CCA-Secure Leakage-Resilient Identity-Based Key-Encapsulation from Simple (not q-type) Assumptions*

Toi Tomita[1], Wakaha Ogata[1], Kaoru Kurosawa[2], Ryo Kuwayama[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
[2] Ibaraki University, Ibaraki, Japan

Abstract

In this paper, we propose a new leakage-resilient identity-based encryption (IBE) scheme that is secure against chosen-ciphertext attacks (CCA) in the bounded memory leakage model. It is the first CCA-secure leakage-resilient IBE scheme which does not depend on q-type assumptions. More precisely, it is secure under the external k-linear assumption. The leakage rate 1/10 is achieved under the XDLIN assumption ($k = 2$).

1 Introduction

1.1 Background.

Most of the encryption schemes known so far have been proven secure by assuming that the secret key is completely hidden. However, in the real world, a partial information of the secret key may leak by side-channel attacks [19, 13, 5] or the cold-boot attack [14]. In recent years, extensive research effort has been invested in providing encryption schemes which are provably secure even in this setting. Such schemes are said to be leakage-resilient.

Akavia et al. [1] introduced the bounded memory leakage model in which a bounded amount of information of the secret key is leaked to the adversary. Naor and Segev [24] showed how to construct leakage-resilient public-key encryption schemes from hash proof systems (HPS) in this model. (Other constructions were given by [17, 3, 21].) Qin et al. [26] showed a generic method to construct a CCA-secure leakage-resilient encryption scheme from a tag-based strongly universal$_2$ HPS.

Regarding identity-based encryption (IBE) schemes, CPA-secure leakage-resilient IBE schemes were shown by Akavia et al. [1], Alwen et al. [2] and

*This is an extended version of [29].

Chow et al. [7]. Finally Kurosawa and Phong [20] showed an IBE scheme which achieves the optimum leakage rate $1 - o(1)$ under the DLIN assumption, where the leakage rate is defined as

$$\frac{\text{size of leakage}}{\text{size of secret key}}.$$

On the other hand, CCA-secure leakage-resilient IBE schemes were shown by Alwen et al. [2], Sun et al. [27] and Li et al. [22]. Unfortunately, all these CCA-secure leakage-resilient IBE schemes rely on q-type assumptions. Due to the Cheon attack [6], it is better to avoid such assumptions.

1.2 Our Contribution.

In this paper, we propose the first CCA-secure leakage-resilient IBE scheme which does not depend on q-type assumptions. Namely, the proposed scheme is secure under external k-linear assumption The smaller k is, the better efficiency and the larger leakage rate are obtained while the assumption is stronger. The leakage rate 1/10 is achieved under the external 2-linear assumption (XDLIN assumption)

In fact, we construct a CCA-secure leakage-resilient IB-KEM. A CCA-secure leakage-resilient IBE scheme is obtained by combining our IB-KEM with any CCA-secure symmetric-key encryption scheme (which does not need to be leakage-resilient). Our IB-KEM scheme is obtained by applying the technique of Qin et al. [26] to the CPA-secure leakage-resilient IBE scheme of Kurosawa and Phong [20].

Table 1 shows a comparison of CCA-secure leakage-resilient IBE schemes.

Table 1: Comparison of CCA-secure leakage-resilient IBE schemes

Schemes	Assumption	Leakage rate
Alwen et al. [2]	q-type	1/6
Sun et al. [27]	q-type	1/6
Li et al. [22]	q-type	1/4
Ours (KEM)	external k-LIN	$1/(4k+2)$
$(k = 2)$	XDLIN	1/10

1.3 Various Models for Leakage-resilient.

Several researchers consider some variants of leakage models to capture practical issues. We summarize some leakage models below.

Micali and Reyzin [23] considered the "only computation leak information" model to deal with physical observation via side-channel attacks. However, this model could not capture key leakage attacks, such as a cold-boot attack. To capture key leakage attacks, Akavia et al. [1] proposed the bounded memory leakage

model, in which an adversary can get partial information on secret keys. Brakerski et al. [4] and Dodis et al. [8] presented a new model called continual memory leakage model, which allows leakage on the private key in many periods of time. In this model, the secret key is updated over time and the total leakage over the lifetime of the system is unbounded. Dodis et al. [9] invented the auxiliary input model, in which the entire secret could be leaked information-theoretically, provided that it is computationally infeasible to compute the secret.

All these leakage models only consider leakage occurring before the challenge ciphertext is given to the adversary. In response to this, Halevi and Lin [15] proposed the after-the-fact leakage model, in which an adversary can obtain leaked information after seeing the challenge ciphertext.

1.4 Organization

The rest of the paper is organized as follows. Sec. 2 introduces notations, some building blocks, and computational assumptions. Sec. 3 describes the definition of IB-KEM and the leakage-resilient CCA-security. We present the concrete construction of our CCA-secure leakage-resilient IB-KEM scheme in Sec. 4 and its security proof in Sec. 5. Finally, the conclusion of this paper is given in Sec. 6.

2 Preliminaries

2.1 Notations

We introduce some notations used in this paper. Let $\lambda \in \mathbb{N}$ denote the security parameter. We say that a function $f(\lambda)$ is negligible in λ if it is smaller than all polynomial fractions for a sufficiently large λ. For a finite set $\mathcal{S}$, we use $s \leftarrow_{\$} \mathcal{S}$ to denote the process of sampling an element s from $\mathcal{S}$ uniformly at random and let $|\mathcal{S}|$ denote its cardinality.

Let GGen be a probabilistic polynomial time (PPT) algorithm that on input the security parameter 1^λ returns a description $params = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, g_1, g_2, e)$ of pairing groups, where $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic groups of a prime order q, g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable (non-degenerated) bilinear map. Define $g_T := e(g_1, g_2)$, which is a generator of $\mathbb{G}_T$.

We refer to [28] for a description of types of bilinear groups. There are three types of bilinear groups according to whether efficient isomorphisms exist or not between $\mathbb{G}_1$ and $\mathbb{G}_2$ [12]. In type 1, both the isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ and its inverse $\psi^{-1} : \mathbb{G}_1 \to \mathbb{G}_2$ are efficiently computable, i.e., it can be regarded as $\mathbb{G}_1 = \mathbb{G}_2$. In type 2, the isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ is efficiently computable but its inverse is not. In type 3, there are no efficient isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$. Type 1 pairing groups are called symmetric, and type 2 and 3 pairing groups are called asymmetric. We assume type 3 pairing groups in our

scheme, but our scheme also works in type 1 and 2 setting under appropriate computational assumptions.

We use implicit representation of group elements as introduced in [11]. For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_q$ we define $[a]_s := g_s^a \in \mathbb{G}_s$ as the implicit representation of a in $\mathbb{G}_s$. Similarly, for a matrix

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & \dots & a_{1,m} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,m} \end{pmatrix} \in \mathbb{Z}_q^{n \times m}$$

we define

$$[\mathbf{A}]_s := \begin{pmatrix} g_s^{a_{1,1}} & \dots & g_s^{a_{1,m}} \\ \vdots & \ddots & \vdots \\ g_s^{a_{n,1}} & \dots & g_s^{a_{n,m}} \end{pmatrix} \in \mathbb{G}_s^{n \times m}$$

as the implicit representation of $\mathbf{A}$ in $\mathbb{G}_s$. Note that it is easy to compute $[\mathbf{AB}]_s$ given $([\mathbf{A}]_s, \mathbf{B})$ or $(\mathbf{A}, [\mathbf{B}]_s)$ with appropriate dimensions. We define $[\mathbf{A}]_1 \circ [\mathbf{B}]_2 := e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{AB}]_T$ that can be efficiently computed given $[\mathbf{A}]_1$ and $[\mathbf{B}]_2$.

2.2 External k-Linear Assumption

We assume the following property.

Definition 1 (External k-Linear Assumption: external k-LIN) *Let $s \in \{1, 2\}$, and $k \in \{1, 2, \dots, \}$. We say that the external k-LIN assumption holds relative to GGen in group $\mathbb{G}_s$ if for any PPT adversary D, the following is negligible in λ:*

$$\mathsf{Adv}^{\mathrm{xklin}}_{\mathsf{GGen},\mathsf{D}}(\lambda) :=$$
$$\left|\Pr[\mathsf{D}(\mathit{params}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{A}^\top \mathbf{r}]_s) = 1] - \Pr[\mathsf{D}(\mathit{params}, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{y}]_s) = 1]\right|,$$

where $\mathit{params} \leftarrow \mathsf{GGen}(1^\lambda), a_1, \dots, a_k \leftarrow_\$ \mathbb{Z}_q$, $\mathbf{r} \leftarrow_\$ \mathbb{Z}_q^k$, $\mathbf{y} \leftarrow_\$ \mathbb{Z}_q^{k+1}$, and

$$\mathbf{A} := \begin{pmatrix} a_1 & \cdots & \mathbf{O} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{O} & \cdots & a_k & 1 \end{pmatrix}.$$

The external 2-linear assumption is also called the XDLIN assumption. Note that the external 1-linear assumption does not hold.

2.3 Statistical Distance, Min-entropy and Randomness Extractor

The statistical distance between random variables X, Y over a finite domain $\mathcal{S}$ is defined by

$$\Delta(X, Y) := \frac{1}{2} \sum_{s \in \mathcal{S}} |\Pr[X = s] - \Pr[Y = s]|.$$

The min-entropy of X is defined by

$$H_\infty(X) \coloneqq -\log_2 \left(\max_x \Pr[X = x] \right).$$

Furthermore, average min-entropy of X conditioned on Y is defined by

$$\tilde{H}_\infty(X \mid Y) \coloneqq -\log_2 \left(\sum_y 2^{-H_\infty(X|Y=y)} \Pr[Y = y] \right),$$

as defined in [10], which also proved the following lemma.

Lemma 1 ([10, Lemma 2.2]) *Let ℓ be a positive integer. Let X, Y and Z be random variables. If Y has at most 2^ℓ possible values, then*

$$\tilde{H}_\infty(X \mid Y, Z) \geq \tilde{H}_\infty(X, Y \mid Z) - \ell \geq \tilde{H}_\infty(X \mid Z) - \ell.$$

One of main tools in our construction is a randomness extractor [10].

Definition 2 (Randomness Extractor) *Let n be a positive integer, and $\phi > n, \epsilon_{\mathsf{Ext}}$ be positive reals, and $\mathcal{D}, \mathcal{S}$ be finite sets. A function $\mathsf{Ext} : \mathcal{D} \times \mathcal{S} \to \{0,1\}^n$ is called a $(\phi, \epsilon_{\mathsf{Ext}})$-randomness extractor if for all pairs of random variables (X, I) such that X is a random variable over $\mathcal{D}$ satisfying $\tilde{H}_\infty(X \mid I) \geq \phi$,*

$$\Delta\left((\mathsf{Ext}(X, S), S, I), (R, S, I)\right) \leq \epsilon_{\mathsf{Ext}}$$

holds, where S is uniform over $\mathcal{S}$ and R is uniform over $\{0,1\}^n$.

2.4 Hash Functions

Let $\mathsf{H} : \mathcal{D} \to \mathcal{R}$ be a hash function, where $\mathcal{D} = \mathcal{D}(\lambda)$ and $\mathcal{R} = \mathcal{R}(\lambda)$ are sets. We require the following property of hash functions for our scheme.

Definition 3 (Target Collision Resistance) *We say a hash function H is target collision resistant if for any PPT adversary A,*

$$\mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{H},\mathsf{A}}(\lambda) \coloneqq \Pr[x^* \leftarrow_{\$} \mathcal{D}, x \leftarrow_{\$} \mathsf{A}(x^*) : x \neq x^* \wedge \mathsf{H}(x) = \mathsf{H}(x^*)]$$

is negligible in λ.

2.5 Useful Facts

Here, we introduce useful facts used in our security proof. We use the following lemmas to prove adaptive identity security of our scheme.

Lemma 2 (Programmable hash function [16, Theorem 7]) *Let m, Q be integers. We choose $h = (h_1, \ldots, h_m) \in \mathbb{Z}_q^m$ as follows: (1) set $J = Q^2$,*

(2) sample $u_{i,j} \leftarrow_{\$} \{-1, 0, 1\}$ *for* $i = 1, \ldots, m$ *and* $j = 1, \ldots, J$*, (3) set* $h_i = \sum_{j=1}^{J} u_{i,j}$*. For* $h = (h_1, \ldots, h_m)$*, we define*

$$\beta_h(x) \coloneqq 1 + \sum_{i=1}^{m} x[i]h_i \bmod q,$$

where $x = (x[1], \ldots, x[m]) \in \{0,1\}^m$*. Then, for any distinct* $id_1, \ldots, id_Q, id^* \in \{0,1\}^m$*, we have*

$$\Pr\left[\bigwedge_{j=1}^{Q} (\beta_h(id_j) \neq 0) \wedge (\beta_h(id^*) = 0)\right] \geq \Theta\left(\frac{1}{\sqrt{m}Q}\right),$$

where the probability is taken over the choice of h*.*

Lemma 3 ([30, Lemma 5]) *Let* $x_1, \ldots, x_l \in \mathbb{R}$ *be reals such that*

$$\sum_{i=1}^{l} |x_i| \leq \frac{1}{2}.$$

Furthermore, let $\delta_1, \ldots, \delta_l \in \mathbb{R}$ *be reals such that* $0 < \delta_{\mathrm{low}} \leq \delta_i \leq \delta_{\mathrm{up}}$ *for* $i = 1, \ldots, l$*. Then, we have*

$$\left|\sum_{i=1}^{l} \delta_i x_i\right| \geq \delta_{\mathrm{low}} \left|\sum_{i=1}^{l} x_i\right| - \frac{\delta_{\mathrm{up}} - \delta_{\mathrm{low}}}{2}.$$

3 Identity-Based Key-Encapsulation Mechanism

In this section, we introduce the syntax, the correctness property, and the security notion for IB-KEM.

Syntax. An IB-KEM scheme $\Pi = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ consists of four PPT algorithms.

- $\mathsf{Setup}(1^\lambda) \to (pp, mk)$. The setup algorithm takes as input the security parameter λ, outputs a public parameter pp and a master key mk. We assume that pp implicitly defines an identity space $\mathcal{ID}$, a session key space $\mathcal{K}$, and a secret key space $\mathcal{SK}$.
- $\mathsf{KGen}(mk, id) \to sk_{id}$. The key generation algorithm takes as input the master key mk and an identity $id \in \mathcal{ID}$, outputs a secret key sk_{id} for the id.
- $\mathsf{Encap}(pp, id) \to (ct, K)$. The encapsulation algorithm takes as input the public parameter pp and an $id \in \mathcal{ID}$, outputs a session key $K \in \mathcal{K}$ together with a ciphertext ct with respect to identity id.
- $\mathsf{Decap}(sk_{id}, ct) \to K$ or $\perp$. The decapsulation algorithm takes as input a secret key sk_{id} and a ciphertext ct, outputs a decapsulated key $K \in \mathcal{K}$ or the rejection symbol $\perp$.

Correctness. We require correctness of decapsulation: that is for all λ, all pairs (pp, mk) generated by $\mathsf{Setup}(1^\lambda)$, all identities $id \in \mathcal{ID}$, and all $(ct, K) \leftarrow \mathsf{Encap}(pp, id)$, $\Pr[\mathsf{Decap}(\mathsf{KGen}(mk, id), ct) = K] = 1$.

Security. In this paper, we consider the IB-KEM variant of CCA-security for leakage-resilient IBE in the bounded memory leakage model [2]. Let Π be an IB-KEM scheme. We consider the IND-ID-lrCCA game between a challenger C and an adversary A as follows.

Setup phase: C runs Setup to generate (pp, mk), and sends pp to A.

Query phase 1: The adversary A makes queries of the following types:

- Key generation query $id \in \mathcal{ID}$. C computes and returns the secret key $sk_{id} \leftarrow \mathsf{KGen}(mk, id)$ to A.
- Leakage query (id, f), where $f : \mathcal{SK} \to \{0,1\}$ is an efficiently computable function. C returns $f(sk_{id})$ to A.
- Decapsulation query (id, ct). C returns $\mathsf{Decap}(sk_{id}, ct)$ to A.

Challenge phase: A sends the challenge identity $id^* \in \mathcal{ID}$ to C. It must be that he has never queried id^* as a key generation query. C chooses a bit $b \leftarrow_\$ \{0,1\}$. C runs $\mathsf{Encap}(pp, id^*)$ to generate (ct^*, K_0^*), and chooses a random session key $K_1^* \leftarrow_\$ \mathcal{K}$. Then, he sends (ct^*, K_b^*) to A.

Query phase 2: A makes queries of the following types:

- Key generation query $id \in \mathcal{ID}$, where it must be that $id \neq id^*$.
- Decapsulation query (id, ct), where it must be that $(id, ct) \neq (id^*, ct^*)$.

Guess phase: Finally A outputs a guess $b' \in \{0,1\}$.

Note that, in query phase 1 and 2, C computes sk_{id} the first time that id is queried in a key generation, leakage, or decryption query, and responds to all future queries on the same id with the same sk_{id}.

Definition 4 (IND-ID-lrCCA security) *An IB-KEM scheme Π is ℓ-IND-ID-lrCCA (indistiguishability against adaptive identity leakage-resilient chosen-ciphertext attack) secure if for any PPT adversary A that makes at most ℓ leakage queries, the advantage*

$$\mathsf{Adv}^{\text{IND-ID-lrCCA}}_{\Pi,\mathsf{A}}(\lambda) \coloneqq \left| \Pr[b' = b] - \frac{1}{2} \right|$$

is negligible in λ.

Remark: Challenge-dependent leakage. In the security definition, the adversary is not allowed to obtain the leakage $f(sk_{id})$ after the challenge phase. We note that this restriction is indeed necessary: the adversary can encode the decapsulation algorithm for the challenge ciphertext ct^* and the challenge identity id^*.

4 Construction

In this section, we propose a new CCA-secure leakage-resilient IB-KEM scheme.

Let $params = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, g_1, g_2, e) \leftarrow \mathsf{GGen}(1^\lambda)$, n be the bit-length of a session key (i.e., $\mathcal{K} = \{0,1\}^n$), m be the bit-length of an identity (i.e., $\mathcal{ID} = \{0,1\}^m$), $\ell < \log_2 q$ be any positive integer, $\mathsf{H} : \mathbb{G}_1^{2k+1} \times \mathcal{S} \to \mathbb{Z}_q \backslash \{0\}$ be a target collision resistant hash function, $\mathsf{Ext} : \mathbb{G}_T \times \mathcal{S} \to \{0,1\}^n$ be a $(\log_2 q - \ell, \epsilon_{\mathsf{Ext}})$-randomness extractor. We assume that ϵ_{Ext} is negligible in λ.

Our scheme $\Pi = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ is described as follows.

$\mathsf{Setup}(1^\lambda)$**:** Choose $a_1, \ldots, a_k \leftarrow_\$ \mathbb{Z}_q \backslash \{0\}$ and $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_m \leftarrow_\$ \mathbb{Z}_q^{k \times k}, \mathbf{D} \leftarrow_\$ \mathbb{Z}_q^{k \times 2}$ uniformly at random and set

$$\mathbf{A} \coloneqq \begin{pmatrix} a_1 & \cdots & \mathbf{O} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{O} & \cdots & a_k & 1 \end{pmatrix} \in \mathbb{Z}_q^{k \times (k+1)}.$$

Output $pp = ([\mathbf{A}]_1, [\mathbf{B}_0]_1, [\mathbf{B}_1]_1, \ldots, [\mathbf{B}_m]_1, [\mathbf{D}]_1)$ and $mk = (a_1, \ldots, a_k, \mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_m, \mathbf{D})$.

For an identity $id = (id[1], \ldots, id[m]) \in \{0,1\}^m$, let

$$\mathbf{F}_{id} = \left(\mathbf{A} \middle\| \mathbf{B}_0 + \sum_{i=1}^{m} id[i]\mathbf{B}_i \right) \in \mathbb{Z}_q^{k \times (2k+1)}.$$

$\mathsf{KGen}(mk, id)$**:** Compute a random matrix $\mathbf{S}_{id} \in \mathbb{Z}_q^{(2k+1) \times 2}$ such that

$$\mathbf{F}_{id}\mathbf{S}_{id} = \mathbf{D} \tag{1}$$

as follows. Let

$$\mathbf{F}'_{id} = \begin{pmatrix} 1 & \\ \vdots & \left\| \mathbf{B}_0 + \sum_{i=1}^{m} id[i]\mathbf{B}_i \right. \\ 1 & \end{pmatrix} \in \mathbb{Z}_q^{k \times (k+1)}.$$

Choose $\mathbf{S}' \leftarrow_\$ \mathbb{Z}_q^{(k+1) \times 2}$ at random, compute

$$\mathbf{S}'' = \begin{pmatrix} a_1^{-1} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{O} & \cdots & a_k^{-1} \end{pmatrix} (\mathbf{D} - \mathbf{F}'_{id}\mathbf{S}') \in \mathbb{Z}_q^{k \times 2},$$

and set

$$\mathbf{S}_{id} = \begin{pmatrix} \mathbf{S}'' \\ \mathbf{S}' \end{pmatrix}.$$

Output $sk_{id} = [\mathbf{S}_{id}]_2$ as a secret key for the id.

$\mathsf{Encap}(pp, id)$**:** Choose $\mathbf{r} \leftarrow_{\$} \mathbb{Z}_q^k$ and $sd \leftarrow_{\$} \mathcal{S}$ at random, compute

$$\begin{aligned}
[\mathbf{c}]_1 &= \left[\mathbf{F}_{id}^\top \mathbf{r}\right]_1 \in \mathbb{G}_1^{2k+1}, \\
\alpha &= \mathsf{H}([\mathbf{c}]_1, sd) \in \mathbb{Z}_q, \\
[t_a]_T &= \left[\mathbf{r}^\top \mathbf{D} \left(\begin{smallmatrix} 1 \\ \alpha \end{smallmatrix}\right)\right]_1 \circ [1]_2 \in \mathbb{G}_T, \\
[t_s]_T &= \left[\mathbf{r}^\top \mathbf{D} \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)\right]_1 \circ [1]_2 \in \mathbb{G}_T.
\end{aligned}$$

Output $ct = ([\mathbf{c}]_1, [t_a]_T, sd)$ and $K = \mathsf{Ext}([t_s]_T, sd)$.

$\mathsf{Decap}(sk_{id}, ct)$**:** On input $sk_{id} = [\mathbf{S}_{id}]_2$ and $ct = ([\mathbf{c}]_1, [t]_T, sd)$, compute

$$\begin{aligned}
\alpha &= \mathsf{H}([\mathbf{c}]_1, sd), \\
[t_a]_T &= \left[\mathbf{c}^\top\right]_1 \circ \left[\mathbf{S}_{id} \left(\begin{smallmatrix} 1 \\ \alpha \end{smallmatrix}\right)\right]_2, \\
[t_s]_T &= \left[\mathbf{c}^\top\right]_1 \circ \left[\mathbf{S}_{id} \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)\right]_2.
\end{aligned}$$

Output $\mathsf{Ext}([t_s]_T, sd)$ if $[t]_T = [t_a]_T$, otherwise $\perp$.

Correctness. Let $sk_{id} = [\mathbf{S}_{id}]_2$, $ct = ([\mathbf{c}]_1, [t]_T, sd)$, and $\alpha = \mathsf{H}([\mathbf{c}]_1, sd)$. If $\mathbf{c} = \mathbf{F}_{id}^\top \mathbf{r}$ and $t = \mathbf{r}^\top \mathbf{D} \left(\begin{smallmatrix} 1 \\ \alpha \end{smallmatrix}\right)$ then

$$t_a = \mathbf{c}^\top \mathbf{S}_{id} \left(\begin{smallmatrix} 1 \\ \alpha \end{smallmatrix}\right) = \mathbf{r}^\top \mathbf{F}_{id} \mathbf{S}_{id} \left(\begin{smallmatrix} 1 \\ \alpha \end{smallmatrix}\right) = \mathbf{r}^\top \mathbf{D} \left(\begin{smallmatrix} 1 \\ \alpha \end{smallmatrix}\right) = t$$

in the Decap procedure, and it is similar to t_s. Therefore, our IB-KEM scheme Π satisfies correctness.

5 Security

In this section, we prove the IND-ID-lrCCA security of our scheme.

Theorem 1 *Under the external k-LIN assumption relative to GGen in group $\mathbb{G}_1$, our scheme Π is ℓ-IND-ID-lrCCA secure for any positive integer ℓ satisfying*

$$\ell \leq \log_2 q - n - \eta, \tag{2}$$

where $\eta = \eta(\lambda)$ is a positive integer such that $2^{-\eta}$ is negligible in λ.

In particular, given an efficient adversary A breaking the ℓ-IND-ID-lrCCA secure of Π with advantage $\epsilon_{\mathsf{A}} := \mathsf{Adv}^{\text{IND-ID-lrCCA}}_{\Pi,\mathsf{A}}(\lambda)$, we can construct an adversary D breaking the external k-LIN assumption with advantage $\epsilon_{\mathsf{D}} := \mathsf{Adv}^{\text{xklin}}_{\mathsf{GGen},\mathsf{D}}(\lambda)$ such that

$$\epsilon_{\mathsf{D}} \geq \Theta\left(\frac{1}{\sqrt{m}(Q_{\mathsf{KGen}} + Q_{\mathsf{Dec}})}\right) \epsilon_{\mathsf{A}} - \mathsf{Adv}^{\text{tcr}}_{\mathsf{H}}(\lambda) - \frac{Q_{\mathsf{Dec}}}{2^\eta (1 - Q_{\mathsf{Dec}}/q)} - \frac{3}{q} - \frac{Q_{\mathsf{Dec}}}{q^{2k+1}\,|\mathcal{S}|} - \epsilon_{\mathsf{Ext}},$$

holds for such λ, where $Q_{\mathsf{KGen}} = \mathsf{poly}(\lambda)$ and $Q_{\mathsf{Dec}} = \mathsf{poly}(\lambda)$ are the number of key generation queries and decapsulation queries made by A, respectively.

Remark. Our scheme works also on type 1 or 2 bilinear groups.

Proof: Let A be an efficient adversary on the IND-ID-lrCCA security of Π. Namely, $\epsilon_{\mathsf{A}} \geq 1/\mathsf{poly}(\lambda)$ for infinitely many λ. We will consider a sequence of games, $\mathrm{Game}_0, \ldots, \mathrm{Game}_9$ performed by a challenger and A. At the end of each game, C outputs a bit $\gamma \in \{0,1\}$, which will be described below.

Let W_i be the event such that $\gamma = 1$ in Game_i.

Game_0: This game is the IND-ID-lrCCA game. At the end of the game, C outputs $\gamma = 1$ if $b' = b$, otherwise $\gamma = 0$, where b' is A's guessing bit of b. Thus,

$$\left| \Pr[W_0] - \frac{1}{2} \right| = \epsilon_{\mathsf{A}}. \tag{3}$$

The challenge is (ct^*, K_b^*) where $ct^* = ([\mathbf{c}^*]_1, [t_a^*]_T, sd^*)$. We denote by $\mathbf{r}^*, \alpha^*, t_s^*$ the corresponding intermediate values. The session key K_b^* is $\mathsf{Ext}([t_s^*]_T, sd^*)$ or random over $\{0,1\}^n$, depending on the bit b.

Game_1: This game is the same as Game_0 except that C changes the generation of the public parameter pp and the ciphertext ct^* as follows.

- In the setup phase, choose $\mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_m \leftarrow_{\$} \mathbb{Z}_q^{(k+1)\times k}$ and $\mathbf{E} \leftarrow_{\$} \mathbb{Z}_q^{(k+1)\times 2}$ uniformly at random. Set $J := (Q_{\mathsf{KGen}}+Q_{\mathsf{Dec}})^2$, sample $u_{i,j} \leftarrow_{\$} \{-1, 0, 1\}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, J$, and set $h_i := \sum_{j=1}^{J} u_{i,j}$. The public parameter is defined as

$$\begin{aligned} \mathbf{B}_0 &= \mathbf{A}\mathbf{R}_0 + \mathbf{I}_k, \\ \mathbf{B}_i &= \mathbf{A}\mathbf{R}_i + h_i\mathbf{I}_k \text{ for } i = 1, \ldots, m, \\ \mathbf{D} &= \mathbf{A}\mathbf{E}. \end{aligned}$$

  Output $pp = ([\mathbf{A}]_1, [\mathbf{B}_0]_1, [\mathbf{B}_1]_1, \ldots, [\mathbf{B}_m]_1, [\mathbf{D}]_1)$. C holds $(a_1, a_2, \mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_m, \mathbf{E})$ as a master key in this game.

  In Game_1, the $\mathbf{F}_{id}$ for $id \in \{0,1\}^m$ can be written by

$$\mathbf{F}_{id} = \left(\mathbf{A} \middle\| \mathbf{A}\mathbf{R}_{id} + \beta_h(id)\mathbf{I}_k \right),$$

  where $\mathbf{R}_{id} = \mathbf{R}_0 + \sum_{i=1}^{m} id[i]\mathbf{R}_i$ and $\beta_h(id) = 1 + \sum_{i=1}^{m} id[i]h_i$.
- In the challenge phase, C computes $[t_a^*]_T$ and $[t_s^*]_T$ as follows:

$$\begin{aligned} [t_a^*]_T &= \left[\mathbf{c}^{*\top} \right]_1 \circ \left[\mathbf{S}^* \begin{pmatrix} 1 \\ \alpha^* \end{pmatrix} \right]_2, \\ [t_s^*]_T &= \left[\mathbf{c}^{*\top} \right]_1 \circ \left[\mathbf{S}^* \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right]_2, \end{aligned}$$

  where $[\mathbf{S}^*]_2$ is the secret key for the id^*.

Note that this change does not affect the distributions of the public parameter pp and the challenge (ct^*, K_b^*). Therefore, we have

$$\Pr[W_0] = \Pr[W_1]. \tag{4}$$

Game$_2$: Let id^* be the challenge identity and $id_1, \ldots, id_Q$ be identities that A queries in the key generation query and the decapsulation query, where $Q \leq Q_{\mathsf{KGen}} + Q_{\mathsf{Dec}}$. Define the event

$$\mathcal{FORCEDABORT} : \bigvee_{i=1}^{Q} (\beta_h(id_i) = 0) \vee (\beta_h(id^*) \neq 0),$$

and

$$\eta(\mathbf{id}_{\mathsf{A}}) \coloneqq \Pr[\neg \mathcal{FORCEDABORT}]$$

for $\mathbf{id}_{\mathsf{A}} = (id_1, \ldots, id_Q, id^*)$, where the probability is taken over the choice of h. By Lemma 2, this probability has a minimum value greater than 0. Let η_{low} be the minimum value of $\eta(\mathbf{id}_{\mathsf{A}})$.

In the guess phase, A outputs its guess $b' \in \{0,1\}$ for b. C checks the event $\mathcal{FORCEDABORT}$ occurs for $\mathbf{id}_{\mathsf{A}}$. If yes, C aborts the game and outputs a fresh random bit $\gamma \in \{0,1\}$. Otherwise, C first estimates the probability $\eta(\mathbf{id}_{\mathsf{A}})$ by sampling $(h_1, \ldots, h_m)$ sufficiently large amount of times. Let $\eta'(\mathbf{id}_{\mathsf{A}})$ be the estimation of $\eta(\mathbf{id}_{\mathsf{A}})$. Depending on the estimate $\eta'(\mathbf{id}_{\mathsf{A}})$, C decides γ as follows:

- Case $\eta'(\mathbf{id}_{\mathsf{A}}) \leq \eta_{\mathrm{low}}$: C outputs $\gamma = [b = b']$.
- Case $\eta'(\mathbf{id}_{\mathsf{A}}) > \eta_{\mathrm{low}}$: With probability $\eta_{\mathrm{low}}/\eta'(\mathbf{id}_{\mathsf{A}})$, C outputs $\gamma = [b = b']$. With probability $1 - \eta_{\mathrm{low}}/\eta'(\mathbf{id}_{\mathsf{A}})$, C aborts the game and outputs a fresh random bit $\gamma \in \{0,1\}$.

Lemma 4 in Appendix will show that

$$\frac{\eta_{\mathrm{low}}}{2} \left| \Pr[W_1] - \frac{1}{2} \right| \leq \left| \Pr[W_2] - \frac{1}{2} \right|.$$

From Lemma 2, we have

$$\left| \Pr[W_1] - \frac{1}{2} \right| \leq \Theta\left(\sqrt{m}(Q_{\mathsf{KGen}} + Q_{\mathsf{Dec}})\right) \left| \Pr[W_2] - \frac{1}{2} \right|. \tag{5}$$

Game$_3$: In Game$_3$, we make the following changes to the experiment. When A queries an identity id to the key generation oracle, C checks whether $\beta_h(id) = 0$. If so, C immediately aborts and returns a fresh random bit γ. When A outputs id^* as a challenge identity, if $\beta_h(id^*) \neq 0$, C immediately aborts and returns a fresh random bit γ.

Clearly, the above changes do not affect A's environment if $\mathcal{FORCEDABORT}$ dose not occur. Then, we have

$$\Pr[W_2] = \Pr[W_3]. \tag{6}$$

Game$_4$: This game is the same as Game$_3$ except that C changes the generation of the secret key $sk_{id} = [\mathbf{S}_{id}]_2$ for id as follows.

- Case $\beta_h(id) \neq 0$: C chooses $\mathbf{W} \leftarrow_{\$} \mathbb{Z}_q^{(k+1)\times 2}$, computes $\mathbf{W}' \in \mathbb{Z}_q^{k\times 2}$ satisfying

$$\beta_h(id)\mathbf{W}' = -\mathbf{A}\mathbf{W} + \mathbf{A}\mathbf{E}, \tag{7}$$

and sets

$$\mathbf{S}_{id} = \begin{pmatrix} \mathbf{W} - \mathbf{R}_{id}\mathbf{W}' \\ \mathbf{W}' \end{pmatrix}.$$

This $\mathbf{S}_{id}$ satisfies Eq. (1) because

$$\begin{aligned} \mathbf{F}_{id}\mathbf{S}_{id} &= \left(\mathbf{A} \middle\| \mathbf{A}\mathbf{R}_{id} + \beta_h(id)\mathbf{I}_k\right) \begin{pmatrix} \mathbf{W} - \mathbf{R}_{id}\mathbf{W}' \\ \mathbf{W}' \end{pmatrix} \\ &= \mathbf{A}\left(\mathbf{W} - \mathbf{R}_{id}\mathbf{W}'\right) + \left(\mathbf{A}\mathbf{R}_{id} + \beta_h(id)\mathbf{I}_k\right)\mathbf{W}' \\ &= \mathbf{A}\mathbf{W} + \beta_h(id)\mathbf{W}' \\ &= \mathbf{A}\mathbf{W} - \mathbf{A}\mathbf{W} + \mathbf{A}\mathbf{E} \\ &= \mathbf{D}. \end{aligned}$$

Further, the above $\mathbf{S}_{id}$ has the same distribution as the secret key generated by KGen, because $2k+2$ elements in $\mathbf{W}$ are chosen at random and the remaining are determined uniquely by Eq. (7).

- Case $\beta_h(id) = 0$: C computes $\mathbf{S}_{id} \in \mathbb{Z}_q^{(2k+1)\times 2}$ such that

$$(\mathbf{I}_{k+1} \| \mathbf{R}_{id})\,\mathbf{S}_{id} = \mathbf{E} \tag{8}$$

as follows. C computes $\mathbf{S}'' \coloneqq \mathbf{E} - \mathbf{R}_{id}\mathbf{S}'$ where $\mathbf{S}' \leftarrow_{\$} \mathbb{Z}_q^{k\times 2}$, and sets

$$\mathbf{S}_{id} = \begin{pmatrix} \mathbf{S}'' \\ \mathbf{S}' \end{pmatrix}.$$

It is easy to see that $[\mathbf{S}_{id}]_2$ is the correct secret key for id by multiplying $\mathbf{A}$ from the left to both hand sides of Eq. (8).
We show that the above $\mathbf{S}_{id}$ has the same distribution of the original KGen as seen from A. Now, $\mathbf{S}'$ is chosen randomly. Hence, we need to show that 2 elements in $\mathbf{S}''$ e.g. $\mathbf{e}\mathbf{S}''$ are also random where $\mathbf{e} \coloneqq (0, \ldots, 0, 1)$. It suffices to prove $\mathbf{u} \coloneqq \mathbf{e}\mathbf{E}$ is random even given $\mathbf{A}$ and $\mathbf{D} = \mathbf{A}\mathbf{E}$, since $\mathbf{e}\mathbf{S}'' = \mathbf{e}\mathbf{E} - \mathbf{e}\mathbf{R}_{id}\mathbf{S}'$. It is easy to see that

$$\begin{pmatrix} \mathbf{D} \\ \mathbf{u} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{A} \\ \mathbf{e} \end{pmatrix}}_{\mathbf{A}'} \mathbf{E}. \tag{9}$$

Because $\mathbf{A}'$ is of full rank, the distribution of $\mathbf{u}$ is random and independent from $\mathbf{D}$ that A knows. Hence, $\mathbf{e}\mathbf{S}''$ is also random as seen from A.

Note that this change dose not affect the distribution of the secret key sk_{id} for id. Therefore, we have

$$\Pr[W_3] = \Pr[W_4]. \tag{10}$$

Game$_5$: This game is the same as Game$_4$ except that $[\mathbf{c}^*]_1$ in the challenge is randomly chosen from $\mathbb{G}_1^{2k+1}$. Furthermore, C chooses $[\mathbf{c}^*]_1 \leftarrow_{\$} \mathbb{G}_1^{2k+1}$, $sd^* \leftarrow_{\$} \mathcal{S}$, and computes $\alpha^* = \mathsf{H}([\mathbf{c}^*]_1, sd^*)$ at the beginning of the game. As we will show in Lemma 6, we have that there exists a PPT adversary D such that

$$|\Pr[W_4] - \Pr[W_5]| \leq \mathsf{Adv}^{\mathrm{xklin}}_{\mathsf{GGen},\mathsf{D}}(\lambda) + \frac{1}{q}. \tag{11}$$

The decapsulation oracle in this game is depicted in Fig. 1. We define that a ciphertext $[\mathbf{c}]_1$ is *valid for id* if there exists $\mathbf{r} \in \mathbb{Z}_q^k$ such that $[\mathbf{c}]_1 = \left[\mathbf{F}_{id}^\top \mathbf{r}\right]_1$. With pp and mk, we can efficiently check whether $[\mathbf{c}]_1 = \left[(c_1, \ldots, c_{2k+1})^\top\right]_1$ is valid for id by simply verifying

$$[(c_{k+1}, \ldots, c_{2k+1})]_1 = \left[(c_1, \ldots, c_k) \begin{pmatrix} a_1^{-1} & \cdots & \mathbf{O} \\ \vdots & \ddots & \vdots \\ \mathbf{O} & \cdots & a_k^{-1} \end{pmatrix} \mathbf{F}'_{id}\right]_1.$$

Decapsulation of adversarial query $(id, ct = ([\mathbf{c}]_1, [t]_T, sd))$

1 : Generate $sk_{id} = [\mathbf{S}_{id}]_2$
2 : $\alpha \leftarrow \mathsf{H}([\mathbf{c}]_1, sd)$
3 : **if** $\beta_h(id) \neq 0$ **then**
4 : **return** $K \leftarrow \mathsf{Decap}(sk_{id}, ct)$
5 : **if** $([\mathbf{c}]_1, sd) \neq \left([\mathbf{c}^*]_1, sd^*\right) \wedge \alpha = \alpha^*$ **then**
6 : **return** $K \leftarrow \mathsf{Decap}(sk_{id}, ct)$
7 : **if** $\left([\mathbf{c}]_1, sd\right) = \left([\mathbf{c}^*]_1, sd^*\right)$ **then**
8 : **if** $[t]_T = [t_a^*]_T$ **then return** K_0^*
9 : **else return** $\perp$
10 : **if** $[\mathbf{c}]_1$ is invalid for id **then**
11 : $[t_a]_T \leftarrow \left[\mathbf{c}^\top\right]_1 \circ \left[\mathbf{S}_{id}(1, \alpha)^\top\right]_2$
12 : $[t_s]_T \leftarrow \left[\mathbf{c}^\top\right]_1 \circ \left[\mathbf{S}_{id}(1, 0)^\top\right]_2$
13 : **if** $[t]_T = [t_a]_T$ **then return** $\mathsf{Ext}([t_s]_T, sd)$
14 : **else return** $\perp$
15 : **return** $K \leftarrow \mathsf{Decap}(sk_{id}, ct)$

Figure 1: Decapsulation oracle in Game$_5$

Game$_6$: In this game, at line 6 in Fig. 1, C returns $\perp$. Then we have

$$\begin{aligned}\Pr[W_5] &= \Pr[W_5 \wedge \mathsf{H} \text{ has collision}] + \Pr[W_5 \wedge \mathsf{H} \text{ has no collision}] \\ &\leq \Pr[\mathsf{H} \text{ has collision}] + \Pr[W_5 \wedge \mathsf{H} \text{ has no collision}] \\ &\leq \mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{H}}(\lambda) + \Pr[W_6].\end{aligned}$$

Therefore, we obtain

$$|\Pr[W_5] - \Pr[W_6]| \leq \mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{H}}(\lambda). \tag{12}$$

Game$_7$: In this game, at line 13 in Fig. 1, C returns $\perp$. As we will show in Lemma 7, we have

$$|\Pr[W_6] - \Pr[W_7]| \leq \frac{Q_{\mathsf{Dec}}}{2^{\eta}(1 - Q_{\mathsf{Dec}}/q)} + \frac{1}{q}. \tag{13}$$

Game$_8$: In this game, at line 8 in Fig. 1, C returns $\perp$. $([\mathbf{c}]_1, sd) = ([\mathbf{c}^*]_1, sd^*)$ holds with probability $1/(q^{2k+1} \cdot |\mathcal{S}|)$ before the challenge phase, since A knows nothing about $(\mathbf{c}^*, sd^*)$ chosen randomly. On the other hand, after the challenge phase $(id^*, ct^* = ([\mathbf{c}^*]_1, [t^*_a]_T, sd^*))$ was already announced to A, any adversarial decapsulation query $(id^*, ([\mathbf{c}^*]_1, [t_a]_T, sd^*))$ with $[t]_T = [t_a]_T$ is equal to (id^*, ct^*). Hence, such adversarial decapsulation query is forbidden by the restriction of IND-ID-lrCCA game.

Thus we have

$$|\Pr[W_7] - \Pr[W_8]| \leq \frac{Q_{\mathsf{Dec}}}{q^{2k+1} \cdot |\mathcal{S}|}. \tag{14}$$

Game$_9$: In this game, K^*_0 is chosen at random from $\{0,1\}^n$ instead of using $\mathsf{Ext}([t^*_s]_T, sd^*)$. As we will show in Lemma 8, we have

$$|\Pr[W_8] - \Pr[W_9]| \leq \epsilon_{\mathsf{Ext}} + \frac{1}{q}. \tag{15}$$

In Game$_9$, A does not get any information about bit b because both K^*_0 and K^*_1 are random. Hence, we have

$$\Pr[W_9] = \frac{1}{2}. \tag{16}$$

From Eqs. (3)–(6) and (10)–(16), we have shown that given an adversary A with advantage ϵ_{A}, there exists an adversary D with $\epsilon_{\mathsf{D}} = \mathsf{Adv}^{\mathrm{xklin}}_{\mathsf{GGen},\mathsf{D}}(\lambda)$ such that

$$\begin{aligned}
\epsilon_{\mathsf{A}} &= \left|\Pr[W_0] - \frac{1}{2}\right| \\
&\leq \Theta\left(\sqrt{m}(Q_{\mathsf{KGen}} + Q_{\mathsf{Dec}})\right) \left|\Pr[W_2] - \frac{1}{2}\right| \\
&\leq \Theta\left(\sqrt{m}(Q_{\mathsf{KGen}} + Q_{\mathsf{Dec}})\right) \sum_{i=4}^{8} |\Pr[W_i] - \Pr[W_{i+1}]| \\
&= \Theta\left(\sqrt{m}(Q_{\mathsf{KGen}} + Q_{\mathsf{Dec}})\right) \left(\epsilon_{\mathsf{D}} + \mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{H}}(\lambda) + \frac{Q_{\mathsf{Dec}}}{2^{\eta}(1 - Q_{\mathsf{Dec}}/q)} + \frac{3}{q} + \frac{Q_{\mathsf{Dec}}}{q^{2k+1}\,|\mathcal{S}|} + \epsilon_{\mathsf{Ext}}\right).
\end{aligned}$$

Therefore, we have

$$\epsilon_{\mathsf{D}} \geq \Theta\left(\frac{1}{\sqrt{m}(Q_{\mathsf{KGen}}+Q_{\mathsf{Dec}})}\right)\epsilon_{\mathsf{A}} - \mathsf{Adv}^{\mathrm{tcr}}_{\mathsf{H}}(\lambda) - \frac{Q_{\mathsf{Dec}}}{2^{\eta}(1-Q_{\mathsf{Dec}}/q)} - \frac{3}{q} - \frac{Q_{\mathsf{Dec}}}{q^{2k+1}\,|\mathcal{S}|} - \epsilon_{\mathsf{Ext}}.$$

The right side of the above inequality is non-negligible, since ϵ_{A} and $\Theta(1/\sqrt{m}(Q_{\mathsf{KGen}}+Q_{\mathsf{Dec}}))$ are non-negligible in λ, other terms are negligible in λ. Hence, this contradicts the edternal k-LIN assumption. This completes the proof of Theorem 1. □

6 Conclusion

In this paper, we proposed the first CCA-secure leakage-resilient IB-KEM scheme which does not depend on q-type assumptions. More precisely, the proposed scheme is secure under external k-linear assumption. $k = 2$ gives the best scheme, which is secure under the XDLIN assumption and has leakage rate 1/10.

A CCA-secure leakage-resilient IBE scheme is obtained by combining our IB-KEM with any CCA-secure symmetric-key encryption scheme (which does not need to be leakage-resilient).

A Proof of Lemmas

To complete the proof of Theorem 1, we prove Lemma 4, 6, 7, and 8.

Lemma 4

$$\frac{\eta_{\mathrm{low}}}{2}\left|\Pr[W_1]-\frac{1}{2}\right| \leq \left|\Pr[W_2]-\frac{1}{2}\right|.$$

We introduce a lemma before proving Lemma 4.

Lemma 5 ([18, Claim 6.7]) *Let $0 < \rho < 1$ be a real. For a sequence of identities $\mathbf{id} \in (\mathcal{ID})^{Q+1}$, and $\mathcal{ABORT}$ be the event that C aborts with added rules in $Game_2$. For any fixed $\mathbf{id}$,*

$$\eta_{\mathrm{low}}\,(1-\rho) \leq \Pr[\neg\mathcal{ABORT}] \leq \eta_{\mathrm{low}}\,(1+\rho)\,.$$

Proof: [of Lemma 4] For a sequence of identities $\mathbf{id} \in (\mathcal{ID})^{Q+1}$, we define $\mathcal{Q}(\mathbf{id})$ as the event that A uses the last entry in $\mathbf{id}$ as the challenge and makes key generation queries and decapsulation queries for the remaining identities. Then, we have $\sum_{\mathbf{id}\in(\mathcal{ID})^{Q+1}} \Pr[\mathcal{Q}(\mathbf{id})] = 1$. Let $\delta(\mathbf{id}) = \Pr[\neg\mathcal{ABORT}]$, and

δ_{low} and δ_{up} be reals such that $\delta_{\mathrm{low}} \leq \delta(\mathbf{id}) \leq \delta_{\mathrm{up}}$. Then, we have

$$\begin{aligned}
&\left|\Pr[W_2] - \frac{1}{2}\right| \\
&= \left|\sum_{\mathbf{id}} \Pr[\mathcal{Q}(\mathbf{id})] \Pr[W_2 \mid \mathcal{Q}(\mathbf{id})] - \frac{1}{2}\right| \\
&= \left|\sum_{\mathbf{id}} \Pr[\mathcal{Q}(\mathbf{id})]\Big(\Pr[W_2 \wedge \neg \mathcal{ABORT} \mid \mathcal{Q}(\mathbf{id})] + \Pr[W_2 \wedge \mathcal{ABORT} \mid \mathcal{Q}(\mathbf{id})] - \frac{1}{2}\Big)\right| \\
&= \left|\sum_{\mathbf{id}} \Pr[\mathcal{Q}(\mathbf{id})]\left(\Pr[W_2 \mid \mathcal{Q}(\mathbf{id})]\delta(\mathbf{id}) + \frac{1}{2}(1-\delta(\mathbf{id})) - \frac{1}{2}\right)\right| \\
&= \left|\sum_{\mathbf{id}} \delta(\mathbf{id}) \Pr[\mathcal{Q}(\mathbf{id})]\left(\Pr[W_1 \mid \mathcal{Q}(\mathbf{id})] - \frac{1}{2}\right)\right| \\
&\geq \delta_{\mathrm{low}}\left|\Pr[W_1] - \frac{1}{2}\right| - \frac{\delta_{\mathrm{up}} - \delta_{\mathrm{low}}}{2}.
\end{aligned}$$

The last inequality above follows from Lemma 3, since we have

$$\left|\sum_{\mathbf{id}} \Pr[\mathcal{Q}(\mathbf{id})]\left(\Pr[W_1 \mid \mathcal{Q}(\mathbf{id})] - \frac{1}{2}\right)\right| = \left|\Pr[W_1] - \frac{1}{2}\right|$$

and

$$\sum_{\mathbf{id}} \left|\Pr[\mathcal{Q}(\mathbf{id})]\left(\Pr[W_1 \mid \mathcal{Q}(\mathbf{id})] - \frac{1}{2}\right)\right| \leq \sum_{\mathbf{id}} \Pr[\mathcal{Q}(\mathbf{id})] \cdot \frac{1}{2} = \frac{1}{2}.$$

From Lemma 5, we have $\delta_{\mathrm{up}} - \delta_{\mathrm{low}} \leq \eta_{\mathrm{low}}\rho/2$. Therefore, defining $\rho \coloneqq |\Pr[W_1] - 1/2|$, we obtain

$$\left|\Pr[W_2] - \frac{1}{2}\right| \geq \delta_{\mathrm{low}}\left|\Pr[W_1] - \frac{1}{2}\right| - \frac{\delta_{\mathrm{up}} - \delta_{\mathrm{low}}}{2} \geq \frac{\eta_{\mathrm{low}}}{2}\left|\Pr[W_1] - \frac{1}{2}\right|.$$

□

Lemma 6 *For any PPT algorithm* A*, there exists a PPT algorithm* D *such that*

$$|\Pr[W_4] - \Pr[W_5]| \leq \mathsf{Adv}^{\mathrm{xklin}}_{\mathsf{GGen},\mathsf{D}}(\lambda) + \frac{1}{q}. \tag{17}$$

Proof: Let $([\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{y}]_1) \in \mathbb{G}_1^{k\times(k+1)} \times \mathbb{G}_2^{k\times(k+1)} \times \mathbb{G}_1^{k+1}$ be an external k-LIN instance, where

$$\mathbf{A} = \begin{pmatrix} a_1 & \cdots & \mathbf{O} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{O} & \cdots & a_k & 1 \end{pmatrix},$$
$$\mathbf{y} = \mathbf{A}^\top \mathbf{r}^* \text{ or random.}$$

Then, we build a PPT algorithm D with input $([\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{y}]_1)$ that simulates the IND-ID-lrCCA game with A as follows.

Setup phase: D generates $pp = ([\mathbf{A}]_1, [\mathbf{B}_0]_1, [\mathbf{B}_1]_1, \ldots, [\mathbf{B}_m]_1, [\mathbf{D}]_1)$ as same as C, except that D computes

$$\begin{aligned}[\mathbf{B}_0]_1 &= [\mathbf{A}\mathbf{R}_0 + \mathbf{I}_k]_1, \\ [\mathbf{B}_i]_1 &= [\mathbf{A}\mathbf{R}_i + h_i\mathbf{I}_k]_1 \text{ for } i = 1, \ldots, m, \\ [\mathbf{D}]_1 &= [\mathbf{A}\mathbf{E}]_1.\end{aligned}$$

Finally D sends pp to A.

Query phase: D answers for each query from A as follows.

- Key Generation query id. Assume that $\beta_h(id) \neq 0$. D chooses $\mathbf{S}' \leftarrow_{\$} \mathbb{Z}_q^{(k+1)\times 2}$ at random, computes $[\mathbf{S}'']_2 \in \mathbb{G}_2^{k\times 2}$ such that $[\beta_h(id)\mathbf{S}'']_2 = [-\mathbf{A}\mathbf{S}' + \mathbf{A}\mathbf{E}]_2$, sets
$$[\mathbf{S}_{id}]_2 = \left[\begin{pmatrix}\mathbf{S}' - \mathbf{R}_{id}\mathbf{S}'' \\ \mathbf{S}'\end{pmatrix}\right]_2,$$
and returns $sk_{id} = [\mathbf{S}_{id}]_2$ to A.
- Leakage query (id, f) and decapsulation query (id, ct). If $\beta_h(id) \neq 0$, then D can generate sk_{id} as above. Furthermore, even in that case that $\beta_h(id) = 0$ (i.e., $id = id^*$), D can generate sk_{id} by computing $\mathbf{S}_{id}$ such that $(\mathbf{I}_{k+1} \| \mathbf{R}_{id})\mathbf{S}_{id} = \mathbf{E}$. Thus, D can answer $f(sk_{id})$ and $\mathsf{Decap}(sk_{id}, ct)$ for any identity.

Challenge phase: D generates the challenge $(ct^*, K_b^*) = (([\mathbf{c}^*]_1, [t_a]_T, sd), K_b^*)$ as same as C, except that D computes

$$[\mathbf{c}^*]_1 = \left[\begin{pmatrix}\mathbf{y} \\ \mathbf{R}_{id^*}^\top \mathbf{y}\end{pmatrix}\right]_1$$

instead of $[\mathbf{c}^*]_1 = \left[\mathbf{F}_{id^*}^\top \mathbf{r}\right]_1$. Then, D returns (ct^*, K_b^*) to A.

Finally, D outputs $\gamma = [b = b']$ where $b' \in \{0, 1\}$ is the output of A.

We will show that the distribution of (ct^*, K_b^*) is the same as the challenge in Game_4 if $\mathbf{y} = \mathbf{A}^\top \mathbf{r}^*$, while if $\mathbf{y}$ is a random it is the same as that in Game_5 with overwhelming probability. First suppose that $\mathbf{y} = \mathbf{A}^\top \mathbf{r}^*$. In this case,

$$\mathbf{c}^* = \begin{pmatrix}\mathbf{y} \\ \mathbf{R}_{id^*}^\top \mathbf{y}\end{pmatrix} = \begin{pmatrix}\mathbf{A}^\top \mathbf{r}^* \\ \mathbf{R}_{id^*}^\top \mathbf{A}^\top \mathbf{r}^*\end{pmatrix} = (\mathbf{A} \| \mathbf{A}\mathbf{R}_{id^*})^\top \mathbf{r}^* = \mathbf{F}_{id^*}^\top \mathbf{r}^*,$$

showing that (ct^*, K_b^*) is the challenge in Game_4. Next suppose that $\mathbf{y}$ is random in $\mathbb{Z}_q^{k+1}$. It suffices to prove that $\mathbf{z} \coloneqq \mathbf{R}_{id^*}^\top \mathbf{y}$ is also random in $\mathbb{Z}_q^k$ even given $\mathbf{A}$, $\mathbf{U} \coloneqq \mathbf{A}\mathbf{R}_{id^*}^\top$, and $\mathbf{y}$. It is easy to see that

$$\begin{pmatrix}\mathbf{U} \\ \mathbf{z}^\top\end{pmatrix} = \underbrace{\begin{pmatrix}\mathbf{A} \\ \mathbf{y}^\top\end{pmatrix}}_{\mathbf{V}} \mathbf{R}_{id^*}.$$

Therefore, $\mathbf{z}$ is random because $\mathbf{V}$ is of full rank with probability $1-1/q$. Hence, $[\mathbf{c}^*]_1$ is random as expected.

Thus, Game_4 and Game_5 are indistinguishable under the external k-LIN assumption, so that we have Eq. (17). □

Lemma 7

$$|\Pr[W_6] - \Pr[W_7]| \leq \frac{Q_{\mathsf{Dec}}}{2^\eta(1 - Q_{\mathsf{Dec}}/q)} + \frac{1}{q}. \tag{18}$$

Proof: We assume that all decapsulation queries are made after the challenge phase, but a similar (but slight simpler) argument can be used if A makes queries before the challenge phase. Suppose that $(id^*, ct = ([\mathbf{c}]_1, [t]_T, sd))$ is the first decapsulation query such that $id = id^*$ and the condition at line 13 in Fig. 1 is evaluated. Let $\mathbf{D} = (\mathbf{d}_1 \| \mathbf{d}_2), \mathbf{S}_{id^*} = (\mathbf{s}_1^* \| \mathbf{s}_2^*)$, where $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{Z}_q^2, \mathbf{s}_1^*, \mathbf{s}_2^* \in \mathbb{Z}_q^{2k+1}$. Then, we have

$$\begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ t_a^* \\ t_a \end{pmatrix} = \underbrace{\begin{pmatrix} (\mathbf{A} \| \mathbf{A}\mathbf{R}_{id^*}) & \mathbf{0} \\ \mathbf{0} & (\mathbf{A} \| \mathbf{A}\mathbf{R}_{id^*}) \\ \mathbf{c}^{*\top} & \alpha^* \mathbf{c}^{*\top} \\ \mathbf{c}^\top & \alpha \mathbf{c}^\top \end{pmatrix}}_{\mathbf{M}} \begin{pmatrix} \mathbf{s}_1^* \\ \mathbf{s}_2^* \end{pmatrix},$$

where t_a is computed at line 11 in Fig. 1. From the supposition, we can assume that $\alpha \neq \alpha^*$, $\mathbf{c}^*$ is chosen uniformly at random, and $[\mathbf{c}]_1$ is invalid for id^*. Hence, the matrix $\mathbf{M}$ is of full rank with probability at least $1 - 1/q$, that implies that the distribution of t_a is random and independent from $\mathbf{D}$ and t_a^*. In addition to $\mathbf{D}$ and t_a^*, A knows at most ℓ bit leakage $\{f(sk_{id^*})\}$ and n bit challenge session key K_b^* that is probable to provide information on the value of t_a to A. Let T_a, F, and I denote random variables induced by t_a, $(\{f(sk_{id^*})\}, K_b^*)$, and $(\mathbf{D}, t_a^*)$, respectively. Given t_a, $(\{f(sk_{id^*})\}, K_b^*)$, and $(\mathbf{D}, t_a^*)$ that A knows, we have

$$\tilde{H}_\infty(T_a \mid F, I) \geq \tilde{H}_\infty(T_a \mid I) - (\ell + n) = \log_2 q - \ell - n$$

from Lemma 1 and the above discussion. Thus, for any t_a, we have $\Pr[T_a = t_a] \leq 2^{\ell+n}/q$. Therefore, in the first evaluation of line 11, the condition $t = t_a$ is satisfied with probability at most $2^{\ell+n}/q$. Now assuming $t = t_a$ is not satisfied, the number of possible t_a decreases one. So, in the i-th evaluation of line 11, the probability that $t = t_a$ holds is at most $2^{\ell+n}/(q - i + 1)$, in the case that $t = t_a$ is not satisfied in all previous evaluations. From the above discussion, we have

$$|\Pr[W_6] - \Pr[W_7]| \leq \frac{Q_{\mathsf{Dec}} 2^{\ell+n}}{q - Q_{\mathsf{Dec}}} + \frac{1}{q}.$$

From Eq. (2), we obtain Eq. (18). □

Lemma 8

$$|\Pr[W_8] - \Pr[W_9]| \leq \epsilon_{\mathsf{Ext}} + \frac{1}{q}. \tag{19}$$

Proof: In Game_9, C returns $\perp$ to A at line 13 in Fig. 1. Hence, A does not learn any information on t_s^* via the decapsulation oracle, since A can only get decapsulation results of valid ciphertexts. Now, A knows $\mathbf{D}$, t_a^*, and $\{f(sk_{id^*})\}$ as information about t_s^*. Then, we show that the min-entropy of t_s^* is at least $\log_2 q - \ell$ with probability at least $1 - 1/q$.

First, we have

$$t_s^* = {\mathbf{c}^*}^\top \mathbf{s}_1^*,$$

and then

$$\begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ t_a^* \\ t_s^* \end{pmatrix} = \underbrace{\begin{pmatrix} (\mathbf{A} \| \mathbf{AE}) & \mathbf{0} \\ \mathbf{0} & (\mathbf{A} \| \mathbf{AE}) \\ {\mathbf{c}^*}^\top & \alpha^* {\mathbf{c}^*}^\top \\ {\mathbf{c}^*}^\top & \mathbf{0} \end{pmatrix}}_{\mathbf{N}} \begin{pmatrix} \mathbf{s}_1^* \\ \mathbf{s}_2^* \end{pmatrix}.$$

The matrix $\mathbf{N}$ is of full rank with probability at least $1 - 1/q$, since $\alpha^* \neq 0$ and $[\mathbf{c}^*]_1$ is uniformly at random. Then, the distribution of t_s^* is random and independent from $\mathbf{D}$ and t_a^*. In addition to $\mathbf{D}$ and t_a^*, A knows at most ℓ bit leakage $\{f(sk_{id^*})\}$ that is probable to provide information on the value of t_s^* to A. Let T_s, D, and F denote random variables induced by t_s^*, $(\mathbf{D}, t_a^*)$, and $\{f(sk_{id^*})\}$ respectively. Given t_s^*, $(\mathbf{D}, t_a^*)$, and $\{f(sk_{id^*})\}$ that A knows, we have

$$\tilde{H}_\infty(T_s \mid D, F) \geq \tilde{H}_\infty(T_s \mid D) - \ell = \log_2 q - \ell$$

from Lemma 1 and the discussion when ignoring $\{f(sk_{id^*})\}$. Hence $\mathsf{Ext}(T_s, sd^*)$ is statistically indistinguishable from an n bits random string because Ext is a $(\log_2 q - \ell)$-randomness extractor. Therefore, we have Eq. (19). $\square$

References

[1] Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009). https://link.springer.com/10.1007/978-3-642-00457-5_28

[2] Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134 Springer, Heidelberg (2010). https://link.springer.com/10.1007/978-3-642-13190-5_6

[3] Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009). http://link.springer.com/10.1007/978-3-642-03356-8_3

[4] Brakerski, Z., Kalai, Y.T., Katz, J., Vaikuntanathan, V.: Overcoming the hole in the bucket: public-key cryptography resilient to continual memory leakage. In: FOCS, pp. 501–510 (2010)

[5] Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004). https://link.springer.com/10.1007/978-3-540-28632-5_2

[6] Cheon, J.H.: Security analysis of the strong diffie-hellman problem. In: Vaudenay S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 1–11 (2006). https://link.springer.com/10.1007/11761679_1

[7] Chow, S.S., Dodis, Y., Rouselakis, Y., Waters, B.: Practical leakage-resilient identity-based encryption from simple assumptions. In: ACM CCS, pp. 152–161 (2010)

[8] Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: FOCS, pp. 511–520 (2010)

[9] Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: STOC, pp. 621–630 (2009)

[10] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM J. Comput. 38(1), 97–139 (2008)

[11] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for diffie-hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (2013) https://link.springer.com/10.1007/978-3-642-40084-1_8

[12] Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Appl. Math. 156(16), 3113–3121 (2008)

[13] Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 251–261. Springer, Heidelberg (2001). https://link.springer.com/10.1007/3-540-44709-1_21

[14] Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: cold boot attacks on encryption keys. In: USENIX, pp. 45–60 (2008)

[15] Halevi, S., Lin, H.: After-the-fact leakage in public-key encryption. In: Y. Ishai (ed.) TCC. LNCS, vol. 6597, pp. 107–124. Springer, Heidelberg (2011). https://link.springer.com/10.1007/978-3-642-19571-6_8

[16] Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS vol. 5157, pp. 21–38. Springer, Heidelberg (2008). https://link.springer.com/10.1007/978-3-540-85174-5_2

[17] Katz, J., Vaikuntanathan, V.: Signature schemes with bounded leakage resilience. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 703–720. Springer, Heidelberg (2009) http://link.springer.com/10.1007/978-3-642-10366-7_41

[18] Kiltz, E., Galindo, D.: Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. Theoretical Computer Science 410(47–49), 5093–5111 (2009)

[19] Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). https://link.springer.com/10.1007/3-540-48405-1_25

[20] Kurosawa, K., Trieu Phong, L.: Leakage resilient IBE and IPE under the DLIN assumption. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 487–501. Springer, Heidelberg (2013). https://link.springer.com/10.1007/978-3-642-38980-1_31

[21] Lewko, A., Rouselakis, Y., Waters, B.: Achieving leakage resilience through dual system encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 70–88. Springer, Heidelberg (2011). http://link.springer.com/10.1007/978-3-642-19571-6_6

[22] Li, J., Teng, M., Zhang, Y., Yu, Q.: A leakage-resilient CCA-secure identity-based encryption scheme. Computer Journal **59**(7), 1066–1075 (2016)

[23] Micali, S., Reyzin, L.: Physically Observable Cryptography. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004). https://link.springer.com/10.1007/978-3-540-24638-1_16

[24] Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009) https://link.springer.com/10.1007/978-3-642-03356-8_2

[25] Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010). https://link.springer.com/10.1007/978-3-642-14623-7_11

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

[27] Sun, S.F., Gu, D., Liu, S.: Efficient leakage-resilient identity-based encryption with CCA security. In: Cao Z., Zhang F. (eds.) Pairing-Based Cryptography. LNCS, vol. 8365, pp. 149–167. Springer, Heidelberg (2013). https://link.springer.com/10.1007/978-3-319-04873-4_9

[28] Tomida, J., Abe, M., Okamoto, T.: Efficient functional encryption for inner-product values with full-hiding security. In: Bishop, M., Nascimento, A.C.A. (eds.) ISC 2016. LNCS, vol. 9866, pp. 408–425. Springer, Cham (2016). https://link.springer.com/10.1007/978-3-319-45871-7_24

[29] Tomita, T., Ogata, W., Kurosawa, K.: CCA-Secure Leakage-Resilient Identity-Based Key-Encapsulation from Simple (not q-type) Assumptions. Presented in IWSEC 2019 (2019).

[30] Yamada, S.: Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 32–62. Springer, Heidelberg (2016). https://link.springer.com/10.1007/978-3-662-49896-5_2