Hindawi
Security and Communication Networks
Volume 2022, Article ID 8220259, 12 pages
https://doi.org/10.1155/2022/8220259

Research Article

Fully Secure ID-Based Signature Scheme with Continuous Leakage Resilience

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

[1]*College of Information Engineering, Suqian University, Jiangsu, Suqian 223800, China*
[2]*College of Mathematics and Informatics, Fujian Normal University, Fujian, Fuzhou 350117, China*

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

Received 11 July 2021; Revised 25 November 2021; Accepted 21 December 2021; Published 24 January 2022

Academic Editor: Mohamed Amine Ferrag

The side channel attacks will lead to the destruction of the security of the traditional cryptographic scheme. Leakage-resilient identity-based signature has attracted great attention. Based on the dual system encryption technology, we construct an identity-based signature scheme that can resist continuous private key leakage. In the standard model, the security of the scheme is proved. The key points of our leakage-resilient signature scheme are as follows: (1) The private key can be extended according to the security requirements. In other words, when the leakage is serious, we can select a bigger value n, where n is a parameter related to the leakage rate. (2) An elaborate key update algorithm makes the scheme resist continuous leakage attacks. Furthermore, the updated private key has the same distribution as the previous private key. (3) The proposed scheme is fully secure in the standard model rather than in the random oracle model or in the general group model. In order to achieve this goal, we use dual system encryption technology. Thus, the security of the constructed scheme does not depend on the number of queries of the attacker.

1. Introduction

In recent years, cryptography researchers have found that some side-channel attacks [1–7] can leak the secret information of the cryptosystem to attackers. In side-channel attacks, the attacker can obtain secret information by observing the energy consumption and timing of the cryptosystem.

The traditional provably secure cryptographic system is based on the black-box model, which does not consider secret information leakage. In the case of side-channel attacks, the security of traditional cryptographic schemes is destroyed. Therefore, it is an urgent problem to design leakage-resilient (LR) cryptographic schemes. In recent years, some scholars have been engaged in this field. The research of leakage-resilient cryptography has become a hot topic in cryptography.

1.1. Related Work. The research results of leakage-resilient cryptography mainly focus on public key cryptosystem, which can be divided into the following models.

1.1.1. Only Computation Leaks Information. The model is given by [8]. In this model, the complexity of the leakage function is unlimited and the total leakage is unlimited, but the leakage is only allowed to occur in the active part of the memory required by the current calculation. In particular, the attacker can choose a polynomial time function with bounded output to apply to the currently active state. In each round of calculation, the storage parts that are not accessed do not leak information, and only the storage parts that participate in the calculation have information leakage. A secure stream cipher scheme is proposed in [9]. They resist the information leakage about the internal state during the computation of each output block. In each step, the amount of leakage depends on the strength of the underlying pseudorandom generator. Pietrzak [10] relaxes the requirement of the pseudorandom function generator and only needs a weak pseudorandom function that can output pseudorandom value on random inputs. Based on the weak pseudorandom function, Pietrzak constructs a stream cipher that is simpler than [9].

1.1.2. Bounded Leakage Model. In the cold-boot attacks, leakage does occur not only in the calculation process. In order to obtain security against cold-boot attacks, a bounded leakage model is proposed in [11]. In this model, the attacker can freely choose a valid computable function and get the output of the function. The basic requirement is that the output of the function does not disclose the entire key. In a word, in this model, any adversary can obtain information that is shorter than the key length. Under some trapdoor one-way function, Akavia et al. [11] give a secure leakage-resilient identity-based scheme and a public key encryption scheme. The two schemes do not increase the size of the secret key and do not introduce any complication of the natural encryption and decryption routines. In [12], a leakage-resilient public key encryption scheme (PKE) is obtained through a hash proof system (HPS). They give a generic construction of a public key encryption scheme that can resist key leakage from any hash proof system. The resulting scheme is as efficient as the underlying hash proof system and additional computational assumptions are not needed in their construction. After that, [13–16] give LR encryption schemes by HPS with some characteristics. Chen et al. [13] generalize HPS to include the characteristics of anonymity (anonymous HPS), and then use anonymous HPS to construct a leakage-resilient public key encryption scheme (LR-PKE). The concept of weak HPS is defined in [14], which shows that LR weak pseudorandom function, LR message authentication code and LR symmetric encryption scheme can be obtained if a one-way function exists. In [15], a lattice-based LR-PKE is proposed by using an updatable hash proof system. The work [16] gives an efficient public key cryptosystem with leakage-resilience, where plaintext length is independent of key leakage parameters. In another public key cryptosystem, Yu et al. [17] first proposed a certificate-based encryption scheme with leakage-resilience. In [17], the leakage of almost the entire encapsulated symmetric key can be tolerated.

1.1.3. Continuous Leakage Model. When some information is leaked each time the private key is used, how the security of the schemes can be achieved? References [18, 19] solve the open problem proposed in [8], respectively, and propose the continuous leakage model (CLM). The requirement that only computation leaks information is not required for the continuous leakage considered in [18, 19]. In CLM, the key must be updated periodically and the necessary constraint is that the leakage between any two consecutive updates is bounded. In other words, the amount of key leakage in each period is limited, but the amount of key leakage in the whole operation process is infinite. Reference [18] gives IBE and public key encryption schemes with continual leakage-resilience under the decisional linear assumption or the symmetric external Diffie-Hellman assumption. Their core contribution is to show how to update the key. In [19], the key point is that the user may use some additional fresh local randomness to periodically refresh the secret key and not to affect the public key. They design a relation which is called continuous leakage-resilient (CLR) one-way relation (OWR). By the CLR-OWR, they propose CLR identification scheme, CLR signature, and CLR authenticated key agreement protocol.

In [20], they explore the case that the memory of a system is divided into two parts and each of them works independently. The attacker can only get leakage information from one part at one time period. They call their security model as the split-state model. By split-state technology, they construct a dynamic secret sharing scheme against continuous leakage attack. Based on the same split-state technology, the work [21] shows that discrete log representations can resist continuous leakage attacks.

Dual system encryption [22] gives a new way to achieve security of IBE and some related encryption schemes. In dual system encryption schemes, there are two kinds of ciphertext and key generation algorithms: normal and semifunctional. The key or ciphertext generated by a normal key generation algorithm or normal ciphertext generation algorithm is called normal key or normal ciphertext. The key or ciphertext generated by a semifunctional key generation algorithm or semifunctional ciphertext generation algorithm is called semifunctional key or semifunctional ciphertext. Normal ciphertext can be decrypted by a normal key or semifunctional key. The semifunctional ciphertext cannot be decrypted by the semifunctional key and can be decrypted by a normal key. Inspired by dual system encryption [22], several continuous leakage-resilience encryption schemes with advanced features are given in [23]. They propose fully secure IBE, HIBE, and ABE which are resilient to bounded leakage. These schemes can resist the leakage not only from the private key but also from the master key. In [24], an identity-based broadcast encryption scheme against continuous leakage is proposed. In reference [25], a hierarchical attribute-based encryption scheme with continuous leakage-resilience is proposed. In [26], an identity-based secure scheme against continuous leakage is designed in the standard model. Based on the q-ABDHE assumption, they first propose a CLR-IBE scheme with CPA security in the standard model. Based on their basic CLR-IBE scheme, they give a CLR-CCA secure IBE scheme with continuous leakage amplification. Different from the above schemes, Li et al. [27] extend the length of the key and master key and realize the leakage-resilience by using redundancy. Further, a key-policy attribute-based cryptosystem against continuous auxiliary input leakage is constructed.

1.2. Leakage-Resilient Signature. Signature is an important primitive in cryptography. As an important part of data security, a digital signature is used in data integrity verification, nonrepudiation and other aspects. Digital signature schemes with various characteristics have emerged, such as identity-based signature, proxy resignature, blind signature, designated verifier signature, and so on.

As far as we know, there are few leakage-resilient signature schemes in the literature.

The first secure signature scheme against bounded leakage is given in [28] in the random oracle model. The

relative leakage rate of the scheme is almost 1. Based on general primitives (i.e., one-time signature scheme and noninteractive zero knowledge proof), the work [29] constructs leakage-resilient signature schemes in the standard model. As time goes by, the private key may leak more and more information. When the leakage of the private key comes to a certain value, the scheme may become insecure. In order to solve this problem, a continuous leakage-resilient signature is proposed in [18, 19].

In [18, 19], the continuous leakage does not need the requirement that only computation leaks information. By the noninteractive zero knowledge proof system (NIZK), the work [30] gives another example of a continuous leakage-resilient signature. The key update algorithm in [30] breaks the second preimage resistance of the hash function H. Thus, they must introduce a new notion: (n, k)-independent preimage resistant hash function H, which is stronger than the notion of second preimage resistance.

Galindo and Vivek [31] and Wu et al. [32] give CLR signature schemes under the general bilinear group assumption. It is generally considered that the bilinear group model is weaker than the standard model.

1.3. Motivation. Because identity-based signature does not need a digital certificate to verify the correctness of public keys and the authenticity of user identity, it solves the problem of management and distribution of digital certificates in traditional signature. Thus, it is widely used in wireless communication and other fields [33].

In order to design an identity-based signature scheme against continuous leakage attacks in the standard model, we must consider the following factors. The private key of the scheme must be extensible. The private key can be easily updated. The security proof of the scheme can be obtained in the standard model. Fortunately, we do it.

In this paper, we propose an identity-based signature scheme by using the dual encryption technique. The security of the scheme is proved under the standard model. The scheme can resist continuous key leakage. Referring to the idea of dual system signature [34], our scheme has the normal key and semifunctional keys. The overall concept of our scheme is shown in Figure 1.

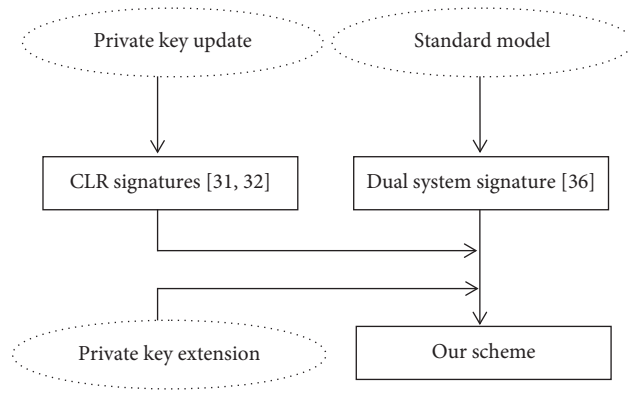

Figure 1: Overall concept of our scheme.

2. Preliminaries

Definition 1. Bilinear map

Let G and $G_{\mathcal{T}}$ be multiplicative cyclic groups with prime order p. Suppose that g is a generator of G. $e: G \times G \longrightarrow G_{\mathcal{T}}$ is called a bilinear map if it satisfies the following three conditions:

(1) Computability: for $g, h \in G$, $e(g, h)$ can be calculated effectively.

(2) Nondegeneracy: $\exists g \in G$, $e(g, g) \neq 1$.

(3) Bilinearity: for $g, h \in G$ and $a, b \in Z^*$, $e(g^a, h^b) = e(g, h)^{ab}$.

Some notations: the size of the term W is denoted by $|W|$. The symbol $\cdot$ is used to denote the product of two vectors. Let $*$ denote the component-wise product of two elements. Let $\langle \cdot, \cdot, \cdot \rangle$ denote vectors. Let $(\cdot, \cdot, \cdot)$ denote the set of some elements. If $g \in G$, $\overrightarrow{v} = < v_1, v_2, \ldots, v_n > \in G^n$, $a \in Z_N$ and $\overrightarrow{b} = < b_1, b_2, \ldots, b_n > \in Z_N^n$, we use $g^{\overrightarrow{b}}$ to denote $\langle g^{b_1}, g^{b_2}, \ldots, g^{b_n} \rangle$ and use $\overrightarrow{v}^a$ to denote $\langle v_1^a, v_2^a, \ldots, v_n^a \rangle$. For $\overrightarrow{v} = < v_1, v_2, \ldots, v_n > \in G^n$ and $\overrightarrow{\iota} = < \iota_1, \iota_2, \ldots, \iota_n > \in G^n$, $e(\overrightarrow{v}, \overrightarrow{\iota}) = \prod_{i=1}^{n} e(v_i, \iota_i) \in G_{\mathcal{T}}$.

In [35], the concept of bilinear groups with composite order is proposed. Suppose Ψ is an algorithm that generates bilinear groups with composite order. Ψ takes the safety parameter as input to generate a bilinear group $\Omega = \{N = p_1 p_2 p_3, \quad G, G_{\mathcal{T}}, e\}$, in which p_1, p_2, and p_3 are three different prime numbers. $\mathrm{Log}(p_1) = \mathrm{Log}(p_2) = \mathrm{Log}(p_3)$. G and $G_{\mathcal{T}}$ are cyclic groups with order N. e is a bilinear mapping.

In addition, for the security parameter λ, both G and $G_{\mathcal{T}}$ are computable in polynomial time. G_{p_1}, G_{p_2} and G_{p_3}, respectively, represents subgroups with order p_1, p_2 and p_3. $G_{p_1 p_2}$ represents subgroup with order $G_{p_1 p_2}$. If $h_i \in G_{p_i}$ and $h_j \in G_{p_j}$ $(i \neq j)$, $e(h_i, h_j)$ is an identity element in G_T. For $h_1 \in G_{p_1}$, $h_2 \in G_{p_2}$ and g is a generator of G, $g^{p_1 p_2}$ can generate G_{p_3}, $g^{p_1 p_3}$ can generate G_{p_2} and $g^{p_2 p_3}$ can generate G_{p_1}. We may find α_1, α_2 such that $h_1 = (g^{p_2 p_3})^{\alpha_1}$, $h_2 = (g^{p_1 p_3})^{\alpha_2}$, and $e(h_1, h_2) = e(g^{p_2 p_3 \alpha_1}, g^{p_1 p_3 \alpha_2}) = e(g^{\alpha_1}, g^{p_3 \alpha_2})^{p_1 p_2 p_3} = 1$.

Three assumptions which are given in [22] will be used in our security proof.

Assumption 1. Given $D^1 = (N, G, G_{\mathcal{T}}, e, g_1, g_3)$, $T_0^1 = g_1^z$ and $T_1^1 = g_1^z g_2^v$ where $z, v \in Z_N$ are randomly selected, any probabilistic polynomial time (PPT) adversary can only distinguish $T_0^1 = g_1^z$ from $T_1^1 = g_1^z g_2^v$ with only negligible advantage.

$\mathrm{Adv1}_{\psi, \mathscr{A}}(\vartheta)$ denotes the advantage that adversary breaks the assumption 1. That is to say, $\mathrm{Adv1}_{\psi, \mathscr{A}}(\vartheta) = |\Pr[\mathscr{A}(D^1, T_0^1) = 1] - \Pr[\mathscr{A}(D^1, T_1^1) = 1]|$.

If $\mathrm{Adv1}_{\psi, \mathscr{A}}(\vartheta)$ is negligible for every PPT adversary, we say that assumption 1 holds.

Assumption 2. Given $D^2 = (N, G, G_T, e, g_1, g_3, g_1^z g_2^v, g_2^u g_3^\varsigma)$ $(z, v, u, \varsigma \in Z_N)$, $T_0 = g_1^\omega g_3^\sigma$ and $T_1 = g_1^\omega g_2^\kappa g_3^\sigma$ $(\omega, \kappa, \sigma \in Z_N)$, any probabilistic polynomial

time adversary can only distinguish $T_0 = g_1^{\omega} g_3^{\sigma}$ from $T_1 = g_1^{\omega} g_2^{\kappa} g_3^{\sigma}$ with only negligible advantage.

$\mathrm{Adv2}_{\psi,\mathscr{A}}(\vartheta)$ denotes the advantage that adversary breaks the assumption 2. That is to say, $\mathrm{Adv2}_{\psi,\mathscr{A}}(\vartheta) = |\Pr[\mathscr{A}(D^2, T_0) = 1] - \Pr[\mathscr{A}(D^2, T_1) = 1]|$.

If $\mathrm{Adv2}_{\psi,\mathscr{A}}(\vartheta)$ is negligible for every PPT adversary, we say that assumption 2 holds.

Assumption 3. Given $D^3 = (N, G, G_{\mathscr{T}}, e, g_1, g_2, g_3, g_1^{\alpha} g_2^{v}, g_1^{s} g_2^{u})$ $(\alpha, s, v, \quad u \in Z_N)$, $T_0 = e(g_1, g_1)^{\alpha s}$ and $T_1 \in G_{p_1}$ $(s \in Z_N)$, any probabilistic polynomial time adversary can only distinguish $T_0 = e(g_1, g_1)^{\alpha s}$ from $T_1 \in G_{p_1}$ with only negligible advantage.

$\mathrm{Adv3}_{\psi,\mathscr{A}}(\vartheta)$ denotes the advantage that adversary breaks the assumption 3. That is to say, $\mathrm{Adv3}_{\psi,\mathscr{A}}(\vartheta) = |\Pr[\mathscr{A}(D^3, T_0) = 1] - \Pr[\mathscr{A}(D^3, T_1) = 1]|$.

If $\mathrm{Adv3}_{\psi,\mathscr{A}}(\vartheta)$ is negligible for every PPT adversary, we say that assumption 3 holds.

3. Formal Description of Continuous Leakage-Resilient Identity-Based Signature

On the basis of [36], the formal description of continuous leakage-resilient identity-based signature scheme (CLR-IBS) is given. Our scheme consists of the following algorithms:

Setup: the algorithm is a probabilistic polynomial time algorithm that is run by a private key generator (PKG). It inputs security parameters ϑ and outputs public parameters params and master key msk.

KeyGen: the algorithm is a probabilistic polynomial time algorithm that is run by a private key generator. The algorithm takes the public parameters msk and the user's identity ID as input and outputs the corresponding private key SK_{ID} of the user.

KeyUpd: the algorithm takes the public parameters msk and the private key SK_{ID} as input and outputs a new private key $\widehat{\mathrm{SK}_{ID}}$.

Sign: the algorithm is a probabilistic polynomial time algorithm that is completed by the signer. It inputs the public parameters params, the message m which is to be signed and the user's private key SK_{ID}. It outputs the signature σ of the message m.

Verify: the algorithm is a probabilistic polynomial time algorithm that is completed by the verifier. It takes the public parameters params, user's identity ID and the signature σ of the message m as input. Then, it judges whether the signature is valid. If the signature is valid, it outputs "accept." Otherwise, it outputs "reject."

KeyGenSF: this algorithm is a probabilistic polynomial time algorithm that is run by PKG. The algorithm takes the public parameters params and the user's identity ID as input. It outputs the semifunctional private key $\widehat{\mathrm{SK}_{\mathrm{ID}}}$.

The algorithm Setup, KeyGen, KeyUpd, and KeyGenSf are generated by PKG, and other algorithms are generated by users. KeyGenSf is only used in security proof.

4. Security Model of CLR-IBS

The security of CLR-IBS is defined by the game Game_R. Our scheme is existentially unforgeable under adaptive chosen message attack in the standard model.

In Game_R, the challenger $\mathscr{B}$ holds a list $\mathscr{L} = \{(\mathscr{H}, \mathscr{I}, \mathscr{SK}, \mathscr{LK})\}$ which consists of handle, collection of identity, secret key and leakage amount, where $\mathscr{H}$, $\mathscr{I}$, $\mathscr{SK}$, and $\mathscr{LK}$ are the handle's space, identity's space, secret key's space, and the leakage amount's space. Suppose $\mathscr{H} = \mathbb{N}$ and $\mathscr{LK} = \mathbb{N}$.

The game Game_R is played by the adversary $\mathscr{A}$ and the challenger $\mathscr{B}$ as follows.

4.1. Initializing. The challenger runs Setup to get the public parameters params and the master key msk. It keeps msk as secret and gives params to the adversary. The handle h is set to 0. It adds an item $(0, 0, 0, 0)$ in $\mathscr{L}$.

The adversary $\mathscr{A}$ can query the following oracles:

$\mathscr{O}$ – Create(ID): given an identity ID, the challenger $\mathscr{B}$ looks up the item with the identity ID in $\mathscr{L}$. If ID is in $\mathscr{L}$, it outputs $\perp$. Otherwise, the challenger $\mathscr{B}$ invokes KeyGen to create the private key $\mathrm{SK}_{\mathrm{ID}}$ and adds the item $(h + 1, \mathrm{ID}, \mathrm{SK}_{\mathrm{ID}}, 0)$ in $\mathscr{L}$. The challenger $\mathscr{B}$ updates the handle $h \leftarrow h + 1$.

$\mathscr{O}$ – Leak(h, f): the adversary queries the leakage for a private key that has the handle h. The adversary selects a function f which takes the private key as input and gives the output with constant size. The function f is computable in polynomial time. If the challenger $\mathscr{B}$ finds an item $(h, \mathrm{ID}, \mathrm{SK}_{\mathrm{ID}}, L)$ which has the handle h in list $\mathscr{L}$, the challenger $\mathscr{B}$ checks if $L + |f(\mathrm{SK}_{\mathrm{ID}})| \le L_{\mathrm{SK}}$, where L_{SK} is the maximum of leakage for the private key. If it is true, it will give $f(\mathrm{SK}_{\mathrm{ID}})$ to the adversary and updates the tuple $(h, \mathrm{ID}, \mathrm{SK}_{\mathrm{ID}}, L)$ with $(h, \mathrm{ID}, \mathrm{SK}_{\mathrm{ID}}, L + |f(\mathrm{SK}_{\mathrm{ID}})|)$ in the list $\mathscr{L}$. Otherwise, the challenger returns $\perp$.

$\mathscr{O}$ – Reveal(h): the adversary $\mathscr{A}$ queries the private key for handle h. The challenger checks whether the item with handle h is in list $\mathscr{L}$. If the item (h, ID, SK_{ID}, L) is in $\mathscr{L}$. The challenger gives the private key $\mathrm{SK}_{\mathrm{ID}}$ to the adversary $\mathscr{A}$ and puts the identity ID into $\mathscr{R}$.

$\mathscr{O}$ – KeyUpd(h): for handle h, the adversary $\mathscr{A}$ queries the updated private key. The challenger checks whether the item with handle h is in list $\mathscr{L}$. If the item $(h, \mathrm{ID}, \mathrm{SK}_{\mathrm{ID}}, L)$ is in $\mathscr{L}$. The challenger $\mathscr{B}$ runs the algorithm KeyUpd. $\mathscr{B}$ gives the updated key $\widehat{\mathrm{SK}_{\mathrm{ID}}}$ to the adversary $\mathscr{A}$ and updates the item $(h, \mathrm{ID}, \mathrm{SK}_{\mathrm{ID}}, L)$ with $(h, \mathrm{ID}, \widehat{\mathrm{SK}_{\mathrm{ID}}}, L)$. Otherwise, $\mathscr{B}$ returns $\perp$.

$\mathscr{O}$ – Sign(m, ID): the adversary selects any identity ID and the message m which will be signed and asks the challenger to generate the signature σ about m by the identity ID.

4.2. Forgery. The adversary generates a forged signature σ^* about a message m^* and the identity ID^*. If the following

conditions are met, it is said that the adversary wins the game:

(1) σ^* is a valid signature of a message m^*. That is to say, it can satisfy the signature verification algorithm Verify.

(2) The private key about the identity ID^* is not asked by the adversary.

(3) The adversary does not ask for the signature $\sigma^* s$ of the message m^*.

If all PPT adversaries can only obtain negligible advantages in Game_R, the CLR-IBS is said to be secure against private key leakage L_{SK}.

5. Construction of CLR-IBS

Based on the composite order group of 3 primes, we propose the CLR-IBS scheme, which consists of six algorithms. Subgroup G_{p_3} is used to randomize private keys. Subgroup G_{p_2} is only used to generate the semifunctional private key in the proof.

5.1. Setup. The algorithm randomly selects $g_1, u_1, h_1, u_2, h_2 \in G_{p_1}$, $g_3 \in G_{p_3}$, $x_1, \ldots, x_n \in Z_N$, $\alpha, r \in Z_N$, $y_1, \ldots, y_n \in Z_N$, $\overrightarrow{\rho} = (\rho_1, \ldots, \rho_{n+1}) \in Z_N^{n+1}$, and $\rho_{n+2} \in Z_N$, where $n \geq 2$ is an integer. The value of n is variable. If n is large, the leakage rate is high accordingly. The leakage rate is the ratio of leakage size to the size of a secret key. If n is small, the public key is short.

The public parameters are $\mathrm{params} = \{N, g_1, g_3, h_1, u_1, h_2, u_2, g_1^{x_1}, \ldots, g_1^{x_n}, e(g_1, g_1)^{\alpha}\}$ and the master key is $\mathrm{msk} = (\overrightarrow{F}_1, F_2) = (< g_1^{y_1}, \ldots, g_1^{y_n}, g_1^{\alpha} \prod_{j=1}^{n} g_1^{-x_j y_j} > * g_3^{\overrightarrow{\rho}_1}, g_3^{\rho_{n+2}})$.

5.2. KeyGen. For the identity $\mathrm{ID} \in Z_N$. The algorithm randomly selects $z_1, \ldots, z_n \in Z_N$, $\overrightarrow{\rho}' = (\rho_1', \ldots, \rho_{n+1}') \in Z_N^{n+1}$, $\rho_{n+2}' \in Z_N$ and $r_{\mathrm{ID}} \in Z_N$. It generates the private key

$$
\begin{aligned}
\mathrm{SK}_{\mathrm{ID}} &= \left(\overrightarrow{K}_1, K_2\right), \\
&= \left(\overrightarrow{F}_1 * < g_1^{z_1}, \ldots, g_1^{z_n}, \left(u_1^{ID} h_1\right)^{r_{ID}} \prod_{j=1}^{n} g_1^{-x_j z_j} > * g_3^{\overrightarrow{\rho}'_1}, F_2 . g_1^{r_{\mathrm{ID}}} . g_3^{\rho_{n+2}'}\right) \\
&= \left(< g_1^{y_1}, \ldots, g_1^{y_n}, g_1^{\alpha} \prod_{j=1}^{n} g_1^{-x_j y_j} > * g_3^{\overrightarrow{\rho}_1} * < g_1^{z_1}, \ldots, g_1^{z_n}, \left(u_1^{ID} h_1\right)^{r_{ID}} \prod_{j=1}^{n} g_1^{-x_j z_j} > * g_3^{\overrightarrow{\rho}'_1}, \; g_3^{\rho_{n+2}} . g_1^{r_{ID}} . g_3^{\rho_{n+2}'}\right) \\
&= \left(< g_1^{y_1+z_1}, \ldots, g_1^{y_n+z_n}, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{\mathrm{ID}}} \prod_{j=1}^{n} g_1^{-x_j \left(y_j+z_j\right)} > * g_3^{\overrightarrow{\rho}_1 + \overrightarrow{\rho}'_1}, \; g_1^{r_{ID}} . g_3^{\rho_{n+2}+\rho_{n+2}'}\right).
\end{aligned}
\tag{1}
$$

Denote $w_j = y_j + z_j (j = \{1, \ldots, n\})$, $\overrightarrow{\rho}'' = \overrightarrow{\rho}_1 + \overrightarrow{\rho}\prime$, and $\rho_{n+2}'' = \rho_{n+2} + \rho_{n+2}'$. We get $\mathrm{SK}_{\mathrm{ID}} = (\overrightarrow{K}_1, K_2)$:

$$
= \left(< g_1^{w_1}, \ldots, g_1^{w_n}, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{ID}} \prod_{j=1}^{n} g_1^{-x_j w_j} > * g_3^{\overrightarrow{\rho}''}, \; g_1^{r_{ID}} * g_3^{\rho_{n+2}''}\right) \tag{2}
$$

5.3. KeyUpd. The algorithm takes public parameters params and the private key $\mathrm{SK}_{\mathrm{ID}}$ as input and outputs a new private key $\widehat{\mathrm{SK}_{\mathrm{ID}}}$. For the private key $\mathrm{SK}_{\mathrm{ID}} = (\overrightarrow{K}_1, K_2) = (< g_1^{w_1}, \ldots, g_1^{w_n}, g_1^{\alpha} \; (u_1^{ID} h_1)^{r_{ID}} \prod_{j=1}^{n} g_1^{-x_j w_j} > * g_3^{\overrightarrow{\rho}''}, \; g_1^{r_{ID}} * g_3^{\rho_{n+2}''})$, the algorithm selects randomly $\Delta w_j \in Z_N$ $(j \in \{1, 2, \ldots, n\})$, $\Delta \overrightarrow{\rho}'' = \{\Delta\rho_1'', \Delta\rho_2'', \ldots, \Delta\rho_{n+1}''\} \in Z_N^{n+1}$, and $\Delta\rho_{n+2}'', \Delta r_{ID} \in Z_N$. It gets the new private key.

$\widehat{\mathrm{SK}_{\mathrm{ID}}} = (< g_1^{w_1+\Delta w_1}, \ldots, g_1^{w_n+} \; \Delta w_n, g_1^{\alpha} (u_1^{ID} \; h_1) - (r_{ID} + \Delta r_{ID}) \prod_{j=1}^{n} g_1^{-x_j (w_j + \Delta w_j)} > * g_3^{\overrightarrow{\rho}'' + \Delta \overrightarrow{\rho}''}, \; g_1^{r_{ID}+\Delta r_{ID}} . g_3^{\Delta\rho_{n+2}''+\Delta\rho_{n+2}''})$. Because $\Delta w_j \in Z_N (j = \{1, \ldots, n\})$, $\Delta \overrightarrow{\rho}'' \in Z_N^{n+1}$, and $\Delta\rho_{n+2}'', \Delta r_{ID} \in Z_N$ are all random, we know that $w_j + \Delta w_j (j = \{1, \ldots, n\})$, $\overrightarrow{\rho}'' + \Delta\overrightarrow{\rho}''$, and $r_{\mathrm{ID}} + \Delta r_{\mathrm{ID}}$ are random. The private keys $\widehat{\mathrm{SK}_{\mathrm{ID}}}$ and $\mathrm{SK}_{\mathrm{ID}}$ have the same distribution. Thus, the private key is updated.

Denote $w_j' = w_j + \Delta w_j (j = \{1, \ldots, n\})$, $r_{\mathrm{ID}}' = r_{\mathrm{ID}} + \Delta r_{\mathrm{ID}}$, $\overrightarrow{\rho}''' = \overrightarrow{\rho}'' + \Delta\overrightarrow{\rho}''$, and $\rho_{n+2}''' = \rho_{n+2}'' + \Delta\rho_{n+2}''$. The updated

private key is written as $\widetilde{SK_{ID}} = (< g_1^{w_1'}, \ldots, g_1^{w_n'}, g_1^{\alpha} (u_1^{ID} h_1)^{-r_{ID}'} \prod_{j=1}^{n} g_1^{-x_j w_j'} > * g_3^{\overrightarrow{\rho}'''}, g_1^{r_{ID}'} * g_3^{\rho'''_{n+2}})$, which has the same form as the original private key SK_{ID}.

5.4. Sign. For the message $m \in Z_N$, the user ID chooses randomly $r_{ID}' \in Z_N$ and $r_m \in Z_N$ and signs the message m with his private key $SK_{ID} = (\overrightarrow{K}_1, K_2)$.

$$\begin{aligned}\sigma &= \left(\overrightarrow{\sigma}_1, \sigma_2, \sigma_3\right) = \left(\overrightarrow{K}_1 * < 1, \ldots, 1, \left(u_1^{ID} h_1\right)^{r_{ID}'} (u_2^m h_2)^{r_m} >, K_2 \cdot g^{r_{ID}'}, g^{r_m}\right) \\ &= \left(< g_1^{w_1}, \ldots, g_1^{w_n}, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{ID} + r_{ID}'} (u_2^m h_2)^{r_m} \prod_{j=1}^{n} g_1^{-x_j w_j} > * g_3^{\overrightarrow{\rho}'}, g_1^{r_{ID} + r_{ID}'} \cdot g_3^{\rho_{n+2}'}, g^{r_m} \right).\end{aligned} \tag{3}$$

5.5. Verify. The receiver receives the signature $\sigma = (\overrightarrow{\sigma}_1, \sigma_2, \sigma_3)$ for the message m from the user ID. Then he verifies whether $e(< g_1^{x_1}, \ldots, g_1^{x_n}, g_1 >, \overrightarrow{\sigma}_1) = e(g_1, g_1)^{\alpha} \cdot e(u_1^{ID} h_1, \sigma_2) \cdot e(u_2^m h_2, \sigma_3)$.

If the equation does not hold, then it outputs "reject." Otherwise, it outputs "accept."

5.6. KeyGenSf. First, the private key generator calls the normal private key generation algorithm to generate the normal private key $SK_{ID} = (\overrightarrow{K}_1, K_2)$. Second, a private key generator randomly selects $\overrightarrow{\gamma} = (\gamma_1, \ldots, \gamma_{n+1}) \in Z_N^{n+1}$ and $\gamma_{n+2} \in Z_N$ to generate the semifunctional private key $\widetilde{SK_{ID}}$.

$$\begin{aligned}\widetilde{SK_{ID}} &= \left(\overrightarrow{K}_1 * g_2^{\overrightarrow{\gamma}}, K_2 * g_2^{\gamma_{n+2}}\right) \\ &= \left(< g_1^{w_1}, \ldots, g_1^{w_n}, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{ID}} \prod_{j=1}^{n} g_1^{-x_j w_j} \right. \\ &\quad \left. > * g_3^{\overrightarrow{\rho}''} * g_2^{\overrightarrow{\gamma}}, g_1^{r_{ID}} g_3^{\overrightarrow{\rho''}_{n+2}} \cdot g_2^{\gamma_{n+2}} \right).\end{aligned} \tag{4}$$

The signatures of a semifunctional private key and normal private key can all pass the verification algorithm. Semifunctional private key is only used for a security proof. In practical application, we will use the normal private key for signature.

Correctness is as follows:

$$\begin{aligned}&e\left(< g_1^{x_1}, \ldots, g_1^{x_n}, g_1 >, \overrightarrow{\sigma}_1 \right) \\ &= e\left(< g_1^{x_1}, \ldots, g_1^{x_n}, g_1 >, < g_1^{w_1}, \ldots, g_1^{w_n}, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{ID} + r_{ID}'} (u_2^m h_2)^{r_m} \prod_{j=1}^{n} g_1^{-x_j w_j} > * g_3^{\overrightarrow{\rho}''} \right) \\ &= \prod_{j=1}^{n} e\left(g_1^{x_j}, g_1^{w_j}\right) \cdot e\left(g_1, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{ID} + r_{ID}'} (u_2^m h_2)^{r_m}\right) \cdot e\left(g_1, \prod_{j=1}^{n} g_1^{-x_j w_j}\right) \\ &= \prod_{j=1}^{n} e\left(g_1, g_1^{x_j w_j}\right) \cdot e\left(g_1, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{ID} + r_{ID}'} (u_2^m h_2)^{r_m}\right) . e\left(g_1, \prod_{j=1}^{n} g_1^{-x_j w_j}\right) \\ &= e\left(g_1, \prod_{j=1}^{n} g_1^{x_j w_j}\right) \cdot e\left(g_1, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{ID} + r_{ID}'} (u_2^m h_2)^{r_m}\right) \cdot e\left(g_1, \prod_{j=1}^{n} g_1^{-x_j w_j}\right) \\ &= e\left(g_1, g_1^{\alpha} \left(u_1^{ID} h_1\right)^{r_{ID} + r_{ID}'} (u_2^m h_2)^{r_m}\right)\end{aligned}$$

$$
\begin{aligned}
&= e(g_1, g_1^{\alpha}) \cdot e\left(g_1, \left(u_1^{ID} h_1\right)^{r_{ID}+r_{ID}'} (u_2^m h_2)^{r_m}\right) \\
&= e(g_1, g_1)^{\alpha} \cdot e\left(g_1, \left(u_1^{ID} h_1\right)^{r_{ID}+r_{ID}'}\right) \cdot e\left(g_1, (u_2^m h_2)^{r_m}\right) \\
&= e(g_1, g_1)^{\alpha} \cdot e\left(g_1^{r_{ID}+r_{ID}'}, u_1^{ID} h_1\right) \cdot e\left(g_1^{r_m}, u_2^m h_2\right) \\
&= e(g_1, g_1)^{\alpha} \cdot e\left(\sigma_2, u_1^{ID} h_1\right) \cdot e\left(\sigma_3, u_2^m h_2\right).
\end{aligned} \tag{5}
$$

6. Security Proof

If the forgery signature $(m^*, \sigma^* = (\vec{\sigma}_1^*, \sigma_2^*, \sigma_3^*))$ is valid, suppose $z \in Z_N$ and $z = (0 \bmod p_1, 1 \bmod p_2, 0 \bmod p_3)$. The forgery signature can be divided into two types. Type I: $(\vec{\sigma}_1^*)^z = (\vec{1})_{n+1}$, $(\sigma_2^*)^z = 1$, $(\sigma_3^*)^z = 1$. Type II: $(\vec{\sigma}_1^*)^z \neq (\vec{1})_{n+1}$ or $(\sigma_2^*)^z \neq 1$ or $(\sigma_3^*)^z \neq 1$.

Theorem 1. *If Assumptions 1–3 are true, the signature scheme given in this paper is leakage-resilient and secure in the standard model. The amount of private key leakage is $L_{SK} = (n - 2\Lambda - 1)\lambda$, where $\lambda = \log p_2$ and $n \geq 2$ is a positive constant integer.*

When n is large, the tolerable leakage rate is high. When n is relatively small, the public parameters is also relatively short. The specific leakage performance analysis is given in Section 8.

In general, we prove the security of the scheme by dual system encryption technology. A series of games are used to complete the proof. Suppose the adversary makes q_1 private key extraction queries, q_2 signature queries. Let $q = q_1 + q_2$. Game_R is the real security game. The other games are modified from the game Game_R. The first game is a real security game and the adversary has only a negligible advantage in winning the last one. The adjacent two games are indistinguishable.

These games are defined as follows:

Game_R:this is a real security game.

Game_0: it is similar to Game_R, but it has some restrictions. When the adversary outputs the forged signature of the message m^*, it needs that $\text{ID}^* \neq \text{ID}_i \bmod p_2$ $(1 \leq i \leq q_1)$ and $(ID^*, m^*) \neq (ID_j, m_j) \bmod p_2$ $(1 \leq i \leq q_2)$, where ID_i $(1 \leq i \leq q_1)$ is the i^{th} inquired identity and (ID_j, m_j) is used for the j^{th} signature query.

$Game_i$ $(i \in [1, q])$: in this game, for the previous i private key query, the challenger answers with the semifunctional private key (if it is a signature query, the challenger first calculates the corresponding semifunctional private key and then uses the signature algorithm to generate the signature). The rest is the same as Game_0.

Proof. Through a series of games Game_R and $Game_i$ $(i \in (0, 1, \ldots, q))$, we use Lemmas 1–6 to prove the security. First, Lemma 1 is used to obtain the leakage bound. Second, we use Lemmas 2–6 to prove that these games are indistinguishable. Thus, the safety can be proved. □

Lemma 1. *The amount of private key leakage about our CLR-IBS can reach $L_{SK} = (n - 2\Lambda - 1)\lambda$.*

Proof. We use a conclusion in [18] to prove this lemma.

Conclusion 1 [18]: suppose that p is a prime number, $n_1 \geq n_2 \geq 2$ $(n_1, n_2 \in N)$ and $\Phi \longleftarrow Z_p^{n_1}$. Let that X be a matrix $(X \longleftarrow Z_p^{n_1 \times n_2})$ and Y be also a matrix with rank 1 $(Y \longleftarrow Rk_1(Z_p^{n_2 \times 1}))$. ε is used to represent a value that can be ignored. If $f: Z_p^{n_1} \longrightarrow W$ is a leakage function where $|W| \leq 4 \cdot (1 - 1/p) \cdot p^{n_2 - 1} \cdot \varepsilon^2$, the statistical distance $SD(X, f(X \cdot Y)), (X, f(\Phi)) \leq \varepsilon$.

From conclusion 1, we can easily get the following Ratiocination 1.

Ratiocination 1: suppose that p is a prime and $n_1 \geq 3$. Select $\vec{\delta} \leftarrow Z_p^{n_1}$, $\vec{\tau} \leftarrow Z_p^{n_1}$ and $\vec{\tau}' \leftarrow Z_p^{n_1}$, such that $\vec{\tau}'$ and $\vec{\delta}$ are orthogonal modulo p by dot product. Let f be a leakage function mapping Z_p^m to W (i.e. $f: Z_p^m \longrightarrow W$). If $|W| \leq 4 \cdot (1 - 1/p) \cdot p^{n-1} \cdot \varepsilon^2$, the statistical distance $SD((\vec{\delta}, f(\vec{\tau}\imath)), (\vec{\delta}, f(\vec{\tau})))$ is negligible. That is to say, $SD((\vec{\delta}, f(\vec{\tau}')), (\vec{\delta}, f(\vec{\tau}))) \leq \varepsilon$ where ε is negligible. □

Proof. By Conclusion 1, we set $n_2 = n_1 - 1$, so $n_1 = n_2 + 1 \geq n_2 \geq 2$. Thus, $\vec{\tau}$ matches to Φ and the basis of the orthogonal space of $\vec{\delta}$ matches to X. So, the distribution of $\vec{\tau}\imath$ is the same as $X \cdot Y$ when $Y \leftarrow Rk_1(Z_p^{(n_1-1)\times 1})$. That is to say, Y is a matrix of $n_1 - 1$ rows and 1 column with rank 1. Since $\vec{\delta}$ is chosen randomly, $X \leftarrow Z_p^{n_1 \times (n_1 - 1)}$ is determined by $\vec{\delta}$. Thus, we conclude that $SD((\vec{\delta}, f(\vec{\tau}\imath)), (\vec{\delta}, f(\vec{\tau}))) = \text{dist}(X, f(X \cdot T)), (X, f(\Phi))$.

Let $n_2 = n$, $p_2 = p$ and $\varepsilon = p_2^{-\Lambda}$. λ denotes $\log p_2$. The leakage size will be up to $\log|W| \leq (n-1)\log p_2 - 2\Lambda \log p_2 = (n - 2\Lambda - 1)\log p_2 = (n - 2\Lambda - 1)\lambda$. Therefore, we conclude that the leakage amount is up to $L_{SK} = L = (n - 2\Lambda - 1)\lambda$. □

Lemma 2. *In the case of private key leakage $L_{SK} = (n - 2\Lambda - 1)\lambda$, if there is an algorithm $\mathcal{A}$ (the adversary) such that $|Adv_{\mathcal{A}}^{\text{Game}_R} - Adv_{\mathcal{A}}^{\text{Game}_0}| \geq \varepsilon$, we can construct an algorithm $\mathcal{B}$ (the challenger) to break assumption 2 with a nonnegligible advantage.*

Proof. First, given $\mathscr{B}$ an instance $D^2 = (N, G, G_T, e, g_1, g_3, g_1^z g_2^v, g_2^u g_3^\varsigma)$, $T_0 = g_1^\omega g_3^\sigma$ and $T_1 = g_1^\omega g_2^\kappa g_3^\sigma$ $(\omega, \kappa, \sigma \in Z_N)$, $\mathscr{B}$ plays Game_R or Game_0 with $\mathscr{A}$. The challenger publishes the system parameters params $= \{N, g_1, g_3, h_1 = g_1^{b_1}, u_1 = g_1^{a_1}, h_2 = g_1^{b_1}, u_2 = g_1^{a_2}, g_1^{x_1}, \ldots, g_1^{x_n}, e(g_1, g_1)^\alpha\}$ to the adversary, where $\alpha, a_1, a_2, b_1, b_2 \in Z_N, x_1, \ldots, x_n \in Z_N$ are randomly selected. For any private key extraction query or signature query of the adversary, the challenger can use the master key to calculate.

Finally, suppose that the adversary generates a forged signature $\sigma^* = (\overrightarrow{\sigma}_1^*, \sigma_2^*, \sigma_3^*)$ of the user ID^* about the message m^*. The adversary hopes that the forged signature can pass the verification. For $1 \le i \le q_1$, $\text{ID}^* = \text{ID}_i \bmod p_2$ and $\text{ID}^* \ne \text{ID}_i \bmod N$, if $a = \gcd(\text{ID}^*, \text{ID}_i)$ and $b = N/a$, three cases are considered:

(1) $a = p_2 p_3, b = p_1$

(2) $a = p_1 p_2, b = p_3$

(3) $a = p_2, b = p_1 p_3$

If $(g_2^u g_3^\varsigma)^a = 1$, the first case occurs and $b = p_1$. Judge whether $e(T^b, g_1^z g_2^v) = 1$. If the equation holds, $T \in G_{p_1 p_3}$. Otherwise, $T \in G_{p_1 p_2 p_3}$.

If $(g_2^u g_3^\varsigma)^a \ne 1$, judge whether $(g_1^z g_2^v)^a = 1$. If the equation holds, the second case occurs and $b = p_3$. Then, judge whether $e(T^b, g_2^u g_3^\varsigma) = 1$. If the equation holds, $T \in G_{p_1 p_3}$. Otherwise, $T \in G_{p_1 p_2 p_3}$.

If $(g_2^u g_3^\varsigma)^a \ne 1$ and $(g_1^z g_2^v)^a \ne 1$, the third case occurs and $b = p_1 p_3$. Judge whether $T^b = 1$. If the equation holds, $T \in G_{p_1 p_3}$. Otherwise, $T \in G_{p_1 p_2 p_3}$.

Similarly, if for some j where $1 \le j \le q_2$, $\text{ID}^* = \text{ID}_j \bmod N$, $m^* = m_j \bmod p_2$ and $m^* \ne m_j \bmod N$, the challenger can still break the assumption 2. The adversary outputs the forged signature $\sigma^* = (\overrightarrow{\sigma}_1^*, \sigma_2^*, \sigma_3^*)$ of the use ID^* about the message m^*, where $\text{ID}^* \ne \text{ID}_i \bmod p_2$ $(1 \le i \le q_1)$ and $(\text{ID}^*, m^*) \ne (\text{ID}_j, m_j) \bmod p_2$ $(1 \le j \le q_2)$.

Probability analysis: if $T \in G_{p_1 p_3}$, $\mathscr{B}$ simulates the game Game_0 properly. If $T \in G$, $\mathscr{B}$ simulates the game Game_R properly. Thus, $|\Pr[\mathscr{B}(D^2, T \in G_{p_1 p_3}) = 0] - \mathscr{B}(D^2, T \in G) = 0]| = |\text{Adv}_{\mathscr{A}}^{\text{Game}_0} - \text{Adv}_{\mathscr{A}}^{\text{Game}_R}| \ge \varepsilon$.

In the case of private key leakage $L_{\text{SK}} = (n - 2\Lambda - 1)\lambda$, if there is an adversary who can distinguish Game_0 from Game_R with an advantage ε that cannot be ignored, $\mathscr{B}$ can break Assumption 2 with an advantage ε that cannot be ignored. This contradicts Assumption 2. Consequently, $|\text{Adv}_{\mathscr{A}}^{\text{Game}_0} - \text{Adv}_{\mathscr{A}}^{\text{Game}_R}| \le \varepsilon$. Thus, Lemma 2 holds. □

Lemma 3. *Under Assumption 1 and in the case of private key leakage $L_{SK} = (n - 2\Lambda - 1)\lambda$, the adversary (an algorithm $\mathscr{A}$) can only output the forged signature of type II with negligible advantage in game Game_0.*

Proof. First, given $\mathscr{B}$ an instance $D^1 = (N, G, G_T, e, g_1, g_3)$, $T_0 = g_1^z$ and $T_1 = g_1^z g_2^v$ $(z, v \in Z_N)$, $\mathscr{B}$ plays Game_0 with $\mathscr{A}$. The challenger publishes the system parameters params $= \{N, g_1, g_3, h_1 = g_1^{b_1}, u_1 = g_1^{a_1}, h_2 = g_1^{b_1}, u_2 = g_1^{a_2}, g_1^{x_1}, \ldots, g_1^{x_n}, e(g_1, g_1)^\alpha\}$ to the adversary, where $\alpha, a_1, a_2, b_1, b_2 \in Z_N$, and $x_1, \ldots, x_n \in Z_N$ are randomly selected. For any private key extraction query or signature query of the adversary, the challenger can use the master key to calculate.

Finally, suppose that the adversary generates the forged signature $\sigma^* = (\overrightarrow{\sigma}_1^*, \sigma_2^*, \sigma_3^*)$ of the user ID^* about the message m^*.

$$\sigma^* = (\overrightarrow{\sigma}_1^*, \sigma_2^*, \sigma_3^*) = (< g_1^{w_1}, \ldots, g_1^{w_n}, \quad g_1^\alpha (u_1^{ID^*} h_1)^{r_{ID}} (u_2^{m^*} h_2)^{r_m} \prod_{j=1}^n g_1^{-x_j w_j} > * \overrightarrow{g_3^{\rho''}} * \overrightarrow{g_2^{s}}_{12}, g_1^{r_{ID}} * g_3^{\rho''_{n+2}} * g_2^{s_{22}}, g_1^{r_m}.g_2^{s_{32}})$$

The adversary hopes that the forged signature can pass the verification.

Furthermore, the challenger checks whether the forged signature satisfies the following equation:

$$e\left(< T^{x_1}, \ldots, T^{x_n}, T >, \overrightarrow{\sigma}^*{}_1\right) = e(g_1, T)^\alpha \cdot e\left(g_1^{a_1 ID^* + b_1}, \sigma_2^*\right) \cdot e\left(g_1^{a_2 ID^* + b_2}, \sigma_3^*\right) \tag{6}$$

Obviously, if $T \in G_{p_1}$, the above equation is always true. If $T \in G_{p_1 p_2}$, the equation also holds when the adversary forges the signature of type I with the probability ε.

If the adversary forges the signature of type II with the probability ε, the equation holds if and only if $\overrightarrow{s}_{12} \cdot < 0, \ldots, 0, 1 > = s_{22}(a_1 ID^* + b_1) + s_{32}(a_2 m^* + b_2) \bmod p_2$. In the whole game, the adversary will not get any information about $(a_1, b_1, a_2, b_2) \bmod p_2$. So, the probability that the equation holds is negligible.

That is to say, in the case of private key leakage $L_{SK} = (n - 2\Lambda - 1)\lambda$, if there is an adversary who can output the forged signature of type II with an advantage ε that cannot be ignored, the equation does not hold. We judge that $T \in G_{p_1 p_2}$. □

Lemma 4. *Under Assumption 2 and in the case of private key leakage $L_{SK} = (n - 2\Lambda - 1)\lambda$, if the adversary (an algorithm $\mathscr{A}$) can output the forged signature of type II with negligible advantage in the game Game_{k-1}, $\mathscr{A}$ can also output the forged signature of type II with negligible advantage in game Game_k.*

Proof. First, given $\mathscr{B}$ an instance $D^2 = (N, G, G_T, e, g_1, g_3, g_1^z g_2^v, g_2^u g_3^\varsigma)$, $T_0 = g_1^\omega g_3^\sigma$, $T_1 = g_1^\omega g_2^\kappa g_3^\sigma$ $(\omega, \kappa, \sigma \in Z_N)$, $Y = (y_1, \ldots, y_{n+1}) \in G_{p_2}^{n+1}$ and $y \in G_{p_2}$, $\mathscr{B}$ plays Game_{k-1} or Game_k with $\mathscr{A}$. The challenger publishes the system parameters params $= \{N, g_1, g_3, h_1 = g_1^{b_1}, u_1 = g_1^{a_1}, h_2 = g_1^{b_1}, u_2 = g_1^{a_2}, g_1^{x_1}, \ldots, g_1^{x_n}, e(g_1, g_1)^\alpha\}$ to the adversary, where $\alpha, a_1, a_2, b_1, b_2 \in Z_N$, and $x_1, \ldots, x_n \in Z_N$ are randomly selected. For any private key extraction query or signature query of the adversary, the challenger can use the master key to calculate.

For the adversary's previous $k - 1$ private key extraction queries, $\mathscr{B}$ will generate a semifunctional private key. $\mathscr{B}$ selects $t_1, \ldots, t_{n+2} \in Z$ randomly and returns $\widetilde{\text{SK}_{\text{ID}}} = (\langle g_1^{w_1} (g_2^u g_3^\varsigma)^{t_1}, \ldots, g_1^{w_n} \quad (g_2^u g_3^\varsigma)^{t_n}, g_1^\alpha (u_i^{ID_i} h_1)^{r_{ID_i}} \prod_{j=1}^n g_1^{-x_j w_j} (g_2^u g_3^\varsigma)^{t_{n+1}} >, \ g_1^{r_{ID_i}} (g_2^u g_3^\varsigma)^{t_{n+2}} \rangle)$. If it is a signature query, the challenger first calculates the corresponding private key and then uses the private key to calculate the corresponding signature. The following cases will be handled in this way.

For the adversary's later $q-k$ private key extraction queries, $\mathcal{B}$ selects $t_1, \ldots, t_{n+2} \in Z$ randomly and returns $\mathrm{SK}_{\mathrm{ID}} = (\langle g_1^{w_1} \quad (g_3^{\varsigma})^{t_1}, \ldots, g_1^{w_n} (g_3^{\varsigma})^{t_n}, g_1^{\alpha} (u_i^{ID_i} h_1)^{r_{ID_i}} \quad \prod_{j=1}^{n} g_1^{-x_j w_j} (g_3^{\varsigma})^{t_{n+1}} \rangle, \; g_1^{r_{ID_i}} (g_3^{\varsigma})^{t_{n+2}} \rangle)$. If it is a signature query, the challenger first calculates the corresponding private key and then uses the private key to calculate the corresponding signature.

For the adversary's k^{th} private key extraction query, $\mathcal{B}$ selects $w_1, \ldots, w_n \in Z_N$, $r_i \in Z_N$ randomly and returns $\mathrm{SK}_{\mathrm{ID}} = (< T^{w_1}, \ldots, T^{w_n}, (g_1^{\alpha} \prod_{j=1}^{n} g_1^{-x_j} \quad w_j T^{(a_1 ID_k + b_1)} > g_3^{\overrightarrow{\rho}},) T g_3^{\rho_{n+2}})$.

If $T \in G_{p_1 p_3}$, the private key is normal. $\mathcal{B}$ simulates the game Game_{k-1} properly. If $T \in G$, the private key is semi-functional. $\mathcal{B}$ simulates the game Game_k properly.

Finally, suppose that the adversary generates the forged signature $\sigma^* = (\overrightarrow{\sigma}^*_1, \sigma_2^*, \sigma_3^*)$ of the user ID^* about the message m^*. $\sigma^* = (\overrightarrow{\sigma}^*_1, \sigma_2^*, \sigma_3^*) = (< g_1^{w_1}, \ldots, g_1^{w_n}, g_1^{\alpha} (u_1^{ID^*} h_1)^{r_{ID}} (u_2^{m^*} h_2)^{r_m} \prod_{j=1}^{n} g_1^{-x_j w_j} > * g_3^{\overrightarrow{\rho'}} * g_2^{\overrightarrow{s}_{12}}, g_1^{r_{ID}} * g_3^{\rho_{n+2}'} * g_2^{s_{22}}, g_1^{r_m} * g_2^{s_{32}})$. The adversary hopes that the forged signature can pass the verification.

Furthermore, the challenger checks whether the forged signature satisfies the following equation:

$$e\Big(< (g_1^z g_2^v)^{x_1}, \ldots, (g_1^z g_2^v)^{x_n}, g_1^z g_2^v >, \overrightarrow{\sigma}^*_1 \Big) = e\big(g_1, (g_1^z g_2^v)\big)^{\alpha} \cdot e\Big(g_1^{a_1 ID^* + b_1}, \sigma_2^*\Big) \cdot e\Big(g_1^{a_2 ID^* + b_2}, \sigma_3^*\Big). \tag{7}$$

Probability analysis: if $T \in G_{p_1 p_3}$, $\mathcal{B}$ simulates the game Game_{k-1} properly. If the adversary outputs a forged signature of type I with an advantage that can be ignored, the equation also holds. Thus, the adversary outputs the forged signature of type II with an advantage that can also be ignored.

If $T \in G_{p_1 p_2 p_3}$, $\mathcal{B}$ simulates the game Game_k properly. If the adversary outputs a forged signature of type I with an advantage ε that can be ignored, the equation also holds.

When the adversary forges the signature of type II with the probability ε if and only if $\overrightarrow{s}_{12}. < 0, \ldots, 0, 1 > = s_{22}(a_1 ID^* + b_1) + s_{32}(a_2 m^* + b_2) \bmod p_2$, the equation also holds. In the whole game, the adversary gets information about $(a_1, b_1, a_2, b_2) \bmod p_2$ only by the k^{th} query. Because $\mathrm{ID}^* \neq \mathrm{ID}_i \bmod p_2 \; (1 \le i \le q_1)$, $(ID^*, m^*) \neq (ID_j, m_j) \bmod p_2 \; (1 \le j \le q_2)$ and $m^* = m_j \bmod p_2$, the equation holds with a probability ε that can also be ignored.

That is to say, in the case of private key leakage $L_{\mathrm{SK}} = (n - 2\Lambda - 1)\lambda$, if there is an adversary in the game Game_k who can output the forged signature of type II with an advantage ε that cannot be ignored such that the equation does not hold, we judge that $T \in G_{p_1 p_2 p_3}$. □

Lemma 5. *Under Assumption 3 and in the case of private key leakage $L_{SK} = (n - 2\Lambda - 1)\lambda$, the adversary (an algorithm $\mathcal{A}$) can only output the forged signature of type I with negligible advantage in the game Game_q.*

Proof. First, given the challenger $\mathcal{B}$ an instance $\underline{D}^3 = (N, G, G_T, e, g_1, g_2, g_3, g_1^{\alpha} g_2^{v}, g_1^{s} g_2^{u})$, $\overrightarrow{Y} = (y_1, \ldots, y_{n+1}) \in G_{p_2}^{n+1}$, $y \in G_{p_2}$ and T ($T = e(g_1, g_1)^{\alpha s}$ or $T \in G_{p_1}$, where $s \in Z_N$), $\mathcal{B}$ plays Game_q with $\mathcal{A}$.

The challenger publishes the system parameters $\mathrm{params} = \{N, g_1, g_3, h_1 = g_1^{b_1}, u_1 = g_1^{a_1}, h_2 = g_1^{b_1}, u_2 = g_1^{a_2}, g_1^{x_1}, \ldots, g_1^{x_n}, e(g_1, g_1)^{\alpha}\}$ to the adversary, where $\alpha, a_1, a_2, b_1, b_2 \in Z_N$, and $x_1, \ldots, x_n \in Z_N$ are randomly selected. For any private key extraction query or signature query of the adversary, the challenger can use the master key to calculate.

Finally, suppose that the adversary generates the forged signature $\sigma^* = (\overrightarrow{\sigma}^*_1, \sigma_2^*, \sigma_3^*)$ of the user ID^* about the message m^*. $\sigma^* = (\overrightarrow{\sigma}^*_1, \sigma_2^*, \sigma_3^*) = (< g_1^{w_1}, \ldots, g_1^{w_n}, g_1^{\alpha} (u_1^{ID^*} h_1)^{r_{ID}} (u_2^{m^*} h_2)^{r_m} \prod_{j=} \quad 1^n g_1^{-x_j w_j} > * g_3^{\overrightarrow{\rho'}} * g_2^{\overrightarrow{s}_{12}}, g_1^{r_{ID}} * g_3^{\rho_{n+2}'} * g_2^{s_{22}}, g_1^{r_m} * g_2^{s_{32}})$. The adversary hopes that the forged signature can pass the verification.

Furthermore, the challenger checks whether the forged signature satisfies the following equation:

$$e\Big(< (g_1^z g_2^v)^{x_1}, \ldots, (g_1^z g_2^v)^{x_n}, g_1^z g_2^v >, \overrightarrow{\sigma}^*_1 \Big) = T \cdot e\Big(g_1^{a_1 ID^* + b_1}, \sigma_2^*\Big) \cdot e\Big(g_1^{a_2 ID^* + b_2}, \sigma_3^*\Big). \tag{8}$$

Probability analysis: according to Lemma 4, if the adversary outputs forged signature of type II in game Game_q with an advantage ε, the adversary outputs forged signature of type I with an advantage ε.

If $T = e(g_1, g_1)^{\alpha s}$, the equation also holds. If $T \neq e(g_1, g_1)^{\alpha s}$, the equation does not hold. Whether the equation holds equals that whether $T = e(g_1, g_1)^{\alpha s}$. That is to say, in the case of private key leakage $L_{SK} = (n - 2\Lambda - 1)\lambda$, the adversary $\mathcal{A}$ can only output the forged signature of type I with negligible advantage in the game Game_q. □

Lemma 6. *Under Assumption 2 and in the case of private key leakage $L_{SK} = (n - 2\Lambda - 1)\lambda$, if the adversary (an algorithm $\mathcal{A}$) can only output the forged signature of type I with negligible advantage in the game Game_k, the adversary can only output the forged signature of type I with negligible advantage in the game Game_{k-1}.*

Proof. First, given the challenger $\mathcal{B}$ an instance $\underline{D}^2 = (N, G, G_T, e, g_1, g_3, g_1^{z} g_2^{v}, g_2^{u} g_3^{\varsigma})$, $\overrightarrow{Y} = (y_1, \ldots, y_{n+1}) \in G_{p_2}^{n+1}$, $y \in G_{p_2}$ and T ($T = g_1^{\omega} g_3^{\sigma}$ or $T = g_1^{\omega} g_2^{\kappa} g_3^{\sigma}$, where $\omega, \kappa, \sigma \in Z_N$), $\mathcal{B}$ plays Game_{k-1} or Game_k with $\mathcal{A}$.

The challenger publishes the system parameters $\mathrm{params} = \{N, g_1, g_3, h_1 = g_1^{b_1}, u_1 = \quad g_1^{a_1}, h_2 = g_1^{b_1}, u_2 = g_1^{a_2}, g_1^{x_1}, \ldots, g_1^{x_n}, e(g_1, g_1)^{\alpha}\}$ to the adversary, where $\alpha, a_1, a_2, b_1, \quad b_2 \in Z_N$, and $x_1, \ldots, x_n \in Z_N$ are randomly selected. For any private key extraction query or signature query of the adversary, the challenger can use the master key to calculate. It is similar to Lemma 4.

Finally, suppose that the adversary generates the forged signature $\sigma^* = (\overrightarrow{\sigma}^*_1, \sigma_2^*, \sigma_3^*)$ of the user ID^* about the

Table 1: The comparisons between our scheme and that in [36].

Performance	IBS of [36]	Our CLR-IBS
Length of private key	$3 \times 3\lambda$	$3(n+3)\lambda$
Amount of leakage-resilience	0	$(n-2\Lambda-1)\lambda$

Table 2: The comparisons of leakage performance and security performance of several schemes and ours.

Schemes	Difficult hypothesis	Relative leakage rate	Leakage model	Security model
[29]-1	UOWHF	1	BLM	STD M
[29]-2	UOWHF	1/4	BLM	STD M
[29]-3	HCRHF	1/2	BLM	STD M
[31]	GBG	1/2	BLM	GGM
[19]	K-linear	$1/(k+1)$	CLM	STD M
[32]	GBG	Log (logP)	CLM	GGM
Ours	A1, A2, A3	1/3	CLM	STD M

message m^*. $\sigma^* = (\overrightarrow{\sigma}_1^*, \sigma_2^*, \sigma_3^*) = (< g_1^{w_1}, \ldots, g_1^{w_n}, g_1^{\alpha}(u_1^{ID^*}h_1)^{r_{ID}}(u_2^{m^*}h_2)^{r_m}\prod_{j=1}^{n} g_1^{-x_j w_j} > * g_3^{\overrightarrow{\rho'}} * g_2^{\overrightarrow{s}_{12}}, g_1^{r_{ID}} * g_3^{\rho'_{n+2}} * g_2^{s_{22}}, g_1^{r_m} * g_2^{s_{32}})$. The adversary hopes that the forged signature can pass the verification.

Probability analysis: according to Lemma 4, the adversary outputs forged signature of type II with an advantage ε. If $T \in G_{p_1p_3}$, $\mathcal{B}$ simulates the game Game_{k-1} properly. Thus, based on Assumption 2, the adversary outputs forged signature of type I with an advantage ε. $\mathcal{B}$ simulates the game Game_k properly if and only if $T \in G_{p_1p_2p_3}$.

That is to say, in the case of private key leakage $L_{SK} = (n-2\Lambda-1)\lambda$, if the adversary (an algorithm $\mathcal{A}$) can only output the forged signature of type I with a non-negligible advantage, we can judge that $T \in G_{p_1p_2p_3}$. □

7. Continuous Leakage Resilience

If the private key of a cryptographic scheme cannot be updated, the leakage will exceed a certain bound with the passage of time, which will break the security of the scheme. In order to keep continuous leakage resilience, the private key must be updated periodically. By the algorithm KeyUpd given in Section 5, our CLR-IBS scheme obtains continuous leakage resilience.

Theorem 2. *Our CLR-IBS scheme can resist continuous leakage attacks of the private key under the standard model.*

Proof. By the algorithm KeyUpd, we can obtain continuous leakage resilience which is similar to that of [25, 37]. The algorithm KeyUpd inputs the private key SK_{ID} and the system parameters params. Then, it outputs a new private key $\widehat{SK}_{ID}$. In the algorithm KeyUpd, some extra values are added to the private key. Because the extra values are randomly selected from Z_N, they have the same distribution with the old ones. Thus, $\widehat{SK}_{ID}$ has the same distribution as the old ones. If the private key is updated periodically, continuous leakage resilience will be obtained. □

8. Leakage Performance and Comparisons

In our scheme, p_1, p_2 and p_3 are all prime numbers of λ bits. The length of the private key is $3(n+3)\lambda$ bits. The amount of private key leakage is at most $(n-2\Lambda-1)\lambda$ bits. Here, $n \geq 2$ is a variable integer, which is used to obtain different leakage resilience. Λ is a normal number. The relative leakage rate of the private key is $(n-2\Lambda-1)\lambda/3(n+3)\lambda = (n-2\Lambda-1)/3(n+3)$.

n is a variable. If we want to get a high relative leakage rate, we can select a larger number n. Thus, the leakage-resilience performance of the system is better. However, the private key becomes longer accordingly. If n is a small number, the relative leakage rate is also small. But the private key is also relatively short.

Table 1 shows the comparisons between the scheme in this paper and that in [36]. Table 1 shows the size and leakage-resilience of the private key.

The paper [36] gives an IBS without leakage-resilience. We propose an IBS which can resist private key continuous leakage.

In specific applications, we can set different values according to the specific requirement of the leakage rate. The leakage rate of the private key depends on the value n. When the value is larger, the private key leakage rate of our scheme can reach 1/3.

References [28, 29, 31] give secure signature schemes against bounded leakage. These schemes can tolerate almost the whole private key leakage. Reference [28] uses Fiat Shamir transform to prove the security under the random oracle model, while reference [29] proves the security by constructing general primitives (i.e. one-time signature scheme and noninteractive zero knowledge proof) under the standard model.

References [18, 19, 32] construct continuous leakage-resilient signature schemes, respectively. In these schemes, the relative leakage rate of the private key is very small. Furthermore, reference [32] gives a continuous leakage-resilient signature scheme that is based on the bilinear group model. It is generally considered that the bilinear group model is weaker than the standard model.

Table 2 shows the comparisons of leakage performance and security performance of several LR signature schemes and ours, in which K-Linear represents K-Linear assumption, UOWHF represents universal one-way hash functions, HCRHF represents a homomorphic collision-resistant hash function family, A1, A2 and A3 stand the three static assumptions of our scheme, GBG represents the generic bilinear group model, GGM represents generic group model, STD M represents the standard model.

9. Conclusion

In this paper, the formal definition and security model of CLR-IBS are given. Furthermore, we propose a CLR-IBS scheme. The scheme can resist continuous leakage of the private key. The security of the scheme is proved by three static assumptions under the composite order bilinear group. The presented scheme has good leakage resilience. The private key leakage rate of the scheme can reach 1/3. Because ring signature also has good properties and application value, it is our next research direction to construct identity-based ring signature scheme with leakage resilience.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

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 (U1736112, 61772009, 61672207, and 6217071832), Jiangsu Provincial Natural Science Foundation of China (BK20161511), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (17KJB520042 and 20KJB413003), Suqian Sci & Tech Program (S201820 and Z2019109), the Cloud Computing and Big Data Security Research Team of Suqian University, and Qing Lan Project.

References

[1] C. Chen, T. Wang, and J. Tian, "Improving timing attack on RSA-CRT via error detection and correction strategy," *Information Sciences*, vol. 232, pp. 464–474, 2013.

[2] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: the case of AES," in *Proceedings of the RSA Conference*, San Jose, CA, USA, February 2005.

[3] O. Acıiçmez, W. Schindler, and Ç. K. Koç, "Cache based remote timing attack on the AES," in *Proceedings of the RSA Conference*, pp. 271–286, San Fancisco, CA, USA, February, 2007.

[4] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, and J. Horn, "Meltdown: reading kernel memory from user space," in *Proceedings of the 27th USENIX Security Symposium*, 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," *Communications of the ACM*, vol. 52, no. 5, pp. 91–98, 2009.

[7] K. Paul, H. Jann, F. Anders et al., "Spectre attacks: exploiting speculative execution," in *Proceedings of the 40th IEEE Symposium on Security and Privacy*, pp. 1–19, San Francisco, CA, USA, May, 2019.

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

[9] S. Dziembowski and K. Pietrzak, "Leakage-resilient cryptography," in *Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pp. 293–302, Philadelphia, PA, USA, October 2008.

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

[11] 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)*, pp. 474–495, San Francisco, CA, USA, March 2009.

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

[13] Y. Chen, Z. Zhang, D. Lin, and Z. Cao, "Generalized (identity-based) hash proof system and its applications," *Security and Communication Networks*, vol. 9, no. 12, pp. 1698–1716, 2016.

[14] C. Hazay, A. López-Alt, H. Wee, and D. Wichs, "Leakage-resilient cryptography from minimal assumptions," *Journal of Cryptology*, vol. 29, no. 3, pp. 514–551, 2016.

[15] Q.-Q. Lai, B. Yang, Y. Yu, Z. Xia, Y.-W. Zhou, and Y. Chen, "Updatable identity-based hash proof system based on lattices and its application to leakage-resilient public-key encryption schemes," *Journal of Computer Science and Technology*, vol. 33, no. 6, pp. 1243–1260, 2018.

[16] S. Liu, J. Weng, and Y. Zhao, "Efficient public key cryptosystem resilient to key leakage chosen ciphertext attacks," in *Proceedings of the RSA Conference*, pp. 84–100, San Francisco, CA, USA, February, 2013.

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

[18] Z. Brakerski, Y. T. Kalai, J. Katz, and V. Vaikuntanathan, "Overcoming the hole in the bucket: public-key cryptography resilient to continual memory leakage," in *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 501–510, Las Vegas, NV, USA, October 2010.

[19] Y. Dodis, K. Haralambiev, A. Lopez-Alt, and D. Wichs, "Cryptography against continuous memory attacks," in *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 511–520, Las Vegas, NV, USA, October 2010.

[20] H. Xiong, C. Zhang, T. H. Yuen, E. P. Zhang, S. M. Yiu, and S. Qing, "Continual leakage-resilient dynamic secret sharing in the split-state model," in *Proceedings of the 14th International Conference on Information and Communications Security*, pp. 119–130, Hong Kong, China, October 2012.

[21] S. Agrawal, Y. Dodis, V. Vaikuntanathan, and D. Wichs, "On continual leakage of discrete log representations," in *Advances in Cryptology - ASIACRYPT 2013*Springer, Berlin, 2013.

[22] B. Waters, "Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions," in *Proceedings of the 29th Annual International Cryptology Conference*, pp. 619–636, Santa Barbara, CA, USA, August 2009.

[23] A. Lewko, Y. Rouselakis, and B. Waters, "Achieving leakage resilience through dual system encryption," in *Proceedings of the Theory of Cryptography Conference*, pp. 70–88, Providence, RI, USA, March 2011.

[24] J. Li, Q. Yu, and Y. Zhang, "Identity-based broadcast encryption with continuous leakage resilience," *Information Sciences*, vol. 429, pp. 177–193, 2018.

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

[26] Y. Zhou, B. Yang, Y. Mu, T. Wang, and X. Wang, "Identity-based encryption resilient to continuous key leakage," *IET Information Security*, vol. 13, no. 5, pp. 426–434, 2019.

[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. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in *Proceedings of the 29th Annual International Cryptology Conference*, pp. 36–54, Santa Barbara, CA, USA, August 2009.

[29] J. Katz and V. Vaikuntanathan, "Signature schemes with bounded leakage resilience," *Advances in Cryptology - ASIACRYPT 2009*, in *Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 703–720, Tokyo, Japan, December 2009.

[30] T. Malkin, I. Teranishi, Y. Vahlis, and M. Yung, "Signatures resilient to continual leakage on memory and computation," in *Proceedings of the 8th Theory of Cryptography Conference*, pp. 89–106, Providence, RI, USA, March 2011.

[31] D. Galindo and S. Vivek, "A practical leakage-resilient signature scheme in the generic group model," in *Proceedings of the 19th International Conference (Selected Areas in Cryptography)*, pp. 50–65, Windsor, ON, Canada, August 2012.

[32] J.-D. Wu, Y.-M. Tseng, S.-S. Huang, and T.-T. Tsai, "Leakage-resilient revocable identity-based signature with cloud revocation authority," *Informatica*, vol. 31, no. 3, pp. 597–620, 2020.

[33] C. Hu, H. Li, Y. Huo, T. Xiang, and X. Liao, "Secure and efficient data communication protocol for wireless body area networks," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 94–107, 2017.

[34] M. Gerbush, A. Lewko, and B. Waters, "Dual form signatures: an approach for proving security from static assumptions," in *Proceedings of the 18th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 25–42, Beijing, China, December 2012.

[35] D. Boneh, E. J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proceedings of the Second Theory of Cryptography Conference*, pp. 325–341, Cambridge, MA, USA, February 2005.

[36] D. Tian and Z. He, "Fully secure ID-based signature scheme in the standard model," *Journal of Information Engineering University*, vol. 20, no. 6, pp. 702–706, 2019.

[37] M. Zhang, W. Shi, C. Wang, Z. Chen, and Y. Mu, "Leakage-resilient attribute-based encryptions with fast decryption: model, analysis and construction," in *Information Security Practice and Experience. ISPEC 2013*Springer, Berlin, 2013.