Hindawi
Security and Communication Networks
Volume 2022, Article ID 6849761, 11 pages
https://doi.org/10.1155/2022/6849761

Research Article

Hierarchical Identity-Based Online/Offline Encryption Scheme with Leakage Resilience

Qihong Yu,[1] **Jiguo Li**,[2] **and Sai Ji**[3]

[1]*College of Information Engineering, Suqian University, Jiangsu, Suqian 223800, China*
[2]*College of Computer and Cyber Security, Fujian Normal University, Fujian, Fuzhou 350117, China*
[3]*College of Information Engineering, Taizhou University, Jiangsu, Taizhou 225300, China*

Correspondence should be addressed to Qihong Yu; yuqhsqu@163.com

Received 19 July 2022; Revised 3 November 2022; Accepted 16 November 2022; Published 30 November 2022

Academic Editor: Barbara Masucci

The adversary is able to obtain some secret information from side channel attacks, which further damages the security for the system. To resolve this issue, we provide a hierarchical identity-based online/offline encryption scheme which resists side channel attacks. In our scheme, most encryption operations are preprocessed in the offline stage and only a small amount of lightweight calculation needs to be performed in the online stage for generating ciphertext. The presented scheme greatly reduces the workload of online encryption and is suitable for the resource-constrained device. The security of the proposed scheme is proved by the dual system technique. The leakage performance analysis shows that the presented scheme is resilient to leakage for almost the whole symmetric key.

1. Introduction

In recent years, many side channel attacks [1–5] have been presented. The adversary is able to obtain some secret information of the system by measuring timing, energy consumptions, and other characteristics from the cryptosystems. Cold start attack [6] is a special form of side channel attacks. The adversary gets part information from internal storage after the machine is shut down. Under these side channel attacks, if some secret information is exposed, the security for the whole cryptosystem is broken. Therefore, the research on leakage-resilient encryption scheme has attracted the attention of many cryptography researchers.

1.1. Related Work. Micali and Reyzin [7] first proposed the model "only computing leaks (OCL)." In the OCL model, an attacker selects an effective and computable function and inputs the internal state and key and outputs some secret information. The complexity of the leakage function is unlimited. The only limitation is that the part which is not involved in the current operation does not leak secret information. Dziembowski and Pietrzak [8, 9] presented two leakage-resilient stream cipher schemes in the OCL model.

Considering that the leakage not only occurs in the calculation process, Akavia et al. [10] presented the bounded leakage model (BLM). In BLM, the leakage function does not expose the whole private key. BLM does not solve the problem that the length of the leaked information obtained by the attacker is longer than the private key information and the amount of useful information is relatively small. To resolve the problem, Naor and Segev [11] presented "entropy bounded leakage model," in which the entropy about leakage information is limited.

Because it can reduce the burden of the key generation center, the hierarchical identity-based encryption (HIBE) schemes have been widely used and attracted the attention of many researchers. Guo et al. [12] proposed an efficient HIBE scheme with high computational efficiency by reasonably reducing the use of parameters. Langrehr and Pan [13] gave two tightly secure HIBE schemes by the matrix Diffie–Hellman assumption. Takayasu [14] presented an efficient adaptively secure revocable hierarchical identity-based encryption (RHIBE) with compact ciphertexts. Emura et al.

[15] gave a generic construction about RHIBE by using HIBE and the complete subtree method.

The hierarchical structure is conducive to protecting the confidentiality of data, and it is suitable for IoT systems, smart home systems, distributed data systems, cloud storage systems, etc. Private key delegation can also reduce the burden on the root private key generator (PKG).

We use Figure 1 to illustrate the hierarchical structure of the national medical diagnosis system. For the national medical system, the root node (the Ministry of Health, which is abbreviated as MH) generates the public key information of the user and acts as the root private key generation center to generate the private key of the secondary node (the Department of Health which is abbreviated as DH and the national hospital which is abbreviated as NH). The secondary node is responsible for generating the private key of the health bureau (HB) and the private key of the city hospital (CH). The third level city node is responsible for generating the private key of the town health center (THC) and the private key of the township hospital (TH). The medical management organization can delegate the private key to the staff, and each hospital can also delegate the private key to the doctors.

For example, one user with the identity of "A1 City: Hospital B1" can delegate the private key to the user with the identity "A1 City: Hospital B1: C1 Doctor" but cannot delegate the private key to the user with the identity of "A1 City: Hospital B2: C1 Doctor."

The delegation solves the problem that the key generation center is overloaded. In fact, the $(k-1)^{\text{th}}$ layer's user with the identity vector $I_{k-1} = (ID_1, ID_2, ..., ID_{k-1})$ generates the private key of the k^{th} layer's user with the identity vector $I_k = (ID_1, ID_2, ..., ID_{k-1}, ID_k)$.

Generally speaking, the "delegation" algorithm is given explicitly or implicitly for the HIBE schemes. The most HIBE schemes explicitly give the delegation algorithm. For example, the "Delegate" algorithm of the paper [12] is the delegation algorithm, and the "Del" algorithm of the paper [13] is the delegation algorithm. In addition, some HIBE schemes implicitly give the delegation algorithm. For example, the subalgorithm "Delegation Key Generation" of the algorithm "GenSK" in the paper [14] acts as the delegation algorithm. The "GenSK" algorithm of the paper [15] has key delegation function. In fact, the two methods are essentially the same. From Figure 1, we can see that the root user (the root key generation center, for short root KGC) mainly runs the private key generation algorithm, and the other level nodes (the subkey generation center) generally only run the private key delegation algorithm.

In order to improve the encryption efficiency, Guo et al. [16] first presented an identity-based online/offline encryption system. They repartitioned encryption processing into two phases: offline phase and online phase. For the offline phase, most encryption preprocessing is carried out and offline ciphertext is produced. In the online stage, a small amount of offline ciphertext is processed to obtain the encrypted ciphertext. Later, some online/offline encryption schemes are presented [17–20]. However, these schemes do not consider the problem of key leakage. Zhang et al. [21] put forward a leakage-resilient identity-based online/offline encryption scheme that is proposed by using the private key extension technique.

1.2. Our Motivations and Contributions. Based on the scheme [22], this paper proposes a leakage-resilient identity-based hierarchical online/offline encryption (**LR-HIOOE**) scheme. By binary extractor technology [23], our scheme can resist the leakage of the symmetric key used for message encryption in the system.

Our **LR-HIOOE** includes six algorithms: **Setup, KeyGen, Delegation, Enc$^{\text{off}}$, Enc$^{\text{on}}$**, and **Decryption.** The specific description of these algorithms is given in Section 3.1. Here, we only use them to explain the idea of our scheme which is shown in Figure 2.

Encryption consists of two phases. A large number of complex operations are carried out in the offline stage, and a small number of simple calculations are carried out in the online stage. It is very suitable for resource-constrained scenes. Offline phase operations are outsourced to cloud service providers (CSP). The online phase operation is completed by the encryptor himself on the lightweight device.

In essence, the binary extractor can transform the input source with a certain entropy into a more uniform output. The binary extractor is a component of many cryptographic primitives. Now, it has become an important tool for designing cryptography schemes against side channel attacks. In such cases, the proof of leakage resilience usually relies on the result that leakage information allowed for every call of the extractor is bounded. Medved and Standaert [24] and Chen et al. [25] show that the hardware implementation of the binary extractor can ensure that bounded leakage is allowed in the case of limited number of measurements. This also gives the reason why the binary extractor can be used as a leakage-resilient cryptography component. Our scheme uses the binary extractor to achieve leakage resilience.

Waters [26] presented the dual system technique. The private key and ciphertext have two appearances: semifunctional appearance and normal appearance. The normal key correctly decrypts two kinds of ciphertext. The semifunctional private key only decrypts the normal ciphertext correctly. Inspired by the works [27–31], we present the syntax description and security model of LR-HIBOOE. To the best of our knowledge, there is no LR-HIBOOE scheme by using the binary extractor in the literature. We propose the first LR-HIBOOE scheme by using the binary extractor. Our scheme is proved to be secure and leakage-resilient by dual system encryption technology. Overall, the binary extractor provides leakage resilience, and the dual system encryption technology provides security.

2. Preliminaries

2.1. Minimum Entropy and Statistical Distance

Definition 1. If random variables U and V are in a finite domain $\mathcal{O}$, $(1/2)\sum_{\sigma\in\mathcal{O}}|\Pr(U=\sigma)-\Pr(V=\sigma)| = SD(U,V)$ is regarded as their statistical distance for U and V.

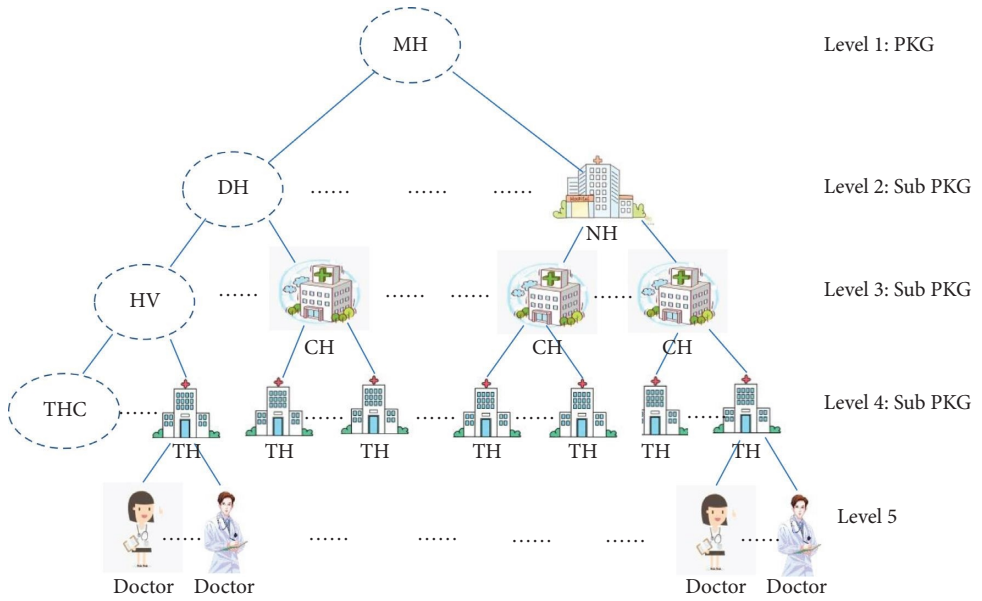

Figure 1: The hierarchical structure of the national medical diagnosis system.

Definition 2. The uncertainty measure of random variables U is called minimum entropy, which is $H_\infty(U) = -\mathrm{Log}(\max_u \Pr(U = u))$. Given variable V, the uncertainty measure of random variables U is called conditional minimum entropy, which is $\widetilde{H}_\infty(U|V) = -\mathrm{Log}(E_{v\leftarrow V}[\max_u \Pr[U = u|V = v]])$.

Conclusion 1. (see [23]): given three random variables U, V, and W in which V is λ bit length, there is $\widetilde{H}_\infty(U|(V,W)) \geq \widetilde{H}_\infty(U|W) - \lambda$. In the paper, λ denotes the leakage amount.

2.2. Binary Extractor

Definition 3. On the condition that as long as $U \in \{0,1\}^\eta$ and $H_\infty(U) > k$, there is $SD((\mathrm{Ext}(U,Q),Q),(D,Q)) \leq \varepsilon$,where ε can be ignored, D is uniform distribution of $\{0,1\}^\zeta$ and Q is uniform distribution of $\{0,1\}^\xi$, the binary function Ext: $\{0,1\}^\eta \times \{0,1\}^\xi \longrightarrow \{0,1\}^\zeta$ is called a (k,ε)-strong binary extractor [23].

2.3. Bilinear Group of Composite Order. For the algorithm which generates a bilinear group of composite order [28], it takes the system security parameter λ as input. It generates the bilinear group $\Omega = \{N = p_1p_2p_3, G, G_T, e\}$ of composite order, where p_1, p_2, and p_3 are three λ-bit length primes. The order of cyclic groups G and G_T is N. Bilinear map e meets the following two conditions.

(1) Bilinearity: For any $g,h \in G$ and $a,b \in Z_N$, we get that $e(g^a,h^b) = e(g,h)^{ab}$.

(2) Non degeneracy: $\exists g \in G$ such that $e(g,g) \neq 1$.

G_{p_1}, G_{p_2}, and G_{p_3} are used to represent subgroups with orders p_1, p_2, and p_3 in G. For example, if $h_i \in G_{p_i}$, $h_j \in G_{p_j}$, and $i \neq j$, $e(h_i,h_j)$ becomes an identity element of G_T. Further, if $h_1 \in G_{p_1}$, $h_2 \in G_{p_2}$, and g is a generator of G, $g^{p_1p_2}$ generates G_{p_3}, $g^{p_1p_3}$ generates G_{p_2}, and $g^{p_2p_3}$ generates G_{p_1}. So, there exists α_1, α_2 such that $h_1 = (g^{p_2p_3})^{\alpha_1}$, $h_2 = (g^{p_1p_3})^{\alpha_2}$, and $e(h_1,h_2) = e(g^{p_2p_3\alpha_1}, g^{p1p3\alpha 2})$, $= e(g^{\alpha_1}, g^{p_3\alpha_2})^{p_1p_2p_3} = 1$. Thus, G_{p_1}, G_{p_2}, and G_{p_3} are orthogonal.

Three assumptions are given to prove the security of our scheme.

Assumption 1. For a composite order bilinear group generation algorithm Ψ: $\Omega = (N = p_1p_2p_3, G, G_T, e) \leftarrow \Psi$, $g \leftarrow G_{p_1}$, $X_3 \leftarrow G_{p3}$, $D = (\Omega, g, X_3)$, $T_1 \leftarrow G_{p_1p_2}$, and $T_2 \leftarrow G_{p_1}$, the advantage that algorithm $\mathscr{A}$ distinguishes T_1 from T_2 is $\mathrm{Adv1}_{\Psi,A}(\lambda) = |\Pr[A(D,T_1) = 1] - \Pr[A(D,T_2) = 1]|$

If the advantages $\mathrm{Adv1}_{\Psi,A}(\lambda)$ of any probabilistic polynomial algorithm can be ignored, it is said that the algorithm Ψ satisfies Assumption 1.

T_1 expresses the product for certain element in G_{p_1} and certain element in G_{p_2}, which are called the part of G_{p_1} and the part of G_{p_2}, respectively.

Assumption 2. For a composite order bilinear group generation algorithm Ψ: $\Omega = (N = p_1p_2p_3, G, G_T, e) \leftarrow \Psi$, $g, X_1 \leftarrow G_{p_1}, X_2, Y_2 \leftarrow G_{p_2}, X_3, Y_3 \leftarrow G_{p3}$, $(D = \Omega, g, X_1X_2, X_3, Y_2Y_3)$, $T_1 \leftarrow G$, and $T_2 \leftarrow G_{p_1p_3}$, the advantage that algorithm $\mathscr{A}$ distinguishes T_1 from T_2 is $\mathrm{Adv2}_{\Psi,A}(\lambda) = |\Pr[A(D,T_1) = 1] - \Pr[A(D,T_2) = 1]|$.

If the advantages $Adv\ 2_{\Psi,A}(\lambda)$ of any probabilistic polynomial algorithm can be ignored, it is said that the algorithm Ψ satisfies Assumption 2.

Assumption 3. For a composite order bilinear group generation algorithm Ψ: $\Omega = (N = p_1p_2p_3, G, G_T, e) \leftarrow \Psi$, $\alpha, s \leftarrow Z_N$, $g \leftarrow G_{p_1}, X_2, Y_2, Z_2 \leftarrow G_{p_2}, X_3 \leftarrow G_{p3}$, $D = (\Omega, g, g^\alpha X_2, X_3, g^s Y_2, Z_2)$, $T_1 \leftarrow e(g,g)^{\alpha s}$, and $T_2 \leftarrow G_T$, the

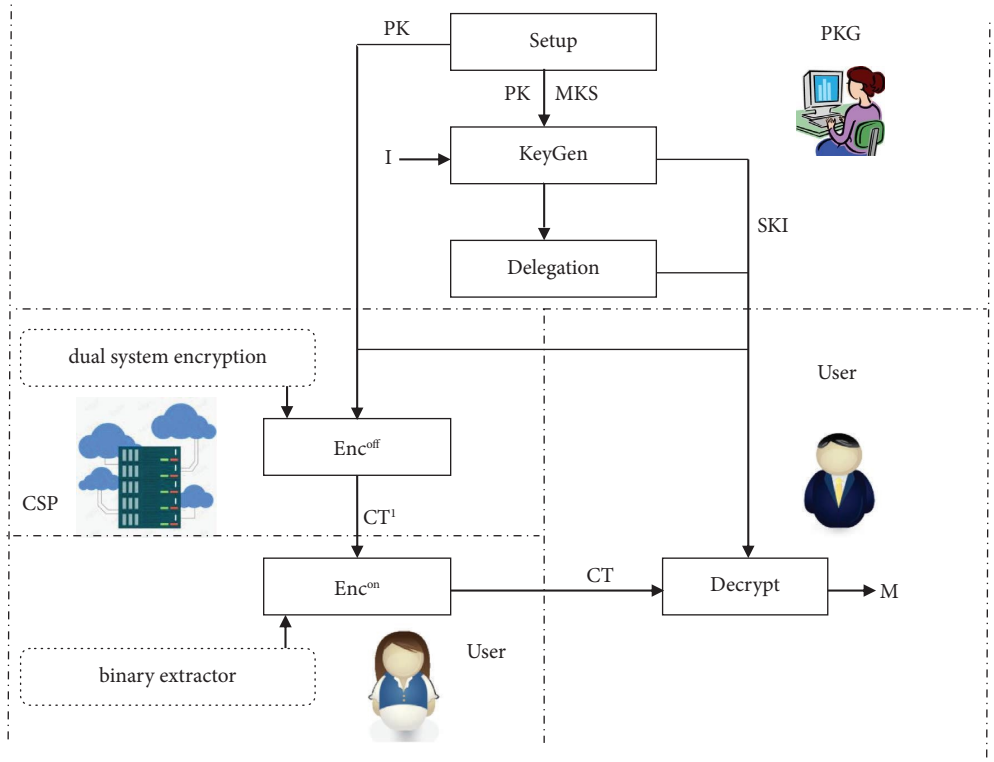

Figure 2: The idea of our LR-HIOOE.

advantage that algorithm $\mathcal{A}$ distinguishes T_1 from T_2 is $\mathrm{Adv3}_{\Psi,A}(\lambda) = |\Pr[A(D, T_1) = 1] - \Pr[A(D, T_2) = 1]|$.

If the advantages $\mathrm{Adv3}_{\Psi,A}(\lambda)$ of any probabilistic polynomial algorithm can be ignored, it is said that the algorithm Ψ satisfies Assumption 3.

3. Syntax and Security Model of LR-HIBOOE

3.1. Syntax of LR-HIBOOE. The presented **LR-HIBOOE** consists of the following six algorithms:

Setup: given security parameter λ, the algorithm generates system public parameter PK and master private key MSK. $\mathrm{Setup}(\lambda) \longrightarrow (PK, MSK)$.

KeyGen: given master private key MSK, public parameter PK, and identity vector $I = (ID_1, \ldots, ID_j)$, the private key generator generates the private key SK_I. $\mathrm{KeyGen}(PK, MSK, I) \longrightarrow SK_I$.

Delegation: the algorithm inputs system public parameter PK, the private key SK_I of identity vector $I = (ID_1, \ldots, ID_j)$, and identity ID_{j+1} and outputs the private key $SK_{I'}$ of identity vector $I' = \{I, ID_{j+1}\}$ with $j+1$ layer. $\mathrm{Delegation}(PK, SK_I, ID_{j+1}) \longrightarrow SK_{I'}$.

$\mathrm{Enc}^{\mathrm{off}}$: given system public parameter PK, the algorithm outputs the offline ciphertext CT'. $Enc^{off}(PK) \longrightarrow CT'$.

$\mathrm{Enc}^{\mathrm{on}}$: the algorithm inputs system public parameter PK, the identity vector I, the offline ciphertext CT', and the message M and produces the final ciphertext CT. $Enc^{on}(PK, M, I, CT') \longrightarrow CT$.

Decryption: the algorithm inputs PK, SK_I, and the final ciphertext CT and outputs the message M. $\mathrm{Decryption}(PK, SK_I, CT) = M$.

We use Figures 3 and 4 to show the technology roadmap of our **LR-HIBOOE**. Figure 3 shows the relations about the main algorithms of **HIBOOE** without leakage resilience. Figure 4 shows the relations about the main algorithms of **LR-HIBOOE**. In the offline encryption, we refresh the symmetric key by using an extractor.

The adversary can obtain the confidential information of the cryptography system through side channel attacks, which leads to the disclosure of the system's secret information, such as the private key information. For example, in a timing attack, the adversary can obtain relevant parameter information through the execution time of the algorithm. When the secret information is leaked, it will mainly cause a certain loss of the private key entropy. That is, the probability that the adversary guesses the private key is greater than the probability of the random guess. When the leaked information reaches a certain amount, the adversary can guess the entire private key. In order to prevent the adversary from guessing the whole private key, it is necessary to make up for the lost entropy. There are usually two ways. One is to extend the private key appropriately so that even if some entropy is lost, certain "entropy" can be retained, which will make it difficult for the adversary to guess the private key correctly. This method will make the private key longer, increase the storage cost, and increase the computing cost. The second way is to make up for the lost entropy in time, which requires an additional function "binary extractor" to rerandomize the

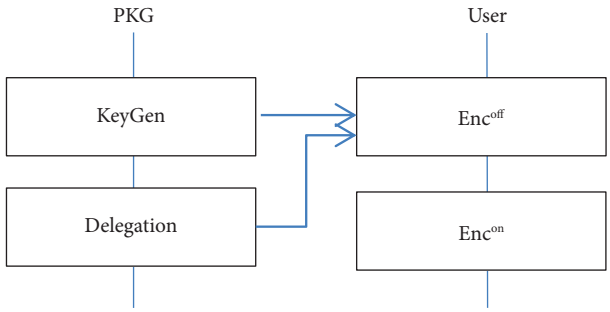

Figure 3: The relations about the main algorithms of HIBOOE without leakage resilience.

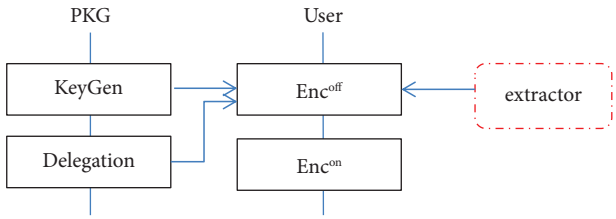

Figure 4: The relations about the main algorithms of LR-HIBOOE.

private key and make up for the lost entropy of the private key. This method does not increase the storage cost of the system, and the computing cost is almost unchanged. In our scheme, the extractor rerandomizes the symmetric key. It is used to compensate for the entropy loss of the symmetric key which is used to encrypt the plaintext.

3.2. Security Model of LR-HIBOOE. The following game Game_R which is played by the attacker and the challenger is used to describe the security of our **LR-HIBOOE** scheme.

Game_R:

Initialize: the challenger calls the algorithm "Setup" to obtain the public parameter PK and master key and gives PK to the attacker. We use W to record private keys that are given to the attacker. The initial value of set W is empty.

Phase 1: the attacker makes some queries as follows:

Private key query. the adversary gives the identity vector I to the challenger. Then, the challenger calls the private key generation algorithm to obtain SK_I and send it to the attacker. The challenger puts it into the set W.

Private key delegation query: a private key SK_I in W and an identity ID are given to the challenger by the attacker. The challenger calls the algorithm delegation to obtain the corresponding private key $SK_{I'}$ ($I' = \{I, ID_{j+1}\}$) and send it to the attacker. The challenger puts it into the set W.

Leakage query: in terms of private key SK_{I^*} for the identity vector I^*, the challenger obtains the corresponding symmetric key K of SK_{I^*} and sends it to the attacker. The attacker selects a function f which is called as leakage function. The challenger sends $f(SK_{I^*})$ to the attacker. Let ξ represent the length of the output value of $f(SK_{I^*})$.

Challenge: the attacker sends the message M_0 or M_1 and a certain identity vector I^* to the challenger, where I^* and each prefix vector of I^* are not in W. The challenger obtains the private key SK_{I^*} and randomly selects $v \in \{0,1\}$ to encrypt M_v. The challenger sends the attacker the ciphertext CT.

Phase 2: for the attacker, he makes the private key inquiry and the private key delegation inquiry with the restraint that the required identity vector cannot be a prefix vector of I^*. In addition, the attacker cannot make leakage query.

Guess: the attacker gives the guess v' about v. If $v = v'$, the attacker wins the game Game_R.

The scheme is said to be secure on the condition that the advantage which is gained by any attacker in the above game is ignored.

4. Construction of LR-HIBOOE

Assuming that the maximum layer depth is l, in the offline encryption stage, each layer needs some random numbers to generate an offline ciphertext and carry some additional information. In the online encryption stage, only l integer operations are needed to get the corresponding ciphertext. Our **LR-HIBOOE** scheme contains the following algorithms.

Setup: the algorithm runs Ψ to obtain a bilinear group G with order $N = p_1p_2p_3$. We use l to represent the maximum depth of identity vector in LR-HIBOOE. Let $\mathcal{M}$ denote message space and $|\mathcal{M}| = 2^m$. H denotes a hash function $H: \{0,1\}^* \longrightarrow \{0,1\}^m$. The algorithm randomly selects $X_3 \in G_{p_3}, h, u_1, \ldots, u_l \in G_{p_1}, \alpha \in Z_N$ and gets system public parameter $PK = \{N, H, g, h, e(g,g)^\alpha, u_1, \ldots, u_l, X_3\}$, where g is a generator of G_{p_1}. The master private key is $MSK = \alpha$.

KeyGen: private key generator (PKG) inputs system public parameters PK and identity vector $(ID_1, ..., ID_j)$. PKG randomly selects $R_3, R_3', R_{j+1}, \ldots, R_l \in G_{p_3}$ and $r \in Z_N$. PKG generates private key $K_1 = g^r R_3, K_2 = g^\alpha (u_1^{ID_1} \cdots u_j^{ID_j} h)^r R_3'$, $E_{j+1} = u_{j+1}^r R_{j+1}, \ldots, E_l = u_l^r R_l$

Delegation: the delegation algorithm inputs the private key $K_1, K_2, E_{j+1}, \ldots, E_l$ of the identity vector $(ID_1, \ldots, ID_j)$. The algorithm randomly selects $r', t' \in Z_N$ and $\widetilde{R}_{j+2}, \ldots, \widetilde{R}_l, \widetilde{R}_3, \widetilde{R}_3' \in G_{p_3}$. The algorithm generates the private key about the identity vector $(ID_1, \ldots, ID_{j+1})$: $\widetilde{K_1} = K_1 g^{r'} \widetilde{R}_3, \widetilde{K_2} = K_2 (u_1^{ID_1} \cdots u_j^{ID_j} h)^{r'} (E_{j+1})^{ID_{j+1}} u_{j+1}^{r' ID_{j+1}} \widetilde{R}_3'$, $\widetilde{K_3} = t'$, $\widetilde{E_{j+2}} = E_{j+2} u_{j+2}^{r'} \widetilde{R}_{j+2}, \ldots, \widetilde{E_l} = E_l u_l^{r'} \widetilde{R}_l$.

Enc^{off}: the algorithm randomly selects $s, t, x_1, x_2, \ldots, x_l \in Z_N$. It calculates $K = e(g,g)^{\alpha s}, C_1 = (u_1^{x_1} \ldots u_l^{x_l} h)^s, C_2 = g^s, C_{3,1} = u_1^{st}, C_{3,2} = u_2^{st}, \ldots, C_{3,l} = u_l^{st}$, $C_4 = H(C_{3,1}, C_{3,2}, \ldots, C_{3,l})$, $C_5 = \text{Ext}(K, C_4)$. It

gains the ciphertext $CT' = (K, C_1, C_2, C_{3,1} \ldots, C_{3,l}, C_5, t, x_1, x_2, \ldots, x_l)$.

Enc^{on}: given the message M, identity vector $I = (ID_1, \ldots, ID_j)$ and CT', it sets $t_1 = t^{-1}(ID_1 - x_1), t_2 = t^{-1}(ID_2 - x_2), \ldots, t_j = t^{-1}(ID_j - x_j), t_{j+1} = t^{-1}x_{j+1}, \ldots, t_l = t^{-1}x_l$. The algorithm calculates $C_6 = C_5 \oplus M$. The final ciphertext is

$$CT = \left(C_1, C_2, C_{3,1} \ldots, C_{3,l}, C_6, t_1, t_2, \ldots, t_l\right). \quad (1)$$

Decryption: if the identity vectors of ciphertext and private key are $(ID_1, \ldots, ID_j)$. The decryption process is as follows:

$$\begin{aligned}
&\frac{e(K_2, C_2)}{e\left(K_1, C_1 C_{3,1}^{t_1} C_{3,2}^{t_2} \cdots C_{l,2}^{t_l}\right)} \\
&= \frac{e\left(g^{\alpha}\left(u_1^{ID_1} \cdots u_l^{ID_l} h\right)^r R_3', g^s\right)}{e\left(g^r R_3, \left(u_1^{x_1} \cdots u_l^{x_l} h\right)^s u_1^{st\left(t^{-1}\left(ID_1 - x_1\right)\right)} \cdots u_j^{st\left(t^{-1}\left(ID_j - x_j\right)\right)} u_{j+1}^{st\left(-t^{-1}x_{j+1}\right)} \cdots u_l^{st\left(-t^{-1}x_l\right)}\right)} \\
&= \frac{e(g, g)^{\alpha s} e\left(u_1^{ID_1} \cdots u_j^{ID_j} h, g\right)^{rs}}{e\left(g, u_1^{ID_1} \cdots u_j^{ID_j} h\right)^{rs}} \\
&= e(g, g)^{\alpha s} = K, \\
M &= C_6 \oplus \mathrm{Ext}\left(K, H\left(C_{3,1}, C_{3,2}, \ldots, C_{3,l}\right)\right).
\end{aligned} \quad (2)$$

We show the specific operations of our leakage-resilient scheme (**LR-HIBOOE**) in Figure 5. As a comparison, we also give the specific operations of the scheme without leakage resilience (**HIBOOE**).

5. Security Proof

By constructing semifunctional private key and ciphertext, we prove the security of our **LR-HIBOOE** scheme.

Semifunctional ciphertext: we randomly select a generator g_2 of G_{p_2}. For identity vector $I = (ID_1, \ldots, ID_j)$, we randomly select $y, v_1, v_2, \ldots, v_l, z_c \in Z_N$ and calculate $K = e(g, g)^{\alpha s}$, $\widehat{C_1} = (u_1^{x_1} \cdots u_l^{x_l} h)^s g_2^{yz_c}$,, $\widehat{C_2} = g^s g_2^y$, $\widehat{C_{3,1}} = u_1^{st} g_2^{ytv_1}, \widehat{C_{3,2}} = u_2^{st} g_2^{ytv_2}, \ldots, \widehat{C_{3,l}} = u_l^{st} g_2^{ytv_l}$, $\widehat{C_4} = H(C_{3,1}, C_{3,2}, \ldots, C_{3,l})$, $\widehat{C_5} = \mathrm{Ext}(K, C_4)$, $t_1 = t^{-1}(ID_1 - x_1), t_2 = t^{-1}(ID_2 - x_2), \ldots, t_j = t^{-1}(ID_j - x_j)$,, $t_{j+1} = t^{-1}ID_{j+1}, \ldots, t_l = t^{-1}ID_l$, and $\widehat{C_6} = \widehat{C_5} \oplus M$. The semifunctional ciphertext is $\widehat{CT} = (\widehat{C_1}, t\widehat{C_2}n, q\widehat{C_{3,1}}h, \ldots x, 7\widehat{C_{3,l}}C, ; \widehat{C_6}, t_1, t_2, \ldots, t_l)$.

Semifunctional private key: for the normal private key $K_1, K_2, E_{j+1}, \ldots, E_l$, we select the random numbers $\gamma, z_k, z_{j+1}, \ldots, z_l \in Z_N$ and generate the semifunctional private key:

$$\begin{aligned}
&\widehat{K_1} = K_1 g_2^{\gamma} = g^r g_2^{\gamma} R_3, \widehat{K_2} = K_2 g_2^{\gamma z_k} = g^{\alpha}\left(u_1^{ID_1} \ldots u_j^{ID_j} h\right)^r g_2^{\gamma z_k} R_3', \\
&\widehat{E_{j+1}} = E_{j+1} g_2^{\gamma z_{j+1}} = u_{j+1}^r g_2^{\gamma z_{j+1}} R_{j+1}, \ldots, \widehat{E}_l = E_l g_2^{\gamma z_l} = u_l^r g_2^{\gamma z_l} R_l.
\end{aligned} \quad (3)$$

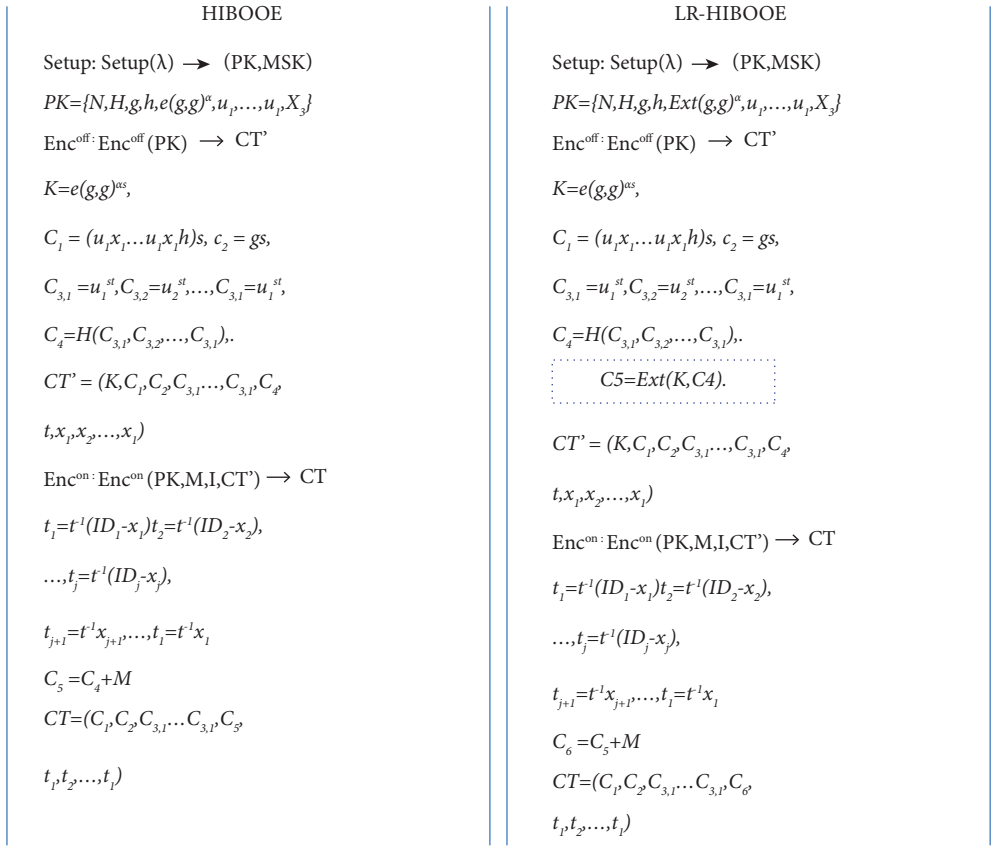

Figure 5: The main operations of our leakage-resilient scheme.

The semifunctional private key correctly decrypts the normal ciphertext via the normal private key. The normal private key correctly decrypts the semifunctional ciphertext.

If the semifunctional private key decrypts the semifunctional ciphertext, we have

$$\frac{e\left(\widehat{K_2},\widehat{C_2}\right)}{e\left(\widehat{K_1},\widehat{C_1}\widehat{C_{3,1}}^{t_1}\widehat{C_{3,2}}^{t_2}\cdots\widehat{C_{3,l}}^{t_l}\right)}$$

$$=\frac{e\left(g^{\alpha}\left(u_1^{ID_1}\cdots u_j^{ID_j}h\right)^{r}g_2^{yz_k}R_3',g^{s}g_2^{y}\right)}{e\left(g^{r}g_2^{y}R_3,\left(u_1^{x_1}\cdots u_l^{x_l}h\right)^{s}g_2^{yz_c}u_1^{st\left(t^{-1}\left(ID_1-x_1\right)\right)}g_2^{ytv_1}\cdots u_j^{st\left(t^{-1}\left(ID_j-x_j\right)\right)}u_{j+1}^{st\left(-t^{-1}x_{j+1}\right)}g_2^{ytv_{j+1}}\cdots u_l^{st\left(-t^{-1}x_l\right)}g_2^{ytv_l}\right)} \tag{4}$$

$$=e(g,g)^{\alpha s}e(g_2,g_2)^{y\gamma\left(z_k-z_c-v_1\left(ID_1-x_1\right)-\cdots-v_j\left(ID_j-x_j\right)+v_{j+1}x_{j+1}+\cdots+v_lx_l\right)}.$$

If $z_k = z_c + v_1(ID_1 - x_1) + \cdots + v_j(ID_j - x_j) - v_{j+1}x_{j+1} - \cdots - v_lx_l$, the ciphertext can be decrypted correctly. At this time, we call the semifunctional private key as nominal one. Although it contains components of G_2, it correctly decrypts the ciphertext.

Our **LR-HIBOOE** scheme is proved to be secure through some games as follows:

Game$_R$. This game is given in Section 3.2.

Game$_{R_l}$. The only difference between Game$_R$ and Game$_{R_l}$ is the way that they respond to the private key

inquiry. In Game_{R_l}, the challenger obtains a private key through the private key generation algorithm, while the challenger obtains a private key through the private key delegation algorithm in Game_R.

$\text{Game}_{\text{Restricted}}$. The game is similar to Game_{R_l}. The only restriction is that the attacker cannot query the prefix vector of the challenging identity vector modulo p_2. This restriction also applies to the following games:

We use q to indicate the number of private key inquiries.

$Game_k$ $((k \in [1, q]))$. The game is similar to $\text{Game}_{\text{Restricted}}$. The ciphertext which is sent to the attacker is the semifunctional one. The first k private key responses are semifunctional ones, and the rest private key responses are normal ones. In particular, in $Game_0$, the challenge ciphertext is in semifunctional form, and the private key is in normal form. In $Game_q$, all private keys and challenge ciphertexts are in semifunctional form.

Game_F. The only difference between Game_F and $Game_q$ is the ciphertext. In Game_F, the challenge ciphertext is a semifunctional one about any random message, while in $Game_q$, the ciphertext is a ciphertext about either of the two submitted messages.

The next five lemmas are given to illustrate that the series of games are not indistinguishable from the point of the attacker.

Lemma 1. *From the point of any attacker $\mathscr{A}$, we have $Game_R Adv_A = Game_{R_l} Adv_A$.*

Proof. The distribution of private keys is exactly the same from the private key generation algorithm and private key delegation algorithm. Therefore, in the view of the attacker, this is not fundamentally different. Thus, the attacker can only gain the same advantage in the two games.

Lemma 2. *If there is an algorithm $\mathscr{A}$ who obtains the advantage ε in differentiating $Game_{R_l}$ from $Game_{\text{Restricted}}$, i.e., $Game_{R_l} Adv_A - Game_{\text{Restricted}} Adv_A = \varepsilon$. An algorithm $\mathscr{B}$ is designed to destroy Assumption 2 over the same advantage ε.*

Proof. In consideration of g, X_1X_2, X_3, Y_2Y_3, the attacker $\mathscr{A}$ and the algorithm $\mathscr{B}$ simulate the game Game_{R_l}. $\mathscr{A}$ has a probability ε to obtain the identity vector I and I^* on the condition that $I \neq I * mod\ N$. $\mathscr{B}$ calculates $a = \gcd(I - I*, N)$ to get one factor of N. Let $b = N/a$. Because a is divided by p_2 and $N = ab = p_1p_2p_3$, there are three cases as follows:

(1) a and b are p_1 and p_2p_3, respectively.

(2) a and b are p_2 and p_1p_3, respectively

(3) a and b are p_3 and p_1p_2, respectively.

Case 1. $\mathscr{B}$ calculates $(Y_2Y_3)^a$ and $(Y_2Y_3)^b$. If $(Y_2Y_3)^a = 1$, $a = p_1$. Otherwise, if $(Y_2Y_3)^b = 1$, $b = p_1$. If $e(T^a, X_1X_2) = 1$, $\mathscr{B}$ can conclude that T does not contain a component of G_{p_2}. Otherwise, T does contain the component of G_{p_2}.

Case 2. $\mathscr{B}$ calculates $(X_1X_2)^a$ and $(X_1X_2)^b$. If they are not unit elements and do not meet Case 1, it meets Case 2. $\mathscr{B}$ calculates g^a and g^b. If $g^a = 1$, $a = p_1p_3$. Otherwise, if $g^b = 1, b = p_1p_3$. Without loss of generality, we suppose that $b = p_1p_3$ and $a = p_2$. In case T^b is a unit element, $\mathscr{B}$ can conclude that T contains the component of G_{p_2}. Otherwise, T does not contain the component of G_{p_2}.

Case 3. $\mathscr{B}$ calculates X_3^a and X_3^b. If $X_3^a = 1$, $a = p_3$. Otherwise, if $X_3^b = 1$, $b = p_3$.

We suppose that $a = p_3$. If $e(T^a, Y_2Y_3)$ is a unit element, $\mathscr{B}$ can conclude that T does not contain the component of G_{p_2}. Otherwise, T contains the component of G_{p_2}. Thus, $\mathscr{B}$ breaks Assumption 2 over the advantage which is larger than $\varepsilon/2$.

Lemma 3. *If there is an algorithm $\mathscr{A}$ who achieves the advantage ε in differentiating $Game_0$ from $Game_{\text{Restricted}}$, i.e., $Game_{\text{Restricted}} Adv_A - Game_0 Adv_A = \varepsilon$. One algorithm $\mathscr{B}$ is designed to destroy Assumption 1 over the same advantage ε.*

Proof. In consideration of g, X_3, T, the attacker $\mathscr{A}$ and the algorithm $\mathscr{B}$ simulate the game $Game_0$ or $\text{Game}_{\text{Restricted}}$. $\mathscr{B}$ randomly selects $a, a_1, \ldots, a_l, b \in Z_N$. $\mathscr{B}$ computes $u_1 = g^{a_1}, \ldots, u_l = g^{a_l}$ and $h = g^b$. $\mathscr{B}$ sends the public parameters $PK = \{N, H, g, h, e(g, g)^\alpha, u_1, \ldots, u_l, X_3\}$ to the attacker $\mathscr{A}$. For the entity vector $I = (ID_1, \ldots, ID_j)$ which is given by the attacker $\mathscr{A}$, $\mathscr{B}$ selects randomly $r, t, w, v_{j_1}, \ldots, v_l \in Z_N$. $\mathscr{B}$ computes $K_1 = g^r X_3^t, K_2 = g^\alpha (u_1^{ID_1} \ldots u_j^{ID_j} h)^r X_3^w, E_{j+1} = u_{j+1}^r X_3^{v_{j+1}}, \ldots, E_l = u_l^r X_3^{v_l}$.

For the message M_0, M_1 and the challenged entity vector $I^* = (ID_1^*, \ldots, ID_j^*)$ which are given by the attacker $\mathscr{A}$, $\mathscr{B}$ randomly selects $t, x_1, x_2, \ldots, x_l \in Z_N$ and $b \in \{0, 1\}$. $\mathscr{B}$ computes $K = e(T, g)^\alpha$, $C_1 = T^{a_1x_1 + a_2x_2 + \ldots + a_lx_l + b}$, $C_2 = T$, $C_{3,1} = T^{a_1t}, C_{3,2} = T^{a_2t}, \ldots, C_{3,l} = T^{a_lt}$, $C_4 = H(C_{3,1}, C_{3,2}, \ldots, C_{3,l})$, $C_5 = \text{Ext}(K, C_4)$, $t_1 = t^{-1}(ID_1^* - x_1), t_2 = t^{-1}(ID_2^* - x_2), \ldots, t_j = t^{-1}(ID_j^* - x_j), t_{j+1} = t^{-1}x_{j+1}, \ldots, t_l = t^{-1}x_l$, and $C_6 = C_5 \oplus M_b$. $\mathscr{B}$ gets the ciphertext:

$$CT = \left(C_1, C_2, C_{3,1} \ldots, C_{3,l}, C_6, t_1, t_2, \ldots, t_l\right). \tag{5}$$

This implicitly sets g^s as the G_{p_1} component of T. Supposing that $T \in G_{p_1p_2}$, the ciphertext is one semifunctional form, where $z_c = a_1x_1 + \ldots + a_lx_l + b$. $\mathscr{A}$ simulates $Game_0$. As for $T \in G_{p_1}$, the ciphertext is one normal ciphertext. The attacker simulates $\text{Game}_{\text{Restricted}}$. Therefore, the algorithm $\mathscr{B}$ breaks Assumption 1 and obtains advantage ε.

Lemma 4. *If there exists an algorithm $\mathscr{A}$ which gains the advantage ε in differentiating $Game_i$ from $Game_{i-1}$, i.e.,*

$Game_{i-1}Adv_A - Game_iAdv_A = \varepsilon$. *One algorithm* $\mathcal{B}$ *is designed to destroy Assumption 2 over the same advantage* ε.

Proof. In consideration of $g, X_1X_2, X_3, Y_2Y_3, T$, $\mathcal{B}$ randomly selects $a, a_1, \ldots, a_l, b \in Z_N$. $\mathcal{B}$ sends the parameters $u_1 = g^{a_1}, \ldots, u_l = g^{a_l}, h = g^b, e(g,g)^a$ to the attacker $\mathcal{A}$.

The attacker $\mathcal{A}$ asks the private key about the p^{th} identity vector $(ID_1, \ldots, ID_j)$.

When $p < i$, $\mathcal{B}$ is given one semifunctional private key. $\mathcal{B}$ selects random number $r, z, t, z_{j+1}, \ldots, z_l \in Z_{N_{ID_j}}$ and calculates $K_1 = g^r(Y_2Y_3)^t, K_2 = g^\alpha(u_1^{ID_1} \ldots u_j^{ID_j}h)^r (Y_2Y_3)^z, E_{j+1} = u_{j+1}^r(Y_2Y_3)^{z_{j+1}}, \ldots, E_l = u_l^r(Y_2Y_3)^{z_l}$, where $g_2^y = Y_2^t$. The semifunctional key is evenly distributed.

When $p > i$, $\mathcal{B}$ produces one normal private key. $\mathcal{B}$ selects random number $r, t, w, z_{j+1}, \ldots, z_l \in Z_{N_{ID_j}}$ and calculates $K_1 = g^r X_3^t, K_2 = g^\alpha(u_1^{ID_1} \ldots u_j^{ID_j}h)^r X_3^w, E_{j+1} = u_{j+1}^r X_3^{z_{j+1}}, \ldots, E_l = u_l^r X_3^{z_l}$.

When $p = i$, the algorithm $\mathcal{B}$ sets $z_k = a_1ID_1 + \ldots + a_jID_j + b$. $\mathcal{B}$ selects random number $r, w, w_{j+1}, \ldots, w_l \in Z_N$ and calculates $[K_1 = T, K_2 = g^\alpha T^{z_k} X_3^w, E_{j+1} = u_{j+1}^r T^{w_{j+1}}, \ldots, E_l = u_l^r T^{w_l}]$.

Supposing that $T \in G_{p_1p_3}$, the private key has normal form, where g^r is the G_{p_1} component of T. As for $T \in G$, the private key takes on semifunctional form.

Challenge. For two message M_0, M_1 and the challenged entity vector $I^* = (ID_1^*, \ldots, ID_j^*)$ which are given by the attacker $\mathcal{A}$, $\mathcal{B}$ randomly selects $t, x_1, x_2, \ldots, x_l \in Z_N$ and $b \in \{0,1\}$. $\mathcal{B}$ calculates $K = e(X_1X_2, g)^\alpha$, $C_1 = (X_1X_2)^{a_1x_1+a_2x_2+\ldots+a_lx_l+b}, C_2 = X_1X_2, C_{3,1} = (X_1X_2)^{a_1t}$, $C_{3,2} = (X_1X_2)^{a_2t}, \ldots, C_{3,l} = (X_1X_2)^{a_lt}$, $C_4 = H(C_{3,1}, C_{3,2}, \ldots, C_{3,l})$, $C_5 = \mathrm{Ext}(K, C_4)$, $t_1 = t^{-1}(ID_1^* - x_1)$, $t_2 = t^{-1}(ID_2^* - x_2), \ldots, t_j = t^{-1}(ID_j^* - x_j), t_{j+1} = t^{-1}$ $x_{j+1}, \ldots, t_l = t^{-1}x_l$, and $C_6 = C_5 \oplus M_b$. $\mathcal{B}$ obtains the ciphertext $CT = (C_1, C_2, C_{3,1} \ldots, C_{3,l}, C_6, t_1, t_2, \ldots, t_l)$.

It indirectly makes that $g^s g_2^y = X_1X_2$ and $z_c = a_1x_1 + a_2x_2 + \ldots + a_lx_l + b$. Because the i^{th} entity vector cannot be a prefix modulo p_2 of the challenged entity vector $I^* = (ID_1^*, \ldots, ID_j^*)$, i.e. $(ID_1^*, \ldots, ID_j^*) \neq (x_1, x_2, \ldots, x_j)$, the attacker $\mathcal{A}$ thinks that z_c, z_k are randomly distributed.

z_c and z_k have an important relationship. When $\mathcal{B}$ needs to judge whether the private key of i^{th} entity vector is semifunctional one, it generates one semifunctional ciphertext.

Because $z_k = z_c + v_1(ID_1 - x_1) + \ldots + v_j(ID_j - x_j) - v_{j+1}x_{j+1} - \ldots - v_lx_l$, even if the private key of i^{th} entity vector is semifunctional, the decryption can succeed, which is equivalent to generate a nominally semifunctional private key.

If $T \in G_{p_1p_3}$, $\mathcal{B}$ simulates Game_{i-1}. If $T \in G$, $\mathcal{B}$ simulates Game_i. Therefore, $\mathcal{B}$ breaks Assumption 2 and obtains advantage ε.

Lemma 5. *Supposing that there exists an attacker* $\mathcal{A}$ *who gains the advantage* ε *in distinguishing* $Game_q$ *and* $Game_F$, *i.e.,* $Game_qAdv_A - Game_FAdv_A = \varepsilon$. *One algorithm* $\mathcal{B}$ *is designed to destroy Assumption 3 over the same advantage* ε.

Proof. In consideration of $g, g^\alpha X_2, X_3, g^s Y_2, Z_2, T$, $\mathcal{B}$ randomly selects $a, a_1, \ldots, a_l, b \in Z_N$ and sends the public parameters $u_1 = g^{a_1}, \ldots, u_l = g^{a_l}, h = g^b$, $e(g,g)^a = e(g^\alpha X_2, g)$ to the attacker $\mathcal{A}$.

When the attacker $\mathcal{A}$ makes one private key inquiry about the identity vector $(ID_1, \ldots, ID_j)$, $\mathcal{B}$ selects random number $c, r, t, w, z, z_{j+1}, \ldots, z_l, w_{j+1}, \ldots, w_l \in Z_N$ and calculates $K_1 = g^r Z_2^z X_3^t, K_2 = g^\alpha X_2(u_1^{ID_1} \cdots u_j^{ID_j}h)^r X_3^w Z_2^c, E_{j+1} = u_{j+1}^r(Y_2Y_3)^{z_{j+1}} X_3^{w_{j+1}}, \ldots, E_l = u_l^r(Y_2Y_3)^{z_l} X_3^{w_l}$.

Challenge. For the message M_0, M_1 and the challenged entity vector $I^* = (ID_1^*, \ldots, ID_j^*)$ which are given by the attacker $\mathcal{A}$, $\mathcal{B}$ randomly selects $t, x_1, x_2, \ldots, x_l \in Z_N$ and $b \in \{0,1\}$. $\mathcal{B}$ calculates $K = T$, $C_1 = (g^sY_2)^{a_1x_1+a_2x_2+\ldots+a_lx_l+b}$, $C_2 = (g^sY_2)^{a_1ID_1+a_2ID_2+\ldots+a_jx_j+b}$, $C_{3,1} = (g^sY_2)^{a_1t}, C_{3,2} = (g^sY_2)^{a_2t}, \ldots, C_{3,l} = (g^sY_2)^{a_lt}$, $C_4 = H(C_{3,1}, C_{3,2}, \ldots, C_{3,l})$, $C_5 = \mathrm{Ext}(K, C_4)$, $t_1 = t^{-1}(ID_1^* - x_1), t_2 = t^{-1}(ID_2^* - x_2), \ldots, t_j = t^{-1}(ID_j^* - x_j), t_{j+1} = t^{-1}x_{j+1}, \ldots, t_l = t^{-1}x_l$, and $C_6 = C_5 \oplus M_b$. $\mathcal{B}$ obtains the ciphertext $CT = (C_1, C_2, C_{3,1} \ldots, C_{3,l}, C_6, t_1, t_2, \ldots, t_l)$.

This implicitly sets that $z_c = a_1x_1 + \cdots + a_lx_l + b$. What is more, z_c is only related to module p_2. Because $u_1 = g^{a_1}, \ldots, u_l = g^{a_l}, h = g^b$ are only the elements of G_{p_1}, $a_1, \ldots, a_l, b$ modulo p_1 are not related to $z_c = a_1x_1 + \cdots + a_lx_l + b$modulo p_2.

In case $T = e(g,g)^{as}$, the challenge ciphertext about the message M_b is semifunctional form. In the case, T is one random number in G_T, and the challenge ciphertext is about one random message and is semifunctional. Therefore, the algorithm $\mathcal{B}$ breaks Assumption 3 over advantage ε.

Theorem 1. *As long as Assumptions 1–3 hold, our scheme is fully secure.*

Proof. Lemmas 1–5 show that the advantages gained by attackers in the game Game_R and Game_F are the same and can be ignored. In addition, b is hidden in the game Game_F. In this way, the advantage of the attacker to break the proposed scheme is ignored.

Specifically, let $\varepsilon_i (i \in \{1,2,3\})$ denote the advantage that the attacker breaks assumption $i (i \in \{1,2,3\})$.

In view of Lemmas 1–5, the difference of the advantage that the attacker may gain in different games is as follows:

$$\begin{aligned} \mathrm{Game}_R\mathrm{Adv}_A &= \mathrm{Game}_{R_t}\mathrm{Adv}_A, \\ Game_{R_t}\mathrm{Adv}_A - \mathrm{Game}_{\mathrm{Restricted}}\mathrm{Adv}_A &\leq \varepsilon_2, \\ \mathrm{Game}_{\mathrm{Restricted}}\mathrm{Adv}_A - Game_0\mathrm{Adv}_A &\leq \varepsilon_1, \\ \mathrm{Game}_{i-1}\mathrm{Adv}_A - Game_i\mathrm{Adv}_A &\leq \varepsilon_2, \\ \mathrm{Game}_q\mathrm{Adv}_A - Game_F\mathrm{Adv}_A &\leq \varepsilon_3. \end{aligned} \tag{6}$$

Furthermore, we have $\mathrm{Game}_R\mathrm{Adv}_A - Game_F\mathrm{Adv}_A \leq \varepsilon_2 + \varepsilon_1 + \varepsilon_2 + \varepsilon_3$. Thus, any attacker only gains negligible advantages.

6. Analysis of Leakage Resilience

Theorem 2. *The relative leakage rate of the encapsulated key of our scheme is* $\rho = Leak/[Log(N)] \approx 1$.

Table 1: The comparisons between our scheme and the schemes.

Schemes	Online encryption	Offline encryption	Decryption	The length of ciphertext	Hierarchy	LR
[21]	$1m + 1M_e$	$(n+4)M_e$	$(n+2)P$	$(n+4)\|G\| + 1\|Z_N\|$	No	Yes
[22]	lm	$(2l+3)M_e$	$2P + (l+1)M_e$	$(l+2)\|G\| + (l+1)\|Z_N\|$	Yes	No
Ours	$lm + E$	$(2l+3)M_e$	$2P + (l+1)M_e$	$(l+2)\|G\| + (l+1)\|Z_N\|$	Yes	Yes

Proof. From the real security game, we know $\tilde{H}_\infty(A|\text{VIEW})) = \text{Log}(N)$. If the leakage information obtained by the attacker through the leakage query is λ bits; that is, Leak has 2^λ values. Then, from Conclusion 1 it can be obtained that $\tilde{H}_\infty(\mathcal{A}|\text{Leak, VIEW})) \geq \tilde{H}_\infty(A|\text{VIEW}) - \lambda = \text{Log}(N) - \lambda$.

Therefore, if the extractor is $(\text{Log}(N) - \lambda, \varepsilon)$ strong, it can be obtained that $SD((\text{Ext}(k, s^*), s^*, \text{Leak, VIEW}), t(U, s^*, \text{Leak, VIEW})) \leq \varepsilon$,where U is uniformly distributed. In fact, when the extractor is good enough, the leakage λ can approach $\text{Log}(N)$. Then, $C_6 = M_b \oplus \text{Ext}(e(g,g)^{\alpha s}, H(C_{3,1}, C_{3,2}, \ldots, C_{3,l}))$ and uniform distribution are indistinguishable. Therefore, the leakage ratio of the encapsulated key $\rho = \text{Leak}/\text{Log(N)} \approx \text{Log(N)}/\text{Log(N)} = 1$. Thus, Theorem 2 holds.

7. Efficiency Evaluation and Experimental Simulation

We use Table 1 to show the comparisons between our scheme and the schemes [21, 22]. We mainly compared the offline encryption, online encryption, decryption, and some other aspects. Let l denote the maximum number of the level. Let P denote the pairing operation in G. Let $|G|$ and $|Z_N|$ denote the length of G and Z_N, respectively. Let M_e denote the exponential operation in G. Let m denote the multiplication operation in G or Z_N. Let E denote the operation time of extractor function. In the scheme [21], n is a parameter which determines the leakage [21, 22] rate.

It can be seen from Table 1 that our scheme and the scheme [22] have the same calculation efficiency about offline encryption and decryption. In the online encryption phase, our scheme has one more extractor operation than the scheme [22]. In particular, the operation time of the extractor is relatively short, so the overall time about online encryption of our scheme and that of the scheme [22] is almost the same. Compared with the literature [21], their scheme has no hierarchal function, which is equivalent to the special case of our scheme when $l = 1$. The efficiency of the scheme [21] is affected by parameter n (n is a parameter which determines the leakage rate). Therefore, under the same conditions, our scheme is more efficient than the scheme [21]. In the paper [21], they obtain the leakage resilience through the private key extension technology. In our scheme, we use the extractor to rerandomize the private key to obtain the leakage resilience. These are two different ideas.

The experimental platform is a PC with 64 bit operating system Windows 10, 3.40 GHz main frequency, 8.00 G RAM and Intel (R) Core (™) i7-6700 CPU. Based on Java Pairing Based Cryptography Library 2.0.0 [32], we use Eclipse 4.4.1 for simulation software. The 160 bit composite order elliptic curve $y^2 = x^3 + x$ is selected for our experiment. When the maximum level is 10, the online encryption time is 0.018 seconds, the offline encryption time is 0.460 seconds, and the decryption time is 0.39 seconds.

8. Conclusions

In this article, we provide an online/offline identity-based hierarchal encryption scheme with leakage resilience which is presented. By using the dual system technique, we prove the security of the scheme. The use of the binary extractor provides the leakage resilience. The entropy leakage ratio of the encapsulated symmetric key is close to 1. Because leakage-resilient cryptography is a relatively new research direction in cryptography, there are still many problems worthy of further research. For example, we will focus on the construction of efficient and leakage-resilient cryptography scheme.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (62172292, 62072104, 61972095, U21A20465), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (17KJB520042, 20KJB413003), the Suqian Sci&Tech Program (S201820, Z2019109), the Jiangsu Province Engineering Research Center of Smart Poultry Farming and Intelligent Equipment, the cloud computing and big data security research team of Suqian University, and sponsored by Qing Lan Project. This work was also supported by the Natural Science Foundation of the Fujian Province, China (2020J01159).

References

[1] Y. S. Won, S. Chatterjee, D. Jap, A. Basu, and S. Bhasin, "WaC: first results on practical side-channel attacks on commercial machine learning accelerator," in *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security*, pp. 111–114, New York, NY, USA, November 2021.

[2] A. Dubey, R. Cammarota, and A. Aysu, "Maskednet: The First Hardware Inference Engine Aiming Power Side-Channel protection," in *Proceedings of the 2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 197–208, IEEE, San Jose, CA, USA, December 2020.

[3] C. S. Chen, T. Wang, and J. Tian, "Improving timing attack on RSA-CRT via error detection and correction strategy," *Information Sciences*, vol. 232, no. 232, pp. 464–474, 2013.

[4] M. Lipp, M. Schwarz, and D. Gruss, T. Prescher, W. Haas, J. Horn et al., Meltdown: reading kernel memory from user space," in *Proceedings of the 27th USENIX Security Symposium*, pp. 973–990, Baltimore, MD, USA, August 2018.

[5] B. J. Van, M. Minkin, O. Weisse et al., "Foreshadow: extracting the keys to the intel SGX kingdom with transient out-of-order execution," in *Proceedings of the 27th USENIX Security Symposium*, pp. 991–1008, Baltimore MD, USA, August 2018.

[6] J. A. Halderman, S. D. Schoen, N. Heninger et al., "Lest we remember: cold-boot attacks on encryption keys," *Communications of the ACM*, vol. 52, no. 5, pp. 91–98, 2009.

[7] S. Micali and L. Reyzin, "Physically observable cryptography," in *Proceedings of the 1st Theory of Cryptography Conference*, pp. 278–296, Cambridge, MA, USA, February 2004.

[8] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in *Proceedings of the FOCS 2008*, pp. 293–302, IEEE, Philadelphia, PA, USA, October 2008.

[9] K. Pietrzak, "A leakage-resilient mode of operation," in *Proceedings of the 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques EUROCRYPT 2009, LNCS 5479*, pp. 462–482, Cologne, Germany, April 2009.

[10] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in *Proceedings of the Sixth Theory of Cryptography Conference, TCC 2009*, vol. 544, pp. 474–495, San Francisco, CA, USA, March 2009.

[11] M. Naor and G. Segev, "Public-key cryptosystems resilient to key leakage," *SIAM Journal on Computing*, vol. 41, no. 4, pp. 772–814, 2012.

[12] L. Guo, J. Wang, and W. C. Yau, "Efficient hierarchical identity-based encryption system for internet of things infrastructure," *Symmetry*, vol. 11, no. 7, p. 913, 2019.

[13] R. Langrehr and J. Pan, "Tightly secure hierarchical identity-based encryption," *Journal of Cryptology*, vol. 33, no. 4, pp. 1787–1821, 2020.

[14] A. Takayasu, "More efficient adaptively secure revocable hierarchical identity-based encryption with compact ciphertexts: achieving shorter keys and tighter reductions," 2021, https://eprint.iacr.org/2021/539.pdf.

[15] K. Emura, A. Takayasu, and Y. Watanabe, "Generic constructions of revocable hierarchical identity-based encryption," 2021, https://eprint.iacr.org/2021/515.pdf.

[16] F. Guo, Y. Mu, and Z. Chen, "Identity-based online/offline encryption," in *Proceedings of the 2008 International Conference on Financial Cryptography and Data Security*, pp. 247–261, Cozumel, Mexico, January 2008.

[17] J. Lai, Y. Mu, F. Guo, and W. Susilo, "Improved identity-based online/offline encryption," in *Proceedings of the 2015Australasian Conference on Information Security and Privacy*, pp. 160–173, Brisbane, Australia, July 2015.

[18] A. Elkhalil, J. Zhang, and R. Elhabob, "An efficient heterogeneous block chain-based online/offline signcryption systems for internet of vehicles," *Cluster Computing*, vol. 24, no. 3, pp. 2051–2068, 2021.

[19] J. Lai, Y. Mu, and F. Guo, "Efficient identity-based online/offline encryption and signcryption with short ciphertext," *International Journal of Information Security*, vol. 16, no. 3, pp. 299–311, 2017.

[20] Z. Wang, H. Ma, and J. Wang, "Attribute-based online/offline encryption with outsourcing decryption," *Journal of Information Science and Engineering*, vol. 32, no. 6, pp. 1595–1611, 2016.

[21] X. Zhang, X. Fu, L. Hong, Y. Liu, and L. Wang, "Provable secure identity-based online/offline encryption scheme with continual leakage resilience for wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 16, no. 6, Article ID 155014772092873, 2020.

[22] Z. Wang, H. Ma, and J. Wang, "Fully secure hierarchical identity-based online/offline encryption," *Journal of Computer Applications*, vol. 35, no. 9, pp. 2522–2526, 2015.

[23] Q. Yu, J. Li, and Y. Zhang, "Leakage-resilient certificate-based encryption," *Security and Communication Networks*, vol. 8, no. 18, pp. 3346–3355, 2015.

[24] M. Medwed and F. X. Standaert, "Extractors against side-channel attacks: weak or strong?" *Journal of Cryptographic Engineering*, vol. 1, no. 3, pp. 231–241, 2011.

[25] D. Chen, Y. Zhou, Y. Han, R. Xue, and Q. He, "On hardening leakage resilience of random extractors for instantiations of leakage-resilient cryptographic primitives," *Information Sciences*, vol. 271, pp. 213–223, 2014.

[26] B. Waters, "Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions," in *Proceedings of the Advances in Cryptology-CRYPTO 2009*, pp. 619–636, Santa Barbara, CA, USA, August 2009.

[27] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," *Information Sciences*, vol. 470, pp. 175–188, 2019.

[28] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Information Sciences*, vol. 484, pp. 113–134, 2019.

[29] J. Li, Y. Zhang, J. Ning, X. Huang, G. S. Poh, and D. Wang, "Attribute based encryption with privacy protection and accountability for CloudIoT," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 762–773, 2022.

[30] J. Li, N. Chen, and Y. Zhang, "Extended file hierarchy access control scheme with attribute based encryption in cloud computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 2, pp. 983–993, 2021.

[31] N. Chen, J. Li, Y. Zhang, and Y. Guo, "Efficient CP-abe scheme with shared decryption in cloud storage," *IEEE Transactions on Computers*, vol. 71, no. 1, pp. 175–184, 2022.

[32] A. De Caro and V. Iovino, "JPBC: java pairing based cryptography," in *Proceedings of the 2011 IEEE symposium on computers and communications (ISCC)*, pp. 850–855, IEEE, Kerkyra, Greece, June 2011.