $G^* = G(U^*, V^*, W^*) = kP$ and $K^* = kQ_2$. Consequently, $C^* = (1, H(T) \oplus M_b, f(i)Q_2, Q_2, kQ_2)$ is generated.
- If $\text{state}^* = 2$, then let $U^* = H(T) \oplus M_b$, $V^* = \gamma Q_2$, and $W^* = Q_2$. A random number $k \in Z_q^*$ is selected, and let $G^* = G(U^*, V^*, W^*) = kP$ and $K^* = kQ_2$. Finally, $C^* = (2, H(T) \oplus M_b, \gamma Q_2, Q_2, kQ_2)$ is generated.
- If $\text{state}^* = 3$ and the access subgroup A intends to attack is f_j^* , then let $U^* = H(T) \oplus M_b$, $V^* = \gamma_j Q_3$, and $W^* = \gamma Q_3$. A random number $k \in Z_q^*$ is selected, and let $G^* = G(U^*, V^*, W^*) = kP$ and $K^* = kQ_3$. Therefore, $C^* = (3, H(T) \oplus M_b, f(i)Q_2, Q_2, kQ_2)$ is generated.
- If $\text{state}^* = 4$, then let $U^* = H(T) \oplus M_b$, $V^* = Q_3$, and $W^* = \gamma Q_3$. A random number $k \in Z_q^*$ is selected, and let $G^* = G(U^*, V^*, W^*) = kP$ and $K^* = kQ_3$. Therefore, $C^* = (4, H(T) \oplus M_b, Q_3, \gamma Q_3, kQ_3)$ is generated.

TABLE 1. Comparison table with existing schemes.

	(1)	(2)	(3)	(4)	(5)
Du et al. (2010) [28]	$O(n)$	$3+2a$	$(n+1)M+$ $(3+2n)E+1BP$	$(n)M+$ $(n+2)E+2BP$	1
Phan et al. (2013) [29]	$O(n)$	2	$(n+1)M+$ $4E+1BP+1H$	$(2n+2)M+$ $2E+2BP+1H$	1
Koti and Purushothama (2016) [13]	$O(1)$	3	$1M+3E+1BP$	$6M+3E+$ $3BP$	1
The proposal (2019)	$O(1)$	4	$3M+1BP+2H$ $+1XOR$	$4BP+1H+$ $1XOR$	4

(1) key storage cost; (2) Length of ciphertext; (3) encryption computation cost; (4) decryption computation cost; (5) supported modes.
H: computation cost of a hash function
M: multiplication computation cost
E: exponentiation computation cost
BP: bilinear pairing computation cost
XOR: computation cost of XOR operation
 n is the number of users in the schemes
 a is the number of ciphertext attributes used in [25] scheme

Phase 2: Phase 2 is similar to Phase 1. However, the ciphertext in the query submitted to the decryption share oracle cannot be the C^* obtained in the challenge phase.

Guess: Based on the value responded by A , B sends the guess value $b' \in \{0, 1\}$ as a response to the DBDH problem. If $b' = b$, then B responds with 1 to represent $T = e(P, P)^{abc}$; otherwise, B responds with 0 to represent $T \neq e(P, P)^{abc}$.

When $T = e(P, P)^{abc}$, the target ciphertext is the correct ciphertext, corresponding to the plaintext M_b . For A , when $T \neq e(P, P)^{abc}$ is a value uniformly distributed in the cyclic group G_2 , the target ciphertext is independent from b . Therefore, the probability of A cracking the BPGOC can be calculated as follows:

$$|Pr[B(P, aP, bP, cP, e(P, P)^{abc}) = 0] - Pr[B(P, aP, bP, cP, T)]| \geq \left| \left(\frac{1}{2} \pm \varepsilon \right) - \frac{1}{2} \right| = \varepsilon$$

This indicates that if A has a high probability of cracking the BPGOC, then B also has a high probability of solving the DBDH problem. Thus, Theorem 1 is proven.

C. COMPARISON

This section provides a comparative analysis of the proposed scheme with existing broadcast encryption schemes. Tables I shows the five ways of comparison, including key storage cost, length of cipher-text, encryption and decryption cost, and the supported mode. Obviously, in terms of key storage space, our system only needs to store one private key and one public key. Besides, for each access subgroup, only two corresponding open values should be disclosed. This leads to the cost only $O(1)$ that is better than the required $O(n)$ proposed by Du et al. [28] and Phan et al. [29]. Next, the four schemes' length of ciphertexts makes little differences. However, the time spent on encryption and decryption of our system is the least, with only five times of bilinear pairing function calculation to complete. (multiplication, hash function, and XOR operation take too little time to be considered.)

Last, the biggest difference among the systems is that our schemes can support four modes of encryption without additional parameters for operation.

D. DISCUSSION

The proposed bilinear pairing-based group-oriented cryptosystem can realize four encryption and decryption models; members only need to store one private key and the sender only must perform encryption one time regardless of the encryption and decryption models adopted. The main design principle is that equality relationships exist between system parameters and a member's private key and the corresponding public key. Therefore, Models 1 and 2 can be achieved without increasing the number of members' private keys.

In conclusion, this group-oriented cryptosystem based on bilinear pairing has the following advantages:

- (1) The cryptosystem is a complete system that simultaneously considers four encryption and decryption models and is applicable to a group environment.
- (2) The cooperative decryption in Model 4 can flexibly set the type of legitimate group according to the distribution of members' decryption weights.
- (3) The cryptosystem requires a low number of keys.
 - a. Any group member needs to store only one private key, and only one public key corresponds to each private key regardless of the number of access subgroups. Thus, the complexity of key management is low.
 - b. For each access subgroup, only two corresponding open values should be disclosed.
 - c. The group only has one public key.
- (4) The amount of calculation for the sender and the amount of transmitted ciphertexts are low.
 - a. The ciphertexts have the same length regardless of the encryption and decryption models adopted: three points in the additive group and one string with the length of l .
 - b. The amount of calculation of the sender only involves the following:
 - Dot product calculation: three times.
 - Bilinear pairing function calculation: one time.
 - XOR calculation: one time.
- (5) Legitimate subgroups can avoid malicious senders when performing decryption. The legitimacy of the ciphertext is verified. If the ciphertext is verified as illegitimate, this indicates that the ciphertext is not derived from the encryption of corresponding plaintext but a meaningless message. Consequently, the decryption is terminated.

V. CONCLUSION

This study proposed a group-oriented cryptosystem based on bilinear pairing that features high flexibility and efficiency. This cryptosystem enables users to flexibly designate

a group member, all members, an access subgroup in a hybrid group, or all legitimate subgroups in a hybrid group as the decryptor. The method has four advantages: (1) It owns four encryption and decryption models and is applicable to a group environment. (2) It requires only a low number of keys. (3) The amount of calculation for the sender and the amount of transmitted ciphertexts are low. (4) Legitimate subgroups can avoid malicious senders when performing decryption since the ciphertext is not derived from the encryption of corresponding plaintext. Therefore, this system is very suitable for Personal Health Records to exchange and sharing securely.

REFERENCES

- [1] S.-W. Chou and C.-H. Chiang, "Understanding the formation of software-as-a-service (SaaS) satisfaction from the perspective of service quality," *Decis. Support Syst.*, vol. 56, no. 1, pp. 148–155, 2013.
- [2] G. K. Landy and A. J. Mastrobattista, "Software as a Service (SaaS)," in *The IT/Digital Legal Companion*. Rockland, MA, USA: Syngress, 2008, pp. 351–374.
- [3] A. Sunyaev, D. Chorny, C. Mauro, and H. Kremer, "Evaluation framework for personal health records: Microsoft healthvault vs. Google health," in *Proc. 43rd Hawaii Int. Conf. Syst. Sci.*, Jan. 2010, pp. 1–10.
- [4] A. Brown, "An Update on Google Health and Google Powermeter." Google Health. [online]. Available: <https://googleblog.blogspot.com/2011/06/update-on-google-health-and-google.html>
- [5] D. Baumer, J. B. Earp, and F. C. Payton, "Privacy of medical records: IT implications of HIPAA," *ACM Comput. Soc.*, vol. 30, no. 4, pp. 40–47, 2000.
- [6] N. Archer, U. Fevrier-Thomas, C. Lokker, K. A. McKibbin, and S. E. Straus, "Personal health records: A scoping review," *J. Amer. Med. Informat. Assoc.*, vol. 18, no. 4, pp. 515–522, 2011.
- [7] I. Carrión, J. L. F. Alemán, and A. Toval, "Assessing the HIPAA standard in practice: PHR privacy policies," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug./Sep. 2011, pp. 2380–2383.
- [8] W. Stallings, *Cryptography and Network Security: Principles and Practice*. London, U.K.: Pearson, 2016.
- [9] I. Carrión, J. L. Fernández-Alemán, and A. Toval, "Usable privacy and security in personal health records," in *HHuman-Computer Interaction—INTERACT (Lecture Notes in Computer Science)*, vol. 6949. Berlin, Germany: Springer, 2011, pp. 36–43.
- [10] J. Li, "Ensuring privacy in a personal health record system," *Computer*, vol. 48, no. 2, pp. 24–31, Feb. 2015.
- [11] S. R. Kessler, S. Pindek, G. Kleinman, S. A. Andel, and P. E. Spector, "Information security climate and the assessment of information security risk among healthcare employees," *Health Inform. J.*, to be published.
- [12] J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu, and R. Buyya, "Ensuring security and privacy preservation for cloud data services," *ACM Comput. Surv.*, vol. 49, no. 1, p. 13, 2016.
- [13] K. Nishat and B. R. Purushothama, "Group-oriented encryption for dynamic groups with constant rekeying cost," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 4120–4137, 2016.
- [14] R. Jiao, H. Ouyang, Y. Lin, Y. Luo, G. Li, Z. Jiang, and Q. Zheng, "A computation-efficient group key distribution protocol based on a new secret sharing scheme," *Information*, vol. 10, no. 5, p. 175, 2019.
- [15] X. Tao, C. Lin, Q. Zhou, Y. Wang, K. Liang, and Y. Li, "Secure and efficient access of personal health record: A group-oriented ciphertext-policy attribute-based encryption," *J. Chin. Inst. Eng.*, vol. 42, no. 1, pp. 80–86, 2019.
- [16] Z. Min, G. Yang, A. K. Sangaiah, S. Bai, and G. Liu, "A privacy protection-oriented parallel fully homomorphic encryption algorithm in cyber physical systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, Jan. 2019, Art. no. 15.
- [17] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.
- [18] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO (Lecture Notes in Computer Science)*, vol. 218. Berlin, Germany: Springer, 1986, pp. 417–426.

- [19] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 2248. Berlin, Germany: Springer, 2001, pp. 514–532.
- [20] S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing," in *Algorithmic Number Theory* (Lecture Notes in Computer Science), vol. 2369. Berlin, Germany: Springer, 2002, pp. 324–337.
- [21] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.
- [22] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Int. Workshop Manag. Requirements Knowl. (MARK)*, Jun. 1979, pp. 313–317.
- [23] E.-H. Lu, W.-Y. Hwang, L. Harn, and J.-Y. Lee, "A conference key distribution system based on the Lagrange interpolating polynomial," in *Proc. 7th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Mar. 1988, pp. 1092–1094.
- [24] J.-B. Feng, H.-C. Wu, C.-S. Tsai, and Y.-P. Chu, "A new multi-secret images sharing scheme using Lagrange's interpolation," *J. Syst. Softw.*, vol. 76, no. 3, pp. 327–339, 2005.
- [25] M. Ito, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," *Electron. Commun. Japan Part III Fundam. Electron. Sci.*, Vol. 72, no. 9, pp. 56–64, 1989.
- [26] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo-random bits," *SIAM J. Comput.*, vol. 13, no. 4, pp. 850–864, 1984.
- [27] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. 1st Annu. Conf. Comput. Commun. Secur.*, Dec. 1993, pp. 62–73.
- [28] Q. Du, G. Wang, and Q. Liu, "A scalable encryption scheme for multi-privileged group communications," in *Proc. IEEE/IFIP 8th Int. Conf. Embedded Ubiquitous Comput. (EUC)*, Hong Kong, Dec. 2010, pp. 597–602.
- [29] D.-P. Phan, D. Pointcheval, S. F. Shahandashti, and M. Strefler, "Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts," *Int. J. Inf. Secur.*, vol. 12, no. 4, pp. 251–265, 2013.



ZHEN-YU WU received the Ph.D. degree in computer science from National Taiwan University, in 2011. He is currently an Associate Professor with the Department of Information Management, National Penghu University of Science and Technology, Taiwan. His current research interests include information security, cryptography, the Internet of Things, and medical information.

...