IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Identity-based encryption resilient to continuous key leakage

ISSN 1751-8709
Received on 6th January 2018
Revised 15th May 2018
Accepted on 2nd November 2018
E-First on 2nd April 2019
doi: 10.1049/iet-ifs.2018.5203
www.ietdl.org

Yanwei Zhou[1,2,3], Bo Yang[1,2,3] ✉, Yi Mu[4], Tao Wang[1,3], Xin Wang[1]

[1]*School of Computer Science, Shaanxi Normal University, Xi'an, People's Republic of China*
[2]*State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, People's Republic of China*
[3]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, People's Republic of China*
[4]*Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, People's Republic of China*
✉ *E-mail: byang@snnu.edu.cn*

Abstract: Leakage of private information has become a threat to the security of computing systems. It has become a common security requirement that a cryptography scheme should withstand various leakage attacks, even the continuous leakage attacks. However, in the current constructions on the (continuous) leakage-resilient identity-based encryption (CLR-IBE) scheme, the leakage parameter is a fixed value. Aiming to solve these problems, in this study, the authors show how to construct the CLR-IBE scheme, and the adaptive chosen-ciphertext attacks security of proposed construction can be proved in the standard model. To further improve the practicability of CLR-IBE scheme, they design an improved IBE scheme with continuous leakage amplified property, and the leakage parameter has an arbitrary length.

1 Introduction

The security of the traditional cryptography scheme is proved in a weaker security model, in which legitimate participants hold internal respective secret states, such as private keys of a user, which are assumed completely inaccessible to the adversary. However, in the real world, an adversary can learn any additional information on the internal secret states through various leakage attacks, such as side-channel attacks, cold-boot attacks and so on. If an adversary obtains a certain amount of leakage on the internal secret states, then the traditional cryptography schemes will not hold their claimed security. Therefore, leakage-resilient cryptography schemes are important to meet the security requirements in the real world. However, most previous leakage-resilient cryptography schemes [1–13] only consider the bounded leakage attacks and do not allow any continuous leakage. In the real world, the leakage is unbounded, because of the adversary can continuously learn some additional information on the private key of the user, and can trivially break the security of cryptography scheme under the continuous leakage attacks. Thus, to further improve the practicability, a cryptography scheme must keep its original security in the continuous leakage setting. For the past few years, several constructions have captured continuous leakage resilience in their security consideration, such as the continuous leakage-resilient public-key encryption (CLR-PKE) [14–16], the continuous leakage-resilient authentication key exchange (CLR-AKE) [17], the continuous leakage-resilient (CLR) signature [18], the CLR certificate-based encryption (CLR-CBE) [10, 11, 19], the CLR certificateless public-key encryption (CLR-CL-PKE) [20] and so on. In the identity-based setting, CLR identity-based encryption (CLR-IBE) schemes with the semantic security were presented in [21–23]. Furthermore, a CLR-IBE scheme is designed in [24], and its chosen-ciphertext attacks (CCA) security is proved in the selective identity model. That is, the previous CLR-IBE schemes either achieve CPA security or obtain CCA security in the selective identity model. Hence, there is no practical IBE scheme with full CCA security in the literature, which has adopted the continuous leakage model. Specially, in the current constructions on the leakage-resilient cryptography schemes, the leakage parameter is a fixed value. That is, the upper bound of permitted leakage keep unchanged even if the actual applications have different leakage requirements. In other words, in the actual application, we cannot resist the various leakage attacks through fixed leakage upper bound. If the upper bound of allowed leakage is smaller than the total length of actual leakage, then, the corresponding cryptography scheme is unusable.

In this paper, we provide a complete solution method for the above problems and show how to construct a CLR-IBE scheme, and its adaptive CCA security is proved, in the standard model, based on a stronger security assumption that depends on the number of private key generation queries made by the adversary. Also, we design a novel construction of CLR-IBE scheme, in which, the upper bound of permitted leakage is depended on the leakage requirements of actual applications.

Prior constructions: To resist the leakage attacks, Alwen *et al.* [25] generalised hash proof systems (HPS) in the identity-based setting and referred to it as identity-based HPS (IB-HPS). Also, they took it as a basic tool to construct CPA secure leakage-resilient IBE (LR-IBE) scheme. That is, based on the IB-HPS, a generic construction of the LR-IBE scheme with CPA security was built. Chow *et al.* [26] proposed three new systems of IB-HPS with the previous identity-based encryption (IBE) schemes [27–29]. Then, three CPA secure LR-IBE schemes can be obtained from the generic method proposed in [25]. Different from these works [25, 26], Yuen *et al.* [23] designed a novel CLR-IBE scheme in the auxiliary input model, which can tolerate a more general form of leakage. Among all, these IBE schemes [23, 25, 26] were only proved CPA secure, and these schemes suffer from the undesirable drawback that the leakage parameter λ and the plaintext length l_m are subject to $\lambda + l_m \leq con$, where con is a constant value. In this case, when the plaintext length l_m approaches to the con, the size of leakage approaches to 0, vice versa. To improve the security level, Li *et al.* [30] presented a new LR-IBE scheme based on Gentry's IBE scheme [31], which can achieve CCA security. In addition, by Liu *et al.*'s [2] conclusion, an LR-IBE scheme with better performance was created by Sun *et al.* [7], in which, the number of bits leaked is $\lambda \leq \log q + \omega(\log \kappa)$, where q is a big prime order of the underlying group and κ is the secret parameter. That is, in [7], the leakage parameter and the plaintext message are independent of each other. The above constructions [7, 25, 26, 30] were designed

in the bounded-leakage model, and cannot keep their claimed security in the continuous leakage setting. Although, Yuen *et al.* [23], Lewko *et al.* [21] and Li *et al.* [22] proposed three CLR-IBE schemes from the dual system encryption technology, which can only achieve CPA security. The CLR-IBE scheme designed in [24] can only obtain the selective identity CCA security. However, adaptive CCA security is a strong and very useful notion of security for IBE scheme. Therefore, in the actual application environment, the CLR-IBE scheme with adaptive CCA security is indispensable.

Our contributions: In the IBE schemes, there exists a bigger gap between the research about how to resist the leakage attacks and the leakage requirements of the real world, because of the bounded leakage resilience is only considered in the current constructions. Also, the existing CLR-IBE schemes either achieve CPA security or obtain selective identity CCA security. In this paper, we focus on the construction of CCA secure CLR-IBE scheme, and the scheme's security is proved, in the standard model, based on the hardness of the decisional augmented bilinear Diffie–Hellman exponent (q-ABDHE) assumption and the following features are the targets of design: (i) we allow continuous leakage on the private key of user; (ii) the round leakage parameter is independent of the plaintext space and has a constant size; (iii) the leakage parameter has arbitrary length, which depends on the leakage requirements of actual applications; and (iv) the adaptive CCA security can be obtained.

To achieve the above objectives, we first propose a CLR-IBE scheme Π with CPA security, and whose security is proved, in the standard model, based on the q-ABDHE assumption. After that, based on our basic construction Π, we design a CLR-CCA secure IBE scheme Π', and its security can be proved with the existing method. Finally, a novel continuous leakage-amplified IBE scheme Π'' is created, and the upper bound of round leakage parameter can be flexibly controlled and is determined by the practical considerations about how much leakage the cryptosystem needs to tolerate. That is, based on our basic construction Π (or Π'), we can develop a CPA (or CCA) secure IBE scheme with continuous leakage amplification.

2 Preliminaries

Due to the space limitation, the basic notions such as statistical distance $\mathtt{SD}(\mathtt{A},\mathtt{B})$, min-entropy $H_\infty(A)$ and average conditional min-entropy $\widetilde{H}_\infty(A|C)$ are omitted in the presentation, and the reader is referred to [2, 32, 33] for the details. Moreover, the formal definition of (continuous) LR-IBE scheme is also omitted, and the reader is referred to [23, 30] for the details.

Let $\kappa \in \mathbb{N}$ denote the security parameter. If S is a string, then $|S|$ denotes its length, while if S is a set then $|S|$ denotes its size and $s \leftarrow_R S$ denotes the operation of picking an element s uniformly at random from S. We denote $y \leftarrow \mathscr{A}(x)$ the operation of running $\mathscr{A}$ with input x and assigning y as a result. We use $\mathrm{negl}(\kappa)$ to denote the set of all functions that are negligible in the security parameter κ.

2.1 Bilinear groups

Let $\mathscr{G}(1^\kappa)$ be a probability polynomial time (PPT) group generation algorithm that takes as input a security parameter κ, and outputs a tuple $\mathbb{G} = (p, G, G_T, e(\cdot,\cdot), P)$, such that (i) G and G_T are two cyclic groups of prime order p; (ii) P is a generator of G; (iii) $e: G \times G \to G_T$ is an efficiently computable bilinear pairing with the following properties:

Bilinear: $e(aU, bV) = e(U, V)^{ab}$, where $a, b \leftarrow_R \mathbb{Z}_p^*$ and $U, V \leftarrow_R G_T$.

Non-degeneracy: $e(P, P) \neq 1_{G_T}$, where 1_{G_T} is the generator of G_T.

Computable: $e(U, V)$ can be computed efficiently for all $U, V \in G$.

2.2 Security assumption

The security of our constructions is based on a complexity assumption that we call the decisional ABDHE assumption (decisional ABDHE). First, we recall the bilinear Diffie–Hellman exponent (q-BDHE) problem [31], which is described as follows:

Given a vector of $2q+1$ elements

$$(P', P, \alpha P, \alpha^2 P, \ldots, \alpha^q P, \alpha^{q+2} P, \ldots, \alpha^{2q} P) \in G^{2q+1}$$

as input (where $\alpha \in \mathbb{Z}_p^*$), and output $e(P, P')^{\alpha^{q+1}} \in G_T$. Since the input vector is missing the term $\alpha^{q+1} P$, the bilinear map does not seem to help compute $e(P, P')^{\alpha^{q+1}}$.

We define the ABDHE (q-ABDHE) problem almost identically:

Given a vector of $2q+2$ elements

$$(P', \alpha^{q+2} P', P, \alpha P, \ldots, \alpha^q P, \alpha^{q+2} P, \ldots, \alpha^{2q} P) \in G^{2q+2}$$

as input, and output $e(P, P')^{\alpha^{q+1}} \in G_T$. Introducing the additional term $\alpha^{q+2} P'$ still does not appear to ease the computation of $e(P, P')^{\alpha^{q+1}}$, since the input vector is missing the term $\alpha^{q+1} P$. Now, we use P_i and P_i' to denote $\alpha^i P$ and $\alpha^i P'$, respectively. The decisional version of truncated q-ABDHE problem is defined as one would expect. Let $T_1 = e(P_{q+1}, P')$ and $T_0 \leftarrow_R G_T$, then, the advantage of an adversary $\mathscr{S}$ in solving q-ABDHE problem is defined as

$$\begin{aligned}\mathtt{Adv}_{\mathscr{S}}^{\mathrm{q\text{-}ABDHE}}(\kappa) = \big|\ &\Pr\left[\mathscr{A}(P', P'_{q+2}, P, P_1, \ldots, P_q, T_1) = 1\right] \\ &- \Pr\left[\mathscr{A}(P', P'_{q+2}, P, P_1, \ldots, P_q, T_0) = 1\right]\big|,\end{aligned}$$

where the probability is over the random choice of generators P, P' in G, the random choice of α in $\mathbb{Z}_p$, the random choice of $T_0 \leftarrow_R G_T$, and the random bits consumed by $\mathscr{S}$.

Definition 1 (q-ABDHE assumption): For $P, P' \leftarrow_R G$ and $\alpha \leftarrow_R \mathbb{Z}_p^*$, the two tuples $\mathscr{T}_1 = (P', P'_{q+2}, P, P_1, \ldots, P_q, T_1)$ and $\mathscr{T}_0 = (P', P'_{q+2}, P, P_1, \ldots, P_q, T_0)$ are computationally indistinguishable, where $P_i = \alpha^i P$, $P_i' = \alpha^i P'$, $T_1 = e(P_{q+1}, P')$ and $T_0 \leftarrow_R G_T$. We say that the q-ABDHE assumption holds if, for all PPT adversaries $\mathscr{S}$, we have

$$\mathtt{Adv}_{\mathscr{S}}^{\mathrm{q\text{-}ABDHE}}(\kappa) \leq \mathtt{negl}(\kappa).$$

2.3 Randomness extractor

By the notion of $\widetilde{H}_\infty(A|C)$ [32], for any adversary $\mathscr{A}$, we obtain

$$\begin{aligned}\Pr\left(\mathscr{A}(C) = A\right) &= \boldsymbol{E}_c[\Pr\left(\mathscr{A}(C) = A\right)] \leq \boldsymbol{E}_c\left[2^{-H_\infty(A|C=c)}\right] \\ &= 2^{-\widetilde{H}_\infty(A|C)}.\end{aligned}$$

Lemma 1: For three variables A, B and C, if $|B| = 2^l$, then we will obtain $\widetilde{H}_\infty(A|(B,C)) \geq \widetilde{H}_\infty(A|C) - l$ [32].

Definition 2 (Randomness extractor): For $U_m \leftarrow_R \{0,1\}^{l_m}$ and $C \leftarrow_R \{0,1\}^{l_t}$, as well as the two variables B and $A \in \{0,1\}^{l_n}$ subject to the constraint $\widetilde{H}_\infty(A|B) \geq k$, if we have $\mathtt{SD}((\mathtt{U_m}, \mathtt{C}, \mathtt{B}), (\mathtt{Ext}(\mathtt{A}, \mathtt{C}), \mathtt{C}, \mathtt{B})) \leq \epsilon$, then $\mathtt{Ext}: \{0,1\}^{l_n} \times \{0,1\}^{l_t} \to \{0,1\}^{l_m}$ is an average-case (k, ϵ)-strong randomness extractor.

Definition 3 (Universal hash function): For $i \in \mathscr{I}$ and all distinct $x_1 \neq x_2 \in \mathscr{X}$, if we have $\Pr_{i \leftarrow \mathscr{I}}[H_i(x_1) = H_i(x_2)] \leq (1/|\mathscr{Y}|)$, then the hash function $\mathscr{H}_{\mathscr{I}}: \mathscr{X} \to \mathscr{Y}$ is universal.

Example 1: The family of functions $\left\{H_{k_1,k_2,\ldots,k_l}: \mathbb{Z}_p^{l+1} \to \mathbb{Z}_p\right\}_{k_i \in \mathbb{Z}_p,\, i=1,\ldots,l}$ is universal, where $H_{k_1,k_2,\ldots,k_l}(x_0, x_1, \ldots, x_l) = x_0 + k_1x_1 + \cdots + k_lx_l$. All operations are in the prime field $\mathbb{F}_p$ [2].

Example 2: Let G be a multiplicative group of prime order p, and $g \in G, g \neq 1$. The family of functions $\left\{H_{k_1,k_2,\ldots,k_l}: G^{l+1} \to G\right\}_{k_i \in \mathbb{Z}_p, i=1,\ldots,l}$ is universal, where $H_{k_1,k_2,\ldots,k_l}(g_0, g_1, \ldots, g_l) = g_0 g_1^{k_1} \cdots g_l^{k_l}$ [2].

Lemma 2 (Leftover hash lemma): Let U_y is a uniform distribution over $\mathcal{Y}$. For two variables $X \in \mathcal{X}$ and C, it holds that $\mathrm{SD}((\mathrm{H_i(X)}, \mathrm{i}), (\mathrm{U_y}, \mathrm{i})) \leq \frac{1}{2}\sqrt{2^{-\mathrm{H}_\infty(\mathrm{X})}\left|\mathcal{Y}\right|}$ and $\mathrm{SD}((\mathrm{H_i(X)}, \mathrm{i}, \mathrm{C}), (U_y, i, C)) \leq \frac{1}{2}\sqrt{2^{-\widetilde{H}_\infty(X|C)}\left|\mathcal{Y}\right|}$ [32].

Lemma 3 (Generalised leftover hash lemma): If two variables C and $A \in \mathcal{X}$ subject to the constraint $\widetilde{H}_\infty(A|C) \geq k$, then, for $S \leftarrow_R \mathcal{S}$ and $U_m \leftarrow_R \mathcal{Y}$, we have $\mathrm{SD}((\mathrm{C}, \mathrm{i}, \mathrm{H_i(A)}), (\mathrm{C}, \mathrm{i}, \mathrm{U_m})) \leq \epsilon$ as long as $l_m \leq k - 2\log\left(\frac{1}{\epsilon}\right)$ [32].

By Lemma 3, we have that, for an index $i \leftarrow_R \mathcal{I}$, the universal hash function $\mathcal{H}_i: \mathcal{X} \to \mathcal{Y}$ can be employed as an average-case strong randomness extractor.

2.4 CCA security model

In the leakage-resilient security model, we model the adversary's leakage attacks on the private key SK_{id}, by giving the adversary access to a leakage oracle $\mathcal{O}_{SK_{id}}^{\lambda,k}()$, and the adversary can query to gain the leakage information about SK_{id}.

Definition 4 (Leakage oracle): A leakage oracle $\mathcal{O}_{SK_{id}}^{\lambda,k}()$ is parameterised by a private key SK_{id}, a leakage parameter λ and a security parameter k. A query to the leakage oracle consists of an efficiently computable leakage function $f_i: \{0,1\}^* \to \{0,1\}^{\lambda_i}$. The leakage oracle $\mathcal{O}_{SK_{id}}^{\lambda,k}()$ checks if the sum of λ_i, over all queries received so far, exceeds the leakage parameter λ and ignores the query if this is the case. The leakage oracle computes the function $f_i(SK_{id})$ for at most polynomial steps, and if the computation completes, responds with the output. Otherwise, it responds with the dummy value $\perp$. Without loss of generality, we can assume that the adversary can access the leakage oracle only once, and obtain at most λ bits leakage.

We require that an IBE scheme $\Pi = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ must remain the original security even in the leakage setting. Thus, in the leakage-resilient security model of the IBE scheme, the adversary can make an addition query other than the key generation queries and the decryption queries, called it leakage queries. That is, an adversary can obtain a certain amount of leakage on the private key of any identity chosen by itself, but the total length of leakage on the same identity must less than the leakage parameter λ. According to the previous works [7, 26, 30], our (continuous) leakage-resilient security definition of IBE scheme only allows leakage attacks against the private keys of the various identities, but not the master secret key. Just as noted by the authors [7, 26, 30, 33], we only allow the adversary to make leakage queries before seeing the challenge ciphertext. This is a necessary restriction as otherwise, the adversary could leak the first bit of the message and easily win the distinguishing game.

For any adversary $\mathcal{A}$, the experiment $\mathrm{Exp}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa)$ of LR-CCA security is described as follows:

$$(Params, S_{msk}) \leftarrow \mathtt{Setup}(1^\kappa)$$

(i) $(state, id^*, M_0, M_1) \leftarrow \mathcal{A}^{\mathcal{O}_{SK_{id}}^{\lambda,\kappa}(), \mathcal{O}^{\mathtt{KeyGen}}(), \mathcal{O}^{\mathtt{Dec}}()}(Params)$, where $|M_0| = |M_1|$ and $M_0, M_1 \in \mathcal{M}$.

(ii) $C_b^* \leftarrow \mathtt{Enc}(\mathtt{id}^*, \mathtt{M_b})$, where $b \leftarrow_R \{0,1\}$.

$$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\neq id^*}^{\mathtt{KeyGen}}(), \mathcal{O}_{\neq (id^*, C_b^*)}^{\mathtt{Dec}}()}(Params, state, C_b^*)$$

(iii) If $b' = b$ return 1. Otherwise, return 0.

We stress that, $\mathcal{M}$ denotes the plaintext space, $\mathcal{O}_{SK_{id}}^{\lambda,\kappa}()$ denotes the leakage oracle of the private key of user, $\mathcal{O}^{\mathtt{KeyGen}}()$ denotes the key generation oracle of any identity, $\mathcal{O}_{\neq id^*}^{\mathtt{KeyGen}}()$ denotes the key generation oracle excluding challenge identity id^*, $\mathcal{O}^{\mathtt{Dec}}()$ denotes the decryption oracle for any identity ciphertext pair, and $\mathcal{O}_{\neq (id^*, C_b^*)}^{\mathtt{Dec}}()$ denotes the decryption oracle excluding challenge identity and the challenge ciphertext pair (id^*, C_b^*).

Then, the advantage $\mathrm{Adv}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa)$ of the adversary $\mathcal{A}$ in the experiment $\mathrm{Exp}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa)$ is defined as

$$\mathrm{Adv}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa) = \left|\Pr[\mathrm{Exp}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa) = 1] - \frac{1}{2}\right|.$$

Definition 5: If the advantage $\mathrm{Adv}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa)$ of the adversary $\mathcal{A}$ in the above experiment $\mathrm{Exp}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa)$ is negligible, then, the corresponding IBE scheme is λ-LR-CCA secure. Also, in the leakage queries, the total length of leakage on the private key of the user must less than the leakage parameter λ.

In the abstract of literature [34], Dodis *et al.* stated that a cryptographic primitive is CLR, if it allows users to refresh their private key, using only fresh local randomness, such that

(i) 'The scheme remains functional after any number of key refreshes, although the public key never changes. Thus, the 'outside world' is neither affected by these key refreshes nor needs to know about their frequency. That is, if the public key is unchanged, then any PPT adversary cannot distinguish the original private key and the updated private key.'
(ii) 'The scheme remains secure even if the adversary can continuously leak arbitrary information about the current secret key, as long as the amount of leaked information is bounded between any two successive key refreshes. There is no bound on the total amount of information that can be leaked during the lifetime of the system. That is, the original cryptographic primitive has bounded leakage resilience. Then, the continuous leakage attacks can be resisted by performing the additional key update algorithm.'

From the above conclusions, we find that the problem of continuous leakage resilience can be reduced to a simpler single-round bounded leakage-resilient problem. That is, a bounded LR-IBE scheme with the key update function can achieve the continuous leakage resilience, if the corresponding key update algorithm with the re-randomisation property, i.e. any PPT adversary cannot distinguish the original private key and the updated private key. Therefore, a CLR-CCA secure IBE scheme can be achieved from an IBE scheme $\Pi = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ with CCA security and bounded leakage resilience by performing an additional key update algorithm Update. Thus, we have the formal definition of CLR-CCA security described in the following definition.

Definition 6: If the advantage $\mathrm{Adv}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa)$ of the adversary $\mathcal{A}$ in the experiment $\mathrm{Exp}_{\mathrm{IBE},\mathcal{A}}^{\mathrm{LR-CCA}}(\lambda,\kappa)$ is negligible, then, the corresponding IBE scheme with key update function is λ-CLR-CCA secure, where the public parameters and the function keep unchanged while the private key of the user is updated. Also, the total number of bits leaked between each update is bounded by a

fixed round leakage parameter λ, while the total amount of leakage throughout its lifetime is not restricted.

We stress that, in the (continuous) LR-CPA security model, the decryption queries for any identity ciphertext pair cannot be submitted by the adversary. That is, the leakage queries and decryption queries can only be performed. Due to the space limitation, the corresponding description is omitted.

3 CLR-CPA secure IBE

In the traditional bounded-leakage model, the total leakage of the private key is bounded by some small constant. However, in the real world, an adversary can continuously learn any additional information on the private key of the user through continuous leakage attacks, and can trivially break the security of cryptographic primitives. Thus, to further improve the practicability, an IBE scheme must maintain its claimed security in the continuous leakage setting. Based on the Dodis *et al.*'s conclusions [34], we need a novel method to push some new randomness into the private key of the user after the decryption operation. To this end, we show how to securely update the private key of the user while the public parameters keep unchanged. That is, we develop an additional key update algorithm Update, which can create a new private key of the user. Thus, the leakage of the previous private key of the user does not contribute to the new private key.

3.1 Constructions

An IBE scheme $\Pi = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Update})$ is constructed in this section, which is secure against continual leakage attacks on the private key of the user.

Setup: On input of the security parameter κ, the setup algorithm $(Params, S_{msk}) \leftarrow \mathtt{Setup}(1^{\kappa})$ is described as follows:

(i) Run the group sampling algorithm $\mathcal{G}(1^{\kappa})$ to obtain $\mathbb{G} = (p, G, G_T, e(\cdot, \cdot), P)$, where p is a prime, G and G_T are two groups of order p, P is a generator of G, and $e: G \times G \rightarrow G_T$ is an efficiently computable non-degenerate bilinear map.

(ii) Choose $\alpha \leftarrow_R \mathbb{Z}_p^*$ and $Q \leftarrow_R G$, and then compute $P_1 = \alpha P$.

(iii) Let $\mathtt{Ext}: \mathtt{G_T} \times \{0,1\}^{\mathtt{l_t}} \rightarrow \{0,1\}^{\mathtt{l_m}}$ be an advantage case $(\log p - \lambda, \varepsilon)$-strong randomness extractor, where λ denotes the leakage parameter and ε is a negligible advantage.

(iv) Let $S_{msk} = \alpha$ be the master secret key and set the public parameters $Params = \langle \mathbb{G}, P_1, Q, \mathtt{Ext} \rangle$ as the common input of the following algorithms. Also, let the identity space be $\mathcal{ID} = Z_p^*$ and the message space be $\mathcal{M} = \{0,1\}^{l_m}$.

Key generation: For any $id \in \mathcal{ID}$, the key generation algorithm $(SK_{id}, tk_{id}) \leftarrow \mathtt{KeyGen}(\mathtt{id}, \mathtt{S_{msk}})$ is described as follows:

(i) Choose $r \leftarrow_R \mathbb{Z}_p^*$, and compute

$$sk_{id,1} = \frac{1}{\alpha - id}(Q - rP) \text{ and } sk_{id,2} = r\,.$$

(ii) Output the private key $SK_{id} = (sk_{id,1}, sk_{id,2})$ associated with the identity id and an update trapdoor $tk_{id} = (P/\alpha - id)$, where tk_{id} is an update key, and is used in the key update algorithm Update. Notice that, id can make tk_{id} is different for each user.

Key update: The algorithm Update can push some new randomness into the private key of the user, and make the private key is random in the adversary's view so that the continual leakage attacks are possible to defend.

For any identity $id \in \mathcal{ID}$, as well as any update index $j \in \mathbb{N}$, the algorithm $SK_{id}^j \leftarrow \mathtt{Update}(\mathtt{SK_{id}^{j-1}}, \mathtt{tk_{id}})$ is described as follows:

(i) Choose $r_j \leftarrow_R \mathbb{Z}_p^*$, and set

$$sk_{id,1}^j = sk_{id,1}^{j-1} - r_j tk_{id} \text{ and } sk_{id,2}^j = sk_{id,2}^{j-1} + r_j\,.$$

Thus, for an update index $j \in \mathbb{Z}_p^*$, we have

$$sk_{id,1}^j = \frac{1}{\alpha - id}(Q - (r + \sum_{i=1}^{j} r_i)P) \text{ and } sk_{id,2}^j = r + \sum_{i=1}^{j} r_i\,.$$

Output the new private key $SK_{id}^j = (sk_{id,1}^j, sk_{id,2}^j)$ subject to the constraints $sk_{id,i}^j \neq sk_{id,i}^{j-1}$ (where $i = 1, 2$) and $\left|SK_{id}^j\right| = \left|SK_{id}^{j-1}\right|$.

Notice that, the new private key SK_{id}^j satisfies the re-randomisation property. That is, SK_{id}^j and SK_{id}^{j-1} are two independent and uniform distributions over the space of private key of the user, i.e. for any PPT adversary, we have

$$\mathtt{SD}(\mathtt{SK_{id}^j}, \mathtt{SK_{id}^{j-1}}) \leq \mathtt{negl}(\kappa)\,.$$

Encryption: For a plaintext $M \in \mathcal{M}$, the encryption algorithm $C \leftarrow \mathtt{Enc}(\mathtt{id}, \mathtt{M})$ is described as follows:

(i) Choose $t \leftarrow_R \mathbb{Z}_p^*$, and compute

$$c_1 = t(P_1 - idP) \text{ and } c_2 = e(P, P)^t\,.$$

(ii) Choose $S \leftarrow_R \{0,1\}^{l_t}$, and compute

$$c_3 = \mathtt{Ext}(\mathtt{e(P, Q)^t, S}) \oplus \mathtt{M}\,.$$

(iii) Output the ciphertext $C = (c_1, c_2, c_3, S)$ associated with the plaintext M.

Decryption: For a ciphertext $C = (c_1, c_2, c_3, S)$, the decryption algorithm $M \leftarrow \mathtt{Dec}(\mathtt{SK_{id}}, \mathtt{C})$ is described as follows:

(i) Output $M = \mathtt{Ext}(\omega, \mathtt{S}) \oplus \mathtt{c_3}$ as the plaintext of C, where $\omega = e(c_1, sk_{id,1})c_2^{sk_{id,2}}$.

(ii) Execute $SK'_{id} \leftarrow \mathtt{Update}(\mathtt{SK_{id}}, \mathtt{tk_{id}})$, and the new private key SK'_{id} will be used in the next round decryption operation.

3.2 Correctness

From the following equation, it is easy for us to see that the decryption algorithm is consistent with the encryption algorithm.

$$\begin{aligned} e(c_1, sk_{id,1})c_2^{sk_{id,2}} &= \mathrm{e}(t(P_1 - idP), \frac{1}{\alpha - id}(Q - rp))e(P, P)^{\mathrm{tr}} \\ &= \mathrm{e}(t(\alpha - id)P, \frac{1}{\alpha - id}(Q - rp))e(P, P)^{\mathrm{tr}} \\ &= \mathrm{e}(tP, Q)e(tP, -rP)e(P, P)^{\mathrm{tr}} \\ &= \mathrm{e}(P, Q)^t\,. \end{aligned}$$

3.3 Proof of security

Based on the Dodis *et al.*'s conclusions [34], we obtain that the CLR-IBE scheme can be obtained from an IBE scheme with the bounded leakage resilience by running an additional key update algorithm. Therefore, in this section, we only proof the bounded leakage-resilient property of our construction, and the continuous leakage resilience is naturally achieved through key update operation.

Theorem 1: Under the q-ABDHE assumption (where $q = q_k + 1$, the adversary submits at most q_k private key generation queries), for any leakage parameter $\lambda \leq 2\log p - l_m - \omega(\log \kappa)$, our construction Π is a CLR-IBE scheme with CPA security.

Proof: If there exists an adversary $\mathcal{A}$ who can break the CLR-CPA security of our construction Π with a non-negligible advantage $\mathtt{Adv}_{\mathcal{A},\Pi}^{\mathrm{LR-CCA}}(\kappa)$, then, we can build a simulator $\mathcal{S}$ who can solve the q-ABDHE assumption with an obvious advantage $\mathtt{Adv}_{\mathcal{S}}^{\mathrm{q-ABDHE}}(\kappa) \geq \mathtt{Adv}_{\mathcal{A},\Pi}^{\mathrm{LR-CCA}}(\kappa) - (1/\mathrm{p})$, where p is a prime order of the underlying group.

$\mathcal{S}$ takes as input a random truncated decision q-ABDHE challenge $\mathcal{T}_v = (P', P'_{q+2}, P, P_1, P_2, \ldots, P_p, T_v)$, where $P_i = \alpha^i P$ and $P'_i = \alpha^i P'$ for an unknown $\alpha \in \mathbb{Z}_p^*$, and T_v is either $e(P_{q+1}, P')$ or a random element of G_T. That is, in the beginning, $\mathcal{S}$ receives a challenge tuple $\mathcal{T}_v = (P', P'_{q+2}, P, P_1, P_2, \ldots, P_q, T_v)$ from the challenger of the q-ABDHE problem.

$\mathcal{S}$ simulates the LR-CPA security game for the adversary $\mathcal{A}$ as follows:

Setup: $\mathcal{S}$ generates random polynomials $f(x) \in \mathbb{Z}_p[x]$ of degree q, and sets $Q = f(\alpha)P$. It sends the public parameter $Params = (P, P_1 = \alpha P, Q = f(\alpha)P, \mathtt{Ext})$ to the adversary $\mathcal{A}$, where $\mathrm{Ext}: G_T \times \{0,1\}^{l_t} \to \{0,1\}^{l_m}$ is an average case strong randomness extractor, and the master secret key is impliedly set as $S_{msk} = \alpha$. Let $\mathcal{ID} = \mathbb{Z}_q^*$ be the identity space and $\mathcal{M} = \{0,1\}^{l_m}$ be the message space.

Notice that, since α is chosen uniformly at random, Q and P_1 are uniformly random and this system public parameter has a distribution identical to that in the actual construction.

Test Stage 1: In this stage, the following two kinds of queries are adaptively submitted by the adversary $\mathcal{A}$, and the query depends on the previous queries, as well as the corresponding responses.

Key generation queries: For the key generation queries of any identity $id \in \mathcal{ID}$, we consider the following two cases:

(i) If $id = \alpha$, $\mathcal{S}$ uses α to solve truncated decision q-ABDHE immediately.

(ii) Otherwise, let $F_{id}(x)$ denote the $(q-1)$-degree polynomial $(f(x) - f(id)/x - id)$. $\mathcal{S}$ outputs the corresponding private key $SK_{id_i} = (F_{id}(\alpha)P, f(id))$. This is a valid private key for the identity id, since

$$F_{id}(\alpha)P = \frac{f(\alpha) - f(id)}{\alpha - id}P = \frac{1}{\alpha - id}(f(\alpha)P - f(id)P)$$
$$= \frac{1}{\alpha - id}(Q - f(id)P).$$

Leakage queries: For the leakage queries of any identity $id \in \mathcal{ID}$, $\mathcal{S}$ operates the leakage oracle $\mathcal{O}^{\lambda,\kappa}_{SK_{id}}(\cdot)$ and returns the corresponding answers $f_i(SK_{id})$ by using the private key SK_{id}, where $f_i: \{0,1\}^* \to \{0,1\}^{\lambda_i}$ is an efficiently computable leakage function submitted by the adversary $\mathcal{A}$ and the private key SK_{id} can be created with the same method as the key generation queries. In the process of leakage queries, the total length of $f_i(SK_{id})$ which all returned from the leakage oracle $\mathcal{O}^{\lambda,\kappa}_{SK_{id}}(\cdot)$ on the same private key SK_{id} must be less than the leakage parameter λ. Otherwise, an invalid answer $\perp$ will be outputted.

Challenge stage: In this stage, the adversary $\mathcal{A}$ will submit a challenging identity $id^* \in \mathcal{ID}$ and two equal length challenge messages $M_0, M_1 \in \mathcal{M}$ to the simulator $\mathcal{S}$, where id^* never appeared in a key generation query and appeared in the leakage queries with at most λ bits leakage. If $id = \alpha$, $\mathcal{S}$ uses α to solve truncated decision q-ABDHE immediately. Otherwise, $\mathcal{S}$ computes $SK_{id^*} = (sk_{id^*,1}, sk_{id^*,2})$ as in the simulation of private key queries for id^*, and does the following operations:

(i) Let $f'(x) = x^{q+2}$, and let $F'_{id^*}(x) = (f'(x) - f'(id^*)/x - id^*)$, which is a polynomial of degree $q+1$. Let $F'_{id^*,i}$ denotes the coefficient of x^i in $F'_{id^*}(x)$.

(ii) Choose $b \leftarrow_R \{0,1\}$, and compute

$$c_1^* = (f'(\alpha) - f'(id^*))P', c_2^* = T_v e\left(P', \sum_{i=0}^{q} F'_{id^*,i}\alpha^i P\right),$$
$$c_3^* = \mathtt{Ext}\left(\mathtt{e}(c_1^*, \mathtt{sk}_{\mathtt{id}^*,1})c_2^{*\mathtt{sk}_{\mathtt{id}^*,2}}, \mathtt{S}^*\right) \oplus \mathtt{M_b}$$

where $S^* \leftarrow_R \{0,1\}^{l_t}$. That is, $\mathcal{S}$ computes the challenge ciphertext $C_b^* = (c_1^*, c_2^*, c_3^*, S^*)$ from $\left(P, P_1, P_2, \ldots, P_q\right)$, and without knowledge of α.

iii) Send $C_b^* = (c_1^*, c_2^*, c_3^*, S^*)$ to the adversary $\mathcal{A}$ as the challenge ciphertext.

Notice that, if there exists $\beta \in \mathbb{Z}_p^*$ such that $P' = \beta P$, then we let $t = \beta F'_{id^*}(\alpha)$. Now, we consider the following two cases:

(i) If $T_v = e(P_{p+1}, P')$, then $c_1^* = t(P_1 - id^*P)$, $c_2^* = e(P,P)^t$ and $c_3^* = \mathtt{Ext}(\mathtt{e}(\mathtt{P}, \mathtt{Q})^{\mathtt{t}}, \mathtt{S}^*) \oplus \mathtt{M_b}$ (where $S^* \leftarrow_R \{0,1\}^{l_t}$), since

$$c_1^* = (f'(\alpha) - f'(id^*))P' = \beta F'_{id^*}(\alpha)(\alpha - id^*)P$$
$$= \beta F'_{id^*}(\alpha)(\alpha P - id^* P) = t(P_1 - id^* P);$$

$$c_2^* = T_v e\left(P', \sum_{i=0}^{q} F'_{id^*,i}\alpha^i P\right) = e\left(P', \sum_{i=0}^{q} F'_{id^*,i}\alpha^i P\right) e(\alpha^{q+1}P, P')$$
$$= e\left(\beta P, \sum_{i=0}^{q+1} F'_{id^*,i}\alpha^i P\right) = e(P,P)^{\beta F'_{id^*}(\alpha)}$$
$$= e(P,P)^t;$$

$$c_3^* = \mathtt{Ext}\left(\mathtt{e}(c_1^*, \mathtt{sk}_{\mathtt{id}^*,1})c_2^{*\mathtt{sk}_{\mathtt{id}^*,2}}, \mathtt{S}^*\right) \oplus \mathtt{M_b}$$
$$= \mathtt{Ext}(\mathtt{e}(\mathtt{P}, \mathtt{Q})^{\mathtt{t}}, \mathtt{S}^*) \oplus \mathtt{M_b}.$$

Thus, the challenge ciphertext $C_b^* = (c_1^*, c_2^*, c_3^*, S^*)$ is a valid encryption ciphertext for id^* and M_b under randomness $t = \beta F'_{id^*}(\alpha)$. Since α and β are uniformly random, t is uniformly random, and so C_b^* is a valid, appropriately-distributed challenge ciphertext.

(ii) If $T_v \leftarrow_R G_T$, then (c_1^*, c_2^*) is a uniformly random and independent element of $G \times G_T$. In this case, the inequalities $c_2^* \neq e(c_1^*, P)^{(1/\alpha - id^*)}$ hold with probability $1 - (1/p)$. When this inequality holds, the value of $\mathrm{e}(c_1^*, sk_{id^*,1})c_2^{*sk\,id^*,2} = \mathrm{e}(c_1^*, (1/\alpha - id^*)(Q - f(id^*)P))c_2^{*\,f(id^*)}$ is uniformly random and independent from $\mathcal{A}$'s view, since $f(id^*)$ is uniformly random and independent from $\mathcal{A}$'s view. Thus, c_3^* is uniformly random and independent, and $C_b^* = (c_1^*, c_2^*, c_3^*, S^*)$ can impart no information regarding the bit b.

Test Stage 2: In this stage, the simulator $\mathcal{S}$ calculates the complete private key of any identity (except the challenge identity id^*) as he did in Test Stage 1. Furthermore, the leakage queries for any identity are unallowed.

Output: Eventually, $\mathcal{A}$ outputs a guess b' of the random value b picked by $\mathcal{S}$.

It is easy to see that the simulation is perfect, and the challenge ciphertext C_b^* is a valid encryption ciphertext of the message M_b if $T_v = \mathrm{e}(P_{p+1}, P')$. On the other hand, if $T_v \leftarrow_R G_T$, then C_b^* is a uniformly random message in the $\mathcal{A}$'s view, and gives no information about the simulator's choice of b, except the probability $(1/p)$.

To sum up, assuming that no queried identity equals α, for $T_0 \leftarrow_R G_T$ and $T_1 = e(P_{p+1}, P')$, we obtain that

$$\mathtt{Adv}^{\mathrm{q-ABDHE}}_{\mathcal{S},\mathcal{T}_0} = \left| \Pr\left[\mathcal{S}(P', P'_{q+2}, P, P_1, \ldots, P_q, T_0) = 0\right] - \frac{1}{2} \right|$$
$$\leq \frac{1}{p};$$

$$\mathtt{Adv}^{\mathrm{q-ABDHE}}_{\mathcal{S},\mathcal{T}_1} = \left| \Pr\left[\mathcal{S}(P', P'_{q+2}, P, P_1, \ldots, P_q, T_1) = 0\right] - \frac{1}{2} \right|$$
$$\geq \mathtt{Adv}^{\mathrm{LR-CCA}}_{\mathcal{A},\Pi}(\kappa).$$

Therefore, we can obtain that if there exists an adversary $\mathcal{A}$ who can break the LR-CPA security of our construction Π with a non-negligible advantage $\mathrm{Adv}_{\mathcal{A},\Pi}^{\mathrm{LR-CPA}}(\kappa)$, and then, we can build a simulator $\mathcal{S}$ who can break the security of decisional version of truncated q-ABDHE assumption with an obvious advantage $\mathrm{Adv}_{\mathcal{S}}^{\mathrm{q-ABDHE}}(\kappa)$, where

$$\begin{aligned}\mathrm{Adv}_{\mathcal{S}}^{\mathrm{q-ABDHE}}(\kappa) &= \mathrm{Adv}_{\mathcal{S},\mathcal{T}_1}^{\mathrm{q-ABDHE}} - \mathrm{Adv}_{\mathcal{S},\mathcal{T}_0}^{\mathrm{q-ABDHE}} \\ &\geq \mathrm{Adv}_{\mathcal{A},\Pi}^{\mathrm{LR-CCA}}(\kappa) - \frac{1}{\mathrm{p}}.\end{aligned}$$

That is, we prove that the advantage of $\mathcal{S}$ in breaking the decisional version of truncated q-ABDHE assumption is negligibly close to the advantage of $\mathcal{A}$ in the LR-CPA security game.

In the continuous leakage setting, the adversary cannot obtain information on the SK_{id^*} from the public parameter *Params*, the challenge plaintexts M_0, M_1, and the challenge identity id^*. Besides the knowledge previously, the adversary also obtains at most λ bits leakage Leak on the private key SK_{id^*}. By Lemma 1, we can obtain

$$\begin{aligned}\widetilde{H}_\infty(sk_{id^*,1}, sk_{id^*,2}|C_b^*, \mathrm{Leak}) &= \widetilde{H}_\infty(sk_{id^*,1}, sk_{id^*,2}|\mathrm{Leak}) \\ &\geq 2\log p - \lambda,\end{aligned}$$

where $sk_{id^*,1} \leftarrow_R G$ and $sk_{id^*,2} \leftarrow_R \mathbb{Z}_p^*$.

In practice, given public parameters *Params*, challenge identity id^*, challenge plaintext M_0, M_1, challenge ciphertext C_b^*, and λ bits leakage on the private key SK_{id^*}, the average min-entropy of the input variable $\mathrm{e}(c_1^*, sk_{id^*,1})c_2^{*sk\,id^*,2}$ of the randomness extractor $\mathrm{Ext}:\mathbb{G}\times\{0,1\}^{l_t}\to\{0,1\}^{l_m}$ is at least $2\log p - \lambda$. Also, $\mathrm{Ext}:\mathbb{G}\times\{0,1\}^{l_t}\to\{0,1\}^{l_m}$ is an average case $(\log p - \lambda, \epsilon)$ strong randomness extractor. Therefore, the average min-entropy of $e(c_1^*, sk_{id^*,1})c_2^{*sk\,id^*,2}$ satisfies the requirement of $\mathrm{Ext}:\mathbb{G}\times\{0,1\}^{l_t}\to\{0,1\}^{l_m}$.

By Lemma 3, we have $l_m \leq 2\log p - \lambda - 2\log(1/\epsilon)$. Taking into account that ϵ is negligible in the security parameter κ, i.e. $2\log(1/\epsilon) = \omega(\log\kappa)$, thus, we have $\lambda \leq 2\log p - l_m - \omega(\log\kappa)$.

Hence, our construction Π is a CLR-CPA secure IBE scheme for any leakage parameter $\lambda \leq 2\log p - l_m - \omega(\log\kappa)$. □

4 CLR-CCA secure IBE

In this section, to further improve the security level, based on our basic construction Π, we design a CLR-IBE scheme $\Pi' = (\mathrm{Setup}', \mathrm{KeyGen}', \mathrm{Enc}', \mathrm{Dec}', \mathrm{Update}')$ with adaptive CCA security.

4.1 Constructions

Setup: On input of the security parameter κ, the setup algorithm $(Params, S_{msk}) \leftarrow \mathrm{Setup}'(1^\kappa)$ is described as follows:

(i) Run the group sampling algorithm $\mathcal{G}(1^\kappa)$ to obtain $\mathbb{G} = (p, G, G_T, e(\cdot,\cdot), P)$.
(ii) Choose $\alpha \leftarrow_R \mathbb{Z}_p^*$ and $Q_1, Q_2 \leftarrow_R G$, and then compute $P_1 = \alpha P$.
(iii) Let $\mathrm{H}:\mathrm{G}\times\mathrm{G_T}\times\mathbb{Z}_\mathrm{p}^*\to\mathbb{Z}_\mathrm{p}^*$ be a universal one-way hash function.
(iv) Let $S_{msk} = \alpha$ be the master secret key and set the public parameters $Params = \langle\mathbb{G}, P_1, Q_1, Q_2, H\rangle$ as the common input of the following algorithms:

Key generation: For any $id \in \mathcal{ID}$, the key generation algorithm $(SK_{id}, tk_{id}) \leftarrow \mathrm{KeyGen}(\mathrm{id}, \mathrm{S_{msk}})$ is described as follows:

(i) Choose $r_1, r_2 \leftarrow_R \mathbb{Z}_p^*$, and compute

$$sk_{id,1} = \frac{1}{\alpha - id}(Q_1 - r_1 P), sk_{id,2} = r_1;$$

$$sk_{id,3} = \frac{1}{\alpha - id}(Q_2 - r_2 P), sk_{id,4} = r_2.$$

(ii) Output the private key $SK_{id} = (sk_{id,1}, sk_{id,2}, sk_{id,3}, sk_{id,4})$ associated with the identity id and an update key $tk_{id} = (P/\alpha - id)$.

Key update: For any identity $id \in \mathcal{ID}$, as well as any update index $j \in \mathbb{N}$, the algorithm $SK_{id}^j \leftarrow \mathrm{Update}'(\mathrm{SK_{id}^{j-1}}, \mathrm{tk_{id}})$ is described as follows:

(i) Choose $r_1^j, r_2^j \leftarrow_R \mathbb{Z}_p^*$, and set

$$sk_{id,1}^j = sk_{id,1}^{j-1} - r_1^j tk_{id}, \quad sk_{id,2}^j = sk_{id,2}^{j-1} + r_1^j,$$

$$sk_{id,3}^j = sk_{id,3}^{j-1} - r_2^j tk_{id}, \quad sk_{id,4}^j = sk_{id,4}^{j-1} + r_2^j.$$

Thus, for an update index $j \in \mathbb{Z}_p^*$, we have

$$sk_{id,1}^j = \frac{1}{\alpha - id}\left(Q_1 - \left(r_1 + \sum_{i=1}^{j} r_1^i\right)P\right), \quad sk_{id,2}^j = r_1 + \sum_{i=1}^{j} r_1^i,$$

$$sk_{id,3}^j = \frac{1}{\alpha - id}\left(Q_2 - \left(r_2 + \sum_{i=1}^{j} r_2^i\right)P\right), \quad sk_{id,4}^j = r_2 + \sum_{i=1}^{j} r_2^i.$$

Output the new private key $SK_{id}^j = (sk_{id,1}^j, sk_{id,2}^j, sk_{id,3}^j, sk_{id,4}^j)$ subject to the constraints $sk_{id,i}^j \neq sk_{id,i}^{j-1}$ (where $i = 1, 2, 3, 4$) and $\left|SK_{id}^j\right| = \left|SK_{id}^{j-1}\right|$.

Encryption: The encryption algorithm $C \leftarrow \mathrm{Enc}'(\mathrm{id}, \mathrm{M})$ is described as follows:

(i) Choose $t \leftarrow_R \mathbb{Z}_p^*$, and compute

$$c_1 = t(P_1 - idP) \text{ and } c_2 = e(P,P)^t.$$

(ii) Choose $n \leftarrow_R \mathbb{Z}_p^*$, and compute

$$c_3 = e(P,Q_1)^{-t}e(P,Q_2)^{-tn} \oplus M \text{ and } c_4 = e(P,Q_1)^{t\mu}e(P,Q_2)^t,$$

where $\mu = \mathrm{H}(\mathrm{c_1}, \mathrm{c_2}, \mathrm{c_3}, \mathrm{n})$.
iii) Output the ciphertext $C = (c_1, c_2, c_3, c_4, n)$ associated with the plaintext M.

Notice that encryption does not require any pairing computations once $e(P,Q_1)$, $e(P,Q_2)$ and $e(P,P)$ have been pre-computed. Alternatively, $e(P,Q_1)$, $e(P,Q_2)$ and $e(P,P)$ can be included in the system parameters, in which cases Q_1 and Q_2 can be dropped. Moreover, in the encryption algorithm, the random extract operation is implemented by a special universal hash function $\mathscr{H}_n(a,b) = ab^n$ as the average-case strong randomness extractor, where $n \in \mathbb{Z}_p^*$, $a = e(P,Q_1)^{-t}$ and $b = e(P,Q_2)^{-t}$. Similarly, the variable c_4 is generated through an implicit universal hash function. Namely, based on Lemma 3, for an index $n \in \mathbb{Z}_p^*$, the universal hash function $\mathscr{H}_n(a,b) = ab^n$ (Example 2 of Section 2.3) can be regarded as an average-case strong randomness extractor.

Decryption: The decryption algorithm $M \leftarrow \mathrm{Dec}'(\mathrm{SK_{id}}, \mathrm{C})$ is described as follows:

(i) Compute

$$\omega_1 = e(c_1, sk_{id,1})c_2^{sk_{id,2}} \text{ and } \omega_2 = e(c_1, sk_{id,3})c_2^{sk_{id,4}}.$$

(ii) Verify whether $c_4 = \omega_1^\mu\omega_2$, if the test fails, then the output is $\perp$; otherwise, output $M = (\omega_1\omega_2^n)c_3$ as the plaintext of C.
(iii) Execute $SK'_{id} \leftarrow \mathrm{Update}'(\mathrm{SK_{id}}, \mathrm{tk_{id}})$, and the new private key SK'_{id} will be used in the next round decryption operation.

4.2 Correctness

From the following equations, it is easy for us to see that the decryption algorithm is consistent with the encryption algorithm:

$$\begin{aligned}\omega_1 &= \mathrm{e}(c_1, sk_{id,1})c_2^{sk_{id,2}} \\ &= \mathrm{e}(t(\alpha - id)P, \frac{1}{\alpha - id}(Q_1 - r_1P))e(P,P)^{\mathrm{tr}_1} \\ &= \mathrm{e}(tP, Q_1)e(tP, -r_1P)e(P,P)^{\mathrm{tr}_1} \\ &= \mathrm{e}(P,Q_1)^t;\end{aligned}$$

$$\begin{aligned}\omega_2 &= e(c_1, sk_{id,3})c_2^{sk_{id,4}} \\ &= e(t(\alpha - id)P, \frac{1}{\alpha - id}(Q_2 - r_2P))e(P,P)^{\mathrm{tr}_2} \\ &= e(tP, Q_2)e(tP, -r_2P)e(P,P)^{\mathrm{tr}_2} \\ &= e(P,Q_2)^t.\end{aligned}$$

Similarly, for the updated private key of the user, our proposal is also correctness. Due to the space limitation, the corresponding description is omitted.

4.3 Proof of security

Theorem 2: Under the q-ABDHE assumption (where $q = q_k + 1$, any adversary submits at most q_k private key generation queries), for any leakage parameter $\lambda' \leq 2\log p - \omega(\log\kappa)$, our construction Π' is a CLR-CCA secure IBE scheme, where the leakage parameter is independent with the length of plaintext.

Proof: The proof is similar to Theorem 2 of Gentry's IBE scheme [31]. Also, Sun *et al.* [7] and Li *et al.* [30] proved the bounded LR-CCA security of their scheme based on this method. Due to the space limitation, the proof of this theorem is omitted, and the reader is referred to [7, 30, 31] for similar proof.

The round leakage parameter λ' is described as follows: By Theorem 1, we have

$$\begin{aligned}&\tilde{H}_\infty(sk_{id,1}, sk_{id,2}, sk_{id,3}, sk_{id,4} | Params, M_0, M_1, C_b, \mathtt{Leak}) \\ &= \tilde{H}_\infty(sk_{id,1}, sk_{id,2}, sk_{id,3}, sk_{id,4} | c_4, \mathtt{Leak}) \\ &\geq 3\log p - \lambda'.\end{aligned}$$

In the ciphertext $C_b = (c_1, c_2, c_3, c_4, n)$, the element c_4 can be written as a function on the private key of the user, and any adversary can obtain a certain amount of leakage on the private key of the user from the corresponding ciphertext.

By Lemma 3 and Theorem 1, we obtain

$$\lambda' \leq 2\log p - \omega(\log\kappa).$$

To sum up, for any round leakage parameter $\lambda \leq 2\log p - \omega(\log\kappa)$, our construction Π' is a CLR-IBE scheme with CCA security. □

4.4 Comparison

Now, we give two comparisons of our construction Π' with the previous works [6–8, 22, 23, 26, 30] in basic performance and computation efficiency. The basic performance is determined by the private key length ($\mathcal{SK}_{\mathrm{Len}}$), ciphertext length ($\mathcal{C}_{\mathrm{Len}}$), security assumption (*Assumption*), leakage model ($\mathcal{L}_{\mathrm{Model}}$), the upper bound of the bit-size of leakage allowed ($\mathcal{U}_\lambda$) and leakage ratio ($\mathcal{L}_{\mathrm{Ratio}}$, i.e. the size of permitted leakage/the size of the private key) which are listed in Table 1. We stress that the leakage ratio can be written as $\mathcal{U}_\lambda/\mathcal{SK}_{\mathrm{Len}}$. The computation efficiency is determined by the computational costs of the algorithms `Enc` and `Dec` which are listed in Table 2.

For presentation simplicity, we will call the corresponding IBE schemes proposed in [1, 6–8, 23, 26, 30] 'Chow-IBE' [26], 'Yuen-IBE' [23], 'Sun-IBE(a)' [7],'Sun-IBE(b)' [6],' Sun-IBE(c)' [8],'Li-IBE(a)' [30] and 'Li-IBE(b)' [1], respectively. Notice that, three LR-IBE schemes are designed in [26], denotes as 'Chow-IBE-I', 'Chow-IBE-II' and 'Chow-IBE-III', respectively. Two LR-IBE schemes are proposed in [8], denotes as 'Sun-IBE(c)-I' and 'Sun-IBE(c)-II', respectively.

Table 1 Comparison of basic parameters with previous works

Scheme	$\mathcal{SK}_{\mathrm{Len}}$	$\mathcal{C}_{\mathrm{Len}}$	*Assumption*
Chow-IBE-I	$2\|G\| + \|q\|$	$4\|G\|$	DBDH
Chow-IBE-II	$2\|G\| + \|q\|$	$4\|G\|$	DBDH
Sun-IBE(a)	$3\|q\| + 3\|G\|$	$4\|G\| + \|q\|$	q-ABDHE
Sun-IBE(c)-I	$3\|q\| + 3\|G\|$	$4\|G\| + \|q\|$	q-ABDHE
Sun-IBE(c)-II	$2\|q\| + 2\|G\|$	$3\|G\| + l_m + \|q\|$	q-ABDHE
Li-IBE(a)	$2\|q\| + 2\|G\|$	$3\|G\| + l_m + \|q\|$	q-ABDHE
our scheme Π'	$2\|q\| + 2\|G\|$	$4\|G\| + \|q\|$	q-ABDHE
scheme	$\mathcal{U}_\lambda$	$\mathcal{L}_{\mathrm{Ration}}$	$\mathcal{L}_{Model}$
Chow-IBE-I	$\log q - l_m - \omega(\log\kappa)$	1/3	BLM
Chow-IBE-II	$\log q - l_m - \omega(\log\kappa)$	1/3	BLM
Sun-IBE(a)	$\log q - \omega(\log\kappa)$	1/6	BLM
Sun-IBE(c)-I	$\log q - \omega(\log\kappa)$	1/6	BLM
Sun-IBE(c)-II	$\log q - l_m - \omega(\log\kappa)$	1/4	BLM
Li-IBE(a)	$\log q - l_m - \omega(\log\kappa)$	1/4	BLM
our scheme Π'	$2\log q - \omega(\log\kappa)$	½	CLM

(1) Let BLM be the bounded-leakage model, and CLM be the continuous-leakage model.

(2) Let $|G|$ denote the length of an element in prime order group G, $|q|$ the length of an element in Z_q^*, and l_m the length of plaintext.

Table 2 Comparison of computation efficiency with previous works

Scheme	Enc	Dec
Chow-IBE-I	$4E_s + 1E_{Ext}$	$1E_s + 2E_e + E_{Ext}$
Chow-IBE-II	$4E_s + 1E_{Ext}$	$1E_s + 2E_e + E_{Ext}$
Chow-IBE-III	$4E_s + 1E_{Ext}$	$1E_s + 2E_e + E_{Ext}$
Sun-IBE(a)	$2E_s + 3E_e$	$5E_d + 2E_e$
Sun-IBE(b)	$(n+1)E_s + 1E_d$	$1E_d + (n+2)E_e$
Sun-IBE(c)-I	$1E_s + 4E_d + 2E_e$	$4E_d + 2E_e$
Sun-IBE(c)-II	$2E_s + 2E_d + 1E_e + 1E_{Ext}$	$2E_d + 2E_e + E_{Ext}$
Li-IBE(a)	$3nE_s + 1E_d$	$2nE_d + E_e$
Li-IBE(b)	$2E_d + 2E_e + 1E_{Ext}$	$3E_d + 2E_e + 1E_{Ext}$
our scheme Π'	$2E_s + 2E_d$	$2E_s + 2E_d + 2E_e$

(1) Let E_{Ext} denote the cost of the randomness extractor operation, E_s the cost of single exponentiation (or multiplication) operation, E_d the cost of double exponentiation (or multiplication) operation, and E_e the cost of the pairing operation.

(2) $n$is a system parameter of the corresponding scheme.

Performance analysis: These constructions [6, 21–23] are proposed based on the composite order bilinear group. Thus, the basic performance and computation efficiency are not the advantages of these IBE schemes, and we mainly compare the security with these constructions. These IBE schemes proposed in [6, 21–23] only achieve CPA security, even if Yuen-IBE can resist the continuous leakage attacks. Our proposal not only can resist the continuous leakage attacks on the private key of user but also can achieve adaptive CCA security.

Table 1 shows that the leakage rate of our construction is improved compared with Chow-IBE, Alwen-IBE and Sun-IBE(a,b,c). Also, Chow-IBE and Yuen-IBE only achieve CPA security. Also, these constructions only achieve bigger leakage by reducing the length of the plaintext message. That is, on the same leakage parameter, Sun-IBE(c)-II and our construction are practical than other schemes, because of the leakage parameter λ is independent of the plaintext space. Therefore, when the length of key leakage approaching to $\mathcal{U}_\lambda$, Sun-IBE(a), Sun-IBE(c)-I and our construction are more efficient, as in this case, the size of plaintext in other constructions approaches to 0.

Efficiency analysis: In Table 2, when evaluating the computation efficiency, the hash function and XOR operations are

omitted. From Table 2, we obtain that our proposal Π' has comparable computational efficiency with the schemes of the other, but our construction with better performance than these schemes.

5 Continuous leakage-amplified IBE

We stress that, in the previous constructions, the upper bound *con* of leakage parameter is fixed, where the *con* is a constant value. In the leakage setting, if the length of bits leaked is bigger than a *con*, then, the corresponding construction will not be valid. However, in real life, for the different actual application environments, the size of allowed leakage is diverse. Hence, how to create a cryptosystem with a variable leakage bound is an important topic of the (continuous) leakage-resilient cryptography. In this section, we propose a CLR-CPA secure IBE scheme with leakage amplification, in which, the leakage parameter has arbitrary length. Namely, the upper bound of round leakage parameter can be flexibly controlled and is determined by the practical considerations about how much leakage the cryptography scheme needs to tolerate.

5.1 Construction

Based on our basic CLR-IBE scheme Π, an improved CLR-IBE scheme $\Pi'' = (\mathtt{Setup''}, \mathtt{KeyGen''}, \mathtt{Update''}, \mathtt{Enc''}, \mathtt{Dec''})$ is described as follows:

Setup. The setup algorithm $(Params, S_{msk}) \leftarrow \mathtt{Setup''}(1^{\kappa})$ is described as follows:

(i) Run the group sampling algorithm $\mathcal{G}(1^{\kappa})$ to obtain $\mathbb{G} = (p, G, G_T, e(\cdot\,,\cdot\,), P)$.
(ii) Choose $l_k \leftarrow_R \mathbb{Z}_q^*$ as the key-size parameter, where l_k is determined by the practical considerations about how much leakage the cryptosystem needs to tolerate.
(iii) Choose $\alpha \leftarrow_R \mathbb{Z}_p^*$ and $Q \leftarrow_R G$, and then compute $P_1 = \alpha P$.
(iv) Let $S_{msk} = \alpha$ be the master secret key and set the public parameters $Params = \langle \mathbb{G}, l_k, P_1, Q, \mathtt{Ext} \rangle$ as the common input of the following algorithms, where $\mathtt{Ext}: \mathtt{G_T} \times \{0,1\}^{\mathtt{l_t}} \rightarrow \{0,1\}^{\mathtt{l_m}}$ is an advantage case $(\log p - \lambda, \varepsilon)$-strong randomness extractor.
Key generation. The key generation algorithm $(SK_{id}, tk_{id}) \leftarrow \mathtt{KeyGen''}(\mathtt{msk}, \mathtt{id})$ is described as follows:

(i) For i from 1 to l_k, compute

$$sk_{id,i} = (s_{id,i}, k_{id,i}) = \left(\frac{1}{\alpha - id}(Q - r_i P), r_i\right),$$

where $r_i \leftarrow_R \mathbb{Z}_p^*$. From now, let i be the location index, and j be the update index.
ii) Output the private key $SK_{id} = (sk_{id,1}, \ldots, sk_{id,l_k})$ associated with the identity id and an update key $tk_{id} = (P/\alpha - id)$.
Key update: For the private key SK_{id}^{j-1} of the user id, the algorithm $SK_{id}^{j} = \mathtt{Update''}(\mathtt{SK_{id}^{j-1}}, \mathtt{tk_{id}})$ is described as follows:

(i) For i from 1 to l_k, compute

$$sk_{id,i}^{j} = \left(s_{id,i}^{j}, k_{id,i}^{j}\right) = \left(s_{id,i}^{j-1} - r_i^j tk_{id}, k_{id,i}^{j} + r_i^j\right).$$

Thus, for an update index $j \in \mathbb{Z}_p^*$, we have

$$\begin{aligned} sk_{id,i}^{j} &= \left(s_{id,i}^{j}, k_{id,i}^{j}\right) \\ &= \left(\frac{1}{\alpha - id}\left(Q - \left(r_i + \sum_{i=1}^{j} r_i^j\right)P\right), r_i + \sum_{i=1}^{j} r_i^j\right). \end{aligned}$$

(ii) Set $SK_{id}^{j} = (sk_{id,1}^{j}, \ldots, sk_{id,l_k}^{j})$, and output the new private key SK_{id}^{j} subject to the constraints $SK_{id}^{j} \neq SK_{id}^{j-1}$ and $\left|SK_{id}^{j}\right| = \left|SK_{id}^{j-1}\right|$.

Encryption: The encryption algorithm $C \leftarrow \mathtt{Enc''}(\mathtt{id}, \mathtt{M})$ is described as follows:

(i) Choose $t \leftarrow_R \mathbb{Z}_p^*$, and compute

$$c_1 = t(P_1 - idP) \text{ and } c_2 = e(P, P)^t.$$

(ii) Choose $S \leftarrow_R \{0,1\}^{l_t}$, and compute

$$c_3 = \mathtt{Ext}(\mathtt{e(P, Q)^{l_k t}, S}) \oplus \mathtt{M}.$$

(iii) Output the ciphertext $C = (c_1, c_2, c_3, S)$ associated with the plaintext M.
Decryption: For the private key $SK_{id} = \left(sk_{id,1}, \ldots, sk_{id,l_k}\right)$ of the user and the ciphertext $C = (c_1, c_2, c_3, S)$, the decryption algorithm $M = \mathtt{Dec''}(\mathtt{SK_{id}}, \mathtt{C})$ is described as follows:
(1) Compute $s_{id} = \sum_{i=1}^{l_k} s_{id,i} = (1/\alpha - id)\left(l_k Q - \sum_{i=1}^{l_k} r_i P\right)$ and $k_{id} = \sum_{i=1}^{l_k} k_{id,i} = \sum_{i=1}^{l_k} r_i$.
(2) Output $M = \mathtt{Ext(e(c_1, s_{id})c_2^{k_{id}}, S)} \oplus \mathtt{c_3}$ as the plaintext of C.
(3) Execute $SK'_{id} \leftarrow \mathtt{Update''}(\mathtt{SK_{id}}, \mathtt{tk_{id}})$, and the new private key SK'_{id} will be used in the next round decryption operation.

5.2 Correctness

The correctness is obtained from the following equations.

$$\begin{aligned} e(c_1, s_{id})c_2^{k_{id}} &= e\left(t(\alpha - id)P, \frac{1}{\alpha - id}\left(l_k Q - \sum_{i=1}^{l_k} r_i P\right)\right) e(P, P)^{t\sum_{i=1}^{l_k} r_i} \\ &= e(tP, l_k Q) e\left(tP, -\sum_{i=1}^{l_k} r_i P\right) e(P, P)^{t\sum_{i=1}^{l_k} r_i} \\ &= e(P, Q)^{l_k t}. \end{aligned}$$

5.3 Proof of security

Theorem 3: Under the q-ABDHE assumption (where $q = q_k + 1$, the adversary submits at most q_k private key generation queries), for any leakage parameter $\lambda' \leq 2l_k \log p - l_m - \omega(\log \kappa)$, our construction Π' is a continuous leakage-amplified IBE scheme with CPA security, where the length of bits leaked can be flexibly controlled by changing the key-size parameter l_k.

Proof: The proof is similar to Theorem 1, due to the space limitation; the proof of this theorem is omitted. The round leakage parameter λ'' is described as follows:
By Theorem 1, we have

$$\begin{aligned} &\tilde{H}_{\infty}(sk_{id,1}, \ldots, sk_{id,l_k} \left| Params, M_0, M_1, C_b, \mathtt{Leak}\right) \\ &= \tilde{H}_{\infty}(sk_{id,1}, \ldots, sk_{id,l_k} \left| \mathtt{Leak}\right) \\ &= \tilde{H}_{\infty}(s_1, k_1, \ldots, s_{l_k}, t_{l_k} \left| \mathtt{Leak}\right) \\ &\geq 2l_k \log p - \lambda''. \end{aligned}$$

By Lemma 3 and Theorem 1, we obtain

$$\lambda'' \leq 2l_k \log p - l_m - \omega(\log \kappa).$$

Specially, the striking advantage of our construction Π'' is that the size of the leakage parameter λ'' can be flexibly controlled by changing the length of the private key. That is, in real life, we can adaptively change the size of the private key to meet the leakage parameter with variable size. □
Similarly, we can design a continuous leakage-amplified IBE scheme with CCA security based on our IBE scheme Π'. Due to the space limitation, concrete construction is omitted.

6 Conclusion

In this paper, we have introduced a new way of constructing a more practical CCA secure CLR-IBE scheme. We gave a concrete construction and proved its security based on the hardness of q-ABDHE assumption in the standard model. Our proposal not only achieves adaptive CCA security but also enjoys better performances. To further tolerate the bigger leakage, new construction of continuous leakage-amplified IBE scheme is proposed, and the striking advantage of our new construction is that the size of round leakage parameter can be flexibly controlled by changing the leakage-size parameter.

In this paper, the CLR-IBE scheme is proposed based on a non-static assumption, which is a stronger security assumption that depends on the number of private key generation queries made by the adversary. Thus, in the next research stage, to further obtain the better performance, we will design the practical CCA secure CLR-IBE scheme under the simple security assumption, such as decisional bilinear Diffie–Hellman (DBDH) assumption. Also, the continuous leakage resilience of the master secret key will be considered.

7 Acknowledgment

The authors thank the anonymous reviewer for your helpful comments. This work was supported by the National Key R&D Program of China (no. 2017YFB0802000), the National Natural Science Foundation of China (61802242, 61572303, 61772326, 61802241, 61872087, 61602290), the Natural Science Basic Research Plan in Shaanxi Province of China (2018JQ6088, 2017JQ6038), the National Cryptography Development Foundation during the 13th Five-year Plan Period (MMJJ20180217), the Foundation of State Key Laboratory of Information Security (2017-MS-03) and the Fundamental Research Funds for the Central Universities (GK201803064).

8 References

[1] Li, J., Guo, Y., Yu, Q., *et al.*: 'Provably secure identity-based encryption resilient to post-challenge continuous auxiliary input leakage', *Secur. Commun. Netw.*, 2016, **9**, (10), pp. 1016–1024

[2] Liu, S, Weng, J, Zhao, Y.: 'Efficient public key cryptosystem resilient to key leakage chosen ciphertext attacks'. Topics in Cryptology – CT-RSA 2013, Berlin Heidelberg, 2013, pp. 84–100

[3] Qin, B., Liu, S.: 'Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter'. Advances in Cryptology – ASIACRYPT 2013, Bengaluru, India, 1–5 December 2013, pp. 381–400

[4] Qin, B., Liu, S.: 'Leakage-flexible cca-secure public-key encryption: simple construction and free of pairing'. Public-Key Cryptography – PKC 2014, Buenos Aires, Argentina, 26–28 March 2014, pp. 19–36

[5] Qin, B., Liu, S., Chen, K.: 'Efficient chosen-ciphertext secure public-key encryption scheme with high leakage-resilience', *IET Inf. Sec.*, 2015, **9**, (1), pp. 32–42

[6] Sun, S., Gu, D., Huang, Z.: 'Fully secure wicked identity-based encryption against key leakage attacks', *Comput. J.*, 2015, **58**, (10), pp. 2520–2536

[7] Sun, S., Gu, D., Liu, S.: 'Efficient leakage-resilient identity-based encryption with CCA security'. Pairing-Based Cryptography – Pairing 2013–6th Int. Conf., Beijing, China, 22–24 November 2013, pp. 149–167

[8] Sun, S., Gu, D., Liu, S.: 'Efficient chosen ciphertext secure identity-based encryption against key leakage attacks', *Secur. Commun. Netw.*, 2016, **9**, (11), pp. 1417–1434

[9] Wang, B.: 'Leakage-resilient message authentication code scheme based on hidden identity weak hash proof system', *IET Inf. Sec.*, 2016, **10**, (4), pp. 173–179

[10] Yu, Q., Li, J., Zhang, Y.: 'Leakage-resilient certificate-based encryption', *Secur. Commun. Netw.*, 2015, **8**, (18), pp. 3346–3355

[11] Yu, Q., Li, J., Zhang, Y., *et al.*: 'Certificate-based encryption resilient to key leakage', *J. Syst. Softw.*, 2016, **116**, pp. 101–112

[12] Zhang, M., Wang, C., Morozov, K.: 'LR-FEAD: leakage-tolerating and attribute-hiding functional encryption mechanism with delegation in affine subspaces', *J. Supercomput.*, 2014, **70**, (3), pp. 1405–1432

[13] Zhou, Y., Yang, B., Zhang, W.: 'Provably secure and efficient leakage-resilient certificateless signcryption scheme without bilinear pairing', *Discrete Appl. Math.*, 2016, **204**, pp. 185–202

[14] Fujisaki, E., Xagawa, K.: 'Public-key cryptosystems resilient to continuous tampering and leakage of arbitrary functions'. Advances in Cryptology – ASIACRYPT 2016, Hanoi, Vietnam, December 4–8 2016, pp. 908–938

[15] Zhou, Y., Yang, B.: 'Continuous leakage-resilient public-key encryption scheme with CCA security', *Comput. J.*, 2017, **60**, (8), pp. 1161–1172

[16] Zhou, Y., Yang, B., Zhang, W., *et al.*: 'CCA2 secure public-key encryption scheme tolerating continual leakage attacks', *Secur. Commun. Netw.*, 2016, **9**, (17), pp. 4505–4519

[17] Toorani, M.: 'On continuous after-the-fact leakage-resilient key exchange'. Proc. of the Second Workshop on Cryptography and Security in Computing Systems, CS2@HiPEAC 2015, Amsterdam, Netherlands, 19–21 January 2015, pp. 31–34

[18] Malkin, T., Teranishi, I., Vahlis, Y., *et al.*: 'Signatures resilient to continual leakage on memory and computation'. Theory of Cryptography – 8th Theory of Cryptography Conf., TCC 2011, Providence, RI, USA, 28–30 March 2011, pp. 89–106

[19] Li, J., Guo, Y., Yu, Q., *et al.*: 'Continuous leakage-resilient certificate-based encryption', *Inf. Sci.*, 2016, **355–356**, pp. 1–14

[20] Zhou, Y., Yang, B.: 'Continuous leakage-resilient certificateless public key encryption with CCA security', *Knowl.-Based Syst.*, 2017, **136**, pp. 27–36

[21] Lewko, A.B., Rouselakis, Y., Waters, B.: 'Achieving leakage resilience through dual system encryption'. Theory of Cryptography – 8th Theory of Cryptography Conf., TCC 2011, Providence, RI, USA, 28–30 March 2011 pp. 70–88

[22] Li, J., Yu, Q., Zhang, Y.: 'Identity-based broadcast encryption with continuous leakage resilience', *Inf. Sci.*, 2018, **429**, (3), pp. 177–193

[23] Yuen, T.H., Chow, S.S.M., Zhang, Y., *et al.*: 'Identity-based encryption resilient to continual auxiliary leakage'. Advances in Cryptology – EUROCRYPT 2012, 2012, pp. 117–134

[24] Zhou, Y., Yang, B., Mu, Y.: 'Continuous leakage-resilient identity-based encryption without random oracles', *Comput. J.*, 2018, **61**, (4), pp. 586–600

[25] Alwen, J, Dodis, Y, Naor, M, *et al.*: 'Public-key encryption in the bounded-retrieval model'. Advances in Cryptology – EUROCRYPT 2010, 2010, pp. 113–134

[26] Chow, S.S.M., Dodis, Y., Rouselakis, Y., *et al.*: 'Practical leakage-resilient identity-based encryption from simple assumptions'. Proc. of the 17th ACM Conf. on Computer and Communications Security ACM, 2010, pp. 152–161

[27] Boneh, D., Boyen, X.: 'Efficient selective-id secure identity-based encryption without random oracles'. Advances in Cryptology – EUROCRYPT 2004, 2004, pp. 223–238

[28] Lewko, A., Waters, B.: 'New techniques for dual system encryption and fully secure hibe with short ciphertexts'. Theory of Cryptography, 2010, pp. 455–479

[29] Waters, B.: 'Efficient identity-based encryption without random oracles'. Advances in Cryptology – EUROCRYPT 2005, 2005, pp. 114–127

[30] Li, J., Teng, M., Zhang, Y., *et al.*: 'A leakage-resilient cca-secure identity-based encryption scheme', *Comput. J.*, 2016, **59**, (7), pp. 1066–1075

[31] Gentry, C.: 'Practical identity-based encryption without random oracles'. Advances in Cryptology – EUROCRYPT 2006, 2006, pp. 445–464

[32] Dodis, Y, Reyzin, L., Smith, A.: 'Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data'. Advances in Cryptology – EUROCRYPT 2004, Interlaken, Switzerland, May 2004, pp. 523–540

[33] Naor, M, Segev, G.: 'Public-key cryptosystems resilient to key leakage', *SIAM J. Comput.*, 2012, **41**, (4), pp. 772–814

[34] Dodis, Y., Haralambiev, K., López-Alt, A., *et al.*: 'Cryptography against continuous memory attacks'. 51th Annual IEEE Symp. on Foundations of Computer Science, FOCS 2010, Las Vegas, Nevada, USA, 23–26 October 2010, pp. 511–520