Hindawi
Security and Communication Networks
Volume 2020, Article ID 1492681, 11 pages
https://doi.org/10.1155/2020/1492681

Research Article

Certificate-Based Encryption Resilient to Continual Leakage in the Standard Model

Yuyan Guo,[1] Jiguo Li,[2] Mingming Jiang,[1] Lei Yu,[1] and Shimin Wei[1]

[1]*School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, Anhui, China*
[2]*College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China*

Correspondence should be addressed to Jiguo Li; ljg1688@163.com

Received 2 February 2020; Revised 31 March 2020; Accepted 3 June 2020; Published 28 June 2020

Academic Editor: Prosanta Gope

The security for many certificate-based encryption schemes was considered under the ideal condition, where the attackers rarely have the secret state for the solutions. However, with a side-channel attack, attackers can obtain partial secret values of the schemes. In order to make the scheme more practical, the security model for the certificate-based encryption which is resilient to continual leakage is first formalized. The attackers in the security model are permitted to get some secret information continuously through the side-channel attack. Based on the certificate-based key encapsulation scheme, a novel certificate-based encryption scheme is proposed, which is resilient to the continual leakage. In the standard model, the new scheme we propose is proved to be secure under the decisional truncated q-augmented bilinear Diffie–Hellman exponent hard problem and the decisional 1-bilinear Diffie–Hellman inversion hard problem. Additionally, the new scheme can resist the chosen-ciphertext attack. Moreover, a comparison is performed with other related schemes, where the proposed solution further considers the continual leakage-resilient property and exhibits less computation cost.

1. Introduction

The certificate-based cryptography (CBC) is a novel public key cryptosystem (PKC) which is proposed by Gentry [1]. CBC combines the traditional PKC and the identity-based cryptosystem to overcome the key escrow and key distribution issues existing in the identity-based cryptosystem, such that the management complexity of the public key certificate can be reduced for the conventional public key infrastructure. In CBC, a public-private key pair will be first generated for every client and applied for a certificate to the trusted certificate authority (CA). Different from the traditional PKC, the CBC provides a hidden certificate mechanism. The certificate of CBC has the function of the traditional public key certificate, and hence it can also be regarded as a part of the secret key for the users [1]. Any user needs to combine his own secret key and certificate to perform decryption or signature operation, and the sender of the message or the signature verifier does not need to pay attention to the certificate status of the communicating party. The implicit certificate mechanism in the CBC eliminates third party inquiries; therefore, CBC offers an efficient method for constructing an efficient and secure public key infrastructure. Due to its good nature, the CBC has been intensively focused on in recent years, and a series of certificate-based encryption (CBE) schemes [2–10] have been proposed. Many certificate-based signature (CBS) proposals [11–14] have also been constructed.

Typically, cryptography is considered to be secure ideally, in which the adversaries do not steal the secret values for the cryptographic system. However, the adversaries are able to access partial secret key by side-channel attack. Therefore, a number of approaches are proposed to model the leakage for such side-channel attacks. Micali and Reyzin [15] constructed the "only computation leaks information" model in 2004. Although this model examines a large type of leak attacks, the disadvantage is that it does not consider the case where the information is leaked from the inactive memory parts, e.g., the cold boot attack [16]. To capture more leaks, Halderman et al. [16] proposed a model named "relative

leakage." However, the major disadvantages are obvious; i.e., the secret key does not have sufficient length, and the allowed leakage number is limited. Akavia et al. [17] proposed a "bounded retrieval" model to make the size of the secret key more flexible without increasing the size of the public key and encryption and decryption time. This model is verified to be more powerful than the one with "only computation leaks information." For the "bounded retrieval" model, the leakage from inactive parts of memory is also taken into account. To further relax the limitations of the secret key-leakage constraint, Dodis et al. [18] and Yang et al. [19] considered the "auxiliary input" model and more kinds of one-way leakage functions. However, the above-mentioned three models do not involve continual leakage attacks. The "continual leakage" model [20–22] was designed to examine attacks where bounded information of the secret internal state is available at the attacker when the cryptographic primitive is invoked.

Researchers have been dedicated to finding a provably secure cryptographic solution to deal with the leakage attack problem, with various proposals. In addition, the "continual leakage" model was applied in many encryption schemes, for example, attribute-based encryption (ABE), public key encryption, and identity-based encryption (IBE). A public key encryption approach was made by Agrawal et al. in [23] aiming to cope with the continual leakage. Yuen et al. [24] proposed an IBE system with the aim of being resilient to continual auxiliary input leakage. Zhou et al. [25] constructed an IBE method with tight security which is resilient to the continuous leakage attacks in the standard model. Then, three continuous leakage-resilient IBE methods [26–28] have been put forward. Leakage amplification was proposed in [26] which constructs continuous leakage-resilient secure IBE scheme, which is considered an arbitrary length of the leakage parameter. The authors in [27] offered a new updatable identity-based hash proof system which is adopted to construct the continuous leakage-resilience identity-based cryptosystem. Zhou et al. [28] designed an improved continuous leakage-resilient IBE scheme with arbitrary length of the parameter leakage. Furthermore, Zhou et al. [29] presented an IBE scheme with leakage-amplified chosen-ciphertext attacks security. Li et al. [30–34] extended IBE to present some attribute-based encryption scheme, which can achieve fine-grained access control in cloud storage and can be applied in social network [35]. However, the above attribute-based encryption schemes did not consider key-leakage problem. In order to solve this problem, Zhang [36] delivered a concrete construction for resilient-leakage ciphertext-policy attribute-based encryption (CP-ABE) and provided a key update procedure to support continual leakage tolerance. Zhang et al. [37] proposed a new notion and construction for attribute-based hash proof system (AB-HPS) in the bounded key-leakage model. They also provided the general leakage-resilient attribute-based encryption construction using the AB-HPS as the primitive without indistinguishable obfuscator. Furthermore, Zhang et al. [38] designed the concrete ABE constructions in the bilinear groups with prime order and the security has been shown in the continual memory leakage model. A key-policy attribute-based encryption was defined and modeled by Li et al. [39], which is resilient to the problem of continual auxiliary input leakage. The proposed approach is also shown to have high security under the static assumptions. Li et al. [40] proposed an efficient extended file hierarchy attribute-based encryption scheme, which is very practical and greatly saves storage space and computation cost for those large institutions or companies. Moreover, Li et al. [41] presented a continuous leakage-resilient hierarchical attribute-based encryption scheme, which is shown to be resilient to the master and secret keys leakage. Zhou and Yang [42] presented a continual leakage-resilient certificateless public key encryption scheme which not only tolerates continual leakage attacks, but also achieves better performances. Li et al. [43] provided a continuous leakage-resilient CBE scheme which is proved secure against adaptive chosen-ciphertext attack in the random oracle model. Authenticated key exchange protocol is used to establish a secure communication channel over a public network. However, it has been demonstrated that some standardized AKE protocols suffer from side-channel and key-leakage attacks. In order to defend against these attacks, Chen et al. [44–46] and Yang et al. [47] presented several leakage-resilient authenticated key exchange protocols.

1.1. Motivations and Contributions. Currently, there are few researches for the certificate-based encryption resilient to continual leakage. Actually, some previous CBE schemes which have been constructed in the ideal setting may be insecure under the continuous leakage attacks. The main reason is that the adversary can recover the complete secret key via continuously accessing the partial information of the secret key. Therefore, it is meaningful for us to construct a CBE scheme to resist the continual leakage attack.

The primary objective of our work is to establish a secure certificate-based encryption scheme which is resilient to continual leakage. Referring to [3–8, 21–25], we design the outline and the security model of CBE resilient to continual leakage. On the basis of the certificate-based key encapsulation method, a CBE scheme is proposed which is shown to be secure in the standard model and can resist the continual leakage attack. The encapsulated symmetric key is randomized using the strong extractor. Furthermore, the encapsulated symmetric key allowing leakage is employed to encrypt the message.

The CBE schemes created in [6, 8] only tolerate the leakage. Further, our CBE scheme added the secret key update algorithm to obtain the continuous leakage-resilience. Our approach can resist a larger leakage by performing the secret key updating algorithm, where the keys are periodically updated and the leakage will not be allowed during updates, but only between the updates. We further consider the leakage limit of the encapsulated symmetric key, and the leakage ratio of our scheme is approximately equal to 1.

We provide a proof to show that our CBE scheme is secure against chosen-ciphertext attack under the hardness of the decisional truncated q-augmented bilinear

Diffie–Hellman exponent (q-ABDHE) problem and the decisional 1-bilinear Diffie–Hellman inversion (1-BDHI) problem.

Compared with the existing CBE schemes, our proposed scheme enhances the continual leakage-resilient property and has a lower communication cost. Therefore, our CBE scheme has obvious advantage. We implement the proposed CBE scheme and the relevant schemes using C++ programming language with the PBC library, and the simulation results show that our scheme has better performance.

1.2. Paper Organization. The required preliminary knowledge is presented in Section 2. In Section 3, we demonstrate the outline and the security model of CBE resilient to continual leakage. In Section 4, a CBE scheme resilient to continual leakage is proposed. Section 5 provides the proof of our CBE scheme. Then, the comparison in terms of the efficiency is shown in Section 6. Finally, we conclude this work in Section 7.

2. Preliminaries

Definition 1. Let $\mathbb{G}$ and $\mathbb{G}_T$ denote multiplicative cyclic groups of the prime order p, respectively. A generator of $\mathbb{G}$ is represented by g. A bilinear map e if $e: \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$ has the following properties, as

(i) Bilinear: $e(P^a, Q^b) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$

(ii) Nondegenerate: $e(g, g) \neq 1 \in \mathbb{G}_T$

(iii) Computable: the map e is efficiently computable

The security of our CBE scheme is resilient to continual leakage depending on the following problems.

Definition 2. The decisional truncated q-ABDHE problem is described as follows: given $\mathscr{D} = (g, g^{\alpha}, \ldots, g^{\alpha^q}, g', g'^{\alpha^{q+2}}) \in \mathbb{G}^{q+3}$, where $\alpha, q \in \mathbb{Z}_p^*$ and $T \in \mathbb{G}_T$, output 1 if $T = e(g, g')^{\alpha^{q+1}}$ and 0 otherwise.

The advantage of a probabilistic polynomial time (PPT) adversary A deciding whether $T = e(g, g')^{\alpha^{q+1}}$ is given as $\mathrm{Adv}_A^{q-\mathrm{ABDHE}} = |\Pr[A(\mathscr{D}, e(g, g')^{\alpha^{q+1}}) = 1] - \Pr[A(\mathscr{D}, T) = 1]|$.

It is said that the decisional truncated q-ABDHE problem is hard if $\mathrm{Adv}_A^{q-\mathrm{ABDHE}}$ is arbitrarily small for all PPT adversaries A.

Definition 3. We define the decisional 1 – BDHI problem as follows: given $\mathscr{D} = (g, g^{\alpha}) \in \mathbb{G}^2$, where $\alpha \in \mathbb{Z}_p^*$ and $T \in \mathbb{G}_T$, output 1 if $T = e(g, g)^{1/\alpha}$ and 0 otherwise.

The advantage that A decides whether $T = e(g, g)^{1/\alpha}$ is given as $\mathrm{Adv}_A^{1-\mathrm{BDHI}} = |\Pr[A(\mathscr{D}, e(g, g)^{1/\alpha}) = 1] - \Pr[A(\mathscr{D}, T)] = 1|$.

We say that the decisional 1 – BDHI problem is hard if $\mathrm{Adv}_A^{1-\mathrm{BDHI}}$ is ignorable for all PPT adversaries A.

Definition 4. The min-entropy of a random variable (RV) X is $H_{\infty}(X) = -\log(\max_x \Pr[X = x])$.

Definition 5. For RV's X and Y, the averaged conditional min-entropy is represented by $\tilde{H}_{\infty}(X \mid Y) = -\log(E_{y \leftarrow\!\!- Y}[\max \Pr[X = x \mid Y = y]]) = -\log(E_{y \leftarrow\!\!- Y}[2^{-H_{\infty}(X \mid Y = y)}])$ with $E_{y \leftarrow\!\!- Y}^{x}$ denoting the expectation of Y.

Lemma 1. *If X, Y, and Z are random variables such that Y contains 2^l ($l \in \mathbb{N}$) potential elements, it has that $H_{\infty}(X \mid (Y, Z)) \geq H_{\infty}(X \mid Z) - l$ [48].*

Definition 6. The statistical distance between random variables X and Y is given by $\mathrm{SD}(X, Y) = 1/2 \sum_x |\Pr[X = x] - \Pr[Y = x]|$, with $x \in F$, where F denotes a finite field.

Definition 7. A random function $\mathrm{Ext}: \mathbb{G} \times \{0,1\}^{\mu} \longrightarrow \{0,1\}^{\eta}$ is regarded as an average-case (m, ε)-strong extractor if $\mu, \eta \in \mathbb{N}$, $X \in \mathbb{G}$, $m \in \mathbb{N}$ and $\tilde{H}_{\infty}(X \mid Y) \geq m$ for all X, Y, we obtain $\mathrm{SD}((\mathrm{Ext}(X, U_{\mu}), U_{\mu}, Y), (U_{\eta}, U_{\mu}, Y)) \leq \varepsilon$, with two variables U_{μ} and U_{η} having uniform distributions over $\{0,1\}^{\mu}, \{0,1\}^{\eta}$ respectively, and ε being negligible.

3. The Outline and Security Model of CBE Resilient to Continual Leakage

3.1. The Outline of CBE Resilient to Continual Leakage. The definition of CBE resilient to continual leakage which is referred in references [3–8] includes a group of algorithms, i.e., Setup, UserKeyGen, CertGen, SymmetricKeyGen, Encrypt, Decrypt, and UpdateSK, which are described as follows:

Setup: for a security parameter 1^n ($n \in \mathbb{N}$), the setup process generates a collection of public parameters params and a corresponding master secret key MSK

UserKeyGen: for the input identity ID, a secret key sk_{ID} and a public key PK_{ID} are produced by the algorithm

CertGen: for the inputs params, an identity ID, MSK, and PK_{ID}, the algorithm generates a certificate Cert_{ID}, which is transmitted to the user

SymmetricKeyGen: taking params, ID, PK_{ID} as its input, the algorithm generates the secret symmetric key K and the intermediate state π

Encrypt: for the inputs params, ID, PK_{ID}, K, π, and the message M, the algorithm returns a ciphertext $C = (\varphi, C_M)$ on the message M, where φ is the encapsulation of K and C_M is the ciphertext of the message M which is encrypted with K

Decrypt: for the inputs params,Cert_{ID}, sk_{ID}, and C, the algorithm generates K and returns either M via K or $\perp$ if C is an invalid ciphertext

UpdateSK: for the inputs sk_{ID} and params, the algorithm produces an updated secret key sk'_{ID} where $|sk'_{ID}| = |sk_{ID}|$

3.2. Security Model for CBE Resilient to Continual Leakage. Inspired by the schemes in references [3–8, 21–25], we propose a security model for the CBE which is resilient to the continual leakage. The model is described through Game-1

and Game-2. We evaluate the security based on these two games resilient to continual leakage and the adaptive chosen-ciphertext attacks (IND-RCL-CCA). In Game-1, an adversary A_1 which simulates the uncertified client is able to substitute the public key and obtain the secret key of any client, but A_1 does not access the MSK. In Game-2, another adversary A_2 which plays an honest-but-curious certifier owns the master key. Such adversary is able to obtain the certificate of any client, but cannot substitute the public keys of any user. The challenger $\mathscr{C}$ interacts with A_1 and A_2 by the following games.

3.2.1. IND-RCL-CCA Game-1

Setup: the challenger $\mathscr{C}$ performs the Setup algorithm, keeps the master secret key MSK, and returns params to A_1

Phase 1: A_1 creates the queries adaptively as follows:

Public key queries: $\mathscr{C}$ holds a list $\mathscr{L}_1 = \{(ID, sk_{ID}, PK_{ID}, \gamma)\}$ to record both the secret and public keys, where $\gamma \in \{0,1\}$, $\gamma = 0$ denotes that PK_{ID} has not been substituted, and $\gamma = 1$ denotes that PK_{ID} has been replaced. $\mathscr{L}_1$ is initially empty. A_1 generates the query for ID, and $\mathscr{C}$ seeks $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$. If $(ID, sk_{ID}, PK_{ID}, \gamma)$ exists, $\mathscr{C}$ returns PK_{ID}. Otherwise, UserKeyGen will be used to produce (sk_{ID}, PK_{ID}), $(ID, sk_{ID}, PK_{ID}, 0)$ is inserted into $\mathscr{L}_1$ and PK_{ID} is returned. For simplicity, for any ID, it is stipulated that A_1 must first make the public key query before making any other queries as follows.

Public key replacing queries: A_1 produces the replace query for (ID, PK'_{ID}). $\mathscr{C}$ looks for $(ID, \bullet, PK_{ID}, \cdot)$ from the list $\mathscr{L}_1$. If it is not found, $\mathscr{C}$ will insert $(ID, \bullet, PK'_{ID}, 1)$ into $\mathscr{L}_1$. Otherwise, $\mathscr{C}$ updates the item $(ID, \bullet, PK_{ID}, \bullet)$ to $(ID, \bullet, PK'_{ID}, 1)$.

Secret key queries: A_1 inputs ID; $\mathscr{C}$ checks $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$. If $\gamma = 0$, $\mathscr{C}$ returns sk_{ID} to A_1. Otherwise $\mathscr{C}$ outputs $\perp$ to A_1.

Certificate queries: A_1 makes the certificate query of ID and gets $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$, and then $\mathscr{C}$ runs algorithm CertGen and outputs Cert_{ID} to A_1.

Leakage queries: $\mathscr{C}$ generates a list $\mathscr{L}_2$ in which the form of the item is (ID, K, cnt), where $\text{cnt} \in \mathbb{N}$ and K is utilized for encrypting the message as the symmetric key. $\mathscr{L}_2$ is initially empty. $\mathscr{C}$ finds (ID, K, cnt) in the list $\mathscr{L}_2$. If the item does not exist, $\mathscr{C}$ will add $(ID, K, 0)$ into $\mathscr{L}_2$. If (ID, K, cnt) is found or after this step, $\mathscr{C}$ will check the condition $\text{cnt} + l_i \le l$, where $l, i \in \mathbb{N}$. If not, $\mathscr{C}$ returns $\perp$. Otherwise, $\mathscr{C}$ selects a leakage function $f_i : \mathbb{G}_T \longrightarrow \{0,1\}^{l_i}$, sets $\text{cnt} \longleftarrow \text{cnt} + l_i$ for (ID, K, cnt), and outputs $f_i(K)$ to A_1.

Decryption queries: For queries on ID and the ciphertext C, $\mathscr{C}$ obtains $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$; if $\gamma = 1$, A_1 has to provide the corresponding secret key; otherwise, $\mathscr{C}$ gets sk_{ID} from $\mathscr{L}_1$. $\mathscr{C}$ makes the certificate queries to get Cert_{ID} and applies Decrypt algorithm to obtain the symmetric key K and uses K to decrypt C. The challenger $\mathscr{C}$ returns either M or $\perp$ to A_1.

Challenge: A_1 gives two messages M_0, M_1 of equal length and a target identity ID^* to $\mathscr{C}$ with the following restriction: A_1 is prohibited from issuing the certificate query for ID^* and does not replace the public key of ID^*. $\mathscr{C}$ executes the SymmetricKeyGen algorithm to get a symmetric key K_1^* and randomly chooses $K_0^* \in \mathbb{G}_T$. $\mathscr{C}$ chooses $\beta \in \{0,1\}$ and $\sigma^* \in \{0,1\}^\mu$ uniformly at random, runs the Encrypt algorithm to encrypt M_β for ID^*, and yields the encapsulation φ^* of K_β^* and the challenge ciphertext $C^* = (\varphi^*, C^*_{M_\beta}, \sigma^*)$ to A_1, where $C^*_{M_\beta} = \text{Ext}(K_\beta^*, \sigma^*) \oplus M_\beta$.

Phase 2: A_1 continually makes the queries similar to Phase 1 with the following constraints: A_1 is prohibited from making the certificate queries for ID^*, as well as the decryption queries on (ID^*, C^*).

Guess: A_1 returns a bit $\beta\prime \in \{0,1\}$. We say that A_1 wins the game if $\beta' = \beta$.

The advantage of A_1 winning the IND-RCL-CCA Game-1 is described as $\text{Adv}_{A_1}^{\text{IND-RCL-CCA}} = |2\Pr[A_1 \text{ wins}] - 1|$.

3.2.2. IND-RCL-CCA Game-2

Setup: the challenger $\mathscr{C}$ performs the Setup algorithm and returns the master secret key MSK and params to A_2.

Phase 1: A_2 adaptively inquires $\mathscr{C}$ for the following queries.

Public key queries: $\mathscr{C}$ holds a list $\mathscr{L}_1 = \{(ID, sk_{ID}, PK_{ID})\}$ to record the secret keys and the public keys. $\mathscr{L}_1$ is empty in the initial step of the game. For the queries about ID, $\mathscr{C}$ finds a tuple (ID, sk_{ID}, PK_{ID}) from $\mathscr{L}_1$. If it exists, $\mathscr{C}$ returns PK_{ID} to A_2. Otherwise, $\mathscr{C}$ uses UserKeyGen to produce sk_{ID} and PK_{ID}, inserts (ID, sk_{ID}, PK_{ID}) into $\mathscr{L}_1$, and returns PK_{ID} to A_2. For simplicity, for any ID, it is stipulated that A_2 must first make the public key query before making any other queries as follows.

Secret key queries: For a secret key query under ID, $\mathscr{C}$ seeks (ID, sk_{ID}, PK_{ID}) from the list $\mathscr{L}_1$ and returns sk_{ID} to A_2.

Leakage queries: $\mathscr{C}$ generates a list $\mathscr{L}_2$ in which the form of the item is (ID, K, cnt), where $\text{cnt} \in \mathbb{N}$ and K represents the symmetric key which is adopted to encrypt the message. $\mathscr{L}_2$ is initially empty. $\mathscr{C}$ finds (ID, K, cnt) from the list $\mathscr{L}_2$. If the item does not exist, $\mathscr{C}$ inserts an item $(ID, K, 0)$ into the list $\mathscr{L}_2$. Following this step or if the item exists, $\mathscr{C}$ decides whether $\text{cnt} + l_i \le l$ where $l, i \in \mathbb{N}$. If not, $\mathscr{C}$ returns $\perp$. Otherwise, $\mathscr{C}$ selects a leakage function $f_i : \mathbb{G}_T \longrightarrow \{0,1\}^{l_i}$, sets $\text{cnt} \longleftarrow \text{cnt} + l_i$ for (ID, K, cnt), and outputs $f_i(K)$ to A_2.

Decryption queries: for queries on ID and ciphertext C, $\mathscr{C}$ obtains $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$. $\mathscr{C}$ conducts the CertGen algorithm to obtain Cert_{ID}, then

performs Decrypt algorithm to obtain the symmetric key K, and adopts K to decrypt C. The challenger $\mathscr{C}$ returns either M or $\perp$ to A_2.

Challenge: A_2 gives two messages M_0 and M_1 with an equal length and a target identity ID^* to $\mathscr{C}$ with the following restrictions: A_2 is not allowed to issue the secret key query for ID^*. $\mathscr{C}$ runs the SymmetricKeyGen algorithm to get a symmetric key K_1^* and randomly chooses $K_0^* \in \mathbb{G}_T$. $\mathscr{C}$ randomly selects $\beta \in \{0,1\}$ and $\sigma^* \in \{0,1\}^\mu$ from the uniform distribution, runs the Encrypt algorithm to encrypt M_β for ID^*, and produces the encapsulation φ^* of K_β^* and the challenge ciphertext $C^* = (\varphi^*, C_{M_\beta}^*, \sigma^*)$ to A_2, where $C_{M_\beta}^* = \mathrm{Ext}(K_\beta^*, \sigma^*) \oplus M_\beta$.

Phase 2: similar to Phase 1, the queries will be continuously made by A_2 under the following constraints: A_2 is prohibited from making the secret key queries for ID^* and the decryption queries on (ID^*, C^*).

Guess: A_2 returns a bit $\beta\prime \in \{0,1\}$. We say that A_2 wins the game if $\beta\prime = \beta$.

The advantage of A_2 winning the IND-RCL-CCA Game-2 is defined to be $\mathrm{Adv}_{A_2}^{\mathrm{IND-RCL-CCA}} = |2\Pr[A_2 \text{ wins}] - 1|$.

Definition 8. A CBE scheme resilient to continual leakage is regarded to be secure under the adaptive chosen-ciphertext attacks, if no PPT adversary has non-negligible advantage in the IND-RCL-CCA Game-1 and IND-RCL-CCA Game-2.

4. Our CBE Scheme Resilient to Continual Leakage

Inspired by the schemes in [3–8, 22, 49], a strong extractor technology is proposed in [48] with a CBE scheme which is resilient to the continual leakage. Seven related algorithms are introduced as follows:

Setup: Define two groups $\mathbb{G}$ and $\mathbb{G}_T$ with prime order p. A bilinear mapping is given by $e: \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$. Let $l = l(n)$ be a bound of all leakages. In this procedure, an average scenario is selected with $(\log|\mathbb{G}_T| - l, \varepsilon_{\mathrm{Ext}})$-strong extractor $\mathrm{Ext}: \mathbb{G}_T \times \{0,1\}^\mu \longrightarrow \{0,1\}^\eta$ where $\mu, \eta \in \mathbb{N}$, and two collision resistant hash functions $H_1: \{0,1\}^* \times \mathbb{G}^3 \longrightarrow \mathbb{Z}_p^*$ and $H_2: \mathbb{G} \times \mathbb{G}_T \longrightarrow \mathbb{Z}_p^*$ are chosen. The message space is $\mathscr{M} = \{0,1\}^\eta$. The algorithm uses random value $\alpha \in \mathbb{Z}_p^*$, $h_1, h_2 \in \mathbb{G}$, and computes $g_1 = g^\alpha$ and $g_T = e(g,g)$, with g being a generator of $\mathbb{G}$. It outputs the master secret key MSK = α and a tuple of public parameter params = $(p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_T, h_1, h_2, e, H_1, H_2, \mathrm{Ext})$.

UserKeyGen: given params, the algorithm picks random numbers $s_1, s_2 \in \mathbb{Z}_p^*$, and sets the secret key $sk_{ID} = (s_1, s_2)$ for the user ID; then it computes the public key $PK_{ID} = (PK_{ID}^{(1)}, PK_{ID}^{(2)}, PK_{ID}^{(3)}) = (g_1^{s_1}, g^{s_1}, g^{s_2})$.

CertGen: given params, MSK, ID, and PK_{ID}, the CertGen algorithm computes $\rho = H_1(ID, PK_{ID})$, selects random numbers $x_1, x_2 \in \mathbb{Z}_p^*$, computes $d_1 = (h_1 g^{-x_1})^{1/(\alpha-\rho)}$ and $d_2 = (h_2 g^{-x_2})^{1/(\alpha-\rho)}$, and outputs $\mathrm{Cert}_{ID} = (\mathrm{Cert}_{ID}^{(1)}, \mathrm{Cert}_{ID}^{(2)}, \mathrm{Cert}_{ID}^{(3)}, \mathrm{Cert}_{ID}^{(4)}) = (x_1, d_1, x_2, d_2)$.

SymmetricKeyGen: Given params, ID, PK_{ID}, the SymmetricKeyGen algorithm computes $\rho = H_1(ID, PK_{ID})$, selects random number $r \in \mathbb{Z}_p^*$, and computes $C_1 = (PK_{ID}^{(1)} \cdot (PK_{ID}^{(2)})^{-\rho})^r$ and $C_2 = g_T^r$. Then, it outputs the secret symmetric key $K = (e(g, h_1 \cdot PK_{ID}^{(2)})^\varphi \cdot e(g, h_2 \cdot PK_{ID}^{(3)}))^r$ where $\varphi = H_2(C_1, C_2)$ and the intermediate state $\pi = (C_1, C_2)$.

Encrypt: Given params, ID, PK_{ID}, K, $\pi = (C_1, C_2)$, and the message M, the algorithm selects a random value $\sigma \in \{0,1\}^\mu$ and calculates $C_3 = \mathrm{Ext}(K, \sigma) \oplus M$. It sets $C_4 = \sigma$ and returns the ciphertext $C = (C_1, C_2, C_3, C_4)$.

Decrypt: Given params, Cert_{ID}, $sk_{ID} = (s_1, s_2)$, and $C = (C_1, C_2, C_3, C_4)$, the algorithm computes $\varphi = H_2(C_1, C_2)$, generates $K = e(C_1, \mathrm{Cert}_{ID}^{(2)\varphi} \cdot \mathrm{Cert}_{ID}^{(4)})^{1/s_1} \cdot C_2^{\varphi \cdot \mathrm{Cert}_{ID}^{(1)} + \mathrm{Cert}_{ID}^{(3)} + \varphi \cdot s_1 + s_2}$, and returns $M = C_3 \oplus \mathrm{Ext}(K, C_4)$.

UpdateSK: Given $sk_{ID} = (s_1, s_2)$, the secret key updating algorithm randomly selects $s_1' \in \mathbb{Z}_p^*$ and $s_2' \in \mathbb{Z}_p^*$. It then generates a new secret key $sk_{ID}' = (s_1 \cdot s_1', s_2 \cdot s_2')$.

Correctness of our scheme. $K = e(C_1, \mathrm{Cert}_{ID}^{(2)\varphi} \cdot \mathrm{Cert}_{ID}^{(4)})^{1/s_1} \cdot C_2^{\varphi \cdot \mathrm{Cert}_{ID}^{(1)} + \mathrm{Cert}_{ID}^{(3)} + \varphi \cdot s_1 + s_2} = e((g_1^{s_1} \cdot g^{-\rho s_1})^r, ((h_1 g^{-x_1})^{1/(\alpha-\rho)})^\varphi \cdot (h_2 g^{-x_2})^{1/(\alpha-\rho)})^{1/s_1} \cdot (e(g, g)^r)^{\varphi x_1 + x_2 + \varphi s_1 + s_2} = e(g, (h_1 g^{-x_1})^\varphi \cdot (h_2 g^{-x_2}))^r \cdot e(g, g)^{r(\varphi x_1 + x_2 + \varphi s_1 + s_2)} = (e(g, h_1 \cdot PK_{ID}^{(2)})^\varphi e(g, h_2 \cdot PK_{ID}^{(3)}))^r$

We have $C_3 \oplus \mathrm{Ext}(K, C_4) = \mathrm{Ext}(K, \sigma) \oplus M \oplus \mathrm{Ext}(K, \sigma) = M$.

5. Security Analysis

Our CBE approaches resilient to continual leakage are proved to be secure under the standard model as follows.

Theorem 1. *If there is a PPT adversary A_1 against the CBE scheme resilient to continual leakage with advantage ε that makes at most q_c certificate queries, q_d decryption queries in the case of l bits entropy leakage for the symmetric key, then there exists a PPT algorithm B against the q-ABDHE problem with an advantage ε, where $q = q_c + q_d + 1$.*

Proof. Given $(\mathbb{G}_T, \mathbb{G}, p, e, g, g^\alpha, \ldots, g^{\alpha^q}, g', g'^{\alpha^{q+2}}, T)$, the algorithm B can be regarded as the challenger $\mathscr{C}$ of IND-RCL-CCA Game-1 to interact with A_1; the target of B is to decide whether $T = e(g, g')^{\alpha^{q+1}}$.

Setup: The algorithm B sets $g_1 = g^\alpha$ and computes $g_T = e(g,g)$; B randomly chooses two q-degree unary polynomials $f_1(x), f_2(x) \in Z_p[x]$, and computes $h_1 = g^{f_1(\alpha)}$ and $h_2 = g^{f_2(\alpha)}$ (B can compute h_1, h_2 based $(g, g^\alpha, \ldots, g^{\alpha^q})$). Two collision resistant hash functions are chosen by B, *i.e.*, $H_1: \{0,1\}^* \times \mathbb{G}^3 \longrightarrow \mathbb{Z}_p^*$, $H_2: \mathbb{G} \times \mathbb{G}_T \longrightarrow \mathbb{Z}_p^*$ and an average-case $(\log|\mathbb{G}_T| - l, \varepsilon_{\mathrm{Ext}})$-strong extractor $\mathrm{Ext}: \mathbb{G}_T \times \{0,1\}^\mu \longrightarrow \{0,1\}^\eta$, $\mu, \eta \in \mathbb{N}$, and sends params = $(p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_T, h_1, h_2, e, H_1, H_2, \mathrm{Ext})$ to A_1.

Phase 1: A_1 makes the following queries adaptively:

Public key queries: A_1 inputs ID; B seeks $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$. If it exists, B returns PK_{ID}. Otherwise, B randomly picks $sk_{ID} = (s_1, s_2) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$, computes $PK_{ID} = (g_1^{s_1}, g^{s_1}, g^{s_2})$, inserts $(ID, sk_{ID}, PK_{ID}, 0)$ into $\mathscr{L}_1$, and returns PK_{ID}.

Public key replace queries: A_1 makes the query for (ID, PK_{ID}') where $PK_{ID}' = (PK_{ID}^{'(1)}, PK_{ID}^{'(2)}, PK_{ID}^{'(3)})$. B checks whether $e(g, PK_{ID}^{'(1)}) = e(PK_{ID}^{'(2)}, g_1)$. If it is not true, B outputs $\perp$, which denotes that this replace query is invalid. Otherwise, B seeks $(ID, \bullet, PK_{ID}, \bullet)$ from $\mathscr{L}_1$; if it is not found, B inserts $(ID, \bullet, PK_{ID}', 1)$ into $\mathscr{L}_1$; otherwise, B updates $(ID, \bullet, PK_{ID}, \bullet)$ to $(ID, \bullet, PK_{ID}', 1)$.

Secret key queries: A_1 inputs ID; B checks $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$. If $\gamma = 0$, B returns sk_{ID} to A_1. Otherwise, B outputs $\perp$ to A_1.

Certificate queries: A_1 inputs ID; B gets $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$ and defines two polynomials $F_1^{ID}(x) = f_1(x) - f_1(\rho)/x - \rho$ and $F_2^{ID}(x) = f_2(x) - f_2(\rho)/x - \rho$, where $\rho = H_1(ID, PK_{ID})$. B computes $\mathrm{Cert}_{ID} = (f_1(\rho), g^{F_1^{ID}(\alpha)}, f_2(\rho), g^{F_2^{ID}(\alpha)})$ and outputs Cert_{ID} to A_1 (obviously, B can compute $g^{F_1^{ID}(\alpha)}, g^{F_2^{ID}(\alpha)}$ based $(g, g^{\alpha}, \ldots, g^{\alpha^q})$. Due to $g^{F_1^{ID}(\alpha)} = (h_1 g^{-f_1(\rho)})^{1/(\alpha-\rho)}$ and $g^{F_2^{ID}(\alpha)} = (h_2 g^{-f_2(\rho)})^{1/(\alpha-\rho)}$, therefore Cert_{ID} is a valid certificate).

Leakage queries: A_1 inputs ID; B finds (ID, K, cnt) from $\mathscr{L}_2$. B adds $(ID, K, 0)$ into $\mathscr{L}_2$ if the item does not exist. If it exists, or in the next step, B decides whether $\mathrm{cnt} + l_i \le l$, where $l \in \mathbb{N}$ is the upper bound of the allowed leak. If not, B returns $\perp$. Otherwise, B selects a leakage function $f_i : \mathbb{G}_T \longrightarrow \{0,1\}^{l_i}$, sets $\mathrm{cnt} \longleftarrow \mathrm{cnt} + l_i$ for (ID, K, cnt), and outputs $f_i(K)$ to A_1.

Decryption queries: For queries on ID and the ciphertext $C = (C_1, C_2, C_3, C_4)$, B obtains $(ID, sk_{ID}, PK_{ID}, \gamma)$ from $\mathscr{L}_1$; if $\gamma = 1$, A_1 has to provide the corresponding secret key; otherwise, B gets the secret key $sk_{ID} = (s_1, s_2)$ from $\mathscr{L}_1$. B makes the certificate queries to gain the certificate $\mathrm{Cert}_{ID} = (\mathrm{Cert}_{ID}^{(1)}, \mathrm{Cert}_{ID}^{(2)}, \mathrm{Cert}_{ID}^{(3)}, \mathrm{Cert}_{ID}^{(4)})$; then he computes the symmetric key $K = e(C_1, \mathrm{Cert}_{ID}^{(2)\gamma} \cdot \mathrm{Cert}_{ID}^{(4)})^{1/s_1} \cdot C_2^{\varphi\gamma \cdot \mathrm{Cert}_{ID}^{(1)} + Cert_{ID}^{(3)} + \varphi \cdot s_1 + s_2}$ where $\varphi = H_2(C_1, C_2)$ and $M = C_3 \oplus \mathrm{Ext}(K, C_4)$. Finally, B returns either M or $\perp$ to A_1.

Challenge: A_1 provides B with two identical-length messages M_0 and M_1, and a target identity ID^* with the following restrictions: A_1 is prohibited from issuing certificate queries for ID^* and does not replace the public key of ID^*. B defines a $q+1$-degree polynomial $F^*(x) = x^{q+2} - \rho^{*\,q+2}/x - \rho^* = \sum_{i=0}^{q+1} F_i^* \cdot x^i$, where F_i^* is the i-term coefficient of $F^*(x)$. B computes $C_1^* = (g'^{\alpha^{q+2}} \cdot g'^{-\rho^{*\,q+2}})^{s_1^*}$, $C_2^* = T^{F_{q+1}^*} \cdot e(\prod_{i=0}^{q} (g^{\alpha^i})^{F_i^*}, g')$, where $\rho^* = H_1(ID^*, PK_{ID^*})$ and $sk_{ID^*} = (s_1^*, s_2^*)$. B then computes $K_1^* = e(C_1^*, \mathrm{Cert}_{ID^*}^{(2)\varphi^*} \cdot \mathrm{Cert}_{ID^*}^{(4)})^{1/s_1^*} \cdot (C_2^*)^{\varphi^* \cdot \mathrm{Cert}_{ID^*}^{(1)} + \mathrm{Cert}_{ID^*}^{(3)} + \varphi^* \cdot s_1^* + s_2^*}$ where $\varphi^* = H_2(C_1^*, C_2^*)$. B randomly selects $K_0^* \in \mathbb{G}_T$, $\beta \in \{0,1\}$ and $\sigma^* \in \{0,1\}^{\mu}$, sets $C_{M_\beta}^* = \mathrm{Ext}(K_\beta^*, \sigma^*) \oplus M_\beta$, $C_4^* = \sigma^*$, and produces the challenge ciphertext $C_\beta^* = (C_1^*, C_2^*, C_{M_\beta}^*, C_4^*)$ to A_1.

Phase 2: A_1 continues making the queries as in Phase 1 with the following restriction: A_1 has no permission to issue certificate queries for ID^*, as well as the decryption queries on (ID^*, C^*).

Guess: A_1 returns the guess $\beta\prime \in \{0,1\}$. If $\beta' = \beta$ holds, B will output 1, indicating that $T = e(g, g')^{\alpha^{q+1}}$. Otherwise, B outputs 0, indicating that $T \neq e(g, g')^{\alpha^{q+1}}$. □

5.1. Probability analysis. If $T = e(g, g')^{\alpha^{q+1}}$, we set $r^* = \log_g g' \cdot F^*(\alpha)$, and we have $C_1^* = (g'^{\alpha^{q+2}} \cdot g'^{-\rho^{*\,q+2}})^{s_1^*} = g^{s_1^* \cdot \log_g g' \cdot (\alpha^{q+2} - \rho^{*\,q+2})} = g^{s_1^* \cdot r^* \cdot (\alpha - \rho^*)} = (PK_{ID^*}^{(1)} \cdot (PK_{ID^*}^{(2)})^{-\rho^*})^{r^*}$, $C_2^* = T^{F_{q+1}^*} \cdot e(\prod_{i=0}^{q} (g^{\alpha^i})^{F_i^*}, g') = (e(g, g')^{\alpha^{q+1}})^{F_{q+1}^*} \cdot e(\prod_{i=0}^{q} (g^{\alpha^i})^{F_i^*}, g') = e(g, g)^{\log_g g' \cdot \sum_{i=0}^{q+1} \alpha^i F_i^*} = g_T^{r^*}$. Thus, $C_\beta^* = (C_1^*, C_2^*, C_{M_\beta}^*, C_4^*)$ is a valid ciphertext, A_1 outputs correct $\beta\prime = \beta$ with the advantage $|2\Pr[A_1 \text{ wins}] - 1| \ge \varepsilon$. If $T \in \mathbb{G}_T$ is a random value, $C_\beta^* = (C_1^*, C_2^*, C_{M_\beta}^*, C_4^*)$ is not a valid ciphertext; it cannot provide useful information for the guess of A_1. Thus, A_1 outputs correct $\beta\prime = \beta$ with the advantage $|\Pr[A_1 \text{ wins}]| = 1/2$.

Thus, B breaks the q-ABDHE problem with advantage $|\Pr[B(g, g^{\alpha}, \ldots, g^{\alpha^q}, g', g'^{\alpha^{q+2}}, e(g, g')^{\alpha^{q+1}}) = 1] - \Pr[B(g, g^{\alpha}, \ldots, g^{\alpha^q}, g', g'^{\alpha^{q+2}}, T) = 1]| \ge |(1/2 \pm \varepsilon) - 1/2| = \varepsilon$.

Theorem 2. *If there is a PPT adversary A_2 against the CBE scheme resilient to continual leakage with advantage ε that makes at most q_{sk} secret key queries, q_d decryption queries with l bits entropy leakage for the symmetric key, then there exists a PPT algorithm B against the 1-BDHI problem with advantage $\varepsilon\prime \ge \varepsilon/q_{sk}$.*

Proof. Given $(\mathbb{G}_T, \mathbb{G}, p, e, g, g^{\alpha}, T)$, the algorithm B performs as the challenger $\mathscr{C}$ of IND-RCL-CCA Game-2 to interact with A_2; the target of B is to decide whether $T = e(g, g)^{1/\alpha}$.

Setup: the algorithm B randomly picks $x, z, x_1, x_2 \in \mathbb{Z}_p^*$ and computes $g_1 = g^x$, $g_T = e(g, g)$, $h_1 = g^{x_1}$ and $h_2 = g^{x_2}$. B chooses two collision resistant hash functions $H_1 : \{0,1\}^* \times \mathbb{G}^3 \longrightarrow \mathbb{Z}_p^*$, $H_2 : \mathbb{G} \times \mathbb{G}_T \longrightarrow \mathbb{Z}_p^*$, and an average-case $(\log|\mathbb{G}_T| - l, \varepsilon_{\mathrm{Ext}})$-strong extractor $\mathrm{Ext} : \mathbb{G}_T \times \{0,1\}^{\mu} \longrightarrow \{0,1\}^{\eta}$, $\mu, \eta \in \mathbb{N}$, and sends $\mathrm{params} = (p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_T, h_1, h_2, e, H_1, H_2, \mathrm{Ext})$ and $\mathrm{MSK} = x$ to A_2.

Phase 1: A_2 asks B for queries adaptively as follows:

Public key queries: A_2 inputs ID_i; B randomly chooses $I \in [1, q_{sk}]$. B checks the tuple $(ID_i, sk_{ID_i}, PK_{ID_i})$ from the $\mathscr{L}_1$. If it exists, B returns PK_{ID_i} to A_2. Otherwise, if $ID_i \neq ID_I$, B randomly picks $sk_{ID_i} = (s_1, s_2) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$, computes $PK_{ID_i} = (g_1^{s_1}, g^{s_1}, g^{s_2})$, inserts the tuple $(ID_i, sk_{ID_i}, PK_{ID_i})$ into $\mathscr{L}_1$, and returns PK_{ID_i}; if $ID_i = ID_I$, B sets $PK_{ID_I} = ((g^{\alpha})^x,$

g^{α}, g^{z}), inserts the tuple $(ID_I, sk_{ID_I}, \bullet)$ into $\mathscr{L}_1$, and outputs PK_{ID_I} to A_2.

Secret key queries: For a secret key query under ID_i, if $ID_i \neq ID_I$, B seeks $(ID_i, sk_{ID_i}, PK_{ID_i})$ from the list $\mathscr{L}_1$ and returns sk_{ID_i} to A_2; otherwise, B ends the game and outputs a failure information.

Leakage queries: A_2 inputs ID; B finds (ID, K, cnt) from $\mathscr{L}_2$. If the item does not exist, B adds $(ID, K, 0)$ into $\mathscr{L}_2$. If the item exists, or in the next step, B decides whether $\text{cnt} + l_i \leq l$ where $l, i \in \mathbb{N}$. If not, B returns $\perp$. Otherwise, B sets $\text{cnt} \longleftarrow \text{cnt} + l_i$ for (ID, K, cnt) and returns $f_i(K)$ to A_2, where f_i is a leakage function and $f_i : \mathbb{G}_T \longrightarrow \{0,1\}^{l_i}$.

Decryption queries: For queries on ID_i and ciphertext $C = (C_1, C_2, C_3, C_4)$. If $ID_i \neq ID_I$, B obtains the secret key sk_{ID_i} and certificate Cert_{ID_i} of ID_i; then B obtains the symmetric key K and uses K to decrypt C. Otherwise, B computes the symmetric key $K = e(C_1, g)^{\varphi/(x-\rho_I)} \cdot C_2^{\varphi \cdot x_1 + x_2 + z}$ where $\varphi = H_2(C_1, C_2)$ and $\rho_I = H_1(ID_I, PK_{ID_I})$ and computes $M = C_3 \oplus \text{Ext}(K, C_4)$. B returns either M or $\perp$ to A_2.

Challenge: M_0 provides B with two identical-length messages M_1 and ID^* and a target identity A_2 with the following restriction: ID^* is not allowed to issue the secret key queries for $ID^* \neq ID_I$. If $r \in \mathbb{Z}_p^*$, B ends the game and outputs failure. Otherwise, B will randomly select $C_1^* = (g_1 \cdot g^{-\rho^*})^r$ and compute $C_2^* = T^r$, $\rho^* = H_1(ID_I, PK_{ID_I})$, where $T = e(g,g)^{1/\alpha}$ and $\beta \in \{0,1\}$. B randomly selects $\sigma^* \in \{0,1\}^{\mu}$ and $K_1^* = T^{r(\varphi^* x_1 + x_2 + z)} \cdot e(g,g)^{\varphi^* r}$ and computes $\varphi^* = H_2(C_1^*, C_2^*)$ where $K_0^* \in \mathbb{G}_T$. B randomly chooses $\beta \in \{0,1\}$, $\sigma^* \in \{0,1\}^{\mu}$, and $C_{M_\beta}^* = \text{Ext}(K_\beta^*, \sigma^*) \oplus M_\beta$, sets $C_4^* = \sigma^*$, $C_\beta^* = (C_1^*, C_2^*, C_{M_\beta}^*, C_4^*)$, and outputs the challenge ciphertext A_2 to A_2.

Phase 2: A_2 continues making the queries which is similar to Phase 1 under the following restriction: ID^* has no permission to perform secret key queries for (ID^*, C^*), as well as the decryption queries on A_2.

Guess: $\beta\prime \in \{0,1\}$ returns the guess $\beta\prime = \beta$. If $T = e(g,g)^{1/\alpha}$ holds, B outputs 1, indicating that $T \neq e(g,g)^{1/\alpha}$. Otherwise, B outputs 0, which indicates $T = e(g,g)^{1/\alpha}$. □

5.2. Probability analysis. If $r^* = r/\alpha$, we set , and we have $C_1^* = (g_1 \cdot g^{-\rho^*})^r = (g_1 \cdot g^{-\rho^*})^{\alpha r^*} = (PK_{ID_I}^{(1)} \cdot (PK_{ID_I}^{(2)})^{-\rho^*})^{r}$, $C_2^* = T^r = (e(g,g)^{1/\alpha})^{\alpha r^*} = g_T^{r^*}$. $K_1^* = T^{r(\varphi^* x_1 + x_2 + z)} \cdot e(g,g)^{\varphi^* r} = e(g,g)^{(\varphi^* x_1 + x_2 + z) r^*} \cdot e(g,g)^{\varphi^* \alpha r^*} = (e(g, h_1 \cdot PK_{ID^*}^{(2)})^{\varphi^*} \cdot e(g, h_2 \cdot PK_{ID^*}^{(3)}))^{r^*}$. Thus, $C_\beta^* = (C_1^*, C_2^*, C_{M_\beta}^*, C_4^*)$ is a valid ciphertext; A_2 outputs correct $\beta\prime = \beta$ with the advantage $|2\Pr[A_2 \text{ wins}] - 1| \geq \varepsilon$. If $T \in \mathbb{G}_T$ is a random value, $C_\beta^* = (C_1^*, C_2^*, C_{M_\beta}^*, C_4^*)$ is an invalid ciphertext; it cannot provide useful information for the guess of A_2. Thus, A_2 outputs correct $\beta\prime = \beta$ with the advantage $|\Pr[A_2 \text{ wins}]| = 1/2$.

Thus, B breaks the 1-BDHI problem with advantage $|\Pr[B(g, g^{\alpha}, e(g,g)^{1/\alpha}) = 1] - \Pr[B(g, g^{\alpha}, T) = 1]| \geq |(1/2 \pm \varepsilon) - 1/2| = \varepsilon$. Furthermore, the probability that A_2 chooses ID_I as the target identity is $1/q_{sk}$. Therefore B breaks the 1-BDHI problem with advantage $\varepsilon\prime \geq \varepsilon/q_{sk}$.

5.3. Leakage Ratio Analysis. We mainly consider the leakage of the symmetric key K. Firstly, a set Z is defined which consists of public parameters, secret keys, and certificates. As an adversary, A acquires at most l bits for leakage of the symmetric key K. Based on Lemma 1, we have $\tilde{H}_{\infty}(K \mid A, (\text{Leak}, Z)) \geq \tilde{H}_{\infty}(A \mid Z) - l = \log|\mathbb{G}_T| - l$, where Leak has 2^l possible values and $l \in \mathbb{N}$ is the leakage length. If we pick the average-case $(\log|\mathbb{G}_T| - l, \varepsilon_{\text{Ext}})$-strong extractor where ε_{Ext} is negligible, we know that $\text{SD}((\text{Ext}(X, U_\mu), U_\mu, Y), (U_\eta, U_\mu, Y)) \leq \varepsilon$, where U_μ and U_η have uniform distributions over $\{0,1\}^{\mu}, \{0,1\}^{\eta}$, respectively. Thus, the ciphertext $C_3 = \text{Ext}(K, \sigma) \oplus M$ and the uniform distribution cannot be distinguished. Moreover, $\log|\mathbb{G}_T| - l$ can be close to zero, the leakage bound l is roughly equal to $\log|\mathbb{G}_T|$, and the leakage ratio of K is $l/\log|\mathbb{G}_T| \approx \log|\mathbb{G}_T|/\log|\mathbb{G}_T| = 1$.

6. Efficiency Comparison

Three CBE schemes [3, 6, 8] are compared with our proposed approach, to evaluate their security and efficiency. The security properties and leakage ratio comparison for four CBE schemes are shown in Table 1.

Table 1 demonstrates that the schemes in [6, 8] and our CBE scheme are leakage-resilient while the scheme in [3] is not. The key-leakage ratio of the scheme [8] is up to 1/3. However, the symmetric key-leakage ratio of scheme [6] and our scheme is close to 1. In addition, our scheme is resistant to continual leakage. In conclusion, our CBE scheme has obvious advantage.

Let $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_3}$ denote the subgroups of orders p_1 and p_3 in $\mathbb{G}$, respectively, where p_1 and p_3 are distinct primes. An NIZK proof is represented by π in [8], and n is an integer. We analyze the communication cost for the four schemes as follows.

From Table 2, the difference of communication performance between the scheme in [6] and our scheme is not obvious. The length of the public/secret key, the certificate, and the ciphertext in the proposed approach is less than that required in [3]. Moreover, the length of the certificate and ciphertext in our proposed approach is also less than that required in [8]. Therefore, our CBE scheme achieves a lower communication cost, compared with the two schemes in [3, 8].

We also implement these schemes in a Windows 10 environment (Intel (R) Core (TM) i7-6500U CPU, 8.00 GB RAM) using C++ language and PBC [50] library, where *a*.param is used as the configuration file, and the message length is fixed at 1024 bits. Note that every comparative scheme is separately run ten times, in order to obtain the average running time. The required running times of the four schemes are listed below.

From both Table 3 and Figure 1, it can be found that although our scheme was added with the secret key update algorithm, the total operating time is shorter than that of the

Table 1: Security properties and leakage ratio comparison.

Schemes	Model	Hard assumption	Leakage-resilience	Continual leakage-resilience	Leakage ratio
Scheme [3]	Standard	q-ABDHE and DBDH	×	×	—
Scheme [6]	Standard	Decisional 3-party Diffie–Hellman	√	×	≈1
Scheme [8]	Standard	Three assumptions in composite order bilinear groups	√	×	1/3
Our scheme	Standard	Decisional truncated q-ABDHE and decisional 1-BDHI	√	√	≈1

Table 2: Comparison of the required communication cost.

Schemes	Public key length	Secret key length	Certificate length	Cipher text length
Scheme [3]	$6\|\mathbb{G}\|$	$6\|\mathbb{Z}_p\|$	$3\|\mathbb{Z}_p\| + 3\|\mathbb{G}\|$	$3\|\mathbb{G}_T\| + \|\mathbb{G}\|$
Scheme [6]	$2\|\mathbb{G}\|$	$\|\mathbb{Z}_p\|$	$3\|\mathbb{G}\|$	$2\|\mathbb{G}\| + \eta + \mu$
Scheme [8]	$\|\mathbb{G}_T\| + \|\pi\|$	$\|\mathbb{Z}_p\|$	$(n+2)(\|\mathbb{G}_{p_1}\| \cdot \|\mathbb{G}_{p_3}\|)$	$\|\mathbb{G}_T\| + (n+2)\|\mathbb{G}_{p_1}\|$
Our scheme	$3\|\mathbb{G}\|$	$2\|\mathbb{Z}_p\|$	$2\|\mathbb{Z}_p\| + 2\|\mathbb{G}\|$	$\|\mathbb{G}\| + \|\mathbb{G}_T\| + \mu + \eta$

Table 3: Running time in microsecond.

Schemes	Setup	UserKeyGen	CertGen	Encrypt	Decrypt	UpdateSK	Total
Scheme [3]	28	33	27	50	5	—	143
Scheme [6]	23	10	25	27	18	—	105
Scheme [8]	104	3	34	11	42	—	195
Our scheme	15	14	29	23	9	3	96

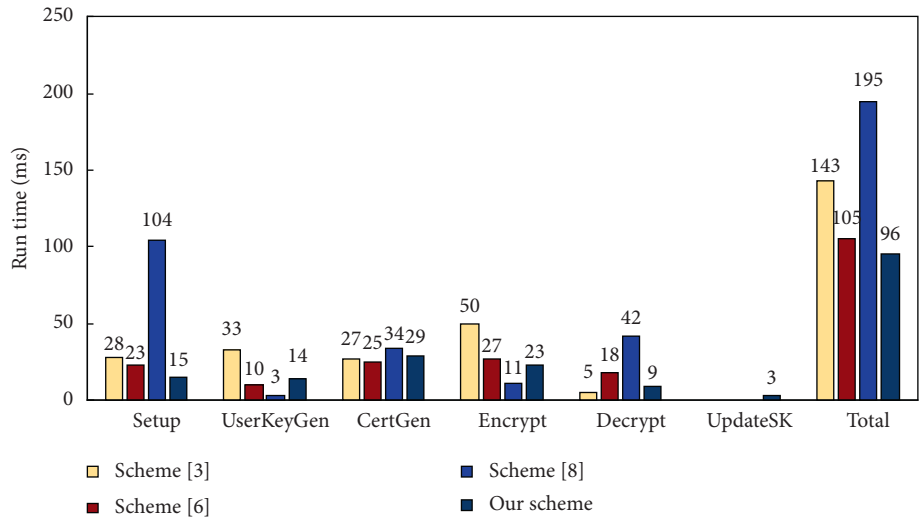

Figure 1: Run time comparison.

other three schemes in [3, 6, 8]. This indicates that our proposed method outperforms the other approaches with higher efficiency.

The message space in scheme [6] and our scheme is $\mathcal{M} = \{0,1\}^{\eta}$; we therefore analyze these two schemes based on the relationship between different lengths of the messages and encryption/decryption running times. Figure 2 shows the relation between encryption running times and the message lengths for scheme [6] and our scheme. The relation between decryption running times and message sizes for both schemes is shown in Figure 3.

According to Figure 2, it can be seen that the required time for encryption in the two approaches and the message sizes are linearly increased. Although the longer the message, the longer the encryption time, the growth of our scheme has a relatively lower amplitude than scheme [6]. From Figure 3, the decryption times for two schemes are not greatly affected by the lengths of the messages. In addition, it can be seen that, for different message lengths, the required time for decryption in our scheme is shown to be less than that of the scheme in [6]. Hence, our proposed scheme has certain advantages from this perspective.

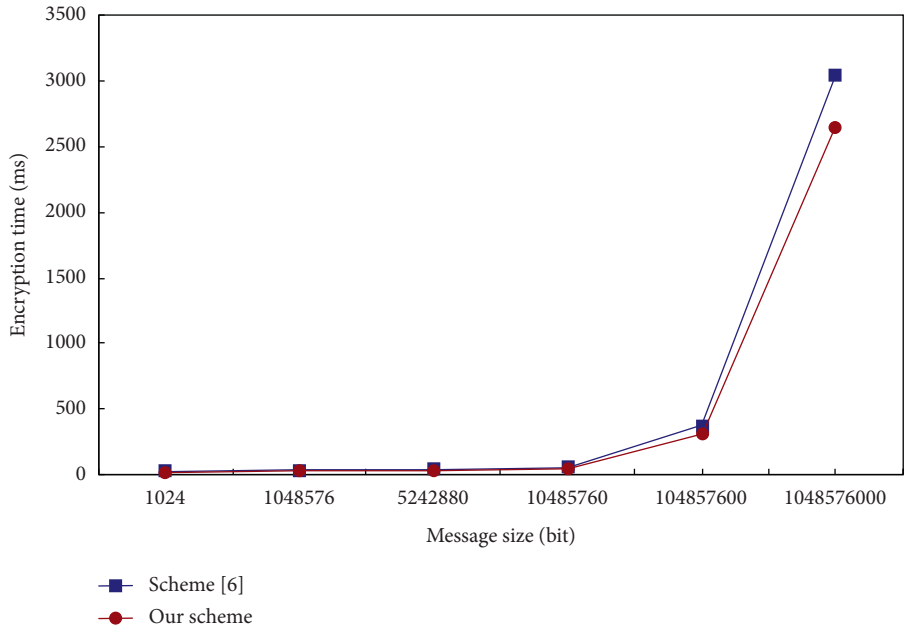

Figure 2: Relation between encryption times and message sizes.

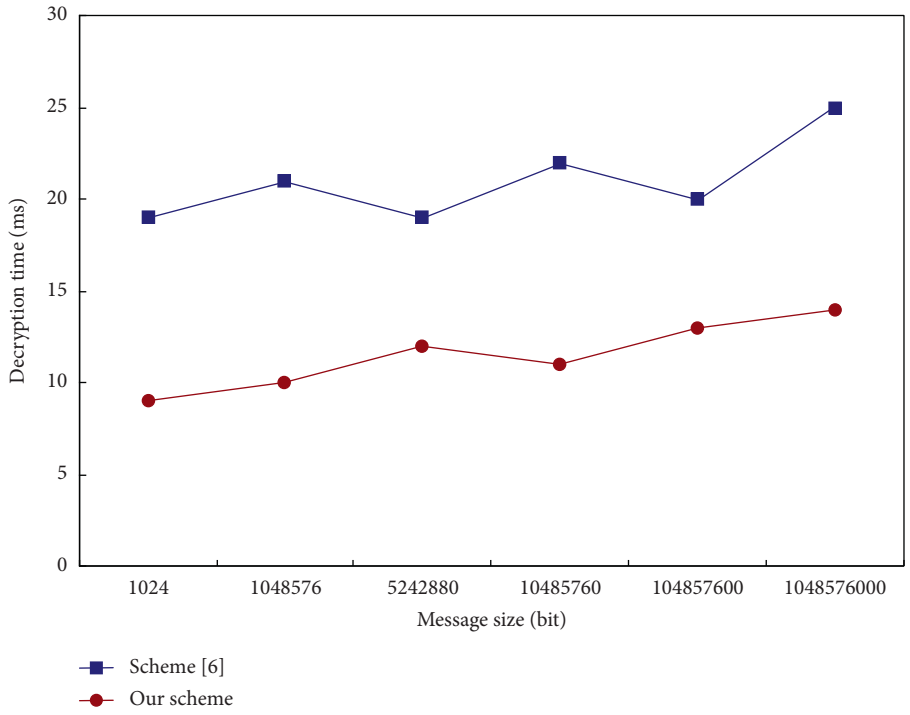

Figure 3: Relation between decryption times and message sizes.

7. Conclusions

In this work, we give a formal definition and the security model for CBE resilient to continual leakage. Besides, we construct a CBE scheme which is resilient to continual leakage. The security of our scheme is reduced to the hardness of the decisional truncated q-augmented bilinear Diffie–Hellman exponent problem and the decisional 1-bilinear Diffie–Hellman inversion problem. Moreover, comparative studies are provided with other existing solutions, in terms of their performance analyses. Our scheme is proved secure against the chosen-ciphertext attack in the standard model. To construct CBE scheme with stronger leakage-resilient property (such as auxiliary inputs, post-challenge leakage, etc.) and leakage-resilient certificateless encryption scheme with keyword search [51], leakage-resilient location determination scheme [52] is left as our future study.

Data Availability

The nature of the data is the c++ language source code. The data used in the finding of this study are included in the article (the environment configuration of the simulation). The corresponding data are attached, which need to invoke the pairing-based cryptography library (PBC) library in [37].

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this article.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China under Grant nos. 61902140, 60573026, 61972095, and U1736112, the Anhui Provincial Natural Science Foundation under Grant nos. 1908085QF288 and 1708085QF154, and the Nature Science Foundation of Anhui Higher Education Institutions under Grant nos. KJ2018A0398, KJ2018A0678, KJ2018A0396, and KJ2019A0605.

References

[1] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Proceedings of the EUROCRYPT*, pp. 272–293, Warsaw, Poland, May 2003.

[2] D. Galindo, P. Morillo, and C. Ràfols, "Improved certificate-based encryption in the standard model," *Journal of Systems and Software*, vol. 81, no. 7, pp. 1218–1226, 2008.

[3] J. K. Liu and J. Zhou, "Efficient certificate-based encryption in the standard model," in *Proceedings of the International Conference on Security and Cryptography For Networks*, pp. 144–155, Amalfi, Italy, September 2008.

[4] Y. Lu, J. Li, and Y. Zhang, "Secure channel free certificate-based searchable encryption withstanding outside and inside keyword guessing attacks," *IEEE Transactions on Services Computing*.

[5] W. Wu, Y. Mu, W. Susilo, X. Huang, and L. Xu, "A provably secure construction of certificate-based encryption from certificateless encryption," *The Computer Journal*, vol. 55, no. 10, pp. 1157–1168, 2012.

[6] Y. Guo, J. Li, Y. Lu, Y. Zhang, and F. Zhang, "Provably secure certificate-based encryption with leakage resilience," *Theoretical Computer Science*, vol. 711, pp. 1–10, 2018.

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

[8] Q. Yu, J. Li, Y. Zhang, W. Wu, X. Huang, and Y. Xiang, "Certificate-based encryption resilient to key leakage," *Journal of Systems & Software*, vol. 116, pp. 101–112, 2016.

[9] W. Yang, J. Weng, A. Yang, C. Xie, and Y. Yang, "Notes on a provably-secure certificate-based encryption against malicious CA attacks," *Information Sciences*, vol. 463, pp. 86–91, 2018.

[10] J. Li, L. Chen, Y. Lu, and Y. Zhang, "Anonymous certificate-based broadcast encryption with constant decryption cost," *Information Sciences*, vol. 454, pp. 110–127, 2018.

[11] J. Li, X. Huang, Y. Mu, and W. Susilo, "Constructions of certificate-based signature secure against key replacement attacks," *Journal of Computer Security*, vol. 18, no. 3, pp. 421–449, 2010.

[12] J. Li, X. Huang, Y. Zhang, and L. Xu, "An efficient short certificate-based signature scheme," *Journal of Systems and Software*, vol. 85, no. 2, pp. 314–322, 2012.

[13] J. Li, Z. Wang, and Y. Zhang, "Provably secure certificate-based signature scheme without pairings," *Information Sciences*, vol. 233, pp. 313–320, 2013.

[14] J. Li, X. Huang, M. Hong, and Y. Zhang, "Certificate-based signcryption with enhanced security features," *Computers and Mathematics with Applications*, vol. 64, no. 6, pp. 1587–1601, 2012.

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

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

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

[18] Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan, "Public-key encryption schemes with auxiliary inputs," in *Proceedings of the Theory of Cryptography Conference*, pp. 361–381, Zurich, Switzerland, February 2010.

[19] G. Yang, Y. Mu, W. Susilo, and D. S. Wong, "Leakage resilient authenticated key exchange secure in the auxiliary input model," in *Proceedings of the International Conference on Information Security Practice And Experience*, pp. 204–217, Lanzhou, China, May 2013.

[20] E. Kiltz and K. Pietrzak, "Leakage resilient elgamal encryption," in *Proceedings of the ASIACRYPT*, pp. 595–612, Singapore, December 2010.

[21] 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 IEEE 16th Annual Symposium on Foundations of Computer Science*, pp. 501–510, Las Vegas, NV, USA, October 2010.

[22] Y. Dodis, K. Haralambiev, A. Lopez-Alt, and D. Wichs, "Cryptography against continuous memory attacks,," in *Proceedings of the IEEE 16th Annual Symposium on Foundations of Computer Science*, pp. 511–520, Las Vegas, NV, USA, October 2010.

[23] S. Agrawal, Y. Dodis, V. Vaikuntanathan, and D. Wichs, "On continual leakage of discrete log representations," in *Proceedings of the ASIACRYPT*, pp. 401–420, Bengaluru, India, December 2013.

[24] T. H. Yuen, S. S. M. Chow, Y. Zhang, and S. M. Yiu, "Identity-based encryption resilient to continual auxiliary leakage," in *Proceedings of the EUROCRYPT*, pp. 117–134, Cambridge, UK, April 2012.

[25] Y. Zhou, B. Yang, H. Hou, L. Zhang, T. Wang, and M. Hu, "Continuous leakage-resilient identity-based encryption with tight security," *The Computer Journal*, vol. 62, no. 8, 2019.

[26] Y. Zhou, B. Yang, and Y. Mu, "Continuous leakage-resilient identity-based encryption with leakage amplification," *Designs Codes & Cryptography*, vol. 87, no. 9, pp. 2061–2090, 2019.

[27] Y. Zhou, B. Yang, and Y. Mu, "The generic construction of continuous leakage-resilient identity-based cryptosystems," *Theoretical Computer Science*, vol. 772, pp. 1–45, 2019.

[28] 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.

[29] Y. Zhou, B. Yang, Z. Xia, M. Zhang, and Y. Mu, "Identity-based encryption with leakage-amplified chosen-ciphertext attacks security," *Theoretical Computer Science*, vol. 809, pp. 277–295, 2020.

[30] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 785–796, 2017.

[31] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1767–1777, 2018.

[32] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Transactions on Services Computing*, vol. 13, no. 3, pp. 478–487, 2020.

[33] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2017.

[34] 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*, 2020.

[35] C. Wang, C. Wang, Z. Wang, X. Ye, J. X. Yu, and B. Wang, "Deepdirect: learning directions of social ties with edge-based network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2277–2291, 2019.

[36] M. Zhang, "New model and construction of ABE: achieving key resilient-leakage and attribute direct-revocation," in *Proceedings of the Australasian Conference on Information Security and Privacy*, pp. 192–208, Wollongong, NSW, Australia, July 2014.

[37] M. Zhang, Y. Zhang, Y. Su, Q. Huang, and Y. Mu, "Attribute-based hash proof system under learning-with-errors assumption in obfuscator-free and leakage-resilient environments," *IEEE System Journal*, vol. 11, no. 2, pp. 1018–1026, 2017.

[38] J. Zhang, J. Chen, J. Gong, A. Ge, and C. Ma, "Leakage-resilient attribute based encryption in prime-order groups via predicate encodings," *Designs Codes & Cryptography*, vol. 86, no. 6, pp. 1–28, 2018.

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

[40] 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*, .

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

[42] Y Zhou and B. Yang, "Continuous leakage-resilient certificateless public key encryption with CCA security," *Knowledge-Based Systems*, vol. 136, pp. 27–36, 2017.

[43] J. Li, Y. Guo, Q. Yu, Y. Lu, Y. Zhang, and F. Zhang, "Continuous leakage-resilient certificate-based encryption," *Information Sciences*, vol. 355, pp. 1–14, 2016.

[44] R. Chen, Y. Mu, G. Yang, W. Susilo, and F. Guo, "Strongly leakage-resilient authenticated key exchange,," in *Proceedings of the Cryptographers' Track at the RSA Conference*, pp. 19–36, San Francisco, CA, USA, February 2016.

[45] R. Chen, Y. Mu, G. Yang, W. Susilo, and F. Guo, "Strong authenticated key exchange with auxiliary inputs," *Designs, Codes and Cryptography*, vol. 85, no. 1, pp. 145–173, 2017.

[46] R. Chen, Y. Mu, G. Yang, W. Susilo, F. Guo, and Z. Yang, "A note on the strong authenticated key exchange with auxiliary inputs," *Designs, Codes and Cryptography*, vol. 85, no. 1, pp. 175–178, 2017.

[47] G. Yang, R. Chen, Y. Mu, W. Susilo, F. Guo, and J. Li, "Strongly leakage-resilient authenticated key exchange, revisited," *Designs, Codes and Cryptography*, vol. 87, no. 12, pp. 2885–2911, 2019.

[48] Y Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Proceedings of the EUROCRYPT*, pp. 523–540, Interlaken, Switzerland, May 2004.

[49] Y. Lu and J. Li, "Efficient and provably-secure certificate-based key encapsulation mechanism in the standard model," *Journal of Computer Research and Development*, vol. 51, no. 7, pp. 1497–1505, 2014, in Chinese.

[50] B. Lynn, *PBC (Pairing-Based Cryptography) Library*, Springer, Berlin, Germany, 2012.

[51] Y. Lu, J. Li, and Y. Zhang, "Privacy-preserving and pairing-free multi-recipient certificateless encryption with keyword search for cloud-assisted IIoTs," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2553–2562, 2020.

[52] H. Shen, M. Zhang, H. Wang, F. Guo, and S. Willy, "A lightweight privacy-preserving fair meeting location determination scheme," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3083–3093, 2020.