A CP-ABE Scheme Based on Lattice LWE and Its Security Analysis

Yunfei Yao 1,*,†, Huiyan Chen 2,*, Linzhi Shen 3,†, Ke Wang 2,† and Qingnan Wang 2,†

1 School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
2 Department of Cryptography Science and Technology, Beijing Electronics Science and Technology Institute, Beijing 102627, China; wangke_unique@163.com (K.W.); 13553094893@163.com (Q.W.)
3 School of Sciences, Civil Aviation University of China, Tianjin 300300, China; lzshen@cauc.edu.cn
* Correspondence: yaoyunfei@bupt.edu.cn (Y.Y.); hychen2001@126.com (H.C.); Tel.: +86-1-533-215-6899 (H.C.)
† These authors contributed equally to this work.

Abstract: In recent years, the research on ciphertext-policy attribute-based encryption (CP-ABE) scheme design based on learning with errors (LWE) has been a challenging problem, and many researchers have made a lot of attempts at it. At EUROCRYPT 2021, Datta, Komargodski, and Waters proposed the first provably secure direct CP-ABE construction that supports NC1 circuit access structures. Improving on their work, we propose a CP-ABE scheme. Technically, we use the new lattice two-stage sampling technique of Lai, Liu, and Wang (EUROCRYPT 2021) in the key generation phase instead of the extended trapdoor sampling technique. In this way, we obtain a CP-ABE scheme, which is at least as secure as the original scheme under the same conditions, and has a shorter ciphertext; we provide an innovative design idea for designing CP-ABE schemes only based on LWE, although this is only a partial theoretical work.

Keywords: CP-ABE; lattice; DNF; linear secret sharing scheme (LSSS); LWE

1. Introduction

In order to better realize the fine-grained access control of encrypted data, attribute-based encryption comes into being as a generalization of traditional public key encryption [1]. Attribute encryption has two variants, including ciphertext strategy and key strategy. In the ciphertext policy ABE, that is, the ciphertext is associated with the access policy, and the key is correlated to the attribute. While in the key policy ABE, the position of the access policy and the attribute is exactly opposite. However, there is one consistent point: decryption will be successful only if the attribute meets the access policy.

Since the concept of ABE was founded by Sahai et al. [2,3], it has been widely studied in cryptography and has many potential applications. Therefore, cryptographic researchers all over the world have been widely concerned by the design of the ABE scheme, which has given birth to a series of achievements that, on the whole, pursue various tradeoffs between expression, efficiency, security, and potential assumptions [4–15].

Most of the work mentioned above has been based on assumptions associated with bilinear mapping. It is necessary and normal to find structures based on other assumptions. On the one hand, they can not only make us more sure about the existence of the corresponding encryption scheme but also stimulate our enthusiasm for studying and developing new technologies, so as to further deepen our understanding of cryptographic primitives. On the other hand, in view of the known attacks of quantum computing on group-based cryptographic schemes [16], it is important to construct lattice-based attribute encryption to realize the security of quantum computing attacks. To this end, there are indeed several ABE schemes [4,5,8,10,11,14,17–19], which rely on other cryptographic assumptions rather than bilinear mappings as their underlying building blocks.
However, it is an indisputable fact that most of the existing ABE schemes based on LWE are ABE with a key strategy. Until the advent of reference [20], only a few LWE-based CP-ABE schemes had been constructed, and all of them were obtained through a special transformation with the help of the KP-ABE. However, this transformation will inevitably lead to some unsatisfactory results. For more details, please refer to [20].

Under the assumption that there is only LWE, improving the CP-ABE structure based on the general-purpose circuits described above has always been a challenging problem. In order to solve this question, several eye-catching jobs have been completed recently. For example, the work of [6,21,22] depends on LWE and bilinear groups. Work [9] lacks a safety certificate. Recently, a CP-ABE project on the basis of LWE was constructed in [22], although an indispensable part of the scheme remained the general circuit size limit. Reference [20] makes a conceptual contribution to this open and challenging issue. Their excellent work also draws on some innovative ideas from the work of [17,23]. In TCC 2013, Boyen et al. [17] proposed the first lattice-based attribute-based encryption construction (ABE) for NC^1 circuits. Unfortunately, soon after, a loophole was discovered in the security; however, because of its technical novelty, it continues to be widely studied by many scholars. In particular, this is the first lattice-based ABE that uses the linear secret sharing scheme (LSSS) as a key tool to implement access control. In the scheme [23], Agrawal et al. proved that the scheme [17] is actually unsafe, and they gave an idea for repairing the security of the scheme by taking the notion of an admissible LSSS and instantiating it as the class of DNFs. After the work of [23], Datta et al. proposed the first CP-ABE scheme constructed in a direct manner and provided the complete security proof. It is a very innovative and creative work; it not only introduces new methods and techniques in concept but also popularizes and constructs new MA-ABE schemes.

The purpose of this paper is to propose an improved attribute-based encryption scheme based on the ciphertext policy (CP-ABE), which is modified on the basis of [20]. Technically, we use the two-stage sampling technique on the lattice in [24] instead of the extended trapdoor sampling technique in [20]. In this way, we obtain a CP-ABE scheme, which is at least as secure as the original scheme and has a shorter ciphertext.

The organization of the work of this article is listed below. In the second part, the relevant preliminary knowledge is given. In the third part, we give our CP-ABE scheme and the related proof of correctness and security.

2. Relevant Basic Knowledge

Here, we present several related concepts to facilitate readers to understand the full text. A mapping \( \text{negl} : \mathbb{N} \rightarrow \mathbb{R} \). If there is a \( N_c > 0 \) such that the formula of \( \text{negl}(x) < 1/f(x) \) is always valid for all \( f(x) > 0 \) and \( x > N_c \), then \( \text{negl}(x) \) is called a negligible function. In addition, for convenience, we use \([n]\) to represent the collection \(\{1, \ldots, n\}\).

Let PPT denote the abbreviation of probabilistic polynomial time. Assuming that \( \psi \) is a distribution, we will use the symbol \( x \leftarrow \psi \) to indicate that \( x \) is randomly sampled on the basis of the distribution \( \psi \). Assuming that \( Y \) is a set, we use the symbol \( y \leftarrow Y \) to indicate that \( y \) is sampled on the elements of the set \( Y \) based on uniform distribution. Without losing generality, we use uppercase bold letters to represent matrices and lowercase bold letters to represent column vectors, such as \( G \) and \( d \). For convenience, we use \( G_l \) to represent the \( l \)th row of matrix \( G \), and, further, \( G_L \) represents a matrix that includes the set of \( G_l \) for all \( l \in L \). Suppose \( b \) is a vector, we use the symbols \( \|b\| \) and \( \|x\|_{\infty} \) to represent its \( l_2 \) norm and \( l_\infty \) norm, respectively.

2.1. Linear Secret Sharing Schemes with Linear Independence

This section is divided into two parts. In the first part, several essential definitions and characteristics of LSSSs are listed. In the second part, a new LSSS is presented, proposed in reference [25], and its definition will be listed later for the integrity of the article. This new
LSSS has some interesting features, which can help us to design a CP-ABE scheme on the basis of the LWE hypothesis for all NC\(^1\) circuits.

2.1.1. Related Contents of LSSS

A secret sharing scheme means that the secret distributor divides the secret \(d\) into \(n\) parts using a specific algorithm and then distributes \(n\) shares to \(n\) participants. Authorized subsets of \(n\) participants can jointly reconstruct secrets, while other unauthorized subsets cannot.

Definition 1 (Access Structures, [25]). A collection \(A \subseteq 2^{[n]}\setminus \emptyset\) of non-empty subsets of \(n\) participants labeled from one to \(n\) is called an access structure. The collections that belong to \(A\) are named the authorized collections, and the collections that do not belong to \(A\) are named the unauthorized collections. An access structure is deemed monotone in the case that the latter condition is satisfied. That is, for any two \(C, E \in 2^{[n]}\), there is an \(E \in A\) if there are \(C \in A\) and \(C \subseteq E\).

A secret sharing scheme corresponding to a monotone access structure \(A\) can be regarded as a randomized algorithm whose input is a secret \(d\) and its output is \(n\) parts \(p_{a_1}, \ldots, p_{a_n}\). These parts satisfy certain restrictions: that is, \(d\) can be determined by \(\{p_{a_i}\}_{i \in A}\) for arbitrary \(A \in A\), while other sets are independent of \(d\).

A secret sharing scheme corresponding to a non-monotone access structure \(A\) can be regarded as a randomized algorithm whose input is a secret \(d\), and its output is \(2n\) shares viewed as \(n\) pairs \((p_{a_1,0}, p_{a_1,1}), \ldots, (p_{a_n,0}, p_{a_n,1})\), these shares satisfy certain restrictions: that is, \(d\) can be determined by \(\{p_{a_i,1}\}_{i \in A} \cup \{p_{a_i,0}\}_{i \notin A}\) for arbitrary \(A \in A\), while other sets are independent of \(d\), which can be regarded as a transformation of the above concepts and algorithms.

Definition 2 (Non-monotone LSSS, [25]). Let \([n]\) be a group of participants. Additionally, suppose \(q \in \mathbb{N}\) is a prime power. The scheme described above is called non-monotone LSSS \(\Pi\) over \(\mathbb{Z}_q\) if the following conditions are true:

1. For \(\rho \in \{0, 1\}\) as well as \(i \in \{1, \ldots, n\}\), each share \(p_{a_i,\rho}\) of a secret \(d\) in \(\mathbb{Z}_q\) makes up a vector whose input is over \(\mathbb{Z}_q\).
2. There must be a share-generating matrix, recorded as \(G \in \mathbb{Z}_q^{n\times 1}\), and there is also a mapping \(\eta\) from \([q]\) to \([2n]\); \(\eta\) establishes an one-to-one mapping from the rows of \(G\) to the participants derived from \([n]\) and their corresponding negation (represented by \([n + 1, \ldots, 2n]\), respectively).

This meets the conditions mentioned later, that is: Given a vector \(v = (d, a_2, \ldots, a_n)^\top \in \mathbb{Z}_q^n\), then according to \(\Pi\), for the matrix \(G\) described above, there must be a vector representation of \(q\) shares of secret \(d\) that is equal to \(pa = G \cdot v \in \mathbb{Z}_q^{q\times 1}\). For \(\rho \in \{0, 1\}\) as well as \(i \in \{1, \ldots, n\}\), the part \(p_{a_i,\rho}\) includes all of the \(p_{a_i}\) values for which \(\eta(i) = n(1 - \rho) + i\) (in this way, all “1 shares” fall in the first \(n\), and all “0 shares” fall in the last \(n\)). Then, \((G, \eta)\) will be named the LSSS policy about the access structure \(A\).

Based on the analysis of correctness and security in [25], we list the following conclusions. Correctness is achieved through the following condition: if \(R \in [n]\) is authorized, the vector \((1, 0, \ldots, 0) \in \mathbb{Z}_q^n\) is in the span of the rows of \(G_R\) (where \(R = R \cup \{i \in \{n + 1, \ldots, 2n\} \mid i - n \notin R\} \subset [2n]\)). Security is achieved through the condition that if \(R \in [n]\) is not authorized, then \((1, 0, \ldots, 0)\) cannot be expressed linearly by the rows of \(G_R\). In addition, in a case without authorization, there is a vector \(t \in \mathbb{Z}_q^{n-1}\), with the condition that its first components are \(t_1 = 1\) and \(G_R t = 0\), of which \(0\) is the all \(0\) vector.
2.1.2. The Non-Monotone LSSS for NC$^1$

We present a new non-monotone linear secret sharing scheme proposed in [25], which is suitable for all access structures that can be represented through NC$^1$ circuits. This scheme possesses some very attractive and practical properties, as shown below:
- Each element of the corresponding policy matrix is very small, and they all come from $\{-1, 0, 1\}$.
- The secret can be rebuilt through a series of small coefficients that belong to $\{0, 1\}$.
- The rows corresponding to the access policy matrix that matches a collection that is not authorized are linearly independent.

The construction: We present the access structure $A$ described by the NC$^1$ circuit, which is adapted from [25]. The circuit can be represented through a Boolean formula with logarithmic depth, which is represented by AND, OR, and NOT gates. Then, we use De Morgan’s law to simplify the Boolean formula to a form with only OR and AND gates, marked by variables or their negation. The next step is to transform each gate node from top to bottom and from left to right, which can be seen as recursive. First, we initialize a global counter variable $c = 2$ and increase the value of $c$ by one after each door is marked. Next, we consider the specific operation flow of one of the doors. The outputs of wire $x$ are marked as $x_1 = (1, 0, \ldots, 0)$ and $x_2 = (0, 1, \ldots, 0)$, and its sub-inputs are $y$ (marked as $y_1$ and $y_2$) and $z$ (marked as $z_1$ and $z_2$). The marker vectors of $y$ and $z$ are calculated from the marker vector of $x$ according to the type of gate (as listed below). We calculate in turn until all the gate nodes are converted and put each vector of the lowest layer into a matrix $G$ from top to down:

- **AND gate**: $y_1 = 0^c || 1^{|O^{k-1}}$, $y_0 = x_0, z_1 = x_1 - y_1, z_0 = x_0 - y_1$.
- **OR gate**: $y_1 = x_1, y_0 = 0^c || 1^{|O^{k-1}}, z_1 = x_1 - y_0, z_0 = x_0 - y_0$.

2.2. The Concept of CP-ABE

A CP-ABE scheme is defined by four PPT algorithms shown below, which have been adapted from the literature [9]:

- **Setup** ($1^\lambda, 1^\mu$): The setup algorithm is an algorithm whose input is the security parameter $\lambda$, which is the input length $l$ of the function $f$. Moreover its outputs are a master secret key MSK and a collection of exposed parameters PK.
- **Encrypt** (PK, $\mu$, $f$): The encryption program is an algorithm whose inputs are a public key PK, a message $\mu$, and a function $f$, and the output is a ciphertext CT.
- **Keygen** (MSK, $x$): The keygen algorithm is an algorithm that outputs the private key SK$^x$ corresponding to the attribute $x$ with the input of the MSK and the attribute $x$.
- **Decrypt** (PK, SK$^x$, CT): The decryption program is an algorithm whose input is the public key PK, a secret key SK$^x$, and a ciphertext CT. If and only if the attribute $x$ satisfies $f(x) = 1$, it will output the message $\mu$; otherwise, decryption fails.

**Definition 3** ([9]). A CP-ABE scheme is correct if the equation shown below holds. Namely,

$$
P_r \left[ \begin{array}{c}
(PK, MSK) \leftarrow Setup(1^\lambda, 1^\mu) \\
SK_x \leftarrow Keygen(MSK, x) \\
CT \leftarrow Encryption(PK, \mu, f) \\
\mu' = Decrypt(PK, CT, SK^x)
\end{array} \right] = 1 - \neg(1^\lambda).
$$

In fact, CP-ABE schemes that implement the access structure $A$ using an LSSS can be seen as an extension of the above definition. Namely, $f$ is transformed into an access structure over the attribute collection. At the same time, the single attribute $x$ in the definition above becomes a set $S$. By the same token, decryption can be successful only when certain conditions are met, that is, the attribute set $S$ satisfies the access policy $A$; otherwise, decryption fails.

**Security:** Next, we present the security model of CP-ABE through the game between the challenger and the attacker shown below.
Setup. After receiving the security parameters, the adversary submits an access structure $\mathcal{A}$, and then the challenger runs the setup algorithm and provides the resulting common parameters to the adversary.

Phase 1. The adversary initiates a polynomial number of private key queries to the challenger, but a limitation of the attribute set is that the access structure $\mathcal{A}$ must not be met, and the challenger responds and gives the matching private key $SK \leftarrow \text{Keygen}(MSK, R)$.

Challenge. The adversary provides two messages of equal lengths to the challenger, and then the challenger chooses a bit $b \in \{0, 1\}$ with the probability of throwing coins at random and returns the ciphertext $CT \leftarrow \text{Encrypt}(PK, \mu_b, \mathcal{A})$ to the adversary.

Phase 2. This step is equivalent to the repetition of Phase 1.

Guess. At the end of the game the adversary outputs a guess $b'$ about $b$.

The probability advantage of an adversary $\mathcal{A}$ in the above game is defined as:

$$\text{Adv}_{\mathcal{A}}^\text{Sel}(\lambda) = |\Pr[b = b'] - 1/2|.$$  

(1)

Definition 4 ([20]). If for an arbitrary PPT adversary $\mathcal{A}$ and all $\lambda \in \mathbb{N}$, there exists a negligible function $\text{negl}(\cdot)$ such that $\text{Adv}_{\mathcal{A}}^\text{Sel} \leq \text{negl}(\cdot)$ holds, then we call the CP-ABE scheme for LSSS-realizable access structures selectively secure.

Definition 5 ([20]). If a relaxed setting is made for the key query phase of the above scheme, that is, the rows in the generating matrix $G$ corresponding to the attribute set sent by the adversary each key query must be linearly independent, namely, those attribute sets that do not meet the access policy are queried. Then it is said that the relaxed scheme is selectively secure under linear independence.

2.3. The Relevant Knowledge of Lattice

Here, we will give some lattice-related knowledge used in our CP-ABE construction. All of the relevant contents are adapted from the corresponding references. In order to save space, only an overview of lattice and Gaussian distributions is given here. For more details, please refer to [24]. A lattice whose dimension is $m$ is a discrete additive subgroup of $\mathbb{R}^m$. For a matrix $A \in \mathbb{Z}_q^{n \times m}$, where $n, m,$ and $q$ are integers greater than 0, $\Lambda_q^A(\mathcal{N}) = \{y \in \mathbb{Z}^n \mid Ay = 0 \mod q\}$ and $\Lambda_q^n(\mathcal{A}) = \{y \in \mathbb{Z}^m \mid Ay = u \mod q\}$ are defined. Assume that $\sigma$ is an arbitrary real number greater than 0. The Gaussian distribution is a distribution interpreted by a probability distribution function $\rho$. For a matrix $A \in \mathbb{R}^m$, the set $\{V \in \mathbb{R}^m \mid \rho_{\sigma, c}(V) = \sum_{y \in V} \rho_{\sigma, c}(y)\}$ is defined as a Gaussian distribution whose parameter is $\sigma$ on the set $V$, represented by the symbol $D_{\sigma, V}$. We use $\bar{D}_{\sigma, V}$ to represent the truncated Gaussian distribution, which is statistically indistinguishable from the Gaussian distribution $D_{\sigma, V}$. The $\bar{D}_{\sigma, V}$ is the same as the $D_{\sigma, V}$ except that when the $l_{\infty}$ norm exceeds $\sqrt{m\sigma}$, it will output 0.

Definition 6 (Decisional LWE [10]). Select a $\lambda$ as the security parameter, then let $n, m,$ and $q$ be functions of $\lambda$ and positive integers at the same time. Therefore, suppose there is a probability distribution $\psi = \psi(\lambda)$ (in general, $\psi$ is a distribution $D_{\sigma, \mathcal{N}}$ with the condition that $\alpha < 1$) over $Z$. The DLWE $n, m, q, \psi$ problem declares that for any $m = \text{poly}(n)$, in the case of $A \leftarrow \mathbb{Z}_q^{n \times m}, s \leftarrow \mathbb{Z}_q^n, u \leftarrow \mathbb{Z}_q^n,$ and $e \leftarrow \psi^n$, there will be a pair of distributions that are computationally indistinguishable as shown below:

$$(A, s^T A + e^T) \approx (A, u^T).$$
Lemma 1 (Leftover Hash Lemma [25]). Let \( n \) and \( q \) be two functions from \( \mathbb{N} \) to \( \mathbb{N} \). Therefore, let \( m > (n + 1) \log q + \omega(\log n) \) and \( l = l(n) \) be a function of \( n \). Then, there are the two distributions shown below are statistically indistinguishable:

\[
\mathcal{D}_1 \equiv \{(B, BS) \mid B \leftarrow \mathbb{Z}_q^{n \times m}, S \leftarrow \{-1, 1\}^{m \times l}\}
\]

\[
\mathcal{D}_2 \equiv \{(B, R) \mid B \leftarrow \mathbb{Z}_q^{n \times m}, R \leftarrow \mathbb{Z}_q^{l \times 1}\}
\]

Lemma 2 (Smudging [24]). Suppose \( n \) belongs to \( \mathbb{N} \). For arbitrary \( c \in \mathbb{Z}^n \) and arbitrary real \( \delta \geq \omega(\sqrt{\log n}) \), there is always \( \text{SD}(\mathcal{D}_{\mathbb{Z}_n^q, \delta}, \mathcal{D}_{\mathbb{Z}_n^q, \delta, \epsilon}) \leq ||c||/\delta \).

Lemma 3 (Lattice Trapdoor [25]). A lattice trapdoor sampler is made up of a pair of algorithms, and the specific meaning and features are shown below:

1. \( \text{TrapGen}(1^n, 1^m, q) \rightarrow (B, T_B) \): The purpose of the algorithm is to output a matrix \( B \) and its corresponding trapdoor \( T_B \), when the input is the dimension \( n, m \) of the matrix and \( q \). The output matrix \( B \) of this algorithm is indistinguishable from the matrix \( C \) of the same dimension which is randomly selected evenly in the same number field.

2. \( \text{SamplePre}(B, T_B, \delta, u) \): The goal of the algorithm is that when the input is matrix \( B \) and its corresponding trapdoor \( T_B \), a parameter \( \delta \in \mathbb{R} \), \( u \), the algorithm can output a vector \( t \in \mathbb{Z}_q^m \), and the vector \( t \) must satisfy two conditions of \( B \cdot t = u \) and \( ||t|| \leq \sqrt{m} \cdot \delta \). The pre-image \( t \) output by the algorithm is indistinguishable from the vector \( d \) selected according to the Gaussian distribution with parameter \( \delta \).

Lemma 4 (SampleLeft [24]). Let \( m > n, q > 2 \) and suppose \( C, D \in \mathbb{Z}_q^{n \times m} \) are two full rank matrices; \( T_C \) is a trapdoor matrix corresponding to \( C \). For a matrix \( P \in \mathbb{Z}_q^{n \times y} \) and \( \theta \geq ||T_C|| \cdot \omega(\sqrt{\log m}) \) (where \( T_C \) is the Gram–Schmidt orthogonalization \( T_C \)), there must be a PPT algorithm \( \text{SampleLeft} (C, T_C, D, P, \theta) \) whose output is a matrix \( Y \in \mathbb{Z}_q^{n \times y} \), and the matrix \( Y \) is statistically indistinguishable from the distribution of \( \mathcal{D}_{\mathbb{Z}_q^{n \times m}}(C|D), \sigma \).

Definition 7 (New Two-Stage Sampling Method [24]). Let \( m > n \geq 1, q \geq 2, \) and all are integers. Consider using \( \delta, \sigma \in \mathbb{R} \) and an arbitrary small norm matrix \( S \in \mathbb{Z}_q^{m \times m} \) as inputs, and the sampler operates sequentially according to the following two stages:

1. The first stage: (this stage has nothing to do with \( S \)).
   - Select a matrix at a random \( A \leftarrow \mathbb{Z}_q^{n \times m} \);
   - Select a vector at a random \( z \leftarrow \mathbb{Z}_q^n \);

2. The second stage:
   - Select a vector \( j \leftarrow \mathcal{D}_{\mathbb{Z}_q^{n \times m}, \delta}; \)
   - Calculate \( k = z - A j \mod q \);
   - Sample a random vector \( k' = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \leftarrow \mathcal{D}_{\mathbb{Z}_q^{m \times m}, \sigma} \) to make the equation \( (A, AS) \)
     \[
     \begin{bmatrix}
     k_1 \\
     k_2
     \end{bmatrix} = k \mod q \) hold;
   - Let \( h = \begin{bmatrix} j + k_1 \\ k_2 \end{bmatrix} \in \mathbb{Z}_q^{2m}, \) make the equation \( (A, AS)h = z \mod q \) hold;
   - Output the tuple \( (A, AS, h, z) \).

   This is an example of a quote.

3. Our Ciphertext-Policy ABE Scheme for NC^1 Circuits Based on LWE

Here, we are going to show our CP-ABE scheme, which supports the access structure described through the NC^1 circuit. The specific features and construction have been shown in Section 2.1.2.
First of all, we give the specific restrictions of each parameter in the scheme according to the proof of correctness and security. The Setup algorithm should select the appropriate parameters and noise distributions \( n, m, \sigma, q \) and \( \psi_{lwe}, \psi_1, \psi_{big} \), which meet the following conditions (fix any \( 0 < \epsilon < 1/2 \)):

- \( \sigma < q, n = f(\lambda), \psi_{lwe} = D_{\mathbb{Z}_{q}, n \cdot q}/\sigma < 2^{n^\epsilon} \) (due to the LWE security)
- \( m > t_{\max} \log q + \omega(\log n) + \lambda \) (due to the leftover hash lemma)
- \( \sigma \geq O(\sqrt{t_{\max} \log q}) \cdot \omega(\sqrt{\log m}) + \lambda \) (due to the trapdoor sampling)
- \( \psi_1 = D_{\mathbb{Z}_{q}^{m-1}}, \psi_2 = D_{\mathbb{Z}_{q}^{m}} \) (due to the trapdoor sampling)
- \( \psi_{big} = D_{\mathbb{Z}_{q}^{m \cdot f}}, \sigma = \sqrt{\frac{\log m}{\lambda}} + \beta \) (due to the smudging/security)
- \( |U| \cdot [\sqrt{2m^{3/2}}(\sqrt{2\sigma^2 + 2\sigma^2 + \sigma^2}) + m^{3/2}] < q/4 \) (due to the correctness)

**Setup(1^1):** Take the security parameter \( \lambda \), the maximum width \( t_{\max} = t_{\max}(\lambda) \) of a linear secret sharing matrix corresponding to the scheme, and the attribute set \( U \) as inputs and perform the following steps in order:

1. Select an LWE instance with the appropriate parameters \( n, m, q, \sigma \) and noise distributions described above.
2. Select a vector \( x \leftarrow \mathbb{Z}_{q}^{n} \) and select a series of random matrices \( \{P_{u}\}_{u \in U} \).
3. Run algorithm \( \text{TrapGen}(1^n, 1^m, q) \) to generate \( \{A_{u}, T_{A_{u}}\}_{u \in U} \).
4. Output the public parameters and master secret keys:

\[
PK = (n, m, q, \psi_{lwe}, \psi_2, \psi_{big}, x, \{A_{u}\}_{u \in U}, \{P_{u}\}_{u \in U}), MSK = (\{T_{A_{u}}\}_{u \in U}).
\]

**KeyGen(MSK, U):** With the master secret key and the collection of attributes \( U \in U \) as inputs, perform the following steps:

1. Sample a \( \tilde{w} \leftarrow \psi_{1} \) and set the vector \( w = (1, \tilde{w}) \in \mathbb{Z}_{q}^{m} \).
2. For each \( u \in U \), sample matrix \( f_{u} \leftarrow D_{\mathbb{Z}_{q}^{m}, \sigma} \), at the same time, and, finally, let \( z_{u} = P_{w} w - A_{u} f_{u} \mod q \).
3. Sample \( S_{u} \leftarrow \{-1, 1\}^{m \times m} \).
4. Sample \( \left[ \begin{array}{c} k_{u} \\ g_{u} \end{array} \right] \leftarrow \text{SampleLeft}(A_{u}, A_{u} S_{u}, T_{A_{u}}, z_{u}) \) for some parameter \( \sigma \). In other words, choose \( \left[ \begin{array}{c} k_{u} \\ g_{u} \end{array} \right] \) to make the equation \( A_{u} A_{u} S_{u} \cdot \left[ \begin{array}{c} k_{u} \\ g_{u} \end{array} \right] = z_{u} \) holds.
5. Let \( h_{u} = \left[ \begin{array}{c} k_{u} + f_{u} \\ g_{u} \end{array} \right] \) and output \( SK_{f} = (w, \{h_{u}\}_{u \in U}) \).

**Enc(PK, \mu, (G, \eta)):** With the public parameters \( PK \), a message \( \mu \in \{0, 1\} \) needs to be encrypted, and an LSSS access policy \((G, \eta)\), generated from the previous transformation, are the inputs (that is to say, where \( G = (G_{i,j})_{y \times t_{\max}} \in \{-1, 0, 1\}^{y \times t_{\max}} \subset \mathbb{Z}_{q}^{y \times t_{\max}} \)). Perform the following steps in order:

1. Sample vector \( s \leftarrow \mathbb{Z}_{q}^{m} \) and error vector \( \{v_{i}\}_{i \in \{2, \ldots, t_{\max}\}} \leftarrow \mathbb{Z}_{q}^{m} \).
2. In addition, continue to sample \( \{e_{i}\}_{i \in \{y\}} \leftarrow \psi_{lwe}^{2m} \) and \( \{\tilde{e}_{i}\}_{i \in \{y\}} \leftarrow \psi_{big}^{2m} \) for every \( i \in \{y\} \). Calculate \( c_{i} \in \mathbb{Z}_{q}^{2m}, \tilde{c}_{i} \in \mathbb{Z}_{q}^{m} \) as described below.

\[
\begin{align*}
c_{i}^T &= s^T (A_{\eta_{y}}(s) A_{\eta_{y}}(s)) + e_{i}^T  \\
\tilde{c}_{i}^T &= G_{i,1}(s, x, 0, \ldots, 0) + \sum_{m-1}^{t_{\max}} G_{i,j}(s, x, 0, \ldots, 0) - s^T P_{\eta_{y}} + \tilde{e}_{i}^T
\end{align*}
\]

3. Output the ciphertext \( ct = ((G, \eta), \{c_{i}\}_{i \in \{y\}}, \{\tilde{c}_{i}\}_{i \in \{y\}}, C = MSB(s^T x) \oplus \mu) \).

**Dec(SK, ct):** With the private key and a ciphertext as inputs, perform the following algorithmic steps in order:

1. In the case that the vector \((1, 0, \ldots, 0)\) cannot be represented by the rows of \( G \) correlated to the attribute set \( U \), the decryption will fail.
3.1. Correctness

Next, we will prove the correctness of the scheme. Given an attribute collection $U \subseteq \mathbb{U}$ and an arbitrary access policy $(G, \eta)$, where $U$ forms an authorized set, calculate the following equation

$$Y' = \sum_{i \in \Delta} \xi_i (c_i^T h_u + \hat{c}_i^T w).$$

By substituting the specific values of $\{ c_i \}_{i \in [y]}$ and $\{ \hat{c}_i \}_{i \in [y]}$ into the above formula, we can obtain

$$Y' = \sum_{i \in \Delta} \xi_i s^T [A_{\eta(i)} \| A_{\eta(i)} S_{\eta(i)}] h_{\eta(i)} + \sum_{i \in \Delta} \xi_i (s^T x, 0, \ldots, 0)w + \sum_{i \in \Delta} \xi_i G_{i}j_v^T w - \sum_{i \in \Delta} \xi_i s^T P_{\eta(i)} w + \sum_{i \in \Delta} \xi_i e_i h_{\eta(i)} + \sum_{i \in \Delta} \xi_i \hat{e}_i w.$$

Here, let us review the previous key generation process. For each $u \in U$, $h_u = \begin{bmatrix} k_u & j_u \\ g_u \end{bmatrix}$, and $[A_u \| A_u S_u] : \begin{bmatrix} k_u \\ g_u \end{bmatrix} = P_u w - A_u j_u$ is established. That is to say, it holds that

$$[A_{\eta(i)} \| A_{\eta(i)} S_{\eta(i)}] h_{\eta(i)} = P_{\eta(i)} w.$$

Further, there will be

$$Y' = \sum_{i \in \Delta} \xi_i s^T P_{\eta(i)} w + \sum_{i \in \Delta} \xi_i G_{i}j_v^T w + \sum_{i \in \Delta} \xi_i G_{i}j_v^T w - \sum_{i \in \Delta} \xi_i s^T P_{\eta(i)} w + \sum_{i \in \Delta} \xi_i e_i h_{\eta(i)} + \sum_{i \in \Delta} \xi_i \hat{e}_i w = \sum_{i \in \Delta} \xi_i G_{i}j_v^T w + \sum_{i \in \Delta} \xi_i G_{i}j_v^T w + \sum_{i \in \Delta} \xi_i e_i h_{\eta(i)} + \sum_{i \in \Delta} \xi_i \hat{e}_i w = \sum_{i \in \Delta} \xi_i G_{i}j_v^T w.$$

According to the above description, the formula $\sum_{i \in \Delta} \xi_i G_{i}j_v = 1$ holds. Moreover, for $1 < l \leq t_{\text{max}}$, $\sum_{i \in \Delta} \xi_i G_{i,j} = 0$. Additionally, because of $w = (1, \hat{w})$, then calculate $(s^T x, 0, \ldots, 0)w = s^T x$. Thus,

$$Y' = s^T x + \sum_{i \in \Delta} \xi_i e_i h_{\eta(i)} + \sum_{i \in \Delta} \xi_i \hat{e}_i w.$$

The correctness is based on the fact that the last two terms of the above two equations are very small and will not affect the MSB of $s^T x$. In order to achieve this, we must make the following inequalities hold with extremely overwhelming probabilities:

- $\| e_i \| \leq \sqrt{2mr}$ : This is because the $2m$ coordinates of $e_i$ are selected according to $D_{\mathbb{Z}, \sigma}$.
- $\| \hat{e}_i \| \leq \sqrt{m\delta}$ : The reason for this is precisely because all the coordinates of $\hat{e}_i$ are selected according to $D_{\mathbb{Z}, \hat{d}}$, where $\delta \geq c\sqrt{m\|R\|} \cdot \lambda^{m(1)}$. 
Appl. Sci. 2023, 13, 8043

- \| \mathbf{h}_{q}(i) \| \leq m \sqrt{2\sigma^2 + 2\sigma\delta + \delta^2} : This is because the equation \( \mathbf{h}_{q}(i) = \left[ \mathbf{k}_{q}(i) + i\mathbf{g}_{q}(i) / b_{q}(i) \right] \) holds where (1) the selection of \( \left[ \mathbf{k}_{u} / b_{u} \right] \) is based on a distribution that is statistically indistinguishable from \( \mathcal{D}_{\mathbb{Z}^{n},\mathcal{g}} \). Additionally, (2) \( j_u \leftarrow \mathcal{D}_{\mathbb{Z}^{n},\mathcal{g}} \).

- \| w \| < m\sigma : The reason is that \( w = (1, \tilde{w}) \), and \( \tilde{w} \) is selected by a distribution \( \mathcal{D}_{\mathbb{Z}^{n-1},\mathcal{g}} \).

- Based on the fact that \( \xi_i \in \{0, 1\} \), we can obtain:

\[
\sum_{i \in \Lambda} \xi_i e_i^\top \mathbf{h}_{q}(i) + \sum_{i \in \Delta} \xi_i e_i^\top w < |U| |\sqrt{2m^{3/2}c(\sqrt{2\sigma^2 + 2\sigma\delta + \delta^2}) + m^{3/2}\sigma\delta}| < q/4.
\]

3.2. Security Analysis

In this part, we will give the security analysis of the scheme in detail. The security proof of our scheme is carried out strictly according to the idea of the security proof in [20].

**Theorem 1** ([20]). Under the condition that the LWE hypothesis is true, the CP-ABE scheme we present for all access structures described through NC^1 circuits is selectively secure.

**Theorem 2.** Under the condition that the LWE assumption is true, the CP-ABE scheme we present is selectively secure under linear independence restriction.

**Proof.** To prove the correctness of Theorem 2, we considered a series of hybrid games, which are slightly different from each other. The first hybrid game matches the real choice of adversaries and the final hybrid game is the one in which the advantage of the opponent is zero. By demonstrating that the advantages of the adversaries have only a completely negligible change between different consecutive hybrid games, we have finally proved the correctness of Theorem 2. Next, we will talk about the specific hybridization and analysis process. In order to facilitate the explanation, we first use 12 algorithm graphs to show, in detail, the process of 12 hybridizations in which all the symbols used are as defined above.

Secondly, we analyze the indistinguishability of any two adjacent hybridization processes in detail. For convenience, we introduce a probability function \( \beta_{A,x}(\lambda) : \mathbb{N} \rightarrow [0, 1] \) for all adversaries \( A \), and any \( x \in \{0, 1, \ldots, 11\} \) whose input is \( l^A \) and output is the probability of the opponent \( A \) correctly guesses the challenge bit \( b \in \{0, 1\} \) in each hybrid program. According to the definition of Algorithm 1, it is known that \( |\beta_{A,0}(\lambda) - 1/2| = \text{Adv}^\text{Sel}_{A}(\lambda) \) for all \( \lambda \in \mathbb{N} \). Furthermore, since the generated challenge ciphertext in Algorithm 12 does not disclose any information about the selection of challenge bits, it is known that \( \beta_{A,11} = 1/2 \) for all \( \lambda \in \mathbb{N} \).

Therefore, through the following series of analysis, we can conclude that for all \( \lambda \in \mathbb{N} \), there is obviously

\[
\text{Adv}^\text{Sel}_{A}(\lambda) \leq \sum_{x \in \{0, 1\}} |\beta_{A,x-1}(\lambda) - \beta_{A,x}(\lambda)| \leq \text{negl}(\lambda) \tag{2}
\]

**Hybrid 0:** This is the original selective weak security model of the CP-ABE scheme we constructed.

**Hybrid 1:** This is similar to Algorithm 1, except for the three parts marked with a wireframe in Algorithm 2, and these changes are only semantic (As in Algorithm 1, \( \{v_1\}_{l \in \{2, \ldots, \text{max}\}} \leftarrow \mathbb{Z}_q^m \)) Algorithm 2 replaces the original vector \( \{v_i\}_{l \in \{2, \ldots, \text{max}\}} \) with \( \tilde{v}_i \mathbf{D}_l \), where \( \{v_i\}_{l \in \{2, \ldots, \text{max}\}} \leftarrow \mathbb{Z}_q^m \) and \( \{\mathbf{D}_l\}_{l \in \{2, \ldots, \text{max}\}} \leftarrow \mathbb{Z}_q^{n \times m} \). The semantic changes involved later are also similar to the methods here), so there is obviously \( \beta_{A,0}(\lambda) = \beta_{A,1}(\lambda) \) for all adversaries \( A \).

**Hybrid 2:** This is similar to Algorithm 2, except for the two parts marked with a wireframe in Algorithm 3, and these changes are only semantic, so there is obviously \( \beta_{A,1}(\lambda) = \beta_{A,2}(\lambda) \) for all adversary \( A \).
Hybrid 3: This is similar to Algorithm 3, except for those parts marked with a wireframe in Algorithm 4. The indistinguishability of Algorithm 3 and Algorithm 4 stems from the leftover hash lemma (as shown in Lemma 1) and the basic fact that the sum of two random matrices is still a random matrix. So, there is obviously \( \| \beta_{A3}(\lambda) - \beta_{A4}(\lambda) \| \leq \text{negl}_{5}(\lambda) \) for all adversaries \( A \).

Hybrid 4: This is similar to Algorithm 4, except for those parts marked with a wireframe in Algorithm 5. It chooses another indistinguishable way to generate \( \{D_j\}_{j \in [2..\text{max}]} \), which stems from the trapdoor lattice sampler. So, there is obviously \( \| \beta_{A5}(\lambda) - \beta_{A6}(\lambda) \| \leq \text{negl}_{4}(\lambda) \) for all adversaries \( A \).

Hybrid 5: This is similar to Algorithm 5, except for several parts marked with wireframes in Algorithm 6, which can be summarized as the way in which \( \begin{bmatrix} k_u \\ g_u \end{bmatrix} \) is generated when answering the private key query of the adversary \( A \) for all \( u \in U \cap \eta(y) \). This indistinguishability stems from the smudging lemma (Lemma 2), so we can deduce that \( \| \beta_{A4}(\lambda) - \beta_{A5}(\lambda) \| \leq Q_{\text{key}}(\lambda) \cdot |U \cap \eta(y)| \cdot 2m \cdot \text{negl}_{\text{smu}}(\lambda) \leq \text{negl}_{5}(\lambda) \) for all adversaries \( A \) where \( \text{negl}_{\text{smu}} \) is a negligible function to measure the indistinguishability of two distributions under smudging and \( Q_{\text{key}} \) represents the number of private key queries made. For example, for all of the \( m \in \mathbb{N} \) and \( \lambda \in \mathbb{N} \), when \( F_1 \equiv \{k_1 | k_1 \leftarrow D_{2^n,\beta} \}, F_2 \equiv \{k_1 + k_2 | k_1 \leftarrow D_{2^n,\beta}, k_2 \leftarrow D_{2^n,\beta} \} \), there is \( \text{SD}(F_1, F_2) \leq m \cdot \text{negl}_{\text{smu}}(\lambda) \) (where \( \delta \geq \sigma \sqrt{m} \cdot \| \cdot \|_{\text{nn}}(1) \)).

Hybrid 6: It is basically similar to Algorithm 6, just except the generation method of the vector \( t \) is different when answering the adversary \( A \). Here, in order to achieve the same purpose, we use the idea of Hybrid 6 in [20]. Namely, let \( b = (b_1, \ldots, b_{\text{max}}) \) be a vector such that \( b_1 = 1 \) and \( \sum_{i \in [2, \ldots, \text{max}]} G_{\eta^{-1}(u)} b_1 = 0 \) for all \( u \in U \). Note that by the game restriction, the set of rows of \( G \) with indices in \( \eta^{-1}(U) \) must be unauthorized with respect to the access policy \( (G, \eta) \), and, hence, the existence of such a vector \( b \) is guaranteed. We can find that the indistinguishability of Algorithm 6 and Algorithm 7 comes from the excellent characteristics of the trapdoor sampling of lattice. So, there is obviously \( \| \beta_{A5}(\lambda) - \beta_{A6}(\lambda) \| \leq \text{negl}_{6}(\lambda) \) for all adversaries \( A \).

Hybrid 7: This is similar to Algorithm 7, except for several parts marked with wireframes in Algorithm 8. Similar to the methods and goals of Hybrid 7 in [24], here, due to the restrictions of the game, for each secret key query of \( A \) corresponding to some attribute set \( U \subset U \), the rows of \( G \) with indices in \( \eta(U) \) are linearly independent. These changes can also be seen as a simple semantic, so they are indistinguishable; therefore, there is obviously \( \| p_{A6}(\lambda) - p_{A7}(\lambda) \| \leq \text{negl}_{7}(\lambda) \) for all adversaries \( A \).

Hybrid 8: This is similar to Algorithm 8, except for several parts marked with wireframes in Algorithm 9, which can be summarized as the way in which \( \begin{bmatrix} k_u \\ g_u \end{bmatrix} \) is generated when answering the private key query of the adversary \( A \). The indistinguishability of Algorithm 8 and Algorithm 9 is derived from the excellent characteristics of the trapdoor sampling of lattice. So, there is obviously \( \| p_{A7}(\lambda) - p_{A8}(\lambda) \| \leq \text{negl}_{8}(\lambda) \) for all adversaries \( A \).

Hybrid 9: This is similar to Algorithm 9, except for the part marked with wireframes in Algorithm 10. Namely, the way to generate the matrices \( \{A_u\}_{u \in \rho(y)} \) has been replaced. The indistinguishability of Algorithm 1 and Algorithm 2 is derived from the excellent characteristics of the trapdoor sampling of lattice as described earlier. So, there is obviously \( \| \beta_{A8}(\lambda) - \beta_{A9}(\lambda) \| \leq \text{negl}_{9}(\lambda) \) for all adversaries \( A \).

Hybrid 10: This is similar to Algorithm 10, except for the two parts marked with wireframes in Algorithm 11. Namely, the way to generate the vectors \( \hat{e}_y \) in the \( \{\hat{e}_y\}_{y \in \rho(y)} \) has been replaced. The indistinguishability of Algorithm 10 and Algorithm 11 is derived from the smudging lemma (Lemma 2). So, there is obviously \( \| \beta_{A9}(\lambda) - \beta_{A10}(\lambda) \| \leq y \cdot m \cdot \text{negl}_{\text{smu}} \leq \text{negl}_{10}(\lambda) \) for all adversaries \( A \).

Hybrid 11: This is basically the same as Algorithm 11, except for a few framed parts, which are summarized as different ways of generating ciphertext in the challenge phase.
Algorithm 1 (Hybrid 0)

Setup phase:
1. $s \leftarrow \mathbb{Z}_q^n$
2. $(\tilde{A}_u, \tilde{T}_u)_{u \in \mathbb{Z}_q^n} \leftarrow \text{TrapGen}(\tilde{A}_u, \tilde{T}_u)$
3. $\{\rho_u\}_{u \in \mathbb{Z}_q^n} \leftarrow \mathbb{Z}_q^n$
4. $\Pi \leftarrow (w, \Pi)$
5. $\eta \leftarrow \mathbb{Z}_q^n$
6. $w \leftarrow U \cdot \eta$ (SampleLeft($A_u, \eta, \Pi, T_u, A_{u'}$, $s$, $\epsilon$)).

Challenge phase:
7. $x \leftarrow \mathbb{Z}_q^n$
8. $\epsilon \leftarrow (w, \Pi)$

Algorithm 2 (Hybrid 1)

Setup phase:
1. $s \leftarrow \mathbb{Z}_q^n$
2. $(\tilde{A}_u, \tilde{T}_u)_{u \in \mathbb{Z}_q^n} \leftarrow \text{TrapGen}(\tilde{A}_u, \tilde{T}_u)$
3. $\{\rho_u\}_{u \in \mathbb{Z}_q^n} \leftarrow \mathbb{Z}_q^n$
4. $\Pi \leftarrow (w, \Pi)$
5. $\eta \leftarrow \mathbb{Z}_q^n$
6. $w \leftarrow U \cdot \eta$ (SampleLeft($A_u, \eta, \Pi, T_u, A_{u'}$, $s$, $\epsilon$)).

Challenge phase:
7. $x \leftarrow \mathbb{Z}_q^n$
8. $\epsilon \leftarrow (w, \Pi)$

Algorithm 3 (Hybrid 2)

Setup phase:
1. $s \leftarrow \mathbb{Z}_q^n$
2. $(\tilde{A}_u, \tilde{T}_u)_{u \in \mathbb{Z}_q^n} \leftarrow \text{TrapGen}(\tilde{A}_u, \tilde{T}_u)$
3. $\{\rho_u\}_{u \in \mathbb{Z}_q^n} \leftarrow \mathbb{Z}_q^n$
4. $\Pi \leftarrow (w, \Pi)$
5. $\eta \leftarrow \mathbb{Z}_q^n$
6. $w \leftarrow U \cdot \eta$ (SampleLeft($A_u, \eta, \Pi, T_u, A_{u'}$, $s$, $\epsilon$)).

Challenge phase:
7. $x \leftarrow \mathbb{Z}_q^n$
8. $\epsilon \leftarrow (w, \Pi)$

In Algorithm 11, $\{e_i\}_{i \in [y]}$ resembles a set of LWE$_{n,k,\epsilon}$ samples, and, here, we really want to use LWE to hide $s$. Assume that for any $i \in [y]$, $[\tilde{A}_u, \tilde{T}_u] = [a_m, a_{m+1}, \ldots, a_{2m}]$, then $c_i = s[a_m, a_{m+1}] + e_i = s[a_{m+1} + e_i]$. Because of the decisional LWE, we have that the selection of $\{e_i\}_{i \in [y]}$ in Algorithm 11 and Algorithm 12 is indistinguishable. In addition, Algorithm 12 makes a semantic transformation with $\tilde{v} = s + v_i$ so that $\{\tilde{v}_i\}_{i \in [y]}$ gets rid of $s$ and has the same distribution as the selection in Algorithm 11. In turn, if we look at $sx^T$, because of $x \leftarrow \mathbb{Z}_q^n$, we can make $a_{2m+1} = x$ and randomly select $e_{2m+1} \leftarrow D_{\mathbb{Z}_q^n}$. Then, there must be MSB$(sx^T) = MSB$(ex$^T$) + e_{2m+1}. Then, according to the decisional LWE$_{n,k,\epsilon}$, we have that the distribution of MSB$(sx^T)$ is indistinguishable from that of MSB$(\tau)$ where $\tau \leftarrow \mathbb{Z}_q^n$. To sum up, the indistinguishability of Algorithm 11 and Algorithm 12 is derived from the LWE assumption. So, there is obviously $\|\beta_{1}(\lambda) - \beta_{11}(\lambda)\| \leq \text{negl}_1(\lambda)$ for all adversaries $A$. □
Algorithm 4 (Hybrid 3)

Setup phase:
1: $s = 2^q$
2: $(A_u, T_{nu})_{u \in U}$ ← $\text{TrapGen}(\mathbb{Z}^m, \mathbb{Z}^q)$.
3: $(d) \in \mathbb{Z}_{2^m-1} \leftarrow 2^q$
4: $(P_u, l_{[u]}) \in \mathbb{Z}^{2^m-1}$
5: $\forall u \in \{u \in U | A_u \neq 0\}$, $P_u = A_u R_u$
6: $u \in \emptyset$
7: $u \in \emptyset$
8: $u \in \emptyset$
9: $\forall u \in \emptyset$
10: $\forall u \in \emptyset$

$\text{QUERY PHASE}$:
1: $(p) \in \mathbb{Z}^m + \psi_{1/2}$
2: $u \leftarrow u$
3: $w = w$
4: $u \leftarrow u$
5: $u \leftarrow u$
6: $\forall u \in \emptyset$
7: $u \leftarrow u$
8: $u \leftarrow u$

Algorithm 5 (Hybrid 4)

Setup phase:
1: $s = 2^q$
2: $(A_u, T_{nu})_{u \in U}$ ← $\text{TrapGen}(\mathbb{Z}^m, \mathbb{Z}^q)$.
3: $(d') = (d_1', \ldots, d_{m-1}') \leftarrow \text{TrapGen}(\mathbb{Z}_{2^m-1}^{m-1}, \mathbb{Z}^q)$
4: $(d') \in \mathbb{Z}_{2^m-1}^{m-1}$
5: $\forall u \in \{u \in U | A_u \neq 0\}$, $d_u' \leftarrow d_u$
6: $u \in \emptyset$
7: $u \in \emptyset$
8: $u \in \emptyset$
9: $\forall u \in \emptyset$
10: $\forall u \in \emptyset$

$\text{QUERY PHASE}$:
1: $(p) \in \mathbb{Z}^m + \psi_{1/2}$
2: $u \leftarrow u$
3: $w = w$
4: $u \leftarrow u$
5: $u \leftarrow u$
6: $\forall u \in \emptyset$
7: $u \leftarrow u$
8: $u \leftarrow u$

Algorithm 6 (Hybrid 5)

Setup phase:
1: $s = 2^q$
2: $(A_u, T_{nu})_{u \in U}$ ← $\text{TrapGen}(\mathbb{Z}^m, \mathbb{Z}^q)$.
3: $(d') = (d_1', \ldots, d_{m-1}') \leftarrow \text{TrapGen}(\mathbb{Z}_{2^m-1}^{m-1}, \mathbb{Z}^q)$
4: $(d') \in \mathbb{Z}_{2^m-1}^{m-1}$
5: $\forall u \in \{u \in U | A_u \neq 0\}$, $d_u' \leftarrow d_u$
6: $u \in \emptyset$
7: $u \in \emptyset$
8: $u \in \emptyset$
9: $\forall u \in \emptyset$
10: $\forall u \in \emptyset$

$\text{QUERY PHASE}$:
1: $(p) \in \mathbb{Z}^m + \psi_{1/2}$
2: $u \leftarrow u$
3: $w = w$
4: $u \leftarrow u$
5: $u \leftarrow u$
6: $\forall u \in \emptyset$
7: $u \leftarrow u$
8: $u \leftarrow u$
9: $\forall u \in \emptyset$
10: $\forall u \in \emptyset$

$\text{CHALLENGE PHASE}$:
1: $s = 2^q$
2: $(d_1, \ldots, d_{m-1}) \leftarrow 2^q$
3: $(r) \in \mathbb{Z}^q$
4: $(r) \leftarrow (r)$
5: $\forall u \in \emptyset$
6: $\forall u \in \emptyset$
7: $\forall u \in \emptyset$
8: $\forall u \in \emptyset$
9: $\forall u \in \emptyset$
10: $\forall u \in \emptyset$
Algorithm 10 (Hybrid 9)

Setup phase:
1: \( x_{z, 0} \leftarrow 2^{n_{\text{z}}} \).
2: \( (A_{\text{z}}, \text{A}_{\text{z}}^{(0)}) \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
3: \( (A_{\text{y}}, \text{A}_{\text{y}}^{(0)}) \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
4: \( \theta' = \{ \theta'_{x, 0} \} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
5: \( \theta'_{x, 0} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
6: \( \theta_{x, 0} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
7: \( \gamma_{x, 0} \leftarrow \gamma^{(0)} \).
8: \( \psi_{x, 0} \leftarrow \psi^{(0)} \).
9: \( \psi_{x, 0} \leftarrow \psi^{(0)} \).
10: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \). 

Key query phase:
1: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \). 
2: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \). 
3: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).

Algorithm 11 (Hybrid 10)

Setup phase:
1: \( x_{z, 0} \leftarrow 2^{n_{\text{z}}} \).
2: \( (A_{\text{z}}, \text{A}_{\text{z}}^{(0)}) \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
3: \( (A_{\text{y}}, \text{A}_{\text{y}}^{(0)}) \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
4: \( \theta' = \{ \theta'_{x, 0} \} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
5: \( \theta'_{x, 0} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
6: \( \theta_{x, 0} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
7: \( \gamma_{x, 0} \leftarrow \gamma^{(0)} \).
8: \( \psi_{x, 0} \leftarrow \psi^{(0)} \).
9: \( \psi_{x, 0} \leftarrow \psi^{(0)} \).
10: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).

Key query phase:
1: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).
2: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).
3: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).

Algorithm 12 (Hybrid 11)

Setup phase:
1: \( x_{z, 0} \leftarrow 2^{n_{\text{z}}} \).
2: \( (A_{\text{z}}, \text{A}_{\text{z}}^{(0)}) \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
3: \( (A_{\text{y}}, \text{A}_{\text{y}}^{(0)}) \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
4: \( \theta' = \{ \theta'_{x, 0} \} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
5: \( \theta'_{x, 0} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
6: \( \theta_{x, 0} \leftarrow \text{SamplePre}^{(0)}(\emptyset) \).
7: \( \gamma_{x, 0} \leftarrow \gamma^{(0)} \).
8: \( \psi_{x, 0} \leftarrow \psi^{(0)} \).
9: \( \psi_{x, 0} \leftarrow \psi^{(0)} \).
10: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).

Key query phase:
1: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).
2: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).
3: \( \text{C}_{\text{HALLENGE PHASE}} = (\text{C}_{\text{HALLENGE PHASE}}) \).
3.3. Comparison

For a long time before, the only known way to implement an LWE-based CP-ABE scheme was to convert a circuit-based KP-ABE scheme into a CP-ABE scheme, in which the access policies and sets of attributes were represented as circuits by using general circuits. However, this transformation essentially results in a CP-ABE with restricted access policy classes, whose parameters are far from ideal.

In recent years, people have tried to make some improvements to the CP-ABE based on the general-purpose circuits described above, while trying to get rid of other assumptions, and have wanted to design solutions only based on LWE. This is a step-by-step process, and, recently, there have been only a few exciting attempts, but these schemes themselves are not very efficient, and they are only attempts at the design level. There has not been too much consideration for specific contents, such as the key size or ciphertext size; therefore, here, we also make a comparison mainly from these levels, as shown below.

The reason why we say that our scheme is at least the same as the scheme [20] under the same parameters is that we can prove that the scheme can indeed achieve selective security. However, also because the use of two-stage sampling technology has been proven able to achieve a strong security, simple function encryption. The so-called strong security means that attackers can decrypt key queries before and after arbitrary challenges are issued in simulated security experiments. Here, we only give proof that the scheme is selectively secure. At the same time, in order to better illustrate the contribution of this scheme, we compare our scheme with some previous CP-ABE schemes, as shown in Table 1 below.

Table 1. Comparison with the construction of some previous LWE-based CP-ABE schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Circuits</th>
<th>Access Structure</th>
<th>Assumption</th>
<th>Direct Design or Indirect</th>
<th>Only Based on LWE or Not</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>[9]</td>
<td>UC *</td>
<td>Circuit encoding of BGG+</td>
<td>LWE</td>
<td>Indirect</td>
<td>Yes</td>
<td>Lack security proof</td>
</tr>
<tr>
<td>[6]</td>
<td>NC¹</td>
<td>Circuit encoding of BGG+</td>
<td>Bilinear maps and LWE</td>
<td>Indirect</td>
<td>No</td>
<td>Ada-INDr</td>
</tr>
<tr>
<td>[21]</td>
<td>NC¹</td>
<td>A nearly LSSS</td>
<td>LWE and KOALA *</td>
<td>Indirect</td>
<td>No</td>
<td>Very sel-security</td>
</tr>
<tr>
<td>[22]</td>
<td>UC *</td>
<td>Circuit encoding of BGG+</td>
<td>LWE</td>
<td>Indirect</td>
<td>Yes</td>
<td>Sel-secure</td>
</tr>
<tr>
<td>[20]</td>
<td>NC¹</td>
<td>LSSS (non-monotone)</td>
<td>LWE</td>
<td>Direct</td>
<td>Yes</td>
<td>Sel-secure</td>
</tr>
<tr>
<td>Ours</td>
<td>NC¹</td>
<td>LSSS (non-monotone)</td>
<td>LWE</td>
<td>Direct</td>
<td>Yes</td>
<td>(at least) Sel-secure</td>
</tr>
</tbody>
</table>

Note *: The UC in the figure above represents the general-purpose circuit, and the KOALA represents knowledge of orthogonality assumption. Direct means that a scheme is constructed directly, while indirect indicates that a scheme needs to be transformed by borrowing other KP-ABE schemes or functional encryption schemes (FE).

Here, under the same parameters, we make a simple comparison between our scheme and the one in [20] in terms of the public key, private key, and ciphertext storage overhead. The results show that the ciphertext can be shortened to a certain extent by two-stage sampling technology. The specific comparison idea is as follows. Since the lengths of parts \((G, \eta)\) and \(C = MSB(s^\top x) \oplus \mu\) in the ciphertext of the two schemes are equal, we only compare parts \(\{c_i\}_{i \in [y]}\) and \(\{\hat{c}_i\}_{i \in [y]}\), as shown in Table 2 (since the value of the parameter “m” in our scheme is half that of the parameter “m” in their scheme, in order to make a unified comparison, we take the parameters in our scheme as the benchmark). In addition,
Table 2. Comparison between our scheme and the scheme in [20] in terms of storage overhead under the same parameters.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Public Key</th>
<th>Private Key</th>
<th>Ciphertext</th>
<th>Plaintext</th>
</tr>
</thead>
<tbody>
<tr>
<td>[20]</td>
<td>4⌈log₂ q⌉ m⌈log₂ (</td>
<td>σ</td>
<td>+</td>
<td>δ</td>
</tr>
<tr>
<td>Ours</td>
<td>2⌈log₂ q⌉ m⌈log₂ (</td>
<td>σ</td>
<td>+</td>
<td>δ</td>
</tr>
</tbody>
</table>

Table 3. Recommended parameters in our CP-ABE scheme.

<table>
<thead>
<tr>
<th>t max</th>
<th></th>
<th>U</th>
<th>λ</th>
<th>n</th>
<th>m</th>
<th>q</th>
<th>e</th>
<th>σ</th>
<th>δ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 *</td>
<td>1</td>
<td>103</td>
<td>512</td>
<td>35,032</td>
<td>2^68</td>
<td>1/7</td>
<td>828</td>
<td>2.988 × 10^9</td>
<td></td>
</tr>
<tr>
<td>1 *</td>
<td>1</td>
<td>128</td>
<td>512</td>
<td>35,082</td>
<td>2^68</td>
<td>1/7</td>
<td>854</td>
<td>3.835 × 10^9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>103</td>
<td>512</td>
<td>327,896</td>
<td>2^100</td>
<td>1/6</td>
<td>2580</td>
<td>1.1 × 10^10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>103</td>
<td>512</td>
<td>327,896</td>
<td>2^100</td>
<td>1/6</td>
<td>2580</td>
<td>1.1 × 10^10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>192</td>
<td>1024</td>
<td>435,595</td>
<td>2^85</td>
<td>1/4</td>
<td>3048</td>
<td>2.55 × 10^11</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>256</td>
<td>1024</td>
<td>737,803</td>
<td>2^90</td>
<td>1/10</td>
<td>4048</td>
<td>7.646 × 10^11</td>
<td></td>
</tr>
</tbody>
</table>

* Note: The data in rows 1 and 2 are for reference only.

3.5. A Simple Example of CP-ABE

Here, we consider a non-monotone Boolean formula \((A \land B) \lor (C \land D)\), whose root has a 1-label \((1, 0, 0, 0, 0)\) and a 0-label \((0, 1, 0, 0, 0)\). According to the construction principle of generating matrix \(G\), we can determine that the policy is

\[
G = \begin{pmatrix}
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 \\
0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & -1 & 0 & 0 \\
1 & 0 & -1 & 0 & -1 \\
0 & 1 & -1 & 0 & -1 \\
\end{pmatrix}
\]

When the parameters \(t_{\text{max}} = 5, |U| = 8, \lambda = 103, n = 512, m = 327,896, e = 1/6, \sigma = 2580, \text{ and } \delta = 1.1 \times 10^{10}\) are selected, respectively, a series of operations are carried out according to the requirements of the four steps of the scheme design in Section 3.1 above because we can easily find a set of \(\{w_i\}_{i \in [8]} = (1, 0, 1, 0, 0, 0, 0, 0)\) to make \(Gw_i = (1, 0, 0, 0, 0, 0, 0, 0)\). Because the ciphertext generated by encryption is \(\{\xi_i\}_{i \in [8]}\) and \(\{\xi_i\}_{i \in [8]}\), therefore, according to the decryption algorithm, it can be calculated \(Y' = \sum_{i \in A} \xi_i (c_i \mathbf{h}_w + \tilde{c}_i \mathbf{w})\). After a series of simplification, we can obtain \(Y' = \mathbf{s}^\top + \mathbf{e}_1^\top \mathbf{h}_{\eta(1)} + \mathbf{e}_1^\top \mathbf{t} + \mathbf{e}_3^\top \mathbf{h}_{\eta(3)} + \mathbf{e}_3^\top \mathbf{h}\). Through the
calculation, it is found that $e_1^\top h_{\eta(1)} + e_1^\top t + e_3^\top h_{\eta(3)} + e_3^\top h < 2.65 \times 10^{19} < q/4$; therefore, $\text{MSB}(Y^\prime) = \text{MSB}(sx^\top)$, so it can be decrypted successfully.

4. Conclusions

In this paper, the CP-ABE scheme of [20] is improved by using a two-stage sampling algorithm, and a selectively secure CP-ABE scheme with shorter ciphertext is obtained. We prove the correctness of the scheme and give strict proof of security. At the same time, we also give a comparison of the computational cost between this scheme and the one in [20], and, in order to better understand the scheme, we recommend several groups of parameters through a series of strict calculations. Finally, we give a simple example to demonstrate the overall operation process of the algorithm. This scheme may provide some ideas for constructing other CP-ABE and multi-authority attribute encryption schemes, and the two-stage sampling algorithm involved in this scheme may be used in other cryptographic schemes to optimize the existing performance.

Author Contributions: Conceptualization, Y.Y. and L.S.; methodology, Y.Y.; validation, Y.Y., L.S., Q.W. and K.W.; security analysis, H.C. and L.S.; investigation, Y.Y., Q.W. and K.W.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.Y. and H.C.; visualization, K.W.; data curation Q.W.; project administration, H.C.; funding acquisition, H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the First-class Discipline Construction Project of Beijing Electronic Science and Technology Institute (No: 3201017) and the National Natural Science Foundation of China (No: 61772047). The APC was funded by the First-class Discipline Construction Project of Beijing Electronic Science and Technology Institute (No: 3201017) and the National Natural Science Foundation of China (No: 61772047).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.