CCA-Secure Deterministic Identity-Based Encryption Scheme

Meijuan Huang
(School of Computer Science, Shaanxi Normal University, Xi’an 710062
School of Mathematics and Information Science
Baoji University of Arts and Sciences, Baoji 721013, China
mjhuang335@foxmail.com)

Bo Yang*
(School of Computer Science, Shaanxi Normal University, Xi’an 710062
State Key Laboratory of Information Security
Institute of Information Engineering, Chinese Academy of Sciences
Beijing 100093, China
byang@snnu.edu.cn
*Corresponding author)

Yi Zhao
(Department of Computer Science, University of Surrey, Guildford, Surrey
GU2 7XH, United Kingdom
yizhaore@hotmail.com)

Kaitai Liang
(Department of Computer Science, University of Surrey, Guildford, Surrey
GU2 7XH, United Kingdom
k.liang@surrey.ac.uk)

Liang Xue
(Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, ON N2L 3G1, Canada
l34xue@uwaterloo.ca)

Xiaoyi Yang
(Department of Computing, The Hong Kong Polytechnic University
Hong Kong, China
aiyooho@sina.com)

Abstract: Deterministic public-key encryption, encrypting a plaintext into a unique ciphertext without involving any randomness, was introduced by Bellare, Boldyreva, and O’Neill (CRYPTO 2007) as a realistic alternative to some inherent drawbacks in randomized public-key encryption. Bellare, Kiltz, Peikert and Waters (EUROCRYPT 2012) bring deterministic public-key encryption to the identity-based setting, and
propose deterministic identity-based encryption scheme (DIBE). Although the constructions of chosen plaintext attack (CPA) secure DIBE scheme have been studied intensively, the construction of chosen ciphertext attack (CCA) secure DIBE scheme is still challenging problems. In this paper, we introduce the notion of identity-based all-but-one trapdoor functions (IB-ABO-TDF), which is an extension version of all-but-one lossy trapdoor function in the public-key setting. We give a instantiation of IB-ABO-TDF under decisional linear assumption. Based on an identity-based lossy trapdoor function and our IB-ABO-TDF, we present a generic construction of CCA-secure DIBE scheme.

Key Words: deterministic identity-based encryption, identity-based lossy trapdoor functions, identity-based all-but-one trapdoor functions, chosen ciphertext security

Category: C.2.0, D.4.6, E.3

1 Introduction

The semantic security for public key encryption requires that the encryption algorithm must be a random process. This creates a significant performance bottleneck if, for example, one wants to perform fast search over many encrypted data items. To address this issue, Bellare, Boldyreva, and O’Neill [Bellare et al. 2007] first introduced the notion of deterministic public-key encryption (DPKE), in which the encryption algorithm does not use randomness, i.e., its encryption algorithm is required to be a deterministic function of the message. The motivating application of deterministic PKE is to perform fast search over many encrypted data items. The technique is more effective in scenarios where frequent search queries are performed over a huge database of unpredictable data items. Deterministic encryption permits logarithmic time search on encrypted data, while randomized encryption only allows linear time search, meaning a search requires scanning the whole database. Moreover, since deterministic encryption does not use randomness, it is an important class of PKE dealing with the subsequently revealed problem of randomness subversion [Bellare et al. 2009]. The DPKE is used as a building block of hedged PKE [Bellare et al. 2015b, Boldyreva et al. 2017] and nonce-based PKE [Huang et al. 2018], which achieve best possible security in the face of bad randomness.

Because the encryption algorithm of DPKE is a deterministic process, of course deterministic public key encryption cannot satisfy the meaningful notion of security of randomized public key encryption. Bellare et al. [Bellare et al. 2007] provided the “strongest possible” notion of security for this primitive, called PRIV, which can be realized for relatively high-entropy plaintext distributions. Constructions of DPKE schemes satisfying the notions of security were proposed in the random oracle model by Bellare et al. [Bellare et al. 2007]. Later, Bellare et al. [Bellare et al. 2008] and Boldyreva et al. [Boldyreva et al. 2008] refined and extended the security notion and presented constructions in the standard model. Especially, Boldyreva et al. [Boldyreva et al. 2008] gave general constructions of both CPA and CCA secure deterministic public key encryption schemes
which are based on lossy trapdoor functions (LTDF) [Peikert et al. 2008]. They showed that any LTDF is a deterministic PKE scheme which is PRIV-secure for high min-entropy block-sources (namely, each message to be encrypted has high min-entropy given the other messages) as long as the lossy mode acts as a universal hash function. Emerging as a practically-motivated notion of theoretical depth and interest, several significant foundational works then further investigated security for deterministic encryption and presented standard model constructions [Brakerski et al. 2011, Mironov et al. 2012, Raghunathan et al. 2013, O’Neill 2010, Cui et al. 2014, Zhang et al. 2014, Wee 2012, Bellare et al. 2015a, Fuller et al. 2012, Koppula et al. 2016].

Identity-based encryption (IBE) is a public key encryption that enables one to encrypt a message using a recipient’s identity, rather than its public key. It simplifies public key and certificate distribution and management and thus has a wide range of applications [Yu et al. 2017, Li et al. 2019]. Due to the inherent advantage of IBE, Bellare, Kiltz, Peikert and Waters [Bellare et al. 2012] extended the notion of deterministic encryption into the identity-based setting. They proposed a CPA-secure deterministic identity-based encryption (DIBE) scheme by first constructing identity-based lossy trapdoor function (IB-LTDF). The DIBE allows quickly logarithmic-time searchable identity-based encryption of database entries while maintaining the maximal possible privacy. Later, Escala et al. [Escala et al. 2014] provided an alternative definition of partial lossiness of IB-LTDF and constructed a hierarchical identity-based lossy trapdoor functions (HIB-LTDF), based on which they achieved DHIBE scheme for block-sources, this DHIBE scheme is secure against chosen plaintext attack (CPA). After that, several follow-up works [Xie et al. 2012, Fang et al. 2016, Zhang et al. 2017] further investigated security of DIBE and presented CPA-secure DIBE schemes from the hardness of learning with error (LWE) problem. So far, existing DIBE schemes only achieve chosen-plaintext security.

Security against adaptive chosen-ciphertext attack (CCA) is a de facto security notion for public-key encryption in practice. Active adversary might also obtain the decryption of ciphertexts under any identity of its choice. Thus it is necessary to consider the stronger security notion of DIBE, i.e., PRIV-ID-CCA security (we will explain it in section 2). Inspired by CHK transformation approach [Canetti et al. 2004] in randomized PKE, which converts IND-CPA secure 2-level hierarchical IBE to IND-ID-CCA secure IBE using a strongly unforgeable one-time signature. The natural idea is to adapt this approach to the deterministic encryption. That is to say, one can construct PRIV-CCA-secure deterministic IBE based on a PRIV-CPA-secure 2-level deterministic HIBE. Unfortunately, we observe that it is not natural to do so. Since in the deterministic setting, the strongly unforgeable one-time signature is replaced by a target collision-resistant hash function of the plaintext. Similarly, following CHK trans-
formation, the hope is that the proof tries to reduce the security of deterministic 
IBE against adaptive chosen-ciphertext attacks to that of the 2-level selective-id 
secure deterministic HIBE against chosen-plaintext attacks. That is, the reduction 
attains to determine which message was actually encrypted in deterministic 
IBE with the help of an adversary who breaks the deterministic HIBE. The anal-
ysis from CHK transformation, however, does not quite work since it crucially 
relies on the fact that the plaintexts corresponding to the challenge ciphertexts 
are chosen by the adversary in randomized encryption setting. While, in the 
deterministic encryption setting, the plaintexts corresponding to the challenge 
ciphertexts are not chosen by the adversary, such that the challenger cannot 
perform the simulations. For the above reasons, in this paper, we attempt to 
construct a deterministic IBE scheme which can achieve CCA security.

1.1 Our contributions

Identity-Based All-But-One TDFs. In STOC’08 [Peikert et al. 2008], Peikert and 
Waters introduced a new powerful primitive called lossy trapdoor functions 
(LTDF) and a richer abstraction called all-but-one trapdoor functions (ABO-
TDF). LTDF operates in one of two possible “mode”, an injective one and an 
un-invertible lossy one, for which the outputs are indistinguishable. ABO-TDF 
is a generalization of the LTDF whose first input is drawn from a set of branch-
es, one of which is lossy. Freeman et al. [Freeman et al. 2010] generalized the 
definition of ABO-TDF by allowing possibly many lossy branches (other than 
one).

We introduce the notion of identity-based all-but-one trapdoor functions (IB-
ABO-TDF), which is an extension of all-but-one trapdoor functions (ABO-TDF) 
in the public key setting. As for identity-based ABO-TDF, which is essentially 
specific ABO-TDF whose identity set can be viewed as the set of branches, 
each function has many lossy branches just as the generalized definition in 
[Freeman et al. 2010], but each branch is now represented by a pair of (id, b). 
The first component id is the user’s identity, and the second component b is the 
tag. That is to say, the IB-ABO-TDF can be viewed as a specific ABO-TDF 
with the set of branch IDS×TagSp, where IDS is identity space and TagSp is 
tag space. The lossy identity id and the lossy tag b determine together the lossy 
branch. Lossy identities are determined by an auxiliary input which is hidden in 
the public parameters. The security requires that it is computationally indistin-
guishable to tell a lossy branch from an injective branch. Meanwhile, given a 
lossy branch, it is hard to find one-more lossy branches without the trapdoor.

Based on the basic IBE scheme of Bellare et al. [Bellare et al. 2012], we 
present a concrete construction of IB-ABO-TDF, and its security is proved in 
the selective-id security model based on the hardness of decisional linear Diffie-
Hellman assumption (DLIN assumption). In our construction, each identity func-
tion takes as input \((mpk, id, b, x)\), where \(mpk\) is public parameter, \(id\) is identity, \(b\) is branch and \(x\) is the input value, and outputs a ciphertext, which is a matrix encryption of the basic IBE scheme.

CCA-secure deterministic IBE. Based on an IB-LTDF and our IB-ABO-TDF, we present a generic construction of CCA-secure deterministic identity-based encryption scheme in the standard model. Its security is proved under the selective-id security model. In this paper, we stick to the original setting of Bellare et al. [Bellare et al. 2007] and require that the plaintext distributions can not depend on the master public key of the system. In the case of block-sources [Boldyreva et al. 2008], Boldyreva et al. proved that PRIV1-security (i.e., for single-message challenge security) is equivalent to PRIV-security (i.e., for multi-message challenge security) in the sense of indistinguishability-based definitions. In this work, we follow the simplified indistinguishability-based notion, called PRIV1-IND as introduced in [Boldyreva et al. 2008].

Our construction of DIBE scheme builds on the framework of Boldyreva et al. [Boldyreva et al. 2008] for constructing CCA-secure deterministic PKE scheme in the standard model. In our construction, the deterministic encryption algorithm \(E(id, b, m)\), where \(id\) is identity, \(b\) is branch and \(m\) is the message, requires to find a deterministic method to sample the branch. Following [Boldyreva et al. 2008], let \(b = H_{\text{tc}}(m)\), where \(H_{\text{tc}}\) is a universal and target collision-resistant hash function. If message \(m\) has sufficient entropy, the branch looks random due to the Leftover Hash Lemma [Dodis et al. 2004]. The differences are, in the identity-based setting, the simulator must answer the decryption queries for any identity (including the challenge identity) from adversary. Because the branch is a pair \((id, b)\), let \((id^*, b^*)\) be the lossy branch, only if the branch \((id, b) \neq (id^*, b^*)\), with overwhelming probability, the identity-based ABO-TDF works as an injective trapdoor function. Therefore, the simulator can answer the adversary’s decryption queries corresponding to the challenge identity as long as \(b \neq b^*\).

1.2 Related work

Bellare et al. introduced the notion of deterministic identity-based encryption in [Bellare et al. 2012]. They constructed an identity-based lossy trapdoor functions (IB-LTDF) from pairing and built a DIBE scheme with selective-id security as an application of IB-LTDF. Soon afterwards, Escala et al. [Escala et al. 2014] introduced the notion of hierarchical identity-based lossy trapdoor functions (HIB-LTDF), based on which they constructed deterministic hierarchical identity-based encryption scheme (DHIBE) from pairings. In [Xie et al. 2012], Xie et al. considered deterministic identity-based public key encryption in the auxiliary-input setting and proposed a DIBE scheme from lattices that is adaptively secure. Fang et al. [Fang et al. 2016] constructed a selective-id secure DHIBE
scheme based on the hardness of learning with rounding over small modulus [Bogdanov et al. 2016]. In fact, a selective-id secure DHIBE implies a DIBE with a selective security. Recently, Zhang et al. [Zhang et al. 2017] constructed an adaptive secure DIBE scheme with shorter public parameters from partitioning function [Yamada et al. 2017] under the learning with error assumption.

Below we compare our construction with the related works [Xie et al. 2012, Fang et al. 2016, Zhang et al. 2017] in terms of property and security in Table 1. The second column shows whether the scheme is constructed in a generic way. The third to fifth column show the security, the adversary type and the underlying assumption for guaranteeing the security. It can be learnt from Table 1 that only our scheme achieves chosen ciphertext security.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Generic security level</th>
<th>Adversary type</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Bellare et al. 2012]</td>
<td>√</td>
<td>CPA</td>
<td>selective-id DLIN</td>
</tr>
<tr>
<td>[Xie et al. 2012]</td>
<td>×</td>
<td>CPA</td>
<td>adaptive-id LWE</td>
</tr>
<tr>
<td>[Escala et al. 2014]</td>
<td>√</td>
<td>CPA</td>
<td>adaptive-id DBDH+DDH</td>
</tr>
<tr>
<td>[Fang et al. 2016]</td>
<td>×</td>
<td>CPA</td>
<td>selective-id LWR</td>
</tr>
<tr>
<td>[Zhang et al. 2017]</td>
<td>×</td>
<td>CPA</td>
<td>adaptive-id LWE</td>
</tr>
<tr>
<td>Our scheme</td>
<td>√</td>
<td>CCA</td>
<td>selective-id DLIN</td>
</tr>
</tbody>
</table>

Table 1: Comparison of performance

1.3 Organization

The rest of the paper is organized as follows. In section 2, we review some standard notions and cryptographic definitions. We introduce the notion of IB-ABO-TDF and present a concrete construction of IB-ABO-TDF in section 3. In section 4, we propose a generic construction of CCA-secure DIBE scheme based on our IB-ABO-TDF and an IB-LTDF. Finally, we state our conclusion in section 5.

2 Preliminaries

2.1 Notation

We use uppercase Roman letters \( A, B, \ldots \) to represent sets, lowercase Roman letters to elements of a set \( x \in X \), and bold to vectors \( \mathbf{x} \in X^n \). Bold uppercase letters \( A = [a_{ij}] \) represents matrices of scalars. If \( \mathbf{x} \) is a vector, then \(|\mathbf{x}|\) denotes
the number of its coordinates and \( x[i] \) denotes its \( i \)-th coordinate. If \( X \) is a set, \( X^n \) denotes the set of \( n \) dimensional vector over \( X \). \( X^{a \times b} \) denotes the set of \( a \) by \( b \) matrices with entries in \( X \). The \((i, j)\)-th entry of a 2 dimensional matrix \( X \) is denoted by \( X[i, j] \).

If \( S \) is a set, then \( s \leftarrow S \) denotes the operation of sampling an element uniformly at random from \( S \). Notation \( \langle a, b \rangle \) represents standard scalar product of vector \( a \) and \( b \) with equal-length.

The security parameter is denoted by \( \lambda \) throughout the paper. We let negl(\( \lambda \)) denote some unspecified function such that it approaches zero faster than reciprocal of every polynomial \( f(\lambda) \), saying that such a function is negligible.

### 2.2 Hashing

A family of hash functions \( \mathcal{H} = \{ H_i : \{0,1\}^n \rightarrow R \} \) is universal if for all \( x_1 \neq x_2 \in \{0,1\}^n \), \( \Pr[H(x_1) = H(x_2) : H \stackrel{\$}{\leftarrow} \mathcal{H}] \leq \frac{1}{|R|} \).

A hash function \( \mathcal{H} = (K, H) \) is a target collision-resistant (tcr) if for every polynomial time adversary \( A \), the tcr-advantage

\[
\text{Adv}^{\text{tcr}}_{\mathcal{H}}(A) = \Pr[H(k, x_1) = H(k, x_2) : (x_1, st) \leftarrow A; k \leftarrow K; x_2 \leftarrow A(k, st)]
\]

of \( A \) against \( \mathcal{H} \) is negligible [Boldyreva et al. 2008].

### 2.3 Randomness extractor

Here we review a few concepts related probability distributions and extracting uniform bits from weak random sources.

The statistical distance between two probability distributions \( X \) and \( Y \) over the same domain \( D \) is

\[
\Delta(X, Y) = \frac{1}{2} \sum_{a \in D} |\Pr[X = a] - \Pr[Y = a]|.
\]

The min-entropy of a random variable \( X \) is \( H_{\infty}(X) = -\log(\max_a \Pr[X = a]) \). A distribution \( X \) over \( \{0,1\}^l \) is called a \((t, l)\)-source if \( H_{\infty}(X) \geq t \). A distribution \( X \) is \( \epsilon \)-close to a \( t \)-source if there exists a \( t \)-source \( Y \) such that \( \Delta(X, Y) \leq \epsilon \). Average min-entropy, which captures the remaining unpredictability of \( X \) conditioned on the value of \( Y \), is

\[
\tilde{H}_{\infty}(X|Y) = -\log(\mathbb{E}_y[H_{\infty}(X|Y=y)])
\]

**Lemma 2.1** (Generalized leftover hash lemma (LHL) [Dodis et al. 2004]). Let \( H \) be a family of universal hash functions with range \( R \). Let \( X \) and \( Y \) be random variables such that \( X \in \{0,1\}^n \) and \( \tilde{H}_{\infty}(X|Y) \geq \log |R| + 2\log(\frac{1}{\epsilon}) \). Then for \( h \leftarrow H \), we have \( \Delta((Y, h, h(X)), (Y, U, U)) \leq \epsilon \), where \( U \) is the uniform distribution over the range \( R \).

**Lemma 2.2** (Chain Rule [Dodis et al. 2004]). If \( Y \) has \( 2^r \) values and \( Z \) is any random variable, then \( \tilde{H}_{\infty}(X|(Y, Z)) \geq \tilde{H}_{\infty}(X|Z) - r \).
2.4 Identity-based lossy trapdoor functions

The notion of IB-LTDF, introduced by Bellare et al. [Bellare et al. 2012], is an extension of LTDF in the identity-based setting. In an identity-based lossy trapdoor functions collection, which mode (injective or lossy) the function operates depends on the identity. To properly define lossiness in the identity-based setting, Bellare et al. added an auxiliary input from auxiliary input space AuxSp when generating the parameters. Depending on the value of this auxiliary input, it can obtain the injective trapdoor function or lossy function. aux ∈ AuxSp denotes a particular auxiliary input independent of any identity, which results in an injective setup. That is, under the injective setup, the evaluation function is injective for any identity. Aux(·) (called auxiliary input generator in [Bellare et al. 2012]) denotes an algorithm that takes as input an identity from identity space IDSp and returns an auxiliary input in AuxSp. aux(id) denotes an auxiliary input produced by an auxiliary input generator Aux(id) taking as input special identity id, which results in a lossy setup. That is, under the lossy setup, the identity id lead to lossy evaluation functions, used in the security proof. The requirement is that it is hard to distinguish lossy identities from injective ones. Next, we review the notion of identity-based lossy trapdoor functions proposed by Bellare et al. [Bellare et al. 2012] in the selective-id case.

**Definition 1 (IB-LTDF).** A collection of identity-based (n, k)-lossy trapdoor functions with the identity space IDSp, input space InSp, and auxiliary input space AuxSp is a tuple of (possibly probabilistic) polynomial time algorithms (Setup, KG, Eval, Inv) with the following specifications:

- **Setup**(1^λ, aux). For fixed particular auxiliary input aux ∈ AuxSp, the algorithm outputs (mpk, msk), where mpk is public parameters and msk is its master secret key.

- **KG**(mpk, msk, id). Given mpk, msk, identity id ∈ IDSp the probabilistic algorithm outputs a private key sk_id with respect to the given id.

- **Eval**(mpk, id, x). The deterministic algorithm takes as input mpk, identity id ∈ IDSp, and x ∈ \{0, 1\}^n, outputs a value y.

- **Inv**(mpk, sk_id, y). The deterministic algorithm takes as input mpk, private key sk_id, and a value y, outputs either x ∈ \{0, 1\}^n or ⊥.

We require that the following correctness and lossiness requirements hold:

- **Injective correctness and invertibility.** For any id ∈ IDSp, the evaluation algorithm computes a deterministic injective function over the input space \{0, 1\}^n, which can be inverted using the private key sk_id corresponding to the given id. Formally, for aux ∈ AuxSp, (mpk, msk) ← Setup(1^λ, aux), sk_id ← KG(mpk, msk, id), and x ∈ \{0, 1\}^n,

\[
\Pr[\text{Inv}(mpk, sk_id, \text{Eval}(mpk, id, x)) \neq x] \leq \text{negl}(\lambda)
\]
- Lossiness. Sample an identity $id \leftarrow \text{IDSp}$, for auxiliary input $aux(id) \leftarrow \text{Aux}(1^\lambda, aux(id))$, the image of the algorithm $\text{Eval}(mpk, id, \cdot)$ has size at most $2^{n-k}$. That is $|\text{Eval}(mpk, id, \cdot)| \leq 2^{n-k}$.

- Indistinguishability of lossy identities and injective identities. For the auxiliary inputs $aux, aux(id) \in \text{AuxSp}$, the first output $mpk_0$ of $\text{Setup}(1^\lambda, aux)$ and the first output $mpk_1$ of $\text{Setup}(1^\lambda, aux(id))$ are computationally indistinguishable. That is, for every probabilistic polynomial time adversary $A$

$$\text{Adv}_{\text{IND-LTDF}, A}(\lambda) = |\Pr[A(mpk_0, 1^\lambda)_{\text{KG}(\cdot)} = 1] - \Pr[A(mpk_1, 1^\lambda)_{\text{KG}(\cdot)} = 1]|$$

is negligible, where $\text{KG}(\cdot)$ denotes that $A$ can make private key query on identity $id$ by calling $\text{KG}$ algorithm. For $d \in \{0, 1\}$, $A(mpk_d, 1^\lambda)$ is defined as follows:

1. The adversary $A$ outputs an identity $id^*$ as the target identity.
2. The challenger $C$ samples $aux \in \text{AuxSp}$ and $aux(id^*) \leftarrow \text{Aux}(id^*)$, and computes $(mpk_0, msk_0) \leftarrow \text{Setup}(1^\lambda, aux), (mpk_1, msk_1) \leftarrow \text{Setup}(1^\lambda, aux(id^*))$. The challenger then sends $mpk_d$ to $A$.
3. The adversary $A$ makes private secret query for identity $id$ with restriction that $id \neq id^*$. $C$ returns $sk_{id}$ to $A$ by calling the algorithm $\text{KG}$.
4. $A$ outputs a guess $d' \in \{0, 1\}$.

2.5 Deterministic identity-based encryption

An deterministic identity-based encryption (DIBE) scheme $\text{DE}_{\text{IB}}$ is a tuple of polynomial time algorithm ($\text{DIB. Setup}, \text{DIB. Der}, \text{DIB. Enc}, \text{DIB. Dec}$). The probability algorithm $\text{DIB. Setup}$ takes as input a security parameter $1^\lambda$, and outputs a master key pair $(mpk, msk)$, where $mpk$ is master public key and $msk$ is master secret key. The key derivation algorithm $\text{DIB. Der}$ takes as input an identity $id$ and master secret key. It returns the private key $sk_{id}$ associated with the identity $id$. The deterministic encryption algorithm $\text{DIB. Enc}$ takes as input the master public key $mpk$, and identity $id$ and a message $m$. It outputs a ciphertext $C$. The decryption algorithm $\text{DIB. Dec}$ takes as input identity $id$, its associated private key $sk_{id}$, and a ciphertext $C$. It returns a message $m$ or the symbol $\bot$.

**PRIV1-IND-ID-CCA security.** An DIBE scheme $\text{DE}_{\text{IB}}$ is PRIV1 selective-id secure against chosen-ciphertext attack for $(t, n)$-source $M_0$ and $M_1$ and all polynomial time adversary $A = (A_1, A_2)$, the advantage

$$\text{Adv}_{\text{PRIV1-IND-ID-CCA}, \text{DE}_{\text{IB}}, A}(\lambda) = |\Pr[\text{Guess}^\text{CCA}_{\text{DE}_{\text{IB}}, A}(M_0) = 1] - \Pr[\text{Guess}^\text{CCA}_{\text{DE}_{\text{IB}}, A}(M_1) = 1]|$$

of $A$ against $\text{DE}_{\text{IB}}$ is negligible. Where $\text{Guess}^\text{CCA}_{\text{DE}_{\text{IB}}, A}(M_b)$ for $b \in \{0, 1\}$ is defined as follows:
The adversary $A$ outputs an identity $id^*$ as the target identity.

The challenger runs $(mpk, msk) \leftarrow \text{DIB.Setup}(1^\lambda)$, and sends $mpk$ to $A_2$.

$A_2$ is allowed to make a number of private key queries and decryption queries for identity $id$:

**Private key query.** The adversary $A_2$ asks for the private key corresponding to any identity $id$ as long as $id \neq id^*$. The challenger correctly generates private key $sk_{id}$ for $id$ and returns to $A_2$.

**Decryption query.** $A_2$ issues decryption queries $C$ for identity $id$. The challenger responds with DIB.Dec($C, id, sk_{id}$) using private key $sk_{id}$ correctly generated for $id$.

The challenger samples $m$ from distribution $M_b ((M_0, M_1, state) \leftarrow A_1(1^\lambda))$, and computes $C^* = \text{DIB.Enc}(mpk, id^*, m)$, and then sends $C^*$ to $A_2$.

$A_2$ outputs its guess $b' \in \{0, 1\}$.

### 3 Identity-based ABO-TDF and its construction

In this section, we first introduce the notion of identity-based ABO-TDF. Then based on the basic IBE scheme of Bellare et al. [Bellare et al. 2012], we propose a concrete construction of IB-ABO-TDF under decisional linear (DLIN) assumption. Let $G$ be a finite cyclic group of prime order $p$ specified by a randomly chosen generator $g$. The DLIN assumption says that $g_{r_1}^{g_{r_2}^{g_{r_3}^{\cdots g_{r_d}^{d+1}}}}$ is pseudorandom given $g_1, \ldots, g_{d+1}, g_1^{r_1}, \ldots, g_d^{r_d}$ where $g_1, \ldots, g_{d+1} \leftarrow G$, $r_1, \ldots, r_d \leftarrow \mathbb{Z}_p$.

#### 3.1 Identity-based ABO-TDF

Identity-based ABO-TDF can be viewed as a specific kind of ABO-TDF with two variable $(id, tag)$ as a branch. The first component $id$ of branch $(id, tag)$ is user’s identity, and the second component $tag$ is the label. If and only if $id$ is lossy identity and $tag$ is lossy label, the branch $(id, tag)$ is lossy branch. In our construction, similarly, we follow the method of Bellare et al. [Bellare et al. 2012] for constructing IB-LTDF. In setup phrase, algorithm takes an additional auxiliary input from an auxiliary input space AuxSp. Lossy identities are determined by an auxiliary input which is hidden in the master public key. The definition of identity-based all-but-one trapdoor functions is described as follows.

**Definition 2** (IB-ABO-TDF). A collection of $(n, k)$-identity-based all-but-one trapdoor functions with the identity space IDSp, auxiliary input space AuxSp and label space TagSp, is a tuple of polynomial time algorithms $(\text{Setup}_{abo}, \text{KG}_{abo}, \text{Eval}_{abo}, \text{Inv}_{abo})$ with the following specifications:
Injective invertibility. For any \( (\mathcal{A}, \mathcal{B}) \) of lossy correctness, one can make private key query on identity \( id \), and \( \mathcal{B} \subset \text{IDSp} \times \text{TagSp} \) is a set of lossy branches.

\( \text{KG}_{\text{abo}}(mpk, msk, id) \). Given \( mpk, msk \), identity \( id \in \text{IDSp} \), the probabilistic algorithm outputs a private key \( sk_{id} \) with respect to the identity \( id \).

\( \text{Eval}_{\text{abo}}(mpk, id, tag, x) \). For any \( tag \in \text{TagSp} \), \( id \in \text{IDSp} \), the algorithm takes as input \( mpk, id, tag \), and \( x \in \{0, 1\}^n \), outputs a value \( C \).

\( \text{Inv}_{\text{abo}}(mpk, sk_{id}, C) \). The deterministic algorithm takes as input \( mpk \), private key \( sk_{id} \), and a value \( C \), outputs either \( x \in \{0, 1\}^n \) or \( \perp \).

We require that the following properties hold:

- Injective correctness and invertibility. For any \( (id, tag) \in \text{IDSp} \times \text{TagSp} \), if \( (id, tag) \notin \mathcal{B} \), where \( (mpk, msk, \mathcal{B}) \leftarrow \text{Setup}_{\text{abo}}(1^\lambda, aux, tag^*) \), the algorithm \( \text{Eval}_{\text{abo}}(mpk, id, tag, \cdot) \) computes a deterministic injective function over the domain \( \{0, 1\}^n \), which can be inverted using the private key \( sk_{id} \) corresponding to the given \( id \). Formally, \( (mpk, msk, \mathcal{B}) \leftarrow \text{Setup}_{\text{abo}}(1^\lambda, aux, tag^*) \), \( sk_{id} \leftarrow \text{KG}_{\text{abo}}(mpk, msk, id) \), \( (id, tag) \notin \mathcal{B} \), and \( x \in \{0, 1\}^n \)

\[
\Pr[\text{Inv}_{\text{abo}}(mpk, sk_{id}, \text{Eval}_{\text{abo}}(mpk, id, tag, x)) \neq x] \leq \operatorname{negl}(\lambda)
\]

- Lossiness. For any \( (id, tag) \in \text{IDSp} \times \text{TagSp} \), if \( (id, tag) \notin \mathcal{B} \), the algorithm \( \text{Eval}_{\text{abo}}(mpk, id, tag, \cdot) \) computes a deterministic function over the domain \( \{0, 1\}^n \) whose image has size at most \( 2^{n-k} \).

- Indistinguishability of lossy branch. For every probabilistic polynomial time algorithm \( \mathcal{A} \), the first output \( mpk_1 \) of \( \text{Setup}_{\text{abo}}(1^\lambda, aux_1, tag_1) \) and the first output \( mpk_2 \) of \( \text{Setup}_{\text{abo}}(1^\lambda, aux_2, tag_2) \) are computationally indistinguishable. Formally, the advantage

\[
\text{Adv}_{\text{IND-ABO-TDF}}^{\text{IND}}(\mathcal{A}) = \left| \Pr[\mathcal{A}(mpk_1, 1^\lambda)^{\text{KG}_{\text{abo}}(\cdot)} = 1] - \Pr[\mathcal{A}(mpk_2, 1^\lambda)^{\text{KG}_{\text{abo}}(\cdot)} = 1] \right|
\]

of \( \mathcal{A} \) is negligible, where \( (mpk_d, msk, \mathcal{B}) \leftarrow \text{Setup}_{\text{abo}}(1^\lambda, aux_d, tag_d) \) for \( d \in \{1, 2\} \). \( \text{KG}_{\text{abo}}(\cdot) \) denotes that \( \mathcal{A} \) can make private key query on identity \( id \) by calling \( \text{KG}_{\text{abo}} \) algorithm.

- Hard to find one-more lossy branch. Any probabilistic polynomial time algorithm \( \mathcal{A} \) that receives \( (mpk, id, tag) \) as input, where \( (id, tag) \in \mathcal{B} \), has only a negligible probability of outputting a pair \( (id', tag') \in \mathcal{B} \setminus \{id, tag\} \).

### 3.2 The construction of identity-based ABO-TDF

Fix a bilinear map \( e : G \times G \rightarrow G_T \), where \( G, G_T \) are groups of prime order \( p \). By \( 1, 1_T \) we denote the identity elements of \( G, G_T \), respectively. By \( G^* = \)
$G = \{1\}$ we denote the set of generators of $G$. For vectors $y = (y_0, y_1) \in \mathbb{Z}_p^2$, $\text{tag} = (tag_0, tag_1)$ and $id \in \mathbb{Z}_p$. We let $f_{id}(y) = (y_0 + y_1id)\mod p$. For any integer $n$ and any identity space $\text{IDSp} \subseteq \mathbb{Z}_p$, message space $\{0, 1\}^n$ and auxiliary input space $\text{AuxSp} \subseteq \mathbb{Z}_p$, the algorithms of IB-ABO-TDF are as follows.

**Setup_{abo}(1^\lambda, y, \text{tag}^*)**. Given auxiliary input $y = (1, 0) \in \mathbb{Z}_p^2$, $\text{tag}^* = (tag_0^*, tag_1^*) \in \mathbb{Z}_p^2$, let $g \overset{\$}{\leftarrow} G^*$, $t \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, $\bar{g} = g^t$. Then let $U \overset{\$}{\leftarrow} G$, $s \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, $H, \hat{H}, V_0, V_1, \hat{V}_0, \hat{V}_1, \mathbb{V}_1 \overset{\$}{\leftarrow} G^n$. It returns master public key $mpk = (g, \bar{g}, G, J, W_0, W_1, H, \hat{H}, V_0, V_1, \hat{V}_0, \hat{V}_1, U)$, where for $1 \leq i, j \leq n$,

$$G[i] = g^{s[i]}, \quad \hat{G}[i] = \bar{g}^{\bar{s}[i]}, \quad J[i, j] = H[j]^{s[i]} \hat{H}[j]^{\bar{s}[i]},$$

$$W_0[i, j] = V_0[j]^{s[i]} \hat{V}_0[j]^{\bar{s}[i]} (U^y g^{tag_0^*} s[i])^{\Delta(i, j)},$$

$$W_1[i, j] = V_1[j]^{s[i]} \hat{V}_1[j]^{\bar{s}[i]} (U^y g^{tag_1^*} s[i])^{\Delta(i, j)}$$

Where $\Delta(i, j) = 1$ if $i = j$ and 0 otherwise. The master secret key $msk = t$ and the set of lossy branches $\tilde{B} = \{(id, \text{tag})|f_{id}(y) = 0\mod p \land \text{tag} = \text{tag}^*\}$.

**KG_{abo}(mpk, msk, id)**. Given $mpk$, $msk$, identity $id \in \mathbb{Z}_p$, the algorithm computes decryption key $sk_{id} = (D_1, D_2, D_3, D_4)$, where $r \overset{\$}{\leftarrow} (\mathbb{Z}_p^*)^n$, $\hat{r} \overset{\$}{\leftarrow} \mathbb{Z}_p^n$, and for all $1 \leq i \leq n$

$$D_1[i] = (V_0[i] \cdot V_1[i]^{id} \cdot r[i] \cdot H[i]^{\tau[i]}), \quad D_2[i] = (\hat{V}_0[i] \cdot \hat{V}_1[i]^{id} \cdot \hat{H}[i]^{\tau[i]}),$$

$$D_3[i] = g^{-t r[i]}, \quad D_4[i] = g^{-t \bar{r}[i]}$$

**Eval_{abo}(mpk, id, tag, x)**. Given $mpk$, identity $id \in \mathbb{Z}_p$, $\text{tag} = (tag_0, tag_1) \in \mathbb{Z}_p^2$, input $x \in \{0, 1\}^n$, the algorithm computes the value $C = (C_1, C_2, C_3, C_4)$ where for $1 \leq j \leq n$, let $S[j] = G[j]^{-s[j]} f_{id}(tag_j)$

$$C_1 = \prod_{i=1}^{n} G[j]^{x[i]}, \quad C_2 = \prod_{i=1}^{n} \hat{G}[j]^{x[i]}$$

$$C_3[j] = S[j] \prod_{i=1}^{n} (W_0[i, j] W_1[i, j]^{id} \cdot x[i]), \quad C_4[j] = \prod_{i=1}^{n} J[i, j]^{x[i]}$$

**Inv_{abo}(mpk, sk_{id}, C)**. Given $mpk$, ciphertext $C = (C_1, C_2, C_3, C_4)$ for identity $id$, the algorithm returns $x \in \{0, 1\}^n$ where for $1 \leq j \leq n$, it sets $x[j] = 0$ if

$$e(C_1, D_1[j]) e(C_2, D_2[j]) e(C_3[j], D_3[j]) e(C_4[j], D_4[j]) = 1$$

and 1 otherwise.
Correctness and invertibility. For $1 \leq j \leq n$, let $I = (U_{f_{id}}(y)g_{f_{id}}(\tag^*))$
\[C_1 = \prod_{i=1}^{n} G[i]^x[i] = g^{(s,x)} \quad C_2 = \prod_{i=1}^{n} G[i]^x[i] = \tilde{g}^{(s,x)}\]
\[C_3[j] = S[j]\prod_{i=1}^{n} (W_0[i,j]W_1[i,j]d_j)x[i] = S[j]e(V_0[j]V_1[j]^d_i s[i]x[i], V_0[j]V_1[j]^d_i \emptyset[i]x[i] \Delta(i,j)) = (V_0[j] \cdot V_1[j]^d_i d_j s[i]x[i])(U_{f_{id}}(y)g_{f_{id}}(\tag^* - \tag))(s[j], x[j])\]
\[C_4[j] = \prod_{i=1}^{n} J[i,j]^x[i] = \prod_{i=1}^{n} H[j]^x[i]x[i] = H[j]^{(s,x)}\]

Thus
\[e(C_1, D_1[j])e(C_2, D_2[j])e(C_3[j], D_3[j])e(C_4[j], D_4[j]) = e((U_{f_{id}}(y)g_{f_{id}}(\tag^* - \tag))s[j], D_3[j])\]

Because we chose $s[i]$ to be non-zero modulo $p$, $f_{id}(y) = \text{1modp} \neq 0$, therefore, $(id, \tag) \notin \hat{B}$, if
\[e(C_1, D_1[j])e(C_2, D_2[j])e(C_3[j], D_3[j])e(C_4[j], D_4[j]) = 1,\]
then $x[j] = 0$, if the result of the pairing is never $1_T$, then the inversion algorithm will correctly recover $x[j] = 1$. 

Lossiness. On input a selective identity $id^* \in \text{IDSp}$, the algorithm Aux($id^*$) returns $y = (-id^*, 1)$. Obviously, $f_{id^*}(y) = \text{0modp}$, when $\tag = \tag^*$, we have $(id^*, \tag) \notin \hat{B}$. We show that if $(id, \tag) \notin \hat{B}$, then algorithm Eval_{abo}($mpk, id, \tag, \cdot$) evaluates a lossy function. Due to $f_{id^*}(y) = \text{0modp}$ and $\tag = \tag^*$, then the dependency of $C_3[j]$ on $x[j]$ vanishes. Examining $(C_1, C_2, C_3, C_4)$, we see that with $mpk$ fixed, the values $(s, x)$, ($\hat{s}, x$) determine the ciphertext $C$. Thus there are at most $p^2$ possible ciphertexts when $(id, \tag) \notin \hat{B}$. This means that $R = |\text{Eval}_{abo}(mpk, id, \tag, \cdot)| \leq p^2$. Moreover, the lossy branch of IB-ABO-TDF is universal. That is, if $x_1 \neq x_2$, for $s, \hat{s} \in \mathbb{Z}_p^n$, $\Pr[\text{Eval}_{abo}(mpk, id, \tag, x_1)] = \text{Eval}_{abo}(mpk, id, \tag, x_2)] = \Pr[(s, x_1), (\hat{s}, x_1)] = \Pr[(s, x_2), (\hat{s}, x_2))] = \frac{1}{p^2} \leq \frac{1}{|\hat{B}|}$. 

Indistinguishability of lossy branches. From Theorem 1, we can see that, under the decision linear (DLIN) assumption, it is hard to distinguish the master public key and random group elements. Therefore, the first output $mpk_1$ of Setup(1^λ, y_1, tag_1) and the first output $mpk_2$ of Setup(1^λ, y_2, tag_2) are computationally indistinguishable. 

Hard to find one-more lossy branch. We show that any probabilistic polynomial time adversary $A$ that receives $(mpk, id, \tag)$ as input, where $(id, \tag) \notin \hat{B}$, outputs a pair $(id', \tag')$ satisfying $(id', \tag') \neq (id, \tag)$ and $(id', \tag') \notin \hat{B}$.
with negligible probability. To see this, observe that the value $y$ and $\text{tag}^*$ are initially hidden by the public parameter $W_0[i,j], W_1[i,j]$, from Theorem 1, we know that $W_0[i,j], W_1[i,j]$ are indistinguishable from a random element in the group $G$ to any probabilistic polynomial time adversary $A$. However, $A$ could obtain the information that $u f_{id}(y) + f_{id}(\text{tag}^* - \text{tag}) = 0$ (let $U = g^u$). There are exactly $p^2$ pairs which satisfy this equation and each of them are equally likely. Therefore, we can conclude that the adversary has only negligible probability of outputing a pair $(id', \text{tag}') \in B \{id, \text{tag}\}$ without master secret key.

In order to prove the indistinguishability of lossy branches (i.e Theorem 1), we first prove the following lemma. That is, the basic ciphertext which contain certain ‘atoms’ from which, given an identity, one can reconstruct ciphertext of the resemble Bellar et al.’s basic IBE scheme, is indistinguishable from random group elements. The concrete games are described in table 2.

**Lemma 3.1.** Game $G_{real}$ and game $G_{random}$ are computationally indistinguishable under DLIN assumption.

**Proof.** This proof via a series of games $G_0, G_1, G_2, G_3$, where game $G_0$ is the game $G_{real}$ and game $G_3$ is the game $G_{random}$. Game $G_1$ is the same as game $G_0$ except $S \leftarrow G$ in challenge phase, Game $G_2$ is the same as game $G_1$ except $W_0 \leftarrow G$ in challenge phase. Game $G_3$ is the same as game $G_2$ except $W_1 \leftarrow G$ in challenge phase. We then show that for $i = 0, 1, 2$, $G_i$ and $G_{i+1}$ are computationally indistinguishable under DLIN assumption. It follows that the Lemma holds. The advantage of a distinguisher $B$ attacking DLIN assumption is denoted by $\text{Adv}_B^{\text{DLIN}}(\lambda)$.

**Claim 1.** $\Pr[G_0 \Rightarrow 1] - \Pr[G_1 \Rightarrow 1] \leq \text{Adv}_B^{\text{DLIN}}(\lambda).

**Proof.** We prove this claim by describing a distinguisher $B_1$ is given $(g, \hat{g}, g^s, \hat{g}^s, H, T)$ where $T$ is either $H^{s+k}$ or random. $B_1$ runs adversary $A$ responding to its queries as follows. When $A$ makes query to Initialize($i^3, y, \text{tag}$), $B_1$ chooses $y \in \mathbb{Z}_p^2$, $\text{tag} \in \mathbb{Z}_p^2$, $u, v \leftarrow \mathbb{Z}_p^2$, $v = (v_0, v_1), \hat{v} = (\hat{v}_0, \hat{v}_1) \leftarrow \mathbb{Z}_p^2$ and computes $\hat{H} = Hg^u, U = \hat{g}^v$, for $k = 0, 1, V_k = U^{-y^u}g^{\text{tag}v}g^{s\hat{v}}, \hat{V}_k = \hat{g}^\hat{v}_k$. $B_1$ returns $(g, \hat{g}, H, \hat{H}, V_0, \hat{V}_1, V_1, U)$ to $A$.

When $A$ makes private-key query to Getsk($id$), $B_1$ does the following: If $f_{id}(y) = 0$ then return $\bot$. Else $B_1$ chooses $r, r' \leftarrow \mathbb{Z}_p$, and computes

$$D_1 = U^{-f_{id}(y)r'} g^{f_{id}(\text{tag})+f_{id}(v)r'} H^{r'(f_{id}(v)+f_{id}(\text{tag}))}$$
$$D_2 = g^{r'r_{id}(v)} H^{r_{id}(y)} H^{uv'}, D_3 = g^{r'} H^{r_{id}(y)}, D_4 = \hat{g}^{-uv'}$$

Then $B_1$ returns $sk_{id} = (D_1, D_2, D_3, D_4)$ to $A$.

We claim that $sk_{id}$ is a correctly distributed and valid random private key for the identity $id$. To see this, let $h$ be such that $H = g^h$ and let $r = r' + \frac{hv'}{r_{id}(y)}$. 


Table 2: Games for the proof of Lemma 3.1. Border areas indicate the difference between the games

and $\hat{r} = ur'$. Then we have

$$D_1 = (V_0 V_1^d)^{tr} H^{tr} = (U^{-y_0} g^{a_{yo}} g^y (U^{-y_0} g^{a_{yo} y_0})^d)^{tr} H^{tr}$$

$$= (U - f_{id}(y) g_f a_d \tag{tag} g f_{id}(y))^{t(x - \frac{t(x + f_{id}(y))}{t(x + f_{id}(y))})} H^{tr}$$

$$= U - f_{id}(y)^r r^{h(x + f_{id}(y))} g(f_{id}(y) + f_{id}(y))^{r'}) g^{f_{id}(y) + f_{id}(y)} H^{-(f_{id}(y) + f_{id}(y))^{r'}} g^{b(x + f_{id}(y))}$$

$$= U - f_{id}(y)^r r^{g(x + f_{id}(y))} g(f_{id}(y) + f_{id}(y))^{r'} H^{-(f_{id}(y) + f_{id}(y))^{r'}} g^{b(x + f_{id}(y))}$$

$$= U - f_{id}(y)^r r^{f_{id}(y) + f_{id}(y))^{r'} H^{-(f_{id}(y) + f_{id}(y))^{r'}} g^{b(x + f_{id}(y))}$$
\[ D_2 = (V_0 V_1 \hat{y}^i)^r \hat{H}^v = (\hat{g}^{vy} \hat{g}^{vz} \hat{y}^i)^r \hat{H}^v = g^{f_{id}(\hat{y}) \left( \frac{v}{r} - \frac{h r'}{t f'_{id}(\hat{y})} \right) \hat{H}^v} \]
\[ = g^{f_{id}(\hat{y}) r' - \frac{h r' f_{id}(\hat{y})}{t f'_{id}(\hat{y})} \hat{H}^v} = g^{f_{id}(\hat{y}) r' \hat{H}^v} \]
\[ D_3 = g^{-tr} = g^{t (\frac{v}{r} - \frac{h r'}{t f'_{id}(\hat{y})})} \]
\[ D_4 = g^{-t \hat{u} \hat{v}^i} = g^{-u \hat{v}^i} \]

Since \( t, f_{id}(\hat{y}) \) are non-zero module \( p \) and \( r', \hat{r}' \) are uniform and independent in \( \mathbb{Z}_p \), \( r, \hat{r} \) are uniform as well. This matches the distribution of private key of identity \( id \) generated by Getsk. Thus, \( sk_{id} \) is a valid private key of the identity \( id \).

When \( A \) makes its Challenge query, \( B_1 \) computes \( S = T \hat{g}^{v^k} \), for \( k = 0, 1 \), do \( W_k = g^{v_k} \hat{g}^{v_k} \) and returns \( (g, \hat{g}^s, S, W_0, W_1) \) to \( A \). We can see that for \( k = 0, 1 \),
\[ W_k = g^{v_k} \hat{g}^{v_k} = (U^{v_k} \hat{g}^{v_k} U^{-v_k} g^{-t v_k} \hat{g}^{v_k}) \hat{g}^{v_k} = (U^{v_k} \hat{g}^{v_k} \hat{V}_k)^{\hat{V}_k} \]
If \( B_1 \) was given a DLIN instance, that is \( T = H^{s+\hat{k}}, \) then we have
\[ S = T \hat{g}^{v_k} = H^{s+\hat{k}} \hat{g}^{v_k} = H^s (H \hat{g}^{v_k}) = H^s \hat{H}^k \]
We see that \( B_1 \) simulates game \( G_0 \). Otherwise, \( B_1 \) was given a non-DLIN instance, i.e \( T \notin G \), then \( S \notin G \), \( B_1 \) simulates game \( G_1 \). Finally, \( A \) outputs \( d' \), \( B_1 \) also outputs \( d' \). So this completes the claim.

Claim 2. \( \Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1] \leq Adv_{B_2}^{DLIN}(\lambda) \).

Proof. Similar to claim 1, we design a simulator \( B_2 \) such that it is given \( (g, \hat{g}, \hat{g}^s, U, T) \) where \( T \) is either \( U^{s+\hat{k}} \) or random. \( B_2 \) runs adversary \( A \) responding to its queries as follows. When \( A \) makes query \( \text{Initialize}(\hat{i} \lambda, \hat{y}, \text{tag}) \), \( B_2 \) chooses \( \hat{y} \in \text{AuxSp}, \text{tag} \in \text{TagSp}, u, h, \hat{h} \triangleleft \mathbb{Z}_p, \hat{\nu} = (v_0, v_1), \bar{\nu} = (\tilde{v}_0, \tilde{v}_1) \triangleleft \mathbb{Z}_p^2, \) and computes \( H \leftarrow \hat{g}^{\hat{h}}, \hat{H} = \hat{g}^{\hat{h}}, U = g^u, \hat{V}_0 = U \hat{g}^{v_0}, \hat{V}_1 = \hat{g}^{v_1} \). \( B_2 \) returns \( (g, \hat{g}, H, \hat{H}, V_0, V_1, U) \) to adversary \( A \).

When \( A \) makes query private-key \( \text{Getsk}(id) \), \( B_2 \) does the following: If \( f_{id}(\hat{y}) = 0 \) then return \( \perp \). Else \( B_2 \) chooses \( r, \hat{r}' \triangleleft \mathbb{Z}_p \), and computes
\[ D_1 = g^{f_{id}(\hat{y}) r} \hat{H}^v, \quad D_2 = \hat{U}^r \hat{g}^{r f_{id}(\hat{y})} \hat{H}^{r'} \hat{U}^{-r}, \quad D_3 = \hat{g}^{-r}, \quad D_4 = g^{-tr} \hat{U}^{-\hat{r}} \]
Then \( B_2 \) returns \( sk_{id} = (D_1, D_2, D_3, D_4) \) to \( A \).

We claim that \( sk_{id} \) is a correctly distributed and valid random private key for the identity \( id \). To see this, let \( \hat{u} \) be such that \( \hat{U} = \hat{g}^\hat{u} \) and let \( \tilde{r} = \tilde{r} - \frac{\hat{u} \hat{r}}{r} \).
Then we have
\[
D_1 = (V_0 V_1^{id})^{tr} H^{t\hat{p}} = (\hat{U}^y g^v)^{tr} H^{t\hat{p}} \\
= (\hat{U}^y g^{f_{id}(v)})^{tr} H^{t\hat{p}} \\
= (\hat{U}^y g^{f_{id}(v)})^{tr} g^{ht(\frac{\hat{p}}{\hat{t}} - \frac{q^0}{\hat{t}})} \\
= (g^{(\hat{y}^t)} g^{f_{id}(v)})^{tr} g^{ht\hat{p}} g^{-ur} \\
= (g^{f_{id}(v)})^{tr} H^{t\hat{p}}
\]

\[
D_2 = (\hat{V}_0 \hat{V}_1^{id})^{tr} H^{\hat{p}} = (\hat{U}^\hat{y} g^{\hat{v}_{id}})^{tr} H^{\hat{p}} \\
= \hat{U}^r g^{f_{id}(\hat{v})} H^{(\frac{\hat{r}}{\hat{t}} - \frac{q^0}{\hat{t}})} \\
= \hat{U}^r g^{f_{id}(\hat{v})} g^{\hat{h}\hat{r}} g^{-\hat{u}r} \\
= \hat{U}^r g^{f_{id}(\hat{v})} H^{\hat{r}} \hat{U}^{-\hat{u}r}.
\]

\[
D_3 = g^{-tr} = \hat{g}^{-r} \\
D_4 = g^{-t\hat{p}} = g^{-t(\frac{\hat{p}}{\hat{t}} - \frac{q^0}{\hat{t}})} = g^{-t\hat{p}} g^{ur} = g^{-t\hat{p}} \hat{U}^{-\hat{u}r}
\]

Since \(t\), \(f_{id}(y)\) are non-zero module \(p\) and \(\hat{r}\) are uniform and independent in \(Z_p\), \(\hat{r}\) are uniform as well. This matches the distribution of private key of identity \(id\) generated by \(\text{Getsk}\). Thus, \(sk_{id}\) is a valid private key of the identity \(id\).

When \(A\) makes its Challenge query, \(B_2\) computes \(W_0 = (g^x)^{\hat{y}_0 + ta_{g^0} + v_0} (\hat{g}^{\hat{v}})^{v_0}\). \(T\), \(W_1 = (g^x)^{uy_1 + ta_{g^1} + v_1} (\hat{g}^{\hat{v}})^{v_1}\) and returns \((g, \hat{g}^{\hat{v}}, S, W_0, W_1)\) to \(A\). We can see that
\[
W_1 = (g^x)^{uy_1 + ta_{g^1} + v_1} (\hat{g}^{\hat{v}})^{v_1} = (U^y g^{ta_{g^1}} V_1^\hat{v}) \hat{V}_1^\hat{v}.
\]

If \(B_2\) was given a DLIN instance, that is \(T = \hat{U}^{s+\hat{s}}\), then we have
\[
W_0 = (g^x)^{uy_0 + ta_{g^0} + v_0} (\hat{g}^{\hat{v}})^{v_0} \cdot T \\
= (U^y g^{ta_{g^0}} \hat{U}^{-v_0} V_0^\hat{v})^{\hat{v}} (U^{-v_0} V_0^\hat{v})^{\hat{v}} \hat{U}^{s+\hat{s}} \\
= (U^y g^{ta_{g^0}} V_0^\hat{v})^{\hat{v}} \hat{U}^{s+\hat{s}} \\
= (U^y g^{ta_{g^0}} V_0^\hat{v})^{\hat{v}} \hat{V}_0^\hat{v}.
\]

We see that \(B_2\) simulates game \(G_1\) in this case. Otherwise, \(B_2\) was given a non-DLIN instance, i.e \(T \not\in \hat{x}\ G\), then \(W_0 \not\in \hat{x}\ G\). \(B_2\) simulates game \(G_2\). Finally, \(A\) outputs \(d'\), \(B_2\) also outputs \(d'\). So this completes the claim.

**Claim 3.** \(\Pr[G_2 \Rightarrow 1] - \Pr[G_3 \Rightarrow 1] \leq Adv_{B_2}^{DLIN}(\lambda)\).

**Proof.** The proof is identical to that of Claim 2. The only differences are, in the initialize(\(\hat{r}, y, \text{tag}\)) query phase, the simulator \(B_2\) lets \(V_0 = g^{\hat{v}_0}, V_1 = \hat{U} g^{\hat{v}_1}, V_0 = \hat{g}^{\hat{v}_0}, V_1 = \hat{U} g^{\hat{v}_1}\), in the private key query phase, lets \(\hat{r} = \frac{\hat{r}}{\hat{t}} - \frac{id\hat{u}r}{\hat{h}r}\), and
in the challenge phrase, the simulator $B_2$ computes $W_0 = (g^r)^{y_0 + t\alpha_0 + v_0(\hat{y})^\delta_0}$ and $W_1 = (g^r)^{y_1 + t\alpha_1 + 1(\hat{y})^\delta_1}$. 

**Theorem 1.** The first output $mpk_1$ of Setupobj($1^\lambda$, $y_1$, $\text{tag}_1$) and the first output $mpk_2$ of Setupobj($1^\lambda$, $y_2$, $\text{tag}_2$) are computationally indistinguishable.

**Proof.** We first prove that $mpk_0$ for $b \in \{0, 1\}$ and random group element matrices are computationally indistinguishable. It is obvious that the theorem follows. To prove $mpk_0$ generated by Setupobj($1^\lambda$, $y$, $\text{tag}$) is indistinguishable from random group parameter matrices and $R_0$ produces a random public parameter matrices. Below we argue that for every $l \in \{1, \ldots, n\}$, any distinguisher $A$ of the two games $R_{l-1}$ and $R_l$ can be used to distinguish $G_{\text{real}}$ from $G_{\text{random}}$. From Lemma 3.1, $R_{l-1}$ and $R_l$ are indistinguishable, therefore, this theorem holds.

Let adversary $A$ be a distinguisher of $R_{l-1}$ and $R_l$, the simulator $B$ is an adversary distinguishing $G_{\text{real}}$ from $G_{\text{random}}$.

**Simulating public parameter.** The simulator $B$ is given $(g, \hat{g}, H, \hat{H}, V_0, V_1, \tilde{V}_0, \tilde{V}_1, U) \leftarrow \text{Initialize}(1^\lambda, y, \text{tag})$ and $(G, \hat{G}, S, W_0, W_1) \leftarrow \text{Challenge}$ where $(G, \hat{G}, S, W_0, W_1)$ is either real or random. $B$ chooses $h, \hat{h} \leftarrow (\mathbb{Z}_p^*)^n$, $v_0, v_1, \tilde{V}_0, \tilde{V}_1$, $\hat{V}_1 \leftarrow \mathbb{Z}_p^n, s \leftarrow (\mathbb{Z}_p^*)^n, \tilde{s} \leftarrow \mathbb{Z}_p^n$. For $i = 1, \ldots, n$

- If $i \neq l$ then $H[i] \leftarrow g^{v_i[i]}, H[i] \leftarrow g^{s_i[i]}, \hat{V}_k[i] \leftarrow g^{v_i[k]}(k = 0, 1), G[i] \leftarrow g^{s_i[i]}, \hat{G}[i] \leftarrow g^{\hat{s}_i[i]}$. For $j = 1, \ldots, n$, $J[i, j] \leftarrow H[j]^s[i]H[j]^{\hat{s}[i]}).

- If $i = l$ then $H[i] \leftarrow H, \hat{H}[i] \leftarrow H, V_k[i] \leftarrow V_k, \hat{V}_k[i] \leftarrow \hat{V}_k(k = 0, 1), G[i] \leftarrow G, \hat{G}[i] \leftarrow \hat{G}$. For $j = 1, \ldots, n$, if $j \neq i$ then $J[i, j] \leftarrow G^{b[j]}G^{\hat{b}[j]}$, if $j = i$ then $J[i, i] \leftarrow S$.

- For $j = 1, \ldots, n$, $k = 0, 1$, if $i = j$ and $i \leq l$ then $W_k[i, j] \leftarrow g^h$; if $i = j$ and $i = l$ then $W_k[i, j] \leftarrow W_l$; otherwise

$$W_k[i, j] \leftarrow V_k[j]^s[i]^h[i]V_k[j]^s[i]^h[i](U^y, g^{t\alpha_0})^s[i, \Delta(i, j)]$$

$B$ then sends $\text{paras} = (g, \hat{g}, H, \hat{H}, V_0, V_1, \tilde{V}_0, \tilde{V}_1, G, \hat{G}, J, W_0, W_1, U)$ to $A$.

**Simulating private key.** When the simulator $B$ simulates the private key $sk_{id}$ for identity $id$ which is chosen by the adversary $A$. $B$ first queries its own Getsk oracle and is given $(D_1, D_2, D_3, D_4) \leftarrow \text{Getsk}(id)$. If $f_{id}(y) = 0$ then $B$ returns $\perp$. Otherwise, $B$ chooses $t' \leftarrow (\mathbb{Z}_p^*)^n, t' \leftarrow (\mathbb{Z}_p^*)^n$. For $i = 1, \ldots, n$

- If $i \neq l$ then $D_1[i] \leftarrow (V_0[i]V_1[i][id]a[i]H[i][t'[i]], D_2[i] \leftarrow g^{f_{id}(y)}t'[i], D_3[i] \leftarrow g^{-r[i]}, D_4[i] \leftarrow g^{-r[i]}$.

- If $i = l$ then $(D_1[i], D_2[i], D_3[i], D_4[i]) \leftarrow (D_1, D_2, D_3, D_4)$.
Where $r[i] = r'[i]$, $\hat{r}[i] = \hat{r}'[i]$ for $i \neq l$. Here $t$ is the master secret key, so that $\hat{g} = g^t$. Because $r'[i], \hat{r}'[i]$ are random, so the above simulation matches the distribution of the private key.

Note that if $(G, \hat{G}, S, W_0, W_1)$ is real, $B$ perfectly simulates the game $R_{l-1}$, but if $(G, \hat{G}, S, W_0, W_1)$ is random, $B$ perfectly simulates the game $R_l$. Finally, $B$ outputs what $A$ outputs. Since $B$ perfectly simulates game $R_{l-1}$ or game $R_l$ depending on the $(G, \hat{G}, S, W_0, W_1)$. This completes the proof of the theorem.

4 CCA-secure DIBE scheme

Given the identity space IDSp and auxiliary input space AuxSp, let $\Pi_{\text{LF}} = (\text{LF.Setup}, \text{LF.KG}, \text{LF.Eval}, \text{LF.Inv})$ be an identity-based lossy trapdoor function with $2^{r_{\text{LF}}}$-bounded lossy function range $R_{\text{LF}}$ (i.e., in the lossy mode, the image of $\Pi_{\text{LF}}$ has size at most $2^{r_{\text{LF}}}$), let $\Pi_{\text{abo}} = (\text{Setup}_{\text{abo}}, \text{KG}_{\text{abo}}, \text{Eval}_{\text{abo}}, \text{Inv}_{\text{abo}})$ be an identity-based all-but-one trapdoor function with branches set $B = \text{IDSp} \times \text{TagSp}$ (TagSp is tag space) and with $2^{r_{\text{abo}}}$-bounded lossy function range $R_{\text{abo}}$, and let $\mathcal{H}_{\text{tc}} = (K_{\text{tc}}, H_{\text{tc}})$ be a target collision-resistant hash function with $2^{r_{\text{tc}}}$-bounded hash range $R_{\text{tc}} \subseteq \text{TagSp}\{\text{tag}^*\}$. We assume that the DIBE scheme has message space $\{0, 1\}^t$. Our deterministic identity-based encryption scheme $\text{DEIB} = (\text{DIB.Setup}, \text{DIB.Der}, \text{DIB.Enc}, \text{DIB.Dec})$ is defined as follows.

**DIB.Setup**$(1^\lambda, aux_0, aux_1) \xleftarrow{\$} \text{AuxSp}, (mpk}_{\text{LF}}, msk_{\text{LF}}) \leftarrow \text{LF.Setup}(1^\lambda, aux_0), (mpk}_{\text{abo}}, msk_{\text{abo}}) \leftarrow \text{Setup}_{\text{abo}}(1^\lambda, aux_1, \text{tag}^*), k_{\text{tc}} \xleftarrow{\$} K_{\text{tc}}. \text{Return } mpk = (mpk}_{\text{abo}}, k_{\text{tc}}, msk = msk_{\text{LF}}.

**DIB.Der**$(mpk, msk, id)$. $sk_{id} \leftarrow \text{KG}(mpk}_{\text{LF}}, msk_{\text{LF}}, id)$. Return $sk_{id}$.

**DIB.Enc**$(mpk, id, m, h \leftarrow H_{\text{tc}}(m), c_1 \leftarrow \text{LF.Eval}(mpk}_{\text{LF}, id, m), c_2 \leftarrow \text{Eval}_{\text{abo}}(mpk}_{\text{abo}, id, h, m)$. Return $C = \|c_1\|c_2$.

**DIB.Dec**$(mpk, id, sk_{id}, C)$. Parse $C$ as $C = h\|c_1\|c_2$, compute $m' \leftarrow \text{LF.Inv}(mpk}_{\text{abo}, sk_{id}, c_1), C' \leftarrow \text{DIB.Enc}(mpk, id, m')$. If $C' = C$ then return $m'$, otherwise return $\bot$.

Note that consistency of the above scheme follows from the fact that the particular auxiliary input $aux_0$ is a constant and independent to the identity, and the range of the tcr hash function does not include the lossy branch of the identity-based ABO-TDF. So both $\text{LF.Eval}(mpk}_{\text{LF}, id, \cdot}$ and $\text{Eval}_{\text{abo}}(mpk}_{\text{abo}, id, h, \cdot}$ are injective trapdoor functions. For all output $(mpk, msk)$ by DIB.Setup and all $m \in \{0, 1\}^t$, there is exactly one string $C$ such that DIB.Dec outputs $m$.

We now turn to security.

**Theorem 2.** If $\Pi_{\text{LF}}$ is a selective-id secure identity-based lossy trapdoor functions with universal lossy mode, $\Pi_{\text{abo}}$ is a universal identity-based all-but-one trapdoor functions, and $\mathcal{H}_{\text{tc}}$ is universal TCR hash function, if for $(t, l)$-sources $M_0, M_1$, and any $\epsilon > 0$ such that $t \geq r_{\text{LF}} + r_{\text{abo}} + r_{\text{tc}} + 2 \log(\frac{1}{\epsilon})$, [insert citation here].
then the deterministic identity-based encryption scheme DIBE described above is selective-id PRIV1-secure against chosen ciphertext attacks.

Proof. The proof proceeds via a sequence of games $G_0, G_1, \ldots, G_6$, where game $G_0$ is the original PRIV1-IND-ID-CCA game, $G_6$ will be independent of the any underlying distribution imposed by the adversary. Then we show that for all $i = 0, \ldots, 5$, game $G_i$ and $G_{i+1}$ are (computationally or statistically) indistinguishable. It follows that the deterministic IBE scheme is PRIV1-IND-ID-CCA secure.

$G_1$: This game is the same as the game $G_0$ except that the auxiliary input and label $(aux_1, tag^*)$ of IB-ABO-TDF is replaced by $(aux_1(id^*), h^*)$, where $h^* = H_{tcr}(m^*)$, $id^*$ is the target identity chosen in advance by the adversary, $m^*$ is chosen from $(t, l)$-source $M_b (b \in \{0, 1\})$ by the simulator, and $aux_1(id^*)$ is generated by an auxiliary input generator $Aux_1(id^*)$ which takes an input an identity in $IDS_p$ and returns an auxiliary input associated with the identity.

$G_2$: This game is identical to game $G_1$ except that the decryption oracle rejects all the ciphertext $C = h||c_1||c_2$ such that $h = h^*$.

$G_3$: This game is the same as the game $G_2$, the only change is to decryption oracle, in which if the adversary submits a ciphertext $C = h||c_1||c_2$ for decryption, such that $(id, h)$ is a lossy branch, then the decryption oracle immediately outputs reject and halts.

$G_4$: This game is the same as game $G_3$ except that the decryption oracle decrypted using the master secret key $msk_{abo}$ of IB-ABO-TDF. That is to say, when the adversary submits a ciphertext $C = h||c_1||c_2$ for decryption, the challenger computes $m' \leftarrow Inv_{abo}(mpk_{abo}, sk_{id}, c_2)$ (Note that the challenger operates the algorithm $KG_{abo}(msk_{abo}, id)$ and generates $sk_{id}$ for identity $id$), and $C' \leftarrow DIB.Enc(mpk, id, m')$. Then it checks whether $C' = C$. If not, it outputs $\perp$, otherwise outputs $m'$.

$G_5$: This game is the same as the game $G_4$, the only change is to the algorithm DIB.Setup, in which we replace the injective function with a lossy one. Formally, in the algorithm DIB.Setup, we replace $(mpk_{lf}, msk_{lf}) \leftarrow LF.Setup(1^\lambda, aux_0)$ with $(mpk_{lf}, msk_{lf}) \leftarrow LF.Setup(1^\lambda, aux_0(id^*))$, where $aux_0(id^*)$ is generated by an auxiliary input generator $Aux_0(id^*)$ and $id^*$ is the target identity chosen in advance by the adversary.

$G_6$: This game is the same as the game $G_5$ except that the challenge ciphertext is sampled uniformly from the ciphertext space instead of encrypting the message $m$ sampled from the $(t, l)$-source $M_b$. Formally, in the challenge phase, the challenger chooses randomly $h^* \leftarrow R_{tcr}$, $c_1^* \leftarrow R_{lf}$, $c_2^* \leftarrow R_{abo, h^*}$ as the challenge ciphertext $C^* = h^*||c_1^*||c_2^*$. \newline

Claim 4. Game $G_0$ and game $G_1$ are computationally indistinguishable, given the indistinguishability of the lossy branch of IB-ABO-TDF.

Proof. We prove this claim by describing an IB-ABO-TDF distinguisher
algorithm $B$ that receives $mpk_{abo}$ as input where $mpk_{abo}$ is either $(mpk_{abo}, msk_{abo}) \leftarrow \text{Setup}_{abo}(1^\lambda, aux_1, \text{tag}^*)$ or $(mpk_{abo}, msk_{abo}) \leftarrow \text{Setup}_{abo}(1^\lambda, aux_1(id^*, h^*), h^*)$, where $h^* = H_{tcr}(m^*)$, $id^*$ is the target identity chosen in advance by the adversary and $m^*$ is chosen from $(t, l)$-source $M_b (b \in \{0, 1\})$ by $B$.

The distinguisher $B$ operates by implementing DIB.Setup, DIB.Der, DIB.Dec and challenge. In the DIB.Setup phase, $B$ runs $(mpk_{LF}, msk_{LF}) \leftarrow \text{LF.Setup}(1^\lambda, aux_0)$ and chooses $k_{tcr} \leftarrow K_{tcr}$. The public key is output as $mpk = (mpk_{LF}, mpk_{abo}, k_{tcr})$. We point out that $B$ knows the injective trapdoor $msk_{LF}$, but does not know the trapdoor $msk_{abo}$. DIB.Der, DIB.Dec are implemented just as game $G_0$ and game $G_1$. Note that the only secret information DIB.Der and DIB.Dec need to operate is $msk_{LF}$, which the distinguisher knows. Likewise, Challenge is implemented just as in all the games. Therefore, any difference in behavior between game $G_0$ and game $G_1$ immediately breaks the hardness of distinguishing a lossy branch from an injective branch of the identity-based ABO trapdoor functions collection.

**Claim 5.** Game $G_1$ and game $G_2$ are computationally indistinguishable, given the target collision-resistant property of the hash function $H_{tcr}$.

**Proof.** We begin by observing that game $G_1$ or game $G_2$ behave equivalently unless an event $E$ happens, which is that the adversary makes a query $C = h \| c_1 \| c_2$ to its decryption oracle, where $h = h^*$. We then show that event $E$ happens with negligible probability. There are two possibilities to consider for its decryption. The first is that $C$ is the ciphertext corresponding to $m^*$. But by the unique encryption property of deterministic encryption that $m^*$ has only one valid ciphertext, namely $C^*$, which the adversary is not allowed to query to its decryption oracle. So in fact this possibility cannot occur. The second possibility is that $C$ decrypts to some $m \neq m^*$. In this case, we can find a valid target-collision $(m, m^*)$ of hash function $H_{tcr}$. By the collision-resistant property of $H_{tcr}$, we conclude that event $E$ happens with negligible probability.

**Claim 6.** Game $G_2$ and game $G_3$ are computationally indistinguishable, given the hardness of finding one-more lossy branch of IB-ABO-TDF.

**Proof.** Let $F$ be the event that the adversary makes a legal decryption query of the form $C = h \| c_1 \| c_2$, such that $(id, h)$ is lossy branch. It is clear that game $G_2$ and game $G_3$ proceed identically until the event $F$ happens. We then show that the event $F$ happens with negligible probability.

Note that, if $h = h^*$, the decryption oracle rejects the ciphertext in both games. Therefore, the $(id, h)$ is a new lossy branch of IB-ABO-TDF. By the hardness of finding one-more lossy branch property of IB-ABO-TDF, the event $F$ happens with negligible probability, and hence the claim follows.

**Claim 7.** Game $G_3$ and game $G_4$ are equivalent.

**Proof.** The only difference between game $G_3$ and game $G_4$ is in the implementation of decryption oracle. We show that decryption oracle is equiv-
alent in the two games. In both games, when the challenger receives a legal decryption query of the form $C = h||c_1||c_2$ from the adversary. It checks that $c_1 \leftarrow \text{LF.Eval}(\text{mpk}_{\text{LF}}, id, x)$, $c_2 \leftarrow \text{Eval}_{abo}(\text{mpk}_{abo}, id, h, x)$ for some $x$ that they compute (in different ways), and outputs $\bot$ if not. It suffices to show that such $x$ is unique. Note that, if $h = h^*$ or $(id, h)$ is lossy branch, the decryption oracle outputs reject. Therefore, $\Pi_{\text{LF}}$ and $\Pi_{abo}$ are both injective, and there is a unique $x$ such that $(c_1, c_2) = (\text{LF.Eval}(\text{mpk}_{\text{LF}}, id, x), \text{Eval}_{abo}(\text{mpk}_{abo}, id, h, x))$. The both implementations of decryption oracle find the $x$.

Claim 8. Game $G_4$ and game $G_5$ are computationally indistinguishable, given the indistinguishability of the injective and lossy functions of IB-LTDF.

Proof. We prove this claim by describing an IB-LTDF distinguisher $\mathcal{B}$ that receives $\text{mpk}_{\text{LF}}$ as input where $\text{mpk}_{\text{LF}}$ was either generated by $\text{LF.Setup}(1^\lambda, \text{aux}_0)$ or generated by $\text{LF.Setup}(1^\lambda, \text{aux}_0(id^*))$. Note that the distinguisher $\mathcal{B}$ knows the trapdoor $\text{msk}_{abo}$ of IB-ABO-TDF, but does not know the trapdoor $\text{msk}_{\text{LF}}$ corresponding to $\text{mpk}_{\text{LF}}$. $\mathcal{B}$ interacts with the adversary as follows.

In the setup phase, $\mathcal{B}$ runs $(\text{mpk}_{abo}, \text{msk}_{abo}) \leftarrow \text{Setup}_{abo}(1^\lambda, \text{aux}_3(id^*)), h^* = H_{tcr}(m^*)$, $K_{tcr} \leftarrow K_{tcr}$ and outputs public key $\text{mpk} = (\text{mpk}_{\text{LF}}, \text{mpk}_{abo}, k_{tcr})$. When the adversary makes a legal private key query for identity $id$ (i.e., $id \neq id^*$), $\mathcal{B}$ obtains $sk_{id}$ by querying its own private key extracting oracle on the identity $id$, then forwards to the adversary. When the adversary makes a legal decryption query $C = h||c_1||c_2$ for any identity $id$, $\mathcal{B}$ can compute the private key for identity $id$ using the trapdoor $\text{msk}_{abo}$, then runs the algorithm $\text{Inv}_{abo}$ and responds the adversary. Challenge phase is implemented just as in both games.

It is easy to see that the distinguisher $\mathcal{B}$ perfectly simulates game $G_4$ or game $G_5$ depending on whether $\text{mpk}_{\text{LF}}$ results in an injective or lossy function (respectively). By the indistinguishability of injective and lossy functions of IB-LTDF, the claim holds.

Claim 9. Game $G_5$ and game $G_6$ are $3\varepsilon$-close.

Proof. We proceed via two sub-games $G_{5,1}, G_{5,2}$. In sub-game $G_{5,1}$, we modify the challenge ciphertext of game $G_5$ so that $h^* \leftarrow H_{tcr}(m^*), c_1^* \leftarrow \text{LF.Eval}(\text{mpk}_{\text{LF}}, id, m^*), c_2^* \leftarrow R_{abo}$. The sub-game $G_{5,2}$ is identical to sub-game $G_{5,1}$ except that $c_1^* \leftarrow R_{LF}$. Blow we will show that game $G_5$ and sub-game $G_{5,1}$, sub-game $G_{5,1}$ and sub-game $G_{5,2}$, sub-game $G_{5,2}$ and game $G_6$ are statistically indistinguishable respectively.

In game $G_5$, let $X = m$ and $Z = h^*||c_1^*$, then we have $|Z| \leq 2^{r_{tcr} + r_{\text{LF}}}$ due to the universal TCR hash function $H_{tcr}$ and the IB-LTDF with the lossy mode. By the hypothesis that $t \geq r_{\text{LF}} + r_{abo} + r_{tcr} + 2\log(\frac{1}{\varepsilon})$ and Chain Rule, we have $H_{\infty}(X|Z) = H_{\infty}(m^*|h^*||c_1^*) \geq r_{abo} + 2\log(\frac{1}{\varepsilon})$, generalized LHL shows that $c_2^*$ is $\varepsilon$-close to uniform on the range of $\Pi_{abo}$ given $h^*||c_1^*$. That is to say, $G_5$ and sub-game $G_{5,1}$ are $\varepsilon$-close. Similarly, in sub-game $G_{5,1}$, we take $X = m$ and $Z = h^*$. According to the Chain Rule and the Generalized LHL, we can show that $c_1^*$ is
$\varepsilon$-close to uniform on the range $R_{LF}$. In sub-game $G_{5,2}$, we take $X = m$, and according to the standard LHL (i.e., the Generalized LHL with empty $Z$), we can conclude that $h^*$ is $\varepsilon$-close to uniform on its range as well. The claim holds.

**Claim 10.** In game $G_6$, adversary has no advantage to win the game.

**Proof.** Obviously, when executed in game $G_6$, $h^*$, $c_1^*$ and $c_2^*$ are chosen uniformly and independent of all other variables, including $b$. It is easy to see that the adversary has no advantage in the game. This claim follows.

### 5 Conclusions

In this paper, we introduced a notion of identity-based all-but-one trapdoor functions, which is an extension of all-but-one trapdoor functions in the identity-based setting. Based on the Bellare et al.’s identity-based lossy trapdoor functions [Bellare et al. 2012], we gave a concrete construction of IB-ABO-TDF and proved its security under DLIN assumption. Based on an IB-LTDF and our IB-ABO-TDF, we proposed a CCA-secure deterministic IBE scheme in the selective-id attack model. A future direction is to construct CCA-secure DIBE scheme in the adaptive case.

### Acknowledgements

This work is supported by the National Natural Science Foundation of China (61802241, 61572303, 61772326, 61402015, 61802242), National Key R and D Program of China (No.2017YFB0802000), the National Cryptography Development Fund during the 13th Five-year Plan Period (MMJJ20180217), the Foundation of State Key Laboratory of Information Security (2017-MS-03), the Fundamental Research Funds for the Central Universities (GK201702004), the Natural Science Basic Research plan in Shannxi Province of China (2017JM6048), the project of science and technology in Baoji City (15RKX-1-5-8), Key project of Baoji University of Arts and Sciences (ZK15027).

### References


