

Improved ciphertext-policy time using short elliptic curve Diffie–Hellman

Pongpisit Wuttidittachotti¹, Parinya Natho²

¹Department of Data Communication and Networking, Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

²Department of Information Technology, Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

Article Info

Article history:

Received Apr 14, 2021

Revised Jan 6, 2023

Accepted Jan 16, 2023

Keywords:

Attribute-based encryption
Ciphertext policy time short
elliptic curve Diffie–Hellman
CPT-SECDH
Ciphertext-policy time
Diffie–Hellman
Elliptic curve

ABSTRACT

Ciphertext-policy attribute-based encryption (CP-ABE) is a suitable solution for the protection of data privacy and security in cloud storage services. In a CP-ABE scheme which provides an access structure with a set of attributes, users can decrypt messages only if they receive a key with the desired attributes. As the number of attributes increases, the security measures are strengthened proportionately, and they can be applied to longer messages as well. The decryption of these ciphertexts also requires a large decryption key which may increase the decryption time. In this paper, we proposed a new method for improving the access time to the CP using a new elliptic curve that enables a short key size to be distributed to the users that allows them to use the defined attributes for encryption and decryption. Each user has a specially created key which uses the defined attributes for encryption and decryption based on the Diffie-Hellman method. After the implement, the results show that this system saves nearly half of the execution time for encryption and decryption compared to previous methods. This proposed system provides guaranteed security by means of the elliptic curve discrete logarithmic problem.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Parinya Natho

Department of Information Technology, King Mongkut's University of Technology North Bangkok

1518 Pracharat Sai 1 Rd. Bangsue, Bangkok 10800, Thailand

Email: s5907011910037@email.kmutnb.ac.th

1. INTRODUCTION

At present, the amount of information is increasing rapidly, so new technology is required to handle the large amounts of data. The cloud [1] system is one of the most popular technologies for data storage. By using the cloud, a user can create a file on their computer and save it to a folder in the cloud. The cloud can be used to store data or share resources in order to perform complex calculations. When using cloud storage, users can keep their data secure and confidential from other unauthorized users or even the cloud service providers. Therefore, users may need to gain more granular control over their data security, for example, by means of a user data access control policy.

Cloud service providers may face an increased demand for the protection of users' data and their privacy and security [2]. One possible answer to this demand is attribute-based access control (ABAC) [3]. access control (AC) is a control approach which allows or denies a user access to the user's resources and secures the required resources and objects by means of an established regulation mechanism. Access control mechanism (ACM) [3] refers to the logical component that safeguards security measures by preventing and detecting unauthorized and authorized access in an automated system. The administrator manages the AC

policy in the system and decides whether the user should be granted access. When a user needs access to a resource, the system will send an access request. Then, the administrator checks whether the user has the right to the resource without conflict with the AC policy and can be given access to the permitted resource. However, the enforcement of access requests is subject to the same factors as the customized organizational policies, such as location and system time.

One of the well-known authentication methods depends on the public key infrastructure (PKI) [4]. Assuming that a user, Alice, is authenticated by another user, Bob, for PKI authentication, Alice receives a private key and a public key from the system Authority. Alice uses the private key to create her signature and sends it to Bob. After Bob receives Alice's signature, he will be able to retrieve the public key and check if Alice's signature is correct. Alice will only be validated if her signature is valid.

A concept of public key cryptography [5] was used in attribute-based encryption (ABE) which has been developed for access control parameters to encrypt data. Cloud data owners can set their own access policies through encrypted files. Thus, sharing information can make data more flexible. The PKI-based authentication mechanism is different from the ABE because of many factors, such as user attributes and object attributes. In addition, the main concept and process of ABE can be summarized in the following steps: i) a user, who must be the cloud data owner, will obtain authorized attribute keys from the cloud service provider authority; ii) the user requests verification from the authenticator; iii) after receiving the authentication request from the user, the authenticator will respond to the user according to the requirements of the attribute; iv) the signature of the authenticated owner will be generated along with the desired attribute key, and the signature will be sent to the investigator; v) to verify the signature, the authentication system retrieves some information from the authority, such as the public key of the attribute. Next, the investigator verifies that the signature is or is not correct; and vi) the authenticators respond to the user with a verified result which is Yes or No.

If we compare ABE and PKI authentication, ABE has three main differences. First, Alice generates a signature based on specific attribute requirements. Second, Alice uses a key attribute set to generate a signature, which is necessary from Bob, instead of Alice using a private key to create a signature. The last difference is between the one-to-one relationship and Alice's private key. With a public key for PKI authentication, the relationship between Alice's public key attribute and the key attribute is changed. Thus, it can be a one-to-one or a one-to-many relationship.

ABE has two different classifications: one is key-policy ABE (KP-ABE) and the other is ciphertext-policy ABE (CP-ABE). Key-Policy ABE work as an important part of the decision process of the access control policy. There are limits on the ability to control and the ability to operate and use the system [6]. CP-ABE is related to the access structure; each set of attributes is provided with a different private key. Users can decrypt the ciphertext if an attribute key is available. In addition, the sender can specify an access control attribute policy for users.

The user can decrypt the message in CP-ABE only if he/her receives the key with the desired attribute because a set of attributes in CP-ABE is correlated. The CP-ABE scheme first introduced the ciphertext policy encryption based on ABE by Bethencourt *et al.* [7]. The ciphertext is linked to the policy access structure, and users' secret keys are associated with the attribute in CP-ABE. A common framework of the CP-ABE scheme consists of setup algorithm, key generation algorithm, encryption algorithm and decryption algorithm. However, the CP-ABE cannot be deployed directly in cloud storage system because the public parameter or key attribute of those schemes depends on the number of attributes. Also, the number of attributes increases, and the keys increase, which can lead to longer messages as well. The decryption of these ciphertexts also requires a large decryption key, which may increase the decryption time [8]. Therefore, the CP-ABE system work is more flexibly adjusted than the KP-ABE.

Our main contributions in this paper can be summarized as follows:

- a) We propose a new method for improving access time to ciphertext policy (CP) encrypted and decrypted data by using a short elliptic curve Diffie–Hellman (SECDH). The main advantage of using the elliptic curve for CP-ABE is its key size and speed performance.
- b) We designed a new elliptic curve that uses a smaller key size to distribute to each user which has a key created to define the attributes for encryption and decryption based on Diffie–Hellman. The research work aims to achieve the ciphertext policy attribute-based encryption by adopting the fast encryption and decryption concept of the SECDH to solve the problem as defined in articles [9]–[12]. The security of the proposed scheme of the elliptic curve discrete logarithmic problem (ECDLP) and the performance speed for encryption and decryption are computed (see sections 3 and 4).

The rest of the paper is organized as follows: section 2 provides a literature review. Section 3 explains our proposed short elliptic curve Diffie–Hellman (CPT-SECDH). Section 4 presents the performance, security, mathematical analyses of the proposed method. Section 5 concludes the paper and suggests future research.

2. LITERATURE REVIEW

It should be re-emphasized here that the main problem studied in this paper is to improved access time to CP-ABE. Cheung and Newport [13] proposed one-time signatures which have been improved from the identity-based encryption (IBE) technique in order to obtain a chosen-ciphertext (CCA) secure extension. It uses AND gate, which are the negative and positive attributes of an access structure. Because it can only be used with the AND gate, this method is not very useful. In this approach, the decisional bilinear Diffie–Hellman (DBDH) assumption serves as security proof.

The work of Goyal *et al.* [14], who proposed an enhanced version of the CP-ABE, is another technique. This version allows for a small access tree with a threshold gate, with security shown using the conventional model and the DBDH assumption. The issue with this model is that the tree depth should be determined during the setup process. As a result, users may be limited to trees with fewer depths than the depths specified during the setup process.

Liang *et al.* [15] introduced a paradigm in 2009 that allowed for quicker encryption, decoding, and a smaller ciphertext. The DBDH assumption was used to test the system. Ibraimi *et al.* [16] suggested a CP-ABE policy that may show any access policies using the AND and OR Boolean operators, as well as the threshold. The CP-ABE model's primary operational issues were the attribute's revocation and ciphertext size. Emura *et al.* [9] provided a fixed constant-length ciphertext model, demonstrating that the number of pairing calculations is likewise fixed. Later, Lewko and Waters [10] introduced a novel model that proved access control for linear sharing matrix over attributes using the linear secret sharing scheme (LSSS). Certain characteristics, such as key size and ciphertext size, were suggested in all these designs.

Another direct method is to assign a part of the decrypt data to the cloud. At mentions the work of Green *et al.* [17] improved an ABE model used outsourced for decryption. Furthermore, the secret key attribute consists of two parts. The first part is the El-Gamal transformation keys and type key. The proxy is the second part, which can decode some ciphertexts using a conversion key, leaving just the El-Gamal ciphertext simple to decrypt for all users. Li *et al.* [18] proposed a multi-authority CP-ABE model with user responsibility, which reduced both the assumption of authorities and their user's trust. The DBDH assumption, the decisional linear (DLIN) assumption, and the q-decisional Diffie–Hellman (q-DDH) assumption were all used to show the security of the standard model. Furthermore, Li *et al.* [19] have modified this system to allow both the key distribution and the decrypted data to be outsourced.

An example of the work of Lewko and Waters [10] presented adaptive security of the multi authority ciphertext policy attribute-based encryption (MA-CP-ABE) scheme. The central authority is not trusted in this system. To improve the optimized ABE algorithms, Odelu *et al.* [20] proposed a unique accessible AND-gate access CP-ABE model with fixed-size keys based on elliptic curve cryptography. The difficult bilinear pairings are replaced with alternative more efficient arithmetic operations in our suggested solution. This was the case until in Li *et al.* [21] presented the threshold multi authority access control scheme in CP-ABE. This has caused the research community to find many authorities can join and manage all the attribute set. None of the authorities are able to obtain the master key for themselves.

Li *et al.* [22] proposed an enhanced CP-ABE by incorporating the ordered binary decision diagram (OBDD) as a new access structure. It is a non-monotonic access structure that permits AND, OR, and NOT between properties in this scheme. The input to a function is made up of Boolean variables X_1, X_2, \dots, X_N . The ordered binary decision algorithm is the name given to this concept. Ding *et al.* [23] developed a new pairing-free data access control architecture for the internet of things (IoT) based on CP-ABE and elliptic curve encryption IoT. Cui *et al.* [12] suggested an express CP-ABE approach with certain concealed access structures in prime-order groups, which enhanced efficiency while maintaining security by deleting the binding model.

Zhang *et al.* [24] proposed privacy protection and an entire hiding access (PPFH-CP-ABE) technique for cloud storage systems that may provide efficient privacy protection. As a result, the ciphertext's access structure is completely obscured. The ciphertext-policy attribute-based encryption (CP-ABESE) approach was recently presented by Yin *et al.* [25]. This technique allows the data subject's owner to specify a data user's fine-grained search permissions. The basic notion is that a data subject owner encrypts an index keyword under a given access policy if and only if the attributes of a data user fulfill the access policy, allowing the data user to search the encrypted index keywords. However, all ciphertext, encryption, and decryption sizes in the CP-ABE paradigm incur overheads in linear increments. For a lightweight mutual authentication, Ayoub *et al.* [26] proposed using elliptic curve cryptography to secure data transit between the cloud and devices. Jasem *et al.* [27] proposed a hot wallet model privacy for bitcoin users on an elliptic curve digital signature algorithm.

3. METHOD

In this section, an improved CPT-SECDH is proposed and described. This research investigates and designs a new method for ciphertext-policy encryption key exchange in public communication. A new key exchange protocol is securely transmits it to another party which one communication party generates a secret key.

Improved ciphertext-policy time using short elliptic curve Diffie–Hellman (Pongpisit Wuttidittachotti)

3.1. System model

The five algorithms that make up the CP-ABE are as follows: first, “*System.Setup*”; second, called an “*Authority.Setup*”; third, key generation is called a “*KeyGen*”; fourth, an encrypted is called an “*Encrypt*”; and fifth, a decrypted is called a “*Decrypt*”. The five algorithms are defined below [10].

- *System.Setup* (k) \rightarrow GP . The parameter k is input security parameter in this algorithm. It outputs the global public parameters GP .
- *Authority.Setup* (GP) \rightarrow SK, PK . The authority takes a generate GP in the first step, which a global public parameter as an input to generate the secret keys SK and public keys PK .
- *KeyGen* (GP, GID, i, SK) \rightarrow $SK_{i,GID}$. Key generation (*KeyGen*) takes a user's parameter list to input the global public parameters in system setup, a global identity GID , attribute i , and the secret key SK of the attribute authority setup algorithm. The key generation algorithm outputs the secret key for attribute $SK_{i,GID}$ with corresponding to GID and a published it to authorized users.
- *Encrypt* ($M, GP, (A, \rho), \{PK_i\}$) \rightarrow CT . The encryption algorithm outputs a ciphertext CT . Given a message M and public key parameters of all the attributes used in the access policies it can gain access to matrix A with ρ mapping rows with attributes.
- *Decrypt* ($GP, CT, \{SK_{i,GID}\}$) \rightarrow M . The decryption algorithm outputs a message M . Given a ciphertext CT if the access matrix satisfies the set of secret keys of an attribute owned by a certain user was able to successfully recover the message M .

Figure 1 show of a model CPT-SECDH which consists of a data owner (DO), attribute authority (AA), data user (DU) and cloud provider (CP).

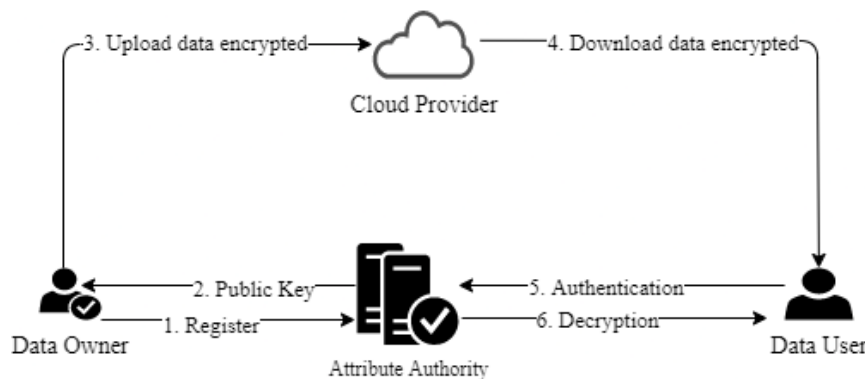


Figure 1. Proposed model for the system

- Data owner (DO). The DO can request access control policies through attributes in the system and the encryption type used to secure the data before outsourcing into the cloud. If the DO decides that some attribute needs to be revoked, he will first inform the responding revoked users list and then send the list to the CP .
- Attribute authority (AA). The AA can create to the system parameters as well as secret keys for the data users because it is only trusted for implementing the system setup algorithm. The attribute authority keeps track of each user's attributes in order to recode reserved attributes. The attribute authority permits data consumers to access a portion of the decryption during the decryption process. Each user is provided with a global identity GID which is registered in the system.
- Data user (DU). The DU can request access to encryption data stored in the CP . If the DU will send his transformation key to the CP for partial decrypted when he wants to access the data in the CP . Only if the attribute mapping between the data user and the access policy is valid can the ciphertext be successfully decrypted by the data user.
- Cloud provider (CP). The CP is responsible for storing encrypted data to implement the data re-encryption from achieving a ciphertext updating and to implementing a partial decryption of the algorithm for the DU .

3.2. Protocol construction

In this section, we designed a new SECDH computation process as shown in Figure 2 can be shown for the encryption and decryption of a message and we explain the work process which consists of the following 3 steps:

- Step 1. Generate key: Alice and Bob aim to exchange information using an elliptic curve $E_p(a, b)$ where p is a prime number. Alice selects a number a and a point A and Bob selects a number b and a point B on the elliptic curve $E_p(a, b)$. Alice computes a public key $PK_A = aG$. Bob computes a public key $PK_B = bG$ where G is a generator of $E_p(a, b)$.
- Step 2. Create a secret key: Alice calculates the secret key $SK_A = a(A + PK_B)$. Bob calculates the secret key $SK_B = b(B + PK_A)$. Alice and Bob publish it as a secret key to each other.
- Step 3. Bob encryption and transport secret key: Bob selects a random point of the shared key $S(x, y)$ on the elliptic curve to be used as the secret key. Bob encryption the shared key $S_E = S + SK_A + bB$, where selects a random point of the shared key $S(x, y)$ on the elliptic curve and shares S_E with Alice. Alice computed S_E and retrieves $S = S_E - SK_B - aA$.
- Encryption: Alice and Bob can send a message $M \in E_p(a, b)$. Alice and Bob can also calculate the ciphertext $C = S + M$ and send it to each other. Alice and Bob agree communication on the elliptic curve now has the same point S as their secret key.
 - Decryption: In this step, Alice and Bob receive C and decryption by calculating $M = C - S = S + M - S$.

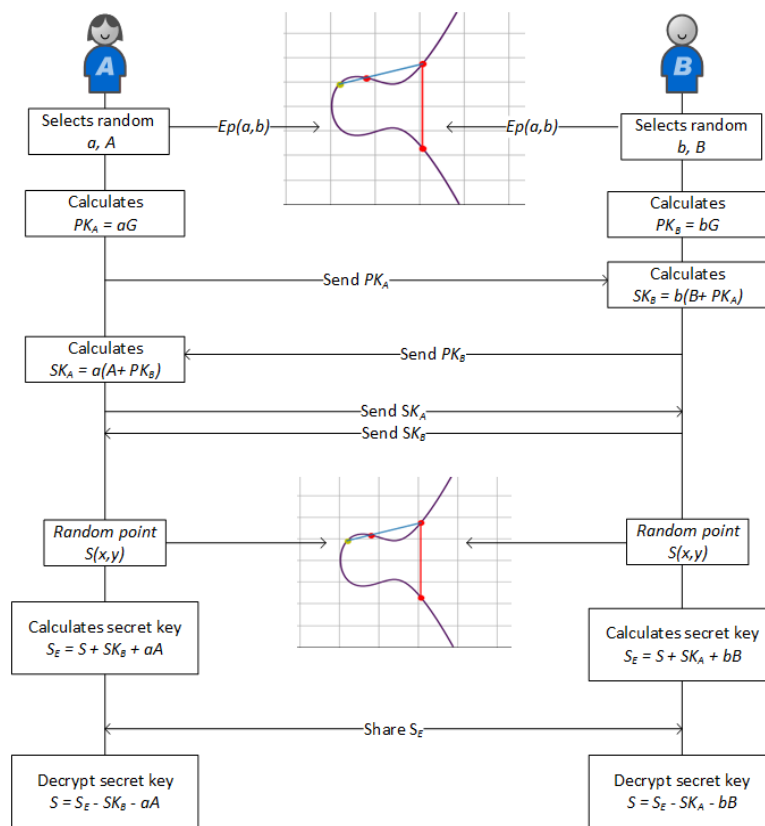


Figure 2. A model of the protocol construction system

The following is an example of a protocol construction called a short elliptic curve of $G = (35, 7)$ which was the generator of the cyclic group $E_{41}(7, 20)$ show as follows. First, Alice computes a secret key choose a number and point on the elliptic curve $a = 3, A = (2, 40)$. Alice computes and publishes it in a public key $PK_A = aG = 3(35, 7) = (19, 40)$. Second, Bob computes a secret key choose a number and point on the elliptic curve $b = 5, B = (9, 19)$. Bob computes and publishes it in a public key $PK_B = bG = 5(35, 7) = (24, 21)$.

Alice computes a secret key $SK_A = a(A + PK_B = 3[(2, 40) + (24, 21)] = (35, 34)$ and publishes to Bob. Bob computes and publishes secret key for Alice $SK_B = b(B + PK_A = 5[(9, 19) + (19, 40)] = (16, 13)$. Bob encrypts secret key and transports $S = (x, y) = (21, 11)$ on the elliptic curve $E_{41}(7, 20)$ for the encrypt and decrypt of a message so he encryption this secret key S as $S_E = S + SK_A + bB = (21, 11) + (35, 34) + 5(9, 19) = (17, 3)$ and sends shares key S_E to Alice. Alice computed the secret key S_E as $S = S_E - SK_B - aA = (17, 3) + (16, 13) + 3(2, 40) = (21, 11)$.

Encryption and decryption of the message: It should be re-emphasized here that the secret key now have the same point S . Alice computes $C = S + M = (21, 11) + (16, 13) = (14, 19)$ and broadcast a message $M = (16, 13) \in E_{41}(7, 20)$ sends to Bob. Bob receives C and computing $M = C - S = (14, 19) - (21, 11) = (16, 13)$.

This example shows how a short elliptic curve consists of defining the number of keys, such as private keys, public keys, as well as points on the elliptic curve and the number of operations and keys for each specific entity. If we compare a new scheme with a normal model elliptic curve [28], the number of public keys are more than in our scheme and the number of operations in the normal model are more than in our scheme. Therefore, our scheme uses a smaller and faster key exchange protocol, which we describe as a SECDH.

3.3. Proposed algorithm

In this section, we give the detailed CPT-SECDH scheme for improved ciphertext time of secured data sharing. First of all, it should be noted that the data owner encrypts the message M with sG , where G is the generator of a circuit subgroup of an elliptic curve with the order r , and s is a random value in \mathbb{Z}_r . We used a simple scalar multiplication on the short elliptic curves, which simplifies the calculation, to improve the efficiency of all the algorithms. The values s and values 0 of the encryption algorithm are divide into λ_x and ω_x in the matrix respectily. To prevent the visibility factor sG , the data user must combine attribute keys of a ciphertext if need to decryption message M . Therefore, it will prevent collusion attacks because each user attribute will bound to a global identity. The decryption algorithm will introduce new form of $H(GID)\omega_x nG$, a public key of authority is nG . In the ω_x shares 0 , the redundant specifications will be cancelled such as the data users have the same identity. Moreover, the different specifications of the $H(GID)\omega_x nG$ form which cannot be eliminated if two users with different identities conspire with each other. This means the recovery of sG will fail, as well as their message M . The CPT-SECDH is consists of the following first, called "System.Setup"; second, called "Authority.Setup"; third, called "Key Generation"; fourth, called "Encryption"; fifth, called "Decryption".

- *System.Setup*. Suppose that $GF(q)$ is the boundary field of q , whilst an elliptic curve is E defined above $GF(q)$, and G is an element of a large prime order r in E . The point G creates a subgroup cycle of E , in which the ECDLP is difficult. Additionally, a map of the GID to elements of \mathbb{Z}_r is chosen from the hash function $H : \{0,1\}^* \rightarrow \mathbb{Z} * r$.
- *Authority.Setup*. The authority will store a list of attributes synonymous to their GID for each data user. The authority uses a random number $n \in \mathbb{Z}_r$ as the master secret key and broadcasts the public key nG . For each attribute i in the system, the authority will randomly select $k_i \in \mathbb{Z}_r$ and publish the public key $PK_i = k_i G$.
- *Key Generation*. The attribute authority can calculate $SK_{i,GID} = k_i + H(GID)n$ and create a key of the attribute i for the user with GID and save this attribute in the list of attributes i .
- *Encryption*. The encrypted algorithm consists of the following steps: First, a plain text message is mapped to the first point M on the elliptic curve E . Second, it selects $s \in \mathbb{Z}_r$ randomly and calculates $C_0 = M + sG$. The encryption algorithm accepts the access policies made by the owner of the data and then exports the $n \times l$ access matrix which has A with ρ as the row mapping for the attributes. Next, the user selects a random vector $v \in \mathbb{Z}_r^l$ with 0 as its selected first item and computes ω_x which means $A_x \cdot v$. Finally, the ciphertext is calculated as $C_{1,x} = \lambda_x G + \omega_x PK_{\rho(x)}$, $C_{2,x} = \omega_x G, \forall x$.
- *Decryption*. The decryption algorithm is a deciphered ciphertext. Data users should find a set of rows A_x of A that satisfies $(1, 0, 0, \dots, 0)$ of these rows and sends its GID with $(C_{2,x}, \rho(x))$ of each x . The authority examines its identity and decides whether these qualities are in attributes accordance with the attribute list. If the request is correct, for each $(C_{2,x}, \rho(x))$, the authority computes $\sum C_{2,x}, SK_{\rho(x)}, GID = \sum (\omega_x G(k_{\rho(x)} + H(GID)n) = \sum (\omega_x k_{\rho(x)} G + \omega_x H(GID)nG)$. Then the authority will send the result to the data user in a secure channel. With the user results, the data user can compute $\sum c_{1,x} - \sum C_{2,x} SK_{\rho(x)}, GID = P(\lambda_x G + \omega_x PK_{\rho(x)}) - \sum (\omega_x k_{\rho(x)} G + \omega_x H(GID)nG) = \sum (\lambda_x G - \omega_x H(GID)nG)$ for everything x . The data user chooses constants $c_x \in \mathbb{Z}_r$ so $\sum_x c_x A_x = (1, 0, \dots, 0)$ and computes $\sum_x c_x (\lambda_x G - \omega_x H(GID)nG)P = sG$, as $v \cdot (1, 0, 0, \dots, 0) = s$ and $u \cdot (1, 0, 0, \dots, 0) = 0$. Finally, the results to message M can calculate $M = C_0 - sG$.

4. RESULTS AND DISCUSSION

The results in this research, the proposed algorithm based on two parts is analyzed. The first part is the speed of the algorithm compared with that of other existing ciphertext-policy attribute-based encryption methods. The second part is security analysis.

Before going into the details of the analyses and results, we explain our test application which used a laptop with an Intel Core i3 1.9 GHz Processor, 8 GB RAM, Ubuntu 16.04 OS, Charm 0.50 and Python 3.5.5 language which formed a framework for rapidly prototyping advanced cryptosystems. In our experiment, firstly, we tested the performance of the basic CP-ABE algorithm and secondly, we tested the performance of CP-ABE using elliptic curve Diffie-Hellman and the proposed new CPT-SECDH. Finally, we compared the performance of all the algorithms.

4.1. Speed analysis

Previous research involved lower least costs for encrypt and decrypt, as the overhead costs of computing does not depend on the number of attributes. The experiment was conducted by evaluating the performance and comparing the proposed new CPT-SECDH (our scheme) with CPABE [9] and CPABE-ECDH [12]. We tested the runtime of the core algorithms for the pairing-based cryptography library, the implements used an elliptic curve group based on curve $y^2 = x^3 + x$ at a 512-bit security level. The number of attributes used several different attributes (from 1 to 100) for encryption and decryption Table 1 shows the time of average values 100 rounds for all algorithms.

The sample data used in the performance testing include.

Policy : ‘THREE and (ONE or TWO)’

Attribute : ‘THREE’, ‘ONE’, ‘TWO’

Summary showing a comparison of the encryption and decryption results in Table 1. Table 1 shows that it required 1.6232 and 0.8449 ms to complete our scheme when the messages were encrypted and decrypted. That means the encryption and decryption performance is better than the CPABE and CPABE-ECDH method because the faster encryption and decryption will result in a faster performance. Even though encryption and decryption in our scheme are faster than in previous research, we can see that it still has a similar speed. Therefore, we analyzed the speed performance between the existing CPABE-ECDH and the new CPT-SECDH (our scheme) which shows a significant relationship when using the paired samples t-test. We also divided our experiment into two main scenarios as follows.

Table 1. Comparison of encryption and decryption results

Scheme	Encryption (ms)	Decryption (ms)
CPABE	586.2652	756.0775
CPABE-ECDH	3.5125	1.4811
Our scheme	1.6232	0.8449

Scenario 1: We compared the computed time incurred in encryption data on the owner in Figure 3 and decryption data in Figure 4, as they both a direct impact on user experience. Figures 3 and 4 show our scheme saves nearly half the time for encryption.

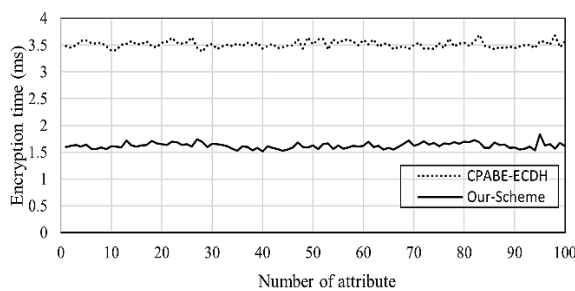


Figure 3. Comparison of encryption times

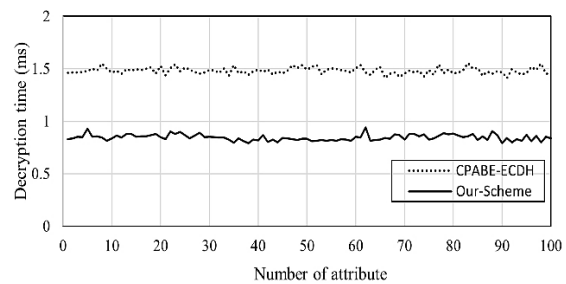


Figure 4. Comparison of decryption times

In Figures 3 and 4 we compare the computing times required with a different number of attributes for encryption and decryption. We can see from these results that our scheme saves almost half the time of encryption and decryption which means that our scheme significantly lowers the computation overheads for encryption and decryption.

Scenario 2: We tested the significance of the relationship between encryption and decryption by determining the confidence interval at a difference value of 99%. The results are summarized in Tables 2 and 3.

In Table 2 we set the significance (α) of 0.01; the average encryption value (\bar{X}) of CPABE-ECDH is equal to 3.5125 which is greater than that of our scheme which is 1.6232. The standard deviation (SD) for the CPABE-ECDH scheme is equal to 0.06471 and our scheme is equal to 0.05465. The mean paired differences (\bar{D}) value is equal to 1.88930. The standard deviation of the paired differences ($SD_{\bar{D}}$) is equal to 0.08415. The t-test statistics (t) are equal to 224.514. Finally, the probability (P) value is equal to 0.00, which means that the CPABE-ECDH schemes are significantly different from our scheme at the 0.01 level. Therefore, our scheme has significantly increased the encryption efficiency.

Table 2. Significance of the relationship of encryption with a 99% confidence interval of the difference

Scheme	\bar{X}	SD	\bar{D}	$SD_{\bar{D}}$	t	P
CPABE-ECDH	3.5125	0.06471	1.88930	0.08415	224.514	0.00
Our scheme	1.6232	0.05465				

In Table 3, we can see that the average decryption value (\bar{X}) of CPABE-ECDH is equal to 1.4811 which is greater than that of our scheme which is 0.8449. The standard deviation (SD) value of the CPABE-ECDH scheme is equal to 0.03100 and the value of our scheme is equal to 0.02910. The mean paired differences (\bar{D}) value is equal to 0.63622. The standard deviation of the paired differences ($SD_{\bar{D}}$) is equal to 0.04487. The t-test statistics (t) are equal to 141.787. Finally, the probability (P) is equal to 0.00. Thus, CPABE-ECDH is significantly different from our scheme at the 0.01 level. Therefore, our scheme has increased the decryption efficiency significantly.

Table 3. Significance of the relationship of decryption with a 99% confidence interval of the difference

Scheme	\bar{X}	SD	\bar{D}	$SD_{\bar{D}}$	t	P
CPABE-ECDH	1.4811	0.03100	0.63622	0.04487	141.787	0.00
Our scheme	0.8449	0.02910				

4.2. Security analysis

The main aim of this research is to improve ciphertext-policy time using SECDH. Therefore, the security aspect of the proposed method is based on the security hardness under the DDH assumption that will need to be tested.

Theorem 1: If there exists a PP adversary A that can destroy the proposed scheme with an unimportant advantage $\varepsilon > 0$, then there will be a PP algorithm β that can differentiate a tuple DDH from a random tuple with the advantage $\frac{\varepsilon}{2}$. Assuming that the generator of the group P is G and the large prime r . First, DDH challenger C randomly uses $a, b \in \mathbb{Z}_r, \beta \in \{0,1\}$ and $R \in P$. We let Z be abG if $\beta = 0$, Otherwise $Z = R$. The challenger C sends the tuple (G, aG, bG, Z) to β . Then β plays the role of challenger instead of C in the following games.

– *Initialize.* First, A selects a challenger access structure (A, ρ) and sends to β . In the setup algorithm to generate the public key for each attribute i in the system as adversary A, B is chosen randomly $k_i \in \mathbb{Z}_r$ and equalizes $PK_i = k_i aG$. For an attribute authority, β chooses random $n \in \mathbb{Z}_r$ and publishes nG as a public key. When k_i is randomly selected; the public parameters will be distributed randomly as well.

Phase 1. An adaptive A submit pairs (i, GID) to β to request a secret key that corresponds to the following restrictions. For each identity GID , we give V_{GID} instead of a subset of rows of A that are labelled with attributes i , which the attacker can query (i, GID) . For each GID , we want the subspace expanded by V_{GID} but it must not include $(1, \dots, 0)$. In other words, the attacker is not able to decrypt because is not able to request the keys that allow decryption. The β responds by recording this attribute i in the list of attributes corresponding to the GID . Then β chooses random $t \in \mathbb{Z}_r$ and calculates $k_i a + t$ as its secret key.

– *Challenge.* An adaptive A selects two messages of equal length $M_0, M_1 \in P$ and sends it to β . β flips the coin β and selects random $s \in \mathbb{Z}_r^l$. This generates $C = M_\beta + sG$. Then β randomly selects the vector $v \in \mathbb{Z}_r^l$ where s is the first item and λ_x represents $A_x \cdot v$, where A_x is the row x of A . A random vector $u \in \mathbb{Z}_r^l$ with 0 is the first item selected and ω_x replaces $A_x \cdot u$. Finally, β creates a ciphertext challenge $C_{2,x} = \omega_x bG$ and $C_{1,x} = \lambda_x G + k_{\rho(x)} \omega_x Z$, adversary A receives ciphertext challenge $CT = \{(A, \rho), C, C_{1,x}, C_{2,x}\}$.

Phase 2. The adversary A may send additional secret key searches (i, GID) without violating the restrictions.

- Guess. An adaptive A outputs a guess β' of β . Then β outputs 0 to indicate that $Z = abG$ in the above game if $\beta' = \beta$; otherwise, β outputs 1 to guess that $Z = R$. Otherwise, $Z = abG$, will be a real ciphertext. In this case, A 's advantage is ε as defined in the assumption. Thus, $Pr[\beta(G, aG, bG, Z = abG) = 0] = \frac{1}{2} + \varepsilon$. If $Z = R$, it is complete randomly from the adversary A 's perspective. Thus, $Pr[B(G, aG, bG, Z = R) = 0] = \frac{1}{2}$. Finally last, β 's the advantage of destroying for security game is $B = \frac{1}{2}(Pr[B(G, aG, bG, Z = abG) = 0] + Pr[B(G, aG, bG, Z = R) = 0]) - \frac{1}{2} = \frac{1}{2}(\frac{1}{2} + \varepsilon + \frac{1}{2}) - \frac{1}{2} = \frac{\varepsilon}{2}$.

4.3. Data security

In CPT-SECDH, only legitimate users who own a certain attribute will receive their corresponding secret key k_i from the attribute authority. In this research, we analyzed data security the protocol based on the elliptic curve discrete logarithm problem (ECDLP). The data users without the attribute, does not they are unable to obtain any information about the secret key k_i from the relevant public key k_iG in polynomial time. Solving ECDLP requires $O(k)$ operation in big O notation with brute force method. It determines the number value of k in the equation $Q = k * P$ for the known points P, Q on the elliptic curve $E_p(a, b)$, where k is a large random number which is less than p .

The message has a meaning in the ciphertext C_0 . If we suppose that M can be mapped to mG , where $m \in \mathbb{Z}_r$, since s is chosen by the owner data randomly, $C_0 = (m + s)G$ is just a random point on the elliptic curve from the attacker's point of view. The attacker is unable to obtain valuable information about M without s . By using secret sharing methods, s is a secret divided by λ_x . Moreover, it can only be recovered when the data user has a satisfactory set of attributes which will enable decryption of the ciphertext. Invalid users, such as those without the attributes claimed by the access policy, will not have attributes corresponding to rows A_x , such as $\sum_x C_x \lambda_x = (1, 0, \dots, 0)$ as to ECDLP. Therefore, s , the first list of vectors v , cannot be calculated.

5. CONCLUSION

Ensuring the security of a ciphertext-policy attribute-based encryption approach has been proved to be a valuable strategy in cloud storage settings for data security and privacy. It allows data owners to upload their data to the cloud in multiple formats while sharing it with users who have the required identification or qualities. The ciphertext generated by the CP-ABE method, on the other hand, has an explicit access structure that may reveal information about privileged receivers or the ciphertext's underlying message. In this paper, we proposed an improved new Ciphertext-Policy Time for encryption and decryption using CPT-SECDH. We surveyed the ciphertext policy attribute-based encryption comprehensively with respect to its access structure. We also designed a new SECDH. The difficulty of the ECDLP employing a key exchange underpins the security of elliptic curve encryption. Our analysis proved that our security scheme and our experiments demonstrated its efficiency. In the future, we plan to work further on how to improve and implementation the efficiency of the CPT-SECDH scheme. The scheme, for example, will be able to make use of the technology and apply it to the healthcare system or IoT system. In addition, attribute management, access policy control updating, ciphertext updating, and user revocation will also be followed-up.





REFERENCES

- [1] N. Li, L.-J. Zhang, P. Xu, L. Wang, J. Zheng, and Y. Guo, "Research on pricing model of cloud storage," in *2013 IEEE Ninth World Congress on Services*, Jun. 2013, pp. 412–419, doi: 10.1109/SERVICES.2013.70.
- [2] P. Natho and P. Kuacharoen, "Cloud storage security based on group key," *Advanced Science Letters*, vol. 21, no. 10, pp. 3156–3160, Oct. 2015, doi: 10.1166/asl.2015.6510.
- [3] V. C. Hu *et al.*, "Guide to attribute based access control (ABAC) definition and considerations," Gaithersburg, MD, Jan. 2014, doi: 10.6028/NIST.SP.800-162.
- [4] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985, doi: 10.1109/TIT.1985.1057074.
- [5] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology - EUROCRYPT 2005*, 2005, pp. 457–473, doi: 10.1007/11426639_27.
- [6] H.-J. Seo and H.-W. Kim, "Attribute-based proxy re-encryption with a constant number of pairing operations," *Journal of information and communication convergence engineering*, vol. 10, no. 1, pp. 53–60, Mar. 2012, doi: 10.6109/jicce.2012.10.1.053.
- [7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007, pp. 321–334, doi: 10.1109/SP.2007.11.
- [8] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of American Federation of Information Processing Societies (AFIPS '79)*, 1979, pp. 313–317.
- [9] K. Emura, A. Miyaji, K. Omote, A. Nomura, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," *International Journal of Applied Cryptography*, vol. 2, no. 1, 2010, doi: 10.1504/IJACT.2010.033798.
- [10] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology - EUROCRYPT 2011*, 2011, pp. 568–588, doi: 10.1007/978-3-642-20465-4_31.
- [11] H. Ma, G. Zeng, Z. Wang, and J. Xu, "Fully secure multi-authority attribute-based traitor tracing," *Journal of Computational Information Systems*, vol. 9, no. 7, pp. 2792–2800, 2013.





- [12] H. Cui, R. H. Deng, J. Lai, X. Yi, and S. Nepal, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited," *Computer Networks*, vol. 133, pp. 157–165, Mar. 2018, doi: 10.1016/j.comnet.2018.01.034.
- [13] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 14th ACM conference on Computer and communications security*, Oct. 2007, pp. 456–465, doi: 10.1145/1315245.1315302.
- [14] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Automata, Languages and Programming*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 579–591, doi: 10.1007/978-3-540-70583-3_47.
- [15] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, Mar. 2009, pp. 343–352, doi: 10.1145/1533057.1533102.
- [16] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *Information Security Practice and Experience*, 2009, pp. 1–12, doi: 10.1007/978-3-642-00843-6_1.
- [17] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *SEC'11: Proceedings of the 20th USENIX conference on Security*, 2011, pp. 1–34.
- [18] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, Mar. 2011, pp. 386–390, doi: 10.1145/1966913.1966964.
- [19] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-grained access control system based on outsourced attribute-based encryption," in *Computer Security - ESORICS 2013*, 2013, pp. 592–609, doi: 10.1007/978-3-642-40203-6_33.
- [20] V. Odelu, A. K. Das, and A. Goswami, "An efficient CP-ABE with constant size secret keys using ECC for lightweight devices," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 1, pp. 1–15, 2016.
- [21] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1484–1496, May 2016, doi: 10.1109/TPDS.2015.2448095.
- [22] L. Li, T. Gu, L. Chang, Z. Xu, Y. Liu, and J. Qian, "A ciphertext-policy attribute-based encryption based on an ordered binary decision diagram," *IEEE Access*, vol. 5, pp. 1137–1145, 2017, doi: 10.1109/ACCESS.2017.2651904.
- [23] S. Ding, C. Li, and H. Li, "A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT," *IEEE Access*, vol. 6, pp. 27336–27345, 2018, doi: 10.1109/ACCESS.2018.2836350.
- [24] Y. Zhang, J. Li, and H. Yan, "Constant size ciphertext distributed CP-ABE scheme with privacy protection and fully hiding access structure," *IEEE Access*, vol. 7, pp. 47982–47990, 2019, doi: 10.1109/ACCESS.2019.2909272.
- [25] H. Yin *et al.*, "CP-ABSE: A ciphertext-policy attribute-based searchable encryption scheme," *IEEE Access*, vol. 7, pp. 5682–5694, 2019, doi: 10.1109/ACCESS.2018.2889754.
- [26] A. Ayoub, R. Najat, and A. Jaafar, "A lightweight secure CoAP for IoT-cloud paradigm using elliptic-curve cryptography," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 20, no. 3, pp. 1460–1470, Dec. 2020, doi: 10.11591/ijeecs.v20.i3.pp1460-1470.
- [27] F. M. Jaseem, A. M. Sagheer, and A. M. Awad, "Enhancement of digital signature algorithm in bitcoin wallet," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 10, no. 1, pp. 449–457, Feb. 2021, doi: 10.11591/eei.v10i1.2339.
- [28] R. R. Ahirwal and M. Ahke, "Elliptic curve Diffie-Hellman key exchange algorithm for securing hypertext information on wide area network," *International Journal of Computer Science and Information Technologies*, vol. 4, no. 2, pp. 363–368, 2013.

BIOGRAPHIES OF AUTHORS



Pongpisit Wuttidittachotti     is currently an associate professor and head of the Department of Data Communication and Networking at the Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. He received his Ph.D. in Networks, Telecommunications, Systems and Architectures from INPT-ENSEEIH, in France. He received an outstanding employee award in social service at the university level in 2019, and an outstanding employee award at the faculty level and the university level in 2020. He owns more than 30 recognized certifications, for example, CISSP, CISM, CISA, CRISC, CGEIT, IRCA ISO/IEC 27001:2013 Lead Auditor, COBIT 5 Foundation, COBIT 2019 Foundation, COBIT 2019 design and implementation, and Data Protection Officer (DPO). So far, Wuttidittachotti has over ten years of working experience covering software development, networks, security, audit, risk management, IT governance, and standards, and compliance. His expertise has been demonstrated as a member of the ISACA Bangkok Chapter committee since 2015, and an Accredited Trainer - COBIT® 2019 Foundation for ISACA Bangkok Chapter. He has conducted and published many research articles in information security and related topics. He can be contacted at email: pongpisit.w@itd.kmutnb.ac.th.



Parinya Natho     is currently a lecturer of the Department of Information System and Business Computer, Faculty of Business Administration and Information Technology, Rajamangala University of Technology Suvarnabhumi (RMUTSB). He received his B.Sc. in Computer Education from Rajamangala University of Technology Thanyaburi. He received an M.Sc. in Computer Science and Information Systems from the National Institute of Development Administration. He is currently studying for his Ph.D. in Information Technology at the Faculty of Information Technology, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. Parinya has also obtained a Digital Literacy Certification Certificate (IC3) to his name. He can be contacted at: s5907011910037@email.kmutnb.ac.th.