Identity-Based Encryption Secure Against Selective Opening Chosen-Ciphertext Attack

Junzuo Lai*, Robert H. Deng**, Shengli Liu***, Jian Weng†, and Yunlei Zhao‡

Abstract. Security against selective opening attack (SOA) requires that in a multi-user setting, even if an adversary has access to all ciphertexts from users, and adaptively corrupts some fraction of the users by exposing not only their messages but also the random coins, the remaining unopened messages retain their privacy. Recently, Bellare, Waters and Yilek considered SOA-security in the identity-based setting, and presented the first identity-based encryption (IBE) schemes that are proven secure against selective opening chosen plaintext attack (SO-CPA). However, how to achieve SO-CCA security for IBE is still open.
In this paper, we introduce a new primitive called extractable IBE, which is a hybrid of one-bit IBE and identity-based key encapsulation mechanism (IB-KEM), and define its IND-ID-CCA security notion. We present a generic construction of SO-CCA secure IBE from an IND-ID-CCA secure extractable IBE with "One-Sided Public Openability"(1SPO), a collision-resistant hash function and a strengthened cross-authentication code. Finally, we propose two concrete constructions of extractable 1SPO-IBE schemes, resulting in the first simulation-based SO-CCA secure IBE schemes without random oracles.

Key words: identity-based encryption, chosen ciphertext security, selective opening security

1 Introduction

Security against Chosen-Plaintext Attack (CPA) and Security against Chosen-Ciphertext Attack (CCA) are now well-accepted security notions for encryption. However, they may not suffice in some scenarios. For example, in a secure multi-party computation protocol, the communications among parties are encrypted, but an adversary may corrupt some parties to obtain not only their messages, but also the random coins used to encrypt the messages. This is the so-called "selective opening attack" (SOA). The traditional CPA (CCA) security does not imply SOA-security [1].

IND-SOA Security vs. SIM-SOA Security. There are two ways to formalize the SOA-security notion [2, 4, 20] for encryption, namely indistinguishability-based one (IND-SOA) and simulation-based one (SIM-SOA). IND-SOA security requires that no probabilistic polynomial-time (PPT) adversary can distinguish an unopened ciphertext from an encryption of a fresh message, which is distributed according to the conditional probability distribution (conditioned on the opened ciphertexts). Such a security notion requires that the joint plaintext distribution should be "efficiently conditionally re-samplable", which restricts SOA security to limited settings. To eliminate this restriction, the so-called full-IND-SOA security [5] was suggested. Unfortunately, there have been no known encryption schemes with full-IND-SOA security up to now. On the other hand, SIM-SOA security requires that anything that can be computed by a PPT adversary from all the ciphertexts and the opened messages together with the corresponding randomness can also be computed by

* Department of Computer Science, Jinan University. Email: laijunzuo@gmail.com
** School of Information Systems, Singapore Management University. Email: robertdeng@smu.edu.sg
*** Department of Computer Science and Engineering, Shanghai Jiao Tong University. Email: slliu@sjtu.edu.cn
† Department of Computer Science, Jinan University. Email: cryptjweng@gmail.com
‡ Software School, Fudan University. Email: yunleizhao@gmail.com

a PPT simulator with only the opened messages. SIM-SOA security imposes no limitation on the message distribution, and it implies IND-SOA security.

The SOA-security (IND-SOA vs. SIM-SOA) is further classified into two notions, security against selective opening chosen-plaintext attacks (IND-SO-CPA vs. SIM-SO-CPA) and that against selective opening chosen-ciphertext attacks (IND-SO-CCA vs. SIM-SO-CCA), depending on whether the adversary has access to a decryption oracle or not.

SOA for PKE. The initial work about SOA security for encryption was done in the traditional public-key encryption (PKE) field. In [2], Bellare, Hofheinz and Yilek showed that any lossy encryption is able to achieve IND-SO-CPA security, and SIM-SOA security is achievable as well if the lossy encryption is "efficiently openable". This result suggests the existence of many IND-SO-CPA secure PKEs based on number-theoretic assumptions, such as the Decisional Diffie-Hellman (DDH), Decisional Composite Residuosity (DCR) and Quadratic Residuosity (QR), and lattices-related assumptions [27, 16, 18, 19, 6, 28, 24]. Later, Hemenway et al. [17] showed that both re-randomizable public-key encryption and statistically-hiding $\binom{2}{1}$-oblivious transfer imply lossy encryption.

In [17], Hemenway et al. also proposed a paradigm of constructing IND-SO-CCA secure PKE from selective-tag weakly secure and separable tag-based PKE with the help of chameleon hashing. Hofheinz [21] showed how to get SO-CCA secure PKE with compact ciphertexts. Fehr et al. [15] proved that sender-equivocable (NC-CCA) security implies SIM-SO-CCA security, and showed how to construct PKE schemes with NC-CCA security based on hash proof systems with explainable domains and L-cross-authentication codes (L-XAC, in short). Recently, Huang et al. [22, 23] showed that using the method proposed in [15] to construct SIM-SO-CCA secure PKE, L-XAC needs to be *strong*.

SOA for IBE. Compared with SOA security for PKE, SOA-secure IBE is lagged behind. The subtlety of proving security for IBE comes from the fact that a key generation oracle should be provided to an adversary to answer private key queries with respect to different identities, and the adversary is free to choose the target identity. It was not until 2011 that the question how to build SOA-secure IBE was answered by Bellare et al. in [3]. Bellare et al. [3] proposed a general paradigm to achieve SIM-SO-CPA security from IND-ID-CPA secure and "One-Sided Publicly Openable" (1SPO) IBE schemes. They also presented two 1SPO IND-ID-CPA IBE schemes without random oracles, one based on the Boyen-Waters anonymous IBE [9] and the other based on Water's dual-system approach [30], yielding two SIM-SO-CPA secure IBE schemes. The second SIM-SO-CPA secure IBE scheme proposed in [3] can be extended to construct the first SIM-SO-CPA secure hierarchical identity-based encryption (HIBE) scheme without random oracles. One may hope to obtain SIM-SO-CCA secure IBEs by applying the BCHK transform [7] to SIM-SO-CPA secure HIBEs. Unfortunately, as mentioned in [3], the BCHK transform [7] does not work in the SOA setting. Consequently, how to construct SIM-SO-CCA secure IBEs has been left as an open question.

Our contribution. We answer the open question of achieving SIM-SO-CCA secure IBE with a new primitive called extractable IBE with One-Sided Public Openability (extractable 1SPO-IBE, in short) and a *strengthened* cross authentication codes (XAC).

- We define a new primitive named extractable 1SPO-IBE and its IND-ID-CCA security notion.

- We define a new property of XAC: *semi-uniqueness*. If an XAC is strong and semi-unique, we say it is a *strengthened* XAC. We also show that the efficient construction of XAC proposed by Fehr et al. [15] is a strengthened XAC actually.
- We propose a paradigm of building $\mathsf{SIM\text{-}SO\text{-}CCA}$ secure IBE from $\mathsf{IND\text{-}ID\text{-}CCA}$ secure extractable 1SPO-IBE, collision-resistant hash function and strengthened XAC. Our approach follows the line of [15], which achieves $\mathsf{SIM\text{-}SO\text{-}CCA}$ secure PKE from hash proof systems with explainable domains and XAC. Our result further highlights the significance of Fehr et al.'s work [15] in achieving $\mathsf{SIM\text{-}SO\text{-}CCA}$ security.
- We construct extractable 1SPO-IBE schemes without random oracles by adapting anonymous IBEs, including the anonymous extension of Lewko-Waters IBE scheme [25] by De Caro, Iovino and Persiano [13] and the Boyen-Waters anonymous IBE [9].

Extractable 1SPO-IBE. Extractable IBE combines one-bit IBE and identity-based key encapsulation mechanism (IB-KEM). The message space of extractable IBE is $\{0, 1\}$. An encryption of 1 under identity ID also encapsulates a session key K, behaving like IB-KEM. More precisely, $(C, K) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 1; R)$ and $C \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0; R')$, where PK_{ex} is the public parameter and R, R' are the randomness used in encryption. If C is from the encryption of 1 under ID, the decryption algorithm, $(b, K) \leftarrow \mathsf{Decrypt}_{ex}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, C)$, is able to use the private key $\mathsf{SK}_{\mathsf{ID}}$ to recover message $b = 1$ as well as the encapsulated session key K. As for an encryption of 0, say $C = \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0; R')$, the decryption algorithm can recover message $b = 0$ but generate a *uniformly random* key K as well.

The security of extractable IBE requires that given a challenge ciphertext C^* and a challenge key K^* under some identity ID^*, no PPT adversary can distinguish, except with negligible advantage, whether C^* is an encryption of 1 under identity ID^* and K^* is the encapsulated key of C^*, or C^* is an encryption of 0 under identity ID^* and K^* is a uniformly random key, even if the adversary has access to a key generation oracle for private key $\mathsf{SK}_{\mathsf{ID}}$ with $\mathsf{ID} \neq \mathsf{ID}^*$ and a decryption oracle to decrypt ciphertexts other than C^* under ID^*. In formula,

$$\mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 1; R) \overset{c}{\approx} \left(\mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0; R'),\ K'\right),$$

for all PPT adversaries getting access to key generation and decryption oracles, where K' is chosen uniformly at random from the session key space. Obviously, the security notion of extractable IBE inherits $\mathsf{IND\text{-}ID\text{-}CCA}$ security of one-bit IBE and $\mathsf{IND\text{-}ID\text{-}CCA}$ security of IB-KEM.

An extractable IBE is called *one-sided publicly openable* (1SPO), if there exists a PPT *public* algorithm POpen as follows: given $C = \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0; R)$, it outputs random coins R' which is uniformly distributed subject to $C = \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0; R')$. Note that, if an extractable 1SPO-IBE scheme is $\mathsf{IND\text{-}ID\text{-}CPA/CCA}$ secure, then given $(\hat{C}, \hat{K}) = \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 1; \hat{R})$, algorithm POpen is able to output random coins $\hat{R}'$, subject to $\hat{C} = \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0; \hat{R}')$. Hence, $\mathsf{IND\text{-}ID\text{-}CPA/CCA}$ secure extractable 1SPO-IBE implies that one can use algorithm POpen to explain/open any valid ciphertext C (i.e., an encryption of 1 or 0) as an encryption of 0. We emphasize that this explanation is done without any secret information.

One-sided public openability [3] is an IBE-analogue of a weak form of deniable PKE [11] (which plays an essential role in the construction of $\mathsf{NC\text{-}CPA/CCA}$ secure PKE in [15], consequently achieving $\mathsf{SIM\text{-}SO\text{-}CPA/CCA}$ secure PKE). In [3], Bellare et al. used one-bit 1SPO-IBE to construct $\mathsf{SIM\text{-}SO\text{-}CPA}$ secure IBE.

SIM-SO-CCA secure IBE from extractable 1SPO-IBE. We follow the line of [15], which achieves $\mathsf{SIM\text{-}SO\text{-}CCA}$ secure PKE from sender-equivocable or weak deniable encryption and XAC.

We give a high-level description on how to construct a $\mathsf{SIM\text{-}SO\text{-}CCA}$ secure IBE scheme from an extractable 1SPO-IBE scheme characterized by $(\mathsf{Encrypt}_{ex}, \mathsf{Decrypt}_{ex})$, with the help of a collision-resistant hash function H and a strengthened $\ell+1$-cross-authentication code XAC.

First, we roughly recall the notion of cross-authentication code XAC, which was introduced in [15]. In an $\ell+1$-cross-authentication code XAC, an authentication tag T can be computed from a *list* of random keys $K_1, \ldots, K_{\ell+1}$ (without a designated message) using algorithm XAuth. The XVer algorithm is used to verify the correctness of the tag T with *any* single key K. If K is from the list, XVer will output 1. If K is uniformly randomly chosen, XVer will output 1 with negligible probability. If an XAC is *strong* and *semi-unique*, we say it is a strengthened XAC. Strongness of XAC means given $(K_i)_{1\le i\le \ell+1, i\ne j}$ and T, a new key $\hat{K}_j$ which is statistically indistinguishable to K_i, can be efficiently sampled. Semi-uniqueness of XAC requires that K can be parsed to (K_a, K_b) and for a fixed T and K_a, there is at most one K_b satisfying $\mathsf{XVer}((K_a, K_b), T) = 1$. The security notion of (strengthened) XAC requires resistance to substitution attacks, i.e., given $T = \mathsf{XAuth}(K_1, \ldots, K_{\ell+1})$ and $(K_i)_{1\le i\le \ell+1, i\ne j}$, the probability that $\mathsf{XVer}(K_j, T') = 1$ is negligible if $T' \ne T$.

Our cryptosystem has message space $\{0,1\}^\ell$, and encryption of an ℓ-bit message $M = m_1\|\cdots\|m_\ell$ for an identity ID is performed bitwise, with one ciphertext element per bit. For each bit m_i, the corresponding ciphertext element C_i is an encryption of m_i under ID, which is generated by the encryption algorithm of the extractable 1SPO-IBE scheme. As shown in [26], a scheme which encrypts long message bit-by-bit is vulnerable to *quoting attacks*. Hence, we use a collision-resistant hash function and a strengthened $\ell+1$-cross-authentication code XAC to bind $C_1, \ldots, C_\ell$ together to resist quoting attacks.

Specifically, let K_a be a public parameter, in our $\mathsf{SIM\text{-}SO\text{-}CCA}$ secure IBE scheme, encryption of an ℓ-bit message $M = m_1\|\cdots\|m_\ell \in \{0,1\}^\ell$ for an identity ID is given by the ciphertext $CT = (C_1, \ldots, C_\ell, T)$, where

$$\begin{cases} (C_i, K_i) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 1) & \text{if } m_i = 1 \\ C_i \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0),\ K_i \leftarrow \mathcal{K} & \text{if } m_i = 0 \end{cases},$$
$$K_b = \mathsf{H}(\mathsf{ID}, C_1, \ldots, C_\ell),\ K_{\ell+1} = (K_a, K_b),\ T = \mathsf{XAuth}(K_1, \ldots, K_{\ell+1}).$$

Here C_i is from the extractable 1SPO-IBE encryption of bit m_i, and K_i is the encapsulated key or randomly chosen key depending on $m_i = 1$ or 0. Finally, XAC tag T glues all the C_is together. Given a ciphertext $CT = (C_1, \ldots, C_\ell, T)$ for identity ID, the decryption algorithm first checks whether $\mathsf{XVer}(K'_{\ell+1}, T) = 1$ or not, where $K'_{\ell+1} = (K_a, \mathsf{H}(\mathsf{ID}, C_1, \ldots, C_\ell))$. If not, it outputs message $\overbrace{0\cdots0}^{\ell}$. Otherwise, it uses $\mathsf{Decrypt}_{ex}$ of the extractable 1SPO-IBE scheme to recover bit m'_i and a session key K'_i from each C_i. If $m'_i = 0$, set $m''_i = 0$, otherwise set $m''_i = \mathsf{XVer}(K'_i, T)$. Finally, it outputs $M'' = m''_1\|\cdots\|m''_\ell$. We assume that the key space $\mathcal{XK}$ of the strengthened XAC and the session key space $\mathcal{K}$ of the extractable 1SPO-IBE are identical (i.e., $\mathcal{K}$=$\mathcal{XK}$), and $\mathcal{K}$ is efficiently samplable and explainable domain.

As for the $\mathsf{SIM\text{-}SO\text{-}CCA}$ security of the IBE scheme, the proving line is to show that encryptions of ℓ ones are "equivocable" ciphertexts, which can be opened to arbitrary messages, and the "equivocable" ciphertexts are computationally indistinguishable from real challenge ciphertexts in an SOA setting, i.e., even if the adversary is given access to a corruption oracle to get the opened messages and randomness, a decryption oracle to decrypt ciphertexts and a key generation oracle to obtain private keys. If so, a PPT SOA-simulator can be constructed to create "equivocable" ciphertexts (i.e., encryptions of ℓ ones) as challenge ciphertexts, then open them accordingly, and $\mathsf{SIM\text{-}SO\text{-}CCA}$ security follows.

To prove a challenge ciphertext $CT = (C_1, \ldots, C_\ell, T)$ under ID, which encrypts $m_1 \| \cdots \| m_\ell$, is indistinguishable from encryption of ℓ ones in the SOA setting, we use hybrid argument. For each $m_i = 0$, we replace (C_i, K_i) (which is used to create CT under ID) with an extractable 1SPO-IBE encryption of 1. If this replacement is distinguishable to an adversary $\mathcal{A}$, then another PPT algorithm $\mathcal{B}$ can simulate SOA-environment for $\mathcal{A}$ by setting (C_i, K_i) to be its own challenge (C^*, K^*) under ID, and use $\mathcal{A}$ to break the $\mathsf{IND\text{-}ID\text{-}CCA}$ security of the extractable 1SPO-IBE. The subtlety lies in how $\mathcal{B}$ deals with $\mathcal{A}$'s decryption query $\widetilde{CT} = (\widetilde{C}_1, \ldots, \widetilde{C}_l, \widetilde{T})$ under ID with $\widetilde{C}_j = C^*$ for some $j \in [\ell]$. Recall that $\mathcal{B}$ is not allowed to issue a private key query $\langle \mathsf{ID} \rangle$ or a decryption query $\langle \mathsf{ID}, C^* \rangle$ to it's own challenger in the extractable 1SPO-IBE security game. In this case, $\mathcal{B}$ will resort to XAC to set $\widetilde{m}''_j = \mathsf{XVer}(K^*, \widetilde{T})$. Observe that, if $(C^*, K^*) = \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 1)$, then $\widetilde{m}''_j = \mathsf{XVer}(K^*, \widetilde{T}) = 1$, which is exactly the same as the output of $\mathsf{Decrypt}$ algorithm. If $C^* = \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0)$ and K^* is random, then $\widetilde{m}''_j = \mathsf{XVer}(K^*, \widetilde{T}) = 0$ except with negligible probability, due to XAC's security against substitution attacks. This is also consistent with the output of the decryption algorithm, except with negligible probability. Hence, with overwhelming probability, $\mathcal{B}$ simulates SOA-environment for $\mathcal{A}$ properly. Note that to apply XAC's security against substitution attacks, we require:

1. $\widetilde{T} \neq T$, which is guaranteed by XAC's *semi-unique* property and *collision resistance* of hash function.
2. K^* should not be revealed to adversary $\mathcal{A}$. Therefore, in the corruption phase, if $\mathcal{B}$ is asked to open (C^*, K^*), it first resamples a $\hat{K}$, which is statistically indistinguishable from K^*. This is guaranteed by the *strongness* of XAC. Then, C will be opened to 0 with algorithm POpen, and $\hat{K}$ (instead of K^*) is opened with a suitable randomness.

Construction of Extractable 1SPO-IBE. In [3], Bellare et al. proposed two one-bit 1SPO-IBEs, one based on the anonymous extension of Lewko-Waters IBE scheme [25] by De Caro, Iovino and Persiano [13] and the other based on the Boyen-Waters anonymous IBE [9]. Both schemes rely on a pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The 1SPO property of the two one-bit IBE schemes is guaranteed by the fact that $\mathbb{G}$ is an *efficiently samplable and explainable domain*, which is characterized by two PPT algorithms Sample and Sample^{-1} for group $\mathbb{G}$. More precisely, Sample chooses an element g from $\mathbb{G}$ uniformly at random, and $\mathsf{Sample}^{-1}(\mathbb{G}, g)$ will output a uniformly distributed R subject to $g = \mathsf{Sample}(\mathbb{G}; R)$. Details of algorithms Sample and Sample^{-1} are given in [3].

A ciphertext of one-bit 1SPO-IBEs in [3] consists of several group elements in $\mathbb{G}$. Those elements are structured if the ciphertext is an encryption of 1 under some identity ID, and this structure can be detected by the private key $\mathsf{SK}_{\mathsf{ID}}$ but not without it[1]. An encryption of 0 is comprised of random elements in $\mathbb{G}$, which are generated by algorithm Sample. Given a ciphertext for ID, the decryption algorithm uses the private key $\mathsf{SK}_{\mathsf{ID}}$ to check whether the ciphertext has a certain structure. If yes, it outputs message 1; otherwise, it outputs message 0. As for the 1SPO property, the algorithm POpen obtains randomness by applying Sample^{-1} to each group element from an encryption of 0.

Unfortunately, the one-bit 1SPO-IBE schemes in [3] are not extractable IBEs. No session keys can be extracted from encryptions of 1, and the schemes are vulnerable to chosen-ciphertext attacks. Therefore, we have to resort to new techniques for extractable 1SPO-IBE.

[1] In fact, in the schemes proposed by Bellare et al. [3], encryptions of a 1 are random group elements in $\mathbb{G}$, and encryptions of a 0 have a certain structure. For ease of description, we exchange them.

We start from anonymous IBE schemes in [13, 9]. Recall that an encryption of a message M for an identity ID in anonymous IBEs [13, 9] takes the form of

$$(\ c_0 = f_0(\mathsf{PK}, s, s_0), \ c_1 = f_1(\mathsf{PK}, \mathsf{ID}, s, s_1), \ c_2 = e(g, g)^{\alpha s} \cdot M \ ), \tag{1}$$

where PK denotes the system's public parameter, α is the master secret key, s, s_0, s_1 are the randomness used in the encryption algorithm, f_0, f_1 are two efficient functions and each of c_0, c_1 denotes one or several elements in $\mathbb{G}$. The private key $\mathsf{SK}_{\mathsf{ID}}$ is structured such that pairings with group elements of (c_1, c_2) result in $e(g, g)^{\alpha s}$, hence the message M can be recovered from c_2.

The idea of constructing extractable 1SPO-IBE is summerized as follows. Firstly, we generate ciphertexts of the form

$$(\ c_0' = f_0'(\mathsf{PK}, s, s_0), \ c_1' = f_1'(\mathsf{PK}, \mathsf{ID}, \mathsf{ID}', s, s_1) \ ), \tag{2}$$

where $\mathsf{ID}' = \mathsf{H}(\mathsf{ID}, c_0')$ and H is a collision-resistant hash function. The ciphertext is similar to Eq.(1), except that it is a 2-hierarchical extension with respect to $(\mathsf{ID}, \mathsf{ID}')$. The structure of (c_0', c_1') is characterized by the shared randomness s and this structure can be publicly verified. The master secret key is now (α, β). Correspondingly the private key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{SK}_{\mathsf{ID},1}, \mathsf{SK}_{\mathsf{ID},2})$, and $\mathsf{SK}_{\mathsf{ID},i} (i = 1, 2)$ are generated by the master secret key α and β respectively, in a similar way as that in the anonymous IBEs [13, 9]. Consequently, $\mathsf{SK}_{\mathsf{ID},1}$ and $\mathsf{SK}_{\mathsf{ID},2}$ help generate $e(g, g)^{\alpha s}$ and $e(g, g)^{\beta s}$ from (c_0', c_1').

Next, we use $e(g, g)^{\alpha s}$ to blind (c_0', c_1') and obtain

$$(\ c_0'' = f_1''(\mathsf{PK}, s, s_0), \ c_1'' = f_1''(\mathsf{PK}, \mathsf{ID}, \mathsf{ID}', s, s_1, e(g, g)^{\alpha s}) \ ), \tag{3}$$

which satisfies the following properties:

1. Without the private key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{SK}_{\mathsf{ID},1}, \ \mathsf{SK}_{\mathsf{ID},2})$ for ID, the relationship between c_0'' and c_1'' (that they share the same s) is hidden from any PPT adversary.
2. With $\mathsf{SK}_{\mathsf{ID},1}$ and $\mathsf{SK}_{\mathsf{ID},2}$, it is still possible to generate $e(g, g)^{\alpha s}$ and $e(g, g)^{\beta s}$ from the blinded ciphertext (c_0'', c_1'').
3. Given the blinded factor $e(g, g)^{\alpha s}$, (c_0'', c_1'') can be efficiently changed back to (c_0', c_1').

Finally, we obtain the extractable 1SPO-IBE with the following features:

$$\mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, b) = \begin{cases} ((c_0'', c_1''), K) = \big((f_1''(\mathsf{PK}, s, s_0), f_1''(\mathsf{PK}, \mathsf{ID}, \mathsf{ID}', s, s_1, e(g, g)^{\alpha s})), e(g, g)^{\beta s}\big) & b = 1 \\ (c_0'', c_1'') \leftarrow \mathsf{Sample}(\mathbb{G}) & b = 0 \end{cases}.$$

– Given a ciphertext $C = (c_0'', c_1'')$ for ID, the decryption algorithm first uses $\mathsf{SK}_{\mathsf{ID},1}$ to compute a blinding factor from (c_0'', c_1''). Then, it uses the blinding factor to retrieve (c_0', c_1') from (c_0'', c_1''). Next, it checks whether (c_0', c_1') have a specific structure. If yes, it outputs message 1 and computes the encapsulated session key from (c_0'', c_1'') using $\mathsf{SK}_{\mathsf{ID},2}$; otherwise, it outputs message 0 and a uniformly random session key.
– Algorithm POpen for 1SPO can be implemented with Sample^{-1}.

We emphasize that the 2-hierarchical IBE structure (when encrypting 1) helps to answer decryption queries in the $\mathsf{IND\text{-}ID\text{-}CCA}$ security proof of the above extractable 1SPO-IBE. In the private key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{SK}_{\mathsf{ID},1}, \mathsf{SK}_{\mathsf{ID},2})$, $\mathsf{SK}_{\mathsf{ID},2}$ is used to generate the encapsulated key $e(g, g)^{\beta s}$ when encrypting 1, and $\mathsf{SK}_{\mathsf{ID},1}$ is used to generate a blind factor $e(g,g)^{\alpha s}$, which helps to convert the publicly verifiable structure of (c'_0, c'_1) to a privately verifiable structure, resulting in $\mathsf{IND\text{-}ID\text{-}CCA}$ secure extractable 1SPO-IBE.

Related Work. Non-committing encryption (NCE) [12] was introduced by Canetti, Feige, Goldreich and Naor [12] to achieve adaptively secure multi-party computation. In NCE schemes there is a simulator, which can generate non-committing ciphertexts, and later open them to any desired message. In [11], Canetti, Dwork, Naor and Ostrovsky extended the notion of NCE to a new primitive which they called deniable encryption. In deniable encryption schemes, a sender may open a ciphertext to an arbitrary message by providing coins produced by a faking algorithm. A weak form of deniable encryption is that encryptions of 1 can be opened as encryptions of 0 even if not vice versa, and 1SPO-IBE is an IBE analogue of this notion. We refer the reader to [3] for more discussions on NCE and deniable encryption.

Organization. The rest of the paper is organized as follows. Some preliminaries are given in Section 2. We introduce the notion and security model of extractable 1SPO-IBE in Section 3. The notion of strengthened XAC and its efficient construction are given in Section 4. We propose a paradigm of building $\mathsf{SIM\text{-}SO\text{-}CCA}$ secure IBE from $\mathsf{IND\text{-}ID\text{-}CCA}$ secure extractable 1SPO-IBE, collision-resistant hash function and strengthened XAC in Section 5. We present two $\mathsf{IND\text{-}ID\text{-}CCA}$ secure extractable 1SPO-IBE schemes in Section 6. The notion of composite order bilinear groups and complexity assumptions we use are given in Appendix A. In Appendix B and C, we give the formal notion of IBE and the simulation-based definition of IBE secure against a selective opening chosen-ciphertext adversary respectively.

2 Preliminaries

If S is a set, then $s_1, \ldots, s_t \leftarrow S$ denotes the operation of picking elements $s_1, \ldots, s_t$ uniformly at random from S. If $n \in \mathbb{N}$ then $[n]$ denotes the set $\{1, \ldots, n\}$. For $i \in \{0,1\}^*$, $|i|$ denotes the bit-length of i. If $x_1, x_2, \ldots$ are strings, then $x_1 \| x_2 \| \cdots$ denotes their concatenation. For a probabilistic algorithm A, we denote $y \leftarrow A(x;R)$ the process of running A on input x and with randomness R, and assigning y the result. Let $\mathcal{R}_A$ denote the randomness space of A, and we write $y \leftarrow A(x)$ for $y \leftarrow A(x;R)$ with R chosen from $\mathcal{R}_A$ uniformly at random. A function $f(\kappa)$ is *negligible*, if for every $c > 0$ there exists a κ_c such that $f(\kappa) < 1/\kappa^c$ for all $\kappa > \kappa_c$.

2.1 Key Derivation Functions

A family of *key derivation functions* [14] $\mathcal{KDF} = \{\mathsf{KDF}_i : \mathcal{X}_i \rightarrow \mathcal{K}_i\}$, indexed by $i \in \{0,1\}^*$, is *secure* if, for all PPT algorithms $\mathcal{A}$ and for sufficiently large i, the distinguishing advantage $Adv^{\mathcal{A}}_{\mathcal{KDF}}(i)$ is negligible (in $|i|$), where

$$\begin{aligned} Adv^{\mathcal{A}}_{\mathcal{KDF}}(i) = |\Pr[\mathcal{A}(\mathsf{KDF}_i, \mathsf{KDF}_i(x)) = 1 \,|\, \mathsf{KDF}_i \leftarrow \mathcal{KDF}, x \leftarrow \mathcal{X}_i] - \\ \Pr[\mathcal{A}(\mathsf{KDF}_i, K) = 1 \,|\, \mathsf{KDF}_i \leftarrow \mathcal{KDF}, K \leftarrow \mathcal{K}_i]| \,. \end{aligned}$$

The above definition is for presentation simplicity. In general, the index i should be generated by a PPT sampler algorithm on the security parameter κ. For notational convenience, we ignore the index i of a key derivation function.

2.2 Efficiently samplable and explainable domain

A domain $\mathcal{D}$ is *efficiently samplable and explainable* [15] iff there exist two PPT algorithms:

- $\mathsf{Sample}(\mathcal{D}; R)$: On input random coins $R \leftarrow \mathcal{R}_{\mathsf{Sample}}$ and a domain $\mathcal{D}$, it outputs an element uniformly distributed over $\mathcal{D}$.
- $\mathsf{Sample}^{-1}(\mathcal{D}, x)$: On input $\mathcal{D}$ and *any* $x \in \mathcal{D}$, this algorithm outputs R that is uniformly distributed over the set $\{R \in \mathcal{R}_{\mathsf{Sample}} | \mathsf{Sample}(\mathcal{D}; R) = x\}$.

3 Extractable IBE with One-Sided Public Openability (Extractable 1SPO-IBE)

Formally, an extractable identity-based encryption (extractable IBE) scheme consists of the following four algorithms:

$\mathsf{Setup}_{ex}(1^\kappa)$ takes as input a security parameter κ. It generates a public parameter PK and a master secret key MSK. The public parameter PK defines an identity space $\mathcal{ID}$, a ciphertext space $\mathcal{C}$ and a session key space $\mathcal{K}$.

$\mathsf{KeyGen}_{ex}(\mathsf{PK}, \mathsf{MSK}, \mathsf{ID})$ takes as input the public parameter PK, the master secret key MSK and an identity $\mathsf{ID} \in \mathcal{ID}$. It produces a private key $\mathsf{SK}_{\mathsf{ID}}$ for the identity ID.

$\mathsf{Encrypt}_{ex}(\mathsf{PK}, \mathsf{ID}, m)$ takes as input the public parameter PK, an identity $\mathsf{ID} \in \mathcal{ID}$ and a message $m \in \{0, 1\}$. It outputs a ciphertext C if $m = 0$, and outputs a ciphertext and a session key (C, K) if $m = 1$. Here $K \in \mathcal{K}$.

$\mathsf{Decrypt}_{ex}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, C)$ takes as input the public parameter PK, a private key $\mathsf{SK}_{\mathsf{ID}}$ and a ciphertext $C \in \mathcal{C}$. It outputs a message $m' \in \{0, 1\}$ and a session key $K' \in \mathcal{K}$.

Correctness. An extractable IBE scheme has completeness error ϵ, if for all κ, $\mathsf{ID} \in \mathcal{ID}$, $m \in \{0, 1\}$, $(\mathsf{PK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}_{ex}(1^\kappa)$, $C/(C, K) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}, \mathsf{ID}, m)$, $\mathsf{SK}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}_{ex}(\mathsf{PK}, \mathsf{MSK}, \mathsf{ID})$ and $(m', K') \leftarrow \mathsf{Decrypt}_{ex}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, C)$:

- The probability that $m' = m$ is at least $1 - \epsilon$, where the probability is taken over the coins used in encryption.
- If $m = 1$ then $m' = m$ and $K' = K$. If $m' = 0$, K' is uniformly distributed in $\mathcal{K}$.

Security. The $\mathsf{IND\text{-}ID\text{-}CCA}$ security of extractable IBE is twisted from $\mathsf{IND\text{-}ID\text{-}CCA}$ security of one-bit IBE and $\mathsf{IND\text{-}ID\text{-}CCA}$ security of identity-based key encapsulation mechanism (IB-KEM). The security notion is defined using the following game between a PPT adversary $\mathcal{A}$ and a challenger.

Setup The challenger runs $\mathsf{Setup}_{ex}(1^\kappa)$ to obtain a public parameter PK and a master secret key MSK. It gives the public parameter PK to the adversary.

Query phase 1 The adversary $\mathcal{A}$ adaptively issues the following queries:

- Key generation query $\langle \mathsf{ID} \rangle$: the challenger runs KeyGen_{ex} on ID to generate the corresponding private key $\mathsf{SK}_{\mathsf{ID}}$, which is returned to $\mathcal{A}$.
- Decryption query $\langle \mathsf{ID}, C \rangle$: the challenger runs KeyGen_{ex} on ID to get the private key, then use the key to decrypt C with $\mathsf{Decrypt}_{ex}$ algorithm. The result is sent back to $\mathcal{A}$.

Challenge The adversary $\mathcal{A}$ submits a challenge identity ID^*. The only restriction is that, $\mathcal{A}$ did not issue a private key query for ID^* in Query phase 1. The challenger first selects a random bit $\delta \in \{0, 1\}$. If $\delta = 1$, the challenger computes $(C^*, K^*) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}, \mathsf{ID}^*, 1)$. Otherwise (i.e., $\delta = 0$), the challenger computes $C^* \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}, \mathsf{ID}^*, 0)$ and chooses $K^* \leftarrow \mathcal{K}$. Then, the challenge ciphertext and session key (C^*, K^*) are sent to the adversary by the challenger.

Query phase 2 This is identical to Query phase 1, except that the adversary does not request a private key for ID^* or the decryption of $\langle \mathsf{ID}^*, C^* \rangle$.

Guess The adversary $\mathcal{A}$ outputs its guess $\delta' \in \{0, 1\}$ for δ and wins the game if $\delta = \delta'$.

The advantage of the adversary in this game is defined as $\mathsf{Adv}^{\mathsf{cca}}_{\mathsf{ex\text{-}IBE},\mathcal{A}}(\kappa) = |\Pr[\delta' = 1|\delta = 1] - \Pr[\delta' = 1|\delta = 0]|$, where the probability is taken over the random bits used by the challenger and the adversary.

Definition 1 *An extractable IBE scheme is IND-ID-CCA secure, if the advantage in the above security game is negligible for all PPT adversaries.*

We say that an extractable IBE scheme is $\mathsf{IND\text{-}sID\text{-}CCA}$ secure if we add an **Init** stage before setup in the above security game where the adversary commits to the challenge identity ID^*.

Definition 2 (Extractable 1SPO-IBE) *An extractable IBE scheme is* One-Sided Publicly Openable *if it is associated with a PPT public algorithm POpen such that for all PK generated by $(\mathsf{PK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}_{ex}(1^\kappa)$, for all $\mathsf{ID} \in \mathcal{ID}$ and any $C \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}, \mathsf{ID}, 0)$, it holds that: the output of $\mathsf{POpen}(\mathsf{PK}, \mathsf{ID}, C)$ distributes uniformly at random over $\mathsf{Coins}(\mathsf{PK}, \mathsf{ID}, C, 0)$, where $\mathsf{Coins}(\mathsf{PK}, \mathsf{ID}, C, 0)$ denotes the set of random coins $\{\tilde{R} \mid C = \mathsf{Encrypt}_{ex}(\mathsf{PK}, \mathsf{ID}, 0; \tilde{R})\}$.*

4 Strengthened Cross-authentication Codes

In this section, we first review the notion and security requirements of cross-authentication codes introduced in [15]. Then we define a new property of cross-authentication codes: *semi-unique*. If a cross-authentication code is *strong* and *semi-unique*, we say it is a *strengthened* cross-authentication code, which will play an important role in our construction of $\mathsf{SIM\text{-}SO\text{-}CCA}$ secure IBE. Finally, we will show that the efficient construction of cross-authentication code proposed by Fehr et al. [15] is actually a strengthened cross-authentication code.

Definition 3 (L-Cross-authentication code.) *For $L \in \mathbb{N}$, an L-cross-authentication code XAC is associated with a key space $\mathcal{XK}$ and a tag space $\mathcal{XT}$, and consists of three PPT algorithms XGen, XAuth and XVer. $\mathsf{XGen}(1^\kappa)$ produces a uniformly random key $K \in \mathcal{XK}$, deterministic algorithm $\mathsf{XAuth}(K_1, \ldots, K_L)$ outputs a tag $T \in \mathcal{XT}$, and deterministic algorithm $\mathsf{XVer}(K, T)$ outputs a decision bit*[2]*. The following is required:*

Correctness. *For all $i \in [L]$, the probability $\mathsf{fail}_{\mathsf{XAC}}(\kappa) := \Pr[\mathsf{XVer}(K_i, \mathsf{XAuth}(K_1, \ldots, K_L)) \neq 1]$, is negligible, where $K_1, \ldots, K_L \leftarrow \mathsf{XGen}(1^\kappa)$ in the probability.*

Security against impersonation and substitution attacks. *$\mathsf{Adv}^{imp}_{\mathsf{XAC}}(\kappa)$ and $\mathsf{Adv}^{sub}_{\mathsf{XAC}}(\kappa)$ as defined below are both negligible: $\mathsf{Adv}^{imp}_{\mathsf{XAC}}(\kappa) := \max_{T'} \Pr[\mathsf{XVer}(K, T') = 1 | K \leftarrow \mathsf{XGen}(1^\kappa)]$, where the*

[2] In Fehr et al.'s original definition [15], algorithm XVer includes an additional input parameter: index i. Let $K_1, \ldots, K_L \leftarrow \mathsf{XGen}(1^\kappa)$ and $T \leftarrow \mathsf{XAuth}(K_1, \ldots, K_L)$. Since $\mathsf{XVer}(K_i, i, T) = \mathsf{XVer}(K_i, j, T)$ in their efficient construction, we only take a key and a tag as input of algorithm XVer for notational convenience.

max *is over all* $T' \in \mathcal{XT}$, *and*

$$\mathsf{Adv}^{\mathsf{sub}}_{\mathsf{XAC}}(\kappa) := \max_{i,K_{\neq i},F} \Pr\left[\begin{array}{c} T' \neq T \wedge \\ \mathsf{XVer}(K_i, T') = 1 \end{array} \middle| \begin{array}{c} K_i \leftarrow \mathsf{XGen}(1^\kappa), \\ T = \mathsf{XAuth}(K_1, \ldots, K_L), \\ T' \leftarrow F(T) \end{array}\right]$$

where the max *is over all* $i \in [L]$, *all* $K_{\neq i} = (K_j)_{j \neq i} \in \mathcal{XK}^{L-1}$ *and all (possibly randomized) functions* $F : \mathcal{XT} \rightarrow \mathcal{XT}$.

Definition 4 (Strengthened XAC.) *An L-cross-authentication code XAC is a* strengthened *XAC, if it enjoys the following additional properties.*

Strongness [22]: *There exists another PPT public algorithm* ReSamp, *which takes as input* i, $(K_j)_{j \neq i}$ *and* T, *with* $K_1, \ldots, K_L \leftarrow \mathsf{XGen}(1^\kappa)$ *and* $T \leftarrow \mathsf{XAuth}(K_1, \ldots, K_L)$, *outputs* $\hat{K}_i$ *(i.e.,* $\hat{K}_i \leftarrow \mathsf{ReSamp}(K_{\neq i}, T)$*), such that* $\hat{K}_i$ *is statistically indistinguishable with* K_i, *i.e., the statistical distance* $Dist(\kappa) := \frac{1}{2} \cdot \sum_{K \in \mathcal{XK}} \left| \Pr[\hat{K}_i = K \,|(K_{\neq i}, T)] - \Pr[K_i = K \,|(K_{\neq i}, T)] \right|$ *is* negligible.

Semi-Uniqueness: *The key space* $\mathcal{XK} = \mathcal{K}_a \times \mathcal{K}_b$. *Given an authentication tag* T *and* $K_a \in \mathcal{K}_a$, *there exists at most* one $K_b \in \mathcal{K}_b$ *such that* $\mathsf{XVer}((K_a, K_b), T) = 1$.

Next, we review the efficient construction of L-cross-authentication code secure against impersonation and substitution attacks proposed by Fehr et al. [15], and show that it is *strong* and *semi-unique* as well, i.e. it is a *strengthened* XAC.

- $\mathcal{XK} = \mathcal{K}_a \times \mathcal{K}_b = \mathbb{F}_q^2$ and $\mathcal{XT} = \mathbb{F}_q^L \cup \{\perp\}$.
- XGen outputs (a, b), which is chosen from $\mathbb{F}_q^2$ uniformly at random.
- $T \leftarrow \mathsf{XAuth}((a_1, b_1), \ldots, (a_L, b_L))$. Let $\mathbf{A} \in \mathbb{F}_q^{L \times L}$ be a matrix with its i-th row $(1, a_i, a_i^2, \ldots, a_i^{L-1})$ for $i \in [L]$. Let $b_1, \ldots, b_L \in \mathbb{F}_q^L$ constitute the column vector $\mathbf{B}$. If $\mathbf{A}T = \mathbf{B}$ has no solution or more than one solution, set $T = \perp$. Otherwise $\mathbf{A}$ is a Vandermonde matrix, and the tag $T = (T_0, \ldots, T_{L-1})$ can be computed efficiently by solving the linear equation system $\mathbf{A}T = \mathbf{B}$.
- Define $poly_T(x) = T_0 + T_1 x + \cdots + T_{L-1} x^{L-1} \in \mathbb{F}_q[x]$ with $T = (T_0, \ldots, T_{L-1})$. $\mathsf{XVer}((a, b), T)$ outputs 1 if and only if $T \neq \perp$ and $poly_T(a) = b$.
- $(a, b) \leftarrow \mathsf{ReSamp}((a_j, b_j)_{j \neq i}, T)$. Choose $a \leftarrow \mathbb{F}_q$ such that $a \neq a_j$ $(1 \leq j \leq \ell, j \neq i)$ and compute $b = poly_T(a)$. Conditioned on $T = \mathsf{XAuth}((a_1, b_1), \ldots, (a_L, b_L))$ $(T \neq \perp)$ and $(a_j, b_j)_{j \neq i}$, both of (a, b) and (a_i, b_i) are uniformly distributed over the same support.
- Fixing $a \in \mathbb{F}_q$ results in a unique $b = poly_T(a)$ such that $\mathsf{XVer}((a, b), T) = 1$, if $T \neq \perp$.

5 Proposed SIM-SO-CCA Secure IBE Scheme

Let $(\mathsf{Setup}_{ex}, \mathsf{KeyGen}_{ex}, \mathsf{Encrypt}_{ex}, \mathsf{Decrypt}_{ex})$ be an extractable 1SPO-IBE scheme with identity space $\mathcal{ID}$, ciphertext space $\mathcal{C}$ and session key space $\mathcal{K} = \mathcal{K}_a \times \mathcal{K}_b$, and (XGen, XAuth, XVer) be a strengthened $\ell + 1$-cross-authentication code XAC with key space $\mathcal{XK} = \mathcal{K} = \mathcal{K}_a \times \mathcal{K}_b$ and tag space $\mathcal{XT}$. We require that key space $\mathcal{K}$ is also an *efficiently samplable and explainable domain*[3] associated with algorithms Sample' and Sample'^{-1}. Our cryptosystem has message space $\{0,1\}^\ell$.

Our scheme consists of the following algorithms:

[3] As mentioned in [15], the *efficiently samplable and explainable* key space $\mathcal{K}$ can be assumed without loss of generality, because $\mathcal{K}$ can always be efficiently mapped into $\mathcal{K}' = \{0,1\}^l$ by means of a suitable (almost) balanced function, such that uniform distribution in $\mathcal{K}$ induces (almost) uniform distribution in $\mathcal{K}'$, and where l is linear in $\log(|\mathcal{K}|)$.

$\mathsf{Setup}(1^\kappa)$: The setup algorithm first chooses $K_a \leftarrow \mathcal{K}_a$ and a collision-resistant hash function $\mathsf{H} : \mathcal{ID} \times \overbrace{\mathcal{C} \times \cdots \times \mathcal{C}}^{\ell} \to \mathcal{K}_b$, and calls Setup_{ex} to obtain $(\mathsf{PK}_{ex}, \mathsf{MSK}_{ex}) \leftarrow \mathsf{Setup}_{ex}(1^\kappa)$. It sets the public parameter $\mathsf{PK} = (\mathsf{PK}_{ex}, \mathsf{H}, K_a)$ and the master secret key $\mathsf{MSK} = \mathsf{MSK}_{ex}$.

$\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathsf{ID} \in \mathcal{ID})$: The key generation algorithm takes as input the public parameter $\mathsf{PK} = (\mathsf{PK}_{ex}, \mathsf{H}, K_a)$, the master secret key $\mathsf{MSK} = \mathsf{MSK}_{ex}$ and an identity ID. It calls KeyGen_{ex} to get

$$\mathsf{SK}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}_{ex}(\mathsf{PK}_{ex}, \mathsf{MSK}_{ex}, \mathsf{ID}),$$

and outputs the private key $\mathsf{SK}_{\mathsf{ID}}$.

$\mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID} \in \mathcal{ID}, M)$: The encryption algorithm takes as input the public parameter $\mathsf{PK} = (\mathsf{PK}_{ex}, \mathsf{H}, K_a)$, an identity ID and a message $M = m_1 \| \cdots \| m_\ell \in \{0,1\}^\ell$. For $i \in [\ell]$, it computes

$$\begin{cases} (C_i, K_i) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 1) & \text{if } \ m_i = 1 \\ C_i \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}, 0), \ K_i \leftarrow \mathsf{Sample}'(\mathcal{K}; R_i^K) & \text{if } \ m_i = 0 \end{cases},$$

where $R_i^K \leftarrow \mathcal{R}_{\mathsf{Sample}'}$. Then, it sets $K_{\ell+1} = (K_a, K_b)$ where $K_b = \mathsf{H}(\mathsf{ID}, C_1, \ldots, C_\ell)$, and computes the tag $T = \mathsf{XAuth}(K_1, \ldots, K_{\ell+1})$. Finally, it outputs the ciphertext $CT = (C_1, \ldots, C_\ell, T)$.

$\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, CT)$: The decryption algorithm takes as input the public parameter $\mathsf{PK} = (\mathsf{PK}_{ex}, \mathsf{H}, K_a)$, a private key $\mathsf{SK}_{\mathsf{ID}}$ for identity ID and a ciphertext $CT = (C_1, \ldots, C_\ell, T)$. This algorithm first computes $K_b' = \mathsf{H}(\mathsf{ID}, C_1, \ldots, C_\ell)$ and checks whether $\mathsf{XVer}(K_{\ell+1}', T) = 1$ with $K_{\ell+1}' = (K_a, K_b')$. If not, it outputs $M'' = \overbrace{0 \cdots 0}^{\ell}$. Otherwise, for $i \in [\ell]$, it computes $(m_i', K_i') \leftarrow \mathsf{Decrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{SK}_{\mathsf{ID}}, C_i)$ and sets

$$m_i'' = \begin{cases} \mathsf{XVer}(K_i', T) & \text{if } \ m_i' = 1 \\ 0 & \text{if } \ m_i' = 0 \end{cases}.$$

Then, it outputs the message $M'' = m_1'' \| \cdots \| m_\ell''$.

Correctness. If $m_i = 1$, then $(m_i', K_i') = (m_i, K_i)$ by correctness of extractable 1SPO-IBE scheme, so $\mathsf{XVer}(K_i', T) = 1$ (hence $m_i'' = 1$) except with probability $\mathsf{fail}_{\mathsf{XAC}}$ by correctness of XAC. On the other hand, if $m_i = 0$, the ϵ-completeness of the extractable 1SPO-IBE guarantees $m_i' = 0$ (hence $m_i'' = 0$) with probability at least $1 - \epsilon$. Consequently, for any $CT \leftarrow \mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID}, M)$, we have $\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, CT) = M$ except with probability at most $\ell \cdot \max\{\mathsf{fail}_{\mathsf{XAC}}, \epsilon\}$.

Theorem 1 *If the extractable 1SPO-IBE scheme is IND-ID-CCA secure, the hash function H is collision-resistant and the strengthened $\ell+1$-cross-authentication code XAC is secure against substitution attacks, then our proposed IBE scheme is SIM-SO-CCA secure.*

Proof. See Appendix D.

6 Proposed IND-ID-CCA Secure Extractable 1SPO-IBE Scheme

In this section, we propose a concrete construction of extractable 1SPO-IBE from the anonymous IBE [13] in a composite order bilinear group. (In Appendix F, we show how to construct an extractable 1SPO-IBE from Boyen-Waters anonymous HIBE [9], which is based on a prime order bilinear group.) The design principle has already been described in the Introduction.

The proposed scheme consists of the following algorithms:

$\mathsf{Setup}_{ex}(1^\kappa)$: Run an N-order group generator $\mathcal{G}(\kappa)$ to obtain a group description $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear map, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1p_2p_3p_4$. Next choose $g, u, v, h \leftarrow \mathbb{G}_{p_1}$, $g_3 \leftarrow \mathbb{G}_{p_3}$, $g_4, W_4 \leftarrow \mathbb{G}_{p_4}$ and $\alpha, \beta \leftarrow \mathbb{Z}_N$. Then choose a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$, and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. The public parameter is

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N),\ u, v, h, W_{14} = gW_4,\ g_4,\ e(g,g)^\alpha,\ e(g,g)^\beta,\ \mathsf{H}, \mathsf{KDF}).$$

The master secret key is $\mathsf{MSK} = (g, g_3, \alpha, \beta)$. We require the group $\mathbb{G}$ be an *efficiently samplable and explainable domain* associated with algorithms Sample and Sample^{-1}. Details on how to instantiate such groups are given in [3].

$\mathsf{KeyGen}_{ex}(\mathsf{PK}, \mathsf{MSK}, \mathsf{ID} \in \mathbb{Z}_N)$: Choose $r, \bar{r} \leftarrow \mathbb{Z}_N$ and $R_3, R'_3, R''_3, \bar{R}_3, \bar{R}'_3, \bar{R}''_3 \leftarrow \mathbb{G}_{p_3}$ (this is done by raising g_3 to a random power). Output the private key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$, where

$$D_0 = g^\alpha (u^{\mathsf{ID}} h)^r R_3,\ D_1 = v^r R'_3,\ D_2 = g^r R''_3,\ \bar{D}_0 = g^\beta (u^{\mathsf{ID}} h)^{\bar{r}} \bar{R}_3,\ \bar{D}_1 = v^{\bar{r}} \bar{R}'_3,\ \bar{D}_2 = g^{\bar{r}} \bar{R}''_3.$$

$\mathsf{Encrypt}_{ex}(\mathsf{PK}, \mathsf{ID} \in \mathbb{Z}_N, m \in \{0,1\})$: If $m = 1$, choose $s, t_4 \leftarrow \mathbb{Z}_N$ and compute

$$c_0 = W_{14}^s g_4^{t_4},\ c_1 = (u^{\mathsf{ID}} v^{\mathsf{ID}'} h)^s g_4^{\mathsf{KDF}(e(g,g)^{\alpha s})},\ K = e(g,g)^{\beta s},$$

where $\mathsf{ID}' = \mathsf{H}(\mathsf{ID}, c_0)$, then output the ciphertext and the session key $(C, K) = ((c_0, c_1), K)$; otherwise (i.e., $m = 0$), choose $c_0, c_1 \leftarrow \mathsf{Sample}(\mathbb{G})$, and output the ciphertext $C = (c_0, c_1)$.

$\mathsf{Decrypt}_{ex}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2), C = (c_0, c_1))$: Compute $\mathsf{ID}' = \mathsf{H}(\mathsf{ID}, c_0)$ and

$$X = e(D_0 D_1^{\mathsf{ID}'}, c_0)/e(D_2, c_1). \tag{4}$$

(One can view $(D_0 D_1^{\mathsf{ID}'}, D_2)$ as a private key associated to the 2-level identity $\widetilde{\mathsf{ID}} = (\mathsf{ID}, \mathsf{ID}')$.) Then, check whether

$$e(c_1/g_4^{\mathsf{KDF}(X)}, W_{14}) = e(c_0,\ u^{\mathsf{ID}} v^{\mathsf{ID}'} h). \tag{5}$$

If not, set $m = 0$ and choose a session key $K \leftarrow \mathbb{G}_T$. Otherwise, set $m = 1$ and compute

$$K = e(\bar{D}_0 \bar{D}_1^{\mathsf{ID}'}, c_0)/e(\bar{D}_2, c_1). \tag{6}$$

Output (m, K).

Correctness. Note that, if $C = (c_0, c_1)$ is an encryption of 1 under identity ID, then

$$X = e(D_0 D_1^{\mathsf{ID}'}, c_0)/e(D_2, c_1) = e(g^\alpha (u^{\mathsf{ID}} v^{\mathsf{ID}'} h)^r,\ g^s)/e(g^r,\ (u^{\mathsf{ID}} v^{\mathsf{ID}'} h)^s) = e(g,g)^{\alpha s},$$
$$e(c_1/g_4^{\mathsf{KDF}(X)}, W_{14}) = e((u^{\mathsf{ID}} v^{\mathsf{ID}'} h)^s, W_{14}) = e(u^{\mathsf{ID}} v^{\mathsf{ID}'} h, W_{14}^s) = e(c_0,\ u^{\mathsf{ID}} v^{\mathsf{ID}'} h),$$
$$K = e(\bar{D}_0 \bar{D}_1^{\mathsf{ID}'}, c_0)/e(\bar{D}_2, c_1) = e(g^\beta (u^{\mathsf{ID}} v^{\mathsf{ID}'} h)^{\bar{r}},\ g^s)/e(g^{\bar{r}},\ (u^{\mathsf{ID}} v^{\mathsf{ID}'} h)^s) = e(g,g)^{\beta s},$$

so decryption always succeeds. On the other hand, if $C = (c_0, c_1)$ is an encryption of 0 under identity ID, then $c_0, c_1 \in \mathbb{G}$ are chosen uniformly at random, thus $\Pr[e(c_1/g_4^{\mathsf{KDF}(X)}, W_{14}) = e(c_0, u^{\mathsf{ID}} v^{\mathsf{ID}'} h)] \leq \frac{1}{2^{2\kappa}}$ where κ is the security parameter. So the completeness error is $\frac{1}{2^{2\kappa}}$.

One-Sided Public Openability (1SPO). If $C = (c_0, c_1)$ is an encryption of 0 under identity ID, then c_0 and c_1 are both randomly distributed in $\mathbb{G}$. Since the group $\mathbb{G}$ is an efficiently samplable and explainable domain associated with Sample and Sample^{-1}, $\mathsf{POpen}(\mathsf{PK}, \mathsf{ID}, C = (c_0, c_1))$ can employ Sample^{-1} to open (c_0, c_1). More precisely, $(R_0, R_1) \leftarrow \mathsf{POpen}(\mathsf{PK}, \mathsf{ID}, (c_0, c_1))$, where $R_0 \leftarrow \mathsf{Sample}^{-1}(\mathbb{G}, c_0)$ and $R_1 \leftarrow \mathsf{Sample}^{-1}(\mathbb{G}, c_1)$.

Security. We now state the security theorem of our proposed extractable IBE scheme.

Theorem 2 *If Assumptions 1, 2, 3, 4, 5 and 6 hold,* H *is a collision-resistant hash function and* KDF *is a secure key derivation function, then the above extractable 1SPO-IBE scheme is* IND-ID-CCA *secure.*

Proof. See Appendix E.

References

1. M. Bellare, R. Dowsley, B. Waters, and S. Yilek. Standard security does not imply security against selective-opening. In *EUROCRYPT*, pages 645–662, 2012.
2. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, pages 1–35, 2009.
3. M. Bellare, B. Waters, and S. Yilek. Identity-based encryption secure against selective opening attack. In *TCC*, pages 235–252, 2011.
4. M. Bellare and S. Yilek. Encryption schemes secure under selective opening attack. *IACR Cryptology ePrint Archive*, 2009:101, 2009.
5. F. Böhl, D. Hofheinz, and D. Kraschewski. On definitions of selective opening security. In *Public Key Cryptography*, pages 522–539, 2012.
6. A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In *CRYPTO*, pages 335–359, 2008.
7. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.
8. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC*, pages 325–341, 2005.
9. X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO*, pages 290–307, 2006.
10. X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). *IACR Cryptology ePrint Archive*, 2006:085, 2006.
11. R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. Deniable encryption. In *CRYPTO*, pages 90–104, 1997.
12. R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *STOC*, pages 639–648, 1996.
13. A. D. Caro, V. Iovino, and G. Persiano. Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts. In *Pairing*, pages 347–366, 2010.
14. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *IACR Cryptology ePrint Archive*, 2001:108, 2001.
15. S. Fehr, D. Hofheinz, E. Kiltz, and H. Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. In *EUROCRYPT*, pages 381–402, 2010.
16. D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. In *Public Key Cryptography*, pages 279–295, 2010.
17. B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *ASIACRYPT*, pages 70–88, 2011.
18. B. Hemenway and R. Ostrovsky. Lossy trapdoor functions from smooth homomorphic hash proof systems. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:127, 2009.
19. B. Hemenway and R. Ostrovsky. Homomorphic encryption over cyclic groups implies chosen-ciphertext security. *IACR Cryptology ePrint Archive*, 2010:99, 2010.
20. D. Hofheinz. Possibility and impossibility results for selective decommitments. *IACR Cryptology ePrint Archive*, 2008:168, 2008.

21. D. Hofheinz. All-but-many lossy trapdoor functions. In *EUROCRYPT*, pages 209–227, 2012.
22. Z. Huang, S. Liu, and B. Qin. Sender equivocable encryption schemes secure against chosen-ciphertext attacks revisited. *IACR Cryptology ePrint Archive*, 2012:473, 2012.
23. Z. Huang, S. Liu, and B. Qin. Sender-equivocable encryption schemes secure against chosen-ciphertext attacks revisited. In *Public Key Cryptography*, pages 369–385, 2013.
24. E. Kiltz, P. Mohassel, and A. O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *EUROCRYPT*, pages 673–692, 2010.
25. A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, pages 455–479, 2010.
26. S. Myers and A. Shelat. Bit encryption is complete. In *FOCS*, pages 607–616, 2009.
27. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 2008.
28. A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.
29. B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.
30. B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

A Composite Order Bilinear Groups

We review the notion of composite order bilinear groups, introduced in [8] firstly.

Let $\mathcal{G}$ be an N-order group generator algorithm that takes as input a security parameter κ and outputs a tuple $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$, where $p_1, p_2, p_3, p_4 \in \{2^{\kappa-1}, \ldots, 2^{\kappa}-1\}$ are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1p_2p_3p_4$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map such that

1. (Bilinear) $\forall g, h \in \mathbb{G}, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$;
2. (Non-degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$.

We further require that multiplication in $\mathbb{G}$ and $\mathbb{G}_T$, as well as the bilinear map e, are computable in time polynomial in κ. For $S \subseteq [4]$ we let $\mathbb{G}_{\prod_{i\in S} p_i}$ denote the unique subgroups of $\mathbb{G}$ having order $\prod_{i\in S} p_i$. Note also that if g and h are group elements of different co-prime order, then $e(g, h) = 1_{\mathbb{G}_T}$.

We now state the complexity assumptions we use. Assumptions 1, 2 ,3 and 6 are some instantiations of the General Subgroup Decision (GSD) assumption defined in [3]. Assumption 4 and 5 are essentially the same as Assumption 2 and 3 in [13].

Assumption 1 *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(\kappa),\ \ N = p_1p_2p_3p_4,$$

$$g, X_1 \leftarrow \mathbb{G}_{p_1},\ \ X_2, Y_2, Z_2 \leftarrow \mathbb{G}_{p_2},\ \ g_3, Y_3 \leftarrow \mathbb{G}_{p_3},\ \ g_4, Z_4 \leftarrow \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, e, N, g, g_3, g_4, X_1X_2, Y_2Y_3, Z_2Z_4),$$

$$T_1 \leftarrow \mathbb{G}_{p_1p_3p_4},\ \ T_2 \leftarrow \mathbb{G}_{p_1p_2p_3p_4}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 is defined as

$$Adv^1_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 5 *we say $\mathcal{G}$ satisfies Assumption 1 if for any polynomial time algorithm $\mathcal{A}$, $Adv^1_{\mathcal{A}}(\kappa)$ is negligible.*

Assumption 2 *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(\kappa),\ N = p_1p_2p_3p_4,$$

$$g \leftarrow \mathbb{G}_{p_1},\ g_3 \leftarrow \mathbb{G}_{p_3},\ g_4 \leftarrow \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, e, N, g, g_3, g_4),$$

$$T_1 \leftarrow \mathbb{G}_{p_1},\ T_2 \leftarrow \mathbb{G}_{p_1p_2}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 is defined as

$$Adv^2_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 6 *we say $\mathcal{G}$ satisfies Assumption 2 if for any polynomial time algorithm $\mathcal{A}$, $Adv^2_{\mathcal{A}}(\kappa)$ is negligible.*

Assumption 3 *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(\kappa),\ N = p_1p_2p_3p_4,$$

$$g, X_1 \leftarrow \mathbb{G}_{p_1},\ X_2, Y_2 \leftarrow \mathbb{G}_{p_2},\ g_3, Y_3 \leftarrow \mathbb{G}_{p_3},\ g_4 \leftarrow \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, e, N, g, X_1X_2, Y_2Y_3, g_3, g_4),$$

$$T_1 \leftarrow \mathbb{G}_{p_1p_3},\ T_2 \leftarrow \mathbb{G}_{p_1p_2p_3}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 is defined as

$$Adv^3_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 7 *we say $\mathcal{G}$ satisfies Assumption 3 if for any polynomial time algorithm $\mathcal{A}$, $Adv^3_{\mathcal{A}}(\kappa)$ is negligible.*

Assumption 4 *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(\kappa),\ N = p_1p_2p_3p_4,$$

$$a, s \in \mathbb{Z}_N,\ g \leftarrow \mathbb{G}_{p_1},$$

$$g_2, X_2, Y_2 \leftarrow \mathbb{G}_{p_2},\ g_3 \leftarrow \mathbb{G}_{p_3},\ g_4 \leftarrow \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, e, N, g, g_2, g_3, g_4, g^aX_2, g^sY_2),$$

$$T_1 = e(g, g)^{as},\ T_2 \leftarrow \mathbb{G}_T.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 4 is defined as

$$Adv^4_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 8 *we say $\mathcal{G}$ satisfies Assumption 4 if for any polynomial time algorithm $\mathcal{A}$, $Adv^4_{\mathcal{A}}(\kappa)$ is negligible.*

Assumption 5 *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(\kappa),\ \ N = p_1p_2p_3p_4,$$

$$s \in \mathbb{Z}_N,\ \ g, u, v, h \leftarrow \mathbb{G}_{p_1}, g_2, A_2 \leftarrow \mathbb{G}_{p_2},$$

$$g_3 \leftarrow \mathbb{G}_{p_3},\ \ g_4, W_4 \leftarrow \mathbb{G}_{p_4},\ \ B_{24}, X_{24}, Y_{24}, E_{24} \leftarrow \mathbb{G}_{p_2p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, e, N, gW_4, gA_2, u, u^sB_{24}, v, v^sX_{24}, h, h^sY_{24}, g_2, g_3, g_4),$$

$$T_1 = g^sE_{24},\ \ T_2 \leftarrow \mathbb{G}_{p_1p_2p_4}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 5 is defined as

$$Adv^5_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 9 *we say $\mathcal{G}$ satisfies Assumption 5 if for any polynomial time algorithm $\mathcal{A}$, $Adv^5_{\mathcal{A}}(\kappa)$ is negligible.*

Assumption 6 *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(\kappa),\ \ N = p_1p_2p_3p_4,$$

$$g \leftarrow \mathbb{G}_{p_1},\ \ g_2, X_2 \leftarrow \mathbb{G}_{p_2},\ \ X_3 \leftarrow \mathbb{G}_{p_3},\ \ g_4 \leftarrow \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, e, N, g, g_2, X_2X_3, g_4),$$

$$T_1 \leftarrow \mathbb{G}_{p_1p_2p_4},\ \ T_2 \leftarrow \mathbb{G}_{p_1p_2p_3p_4}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking Assumption 6 is defined as

$$Adv^6_{\mathcal{A}}(\kappa) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 10 *we say $\mathcal{G}$ satisfies Assumption 6 if for any polynomial time algorithm $\mathcal{A}$, $Adv^6_{\mathcal{A}}(\kappa)$ is negligible.*

Assumption 7 (Computational Diffie-Hellman (CDH) Assumption) *Let $\mathcal{G}$ be as above. We define the following distribution:*

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(\kappa),\ \ N = p_1p_2p_3p_4,$$

$$x, y \leftarrow \mathbb{Z}_N,\ \ g \leftarrow \mathbb{G}_{p_1},\ \ g_2 \leftarrow \mathbb{G}_{p_2},\ \ g_3 \leftarrow \mathbb{G}_{p_3},\ \ g_4 \leftarrow \mathbb{G}_{p_4},$$

$$D = (\mathbb{G}, \mathbb{G}_T, e, p_1, p_2, p_3, p_4, g, g_2, g_3, g_4, g^x, g^y).$$

The advantage of an algorithm $\mathcal{A}$ in breaking CDH Assumption in $\mathbb{G}_{p_1}$ is defined as

$$Adv^{\mathsf{CDH}}_{\mathcal{A}}(\kappa) = \Pr[\mathcal{A}(D) = g^{xy}].$$

Definition 11 *we say $\mathcal{G}$ satisfies CDH Assumption in $\mathbb{G}_{p_1}$ if for any PPT algorithm $\mathcal{A}$, $Adv^{\mathsf{CDH}}_{\mathcal{A}}(\kappa)$ is negligible.*

B Identity-Based Encryption

An IBE scheme consists of the following four algorithms:

$\mathsf{Setup}(1^\kappa)$ takes as input a security parameter κ. It generates a public parameter PK and a master secret key MSK.

$\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathsf{ID})$ takes as input the public parameter PK, the master secret key MSK and an identity ID. It outputs a private key $\mathsf{SK}_{\mathsf{ID}}$ for the identity ID.

$\mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID}, M)$ takes as input the public parameter PK, an identity ID and a message M. It outputs a ciphertext CT.

$\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, CT)$ takes as input the public parameter PK, a private key $\mathsf{SK}_{\mathsf{ID}}$ and a ciphertext CT. It outputs either a message M or a failure symbol $\perp$.

An IBE scheme has completeness error ϵ if $\mathsf{Decrypt}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, \mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID}, M)) = M$ holds with probability at least $1-\epsilon$ for all ID and M, $(\mathsf{PK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\kappa)$ and $\mathsf{SK}_{\mathsf{ID}} \leftarrow \mathsf{KeyGen}(\mathsf{PK}, \mathsf{MSK}, \mathsf{ID})$, where the probability is taken over the coins used in encryption.

C Selective Opening Secure Identity-Based Encryption

We recall a simulation-based definition of IBE secure against a selective opening chosen-ciphertext adversary that was originally formalized in [3]. Note that, the model considers adaptive sender corruptions and non-adaptive receiver corruptions. Here an n-message sampler $\mathcal{M}$ is a randomized algorithm that on input string $\alpha \in \{0,1\}^*$ outputs an n-vector $\mathbf{M} = (M^{(1)}, \ldots, M^{(n)})$ of messages, and a relation R is any randomized algorithm that outputs a single bit.

Definition 12 *An identity-based encryption scheme IBE=(Setup, KeyGen, Encrypt, Decrypt) is simulation-based chosen-ciphertext secure under selective openings (SIM-SO-CCA secure) iff for every PPT n-message sampler $\mathcal{M}$, every PPT relation R, every stateful PPT adversary $\mathcal{A}$, there is a stateful PPT simulator $\mathcal{S}$ such that $\mathsf{Adv}^{\textsf{so-cca}}_{\mathsf{IBE},\mathcal{A},\mathcal{S},n,\mathcal{M},\mathsf{R}}(\kappa)$ is negligible, where*

$$\mathsf{Adv}^{\textsf{so-cca}}_{\mathsf{IBE},\mathcal{A},\mathcal{S},n,\mathcal{M},\mathsf{R}}(\kappa) = \Pr\left[\mathsf{Game}^{\textsf{so-cca-real}}_{\mathsf{IBE},\mathcal{A},n,\mathcal{M},\mathsf{R}}(\kappa) = 1\right] - \Pr\left[\mathsf{Game}^{\textsf{so-ideal}}_{\mathsf{IBE},\mathcal{S},n,\mathcal{M},\mathsf{R}}(\kappa) = 1\right].$$

Note that, we require that $\mathcal{A}$ never query $\mathsf{KeyGen}(\cdot)$ on a challenge identity $\mathsf{ID}^{(i)}$ and $\mathsf{Decrypt}(\cdot,\cdot)$ on a challenge ciphertext $(\mathsf{ID}^{(i)}, CT^{(i)})$.

$\mathsf{Game}^{\textsf{so-cca-real}}_{\mathsf{IBE},\mathcal{A},n,\mathcal{M},\mathsf{R}}(\kappa)$ $(\mathsf{PK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\kappa);$ $((\mathsf{ID}^{(i)})_{i\in[n]}, \alpha) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\cdot),\mathsf{Decrypt}(\cdot,\cdot)}(\mathsf{PK});$ $(M^{(i)})_{i\in[n]} \leftarrow \mathcal{M}(\alpha);$ $R^{(1)}, \ldots, R^{(n)} \leftarrow \mathcal{R}_{\mathsf{Encrypt}};$ $(CT^{(i)})_{i\in[n]} = (\mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID}^{(i)}, M^{(i)}; R^{(i)}))_{i\in[n]};$ $I \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\cdot),\mathsf{Decrypt}(\cdot,\cdot)}((CT^{(i)})_{i\in[n]});$ $out_{\mathcal{A}} \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\cdot),\mathsf{Decrypt}(\cdot,\cdot)}((M^{(i)}, R^{(i)})_{i\in I});$ *return* $\mathsf{R}((\mathsf{ID}^{(i)})_{i\in[n]}, (M^{(i)})_{i\in[n]}, I, out_{\mathcal{A}}).$	$\mathsf{Game}^{\textsf{so-ideal}}_{\mathsf{IBE},\mathcal{S},n,\mathcal{M},\mathsf{R}}(\kappa)$ $((\mathsf{ID}^{(i)})_{i\in[n]}, \alpha) \leftarrow \mathcal{S};$ $(M^{(i)})_{i\in[n]} \leftarrow \mathcal{M}(\alpha);$ $I \leftarrow \mathcal{S};$ $out_{\mathcal{S}} \leftarrow \mathcal{S}((M^{(i)})_{i\in I});$ *return* $\mathsf{R}((\mathsf{ID}^{(i)})_{i\in[n]}, (M^{(i)})_{i\in[n]}, I, out_{\mathcal{S}}).$

D Proof of Theorem 1

Proof. We first show that encryptions of $\overbrace{1 \cdots 1}^{\ell}$ are computational indistinguishable to encryptions of real messages in the SOA setting. We consider the following games (for k from 1 to $n\ell$):

Game_0: This is the real SIM-SO-CCA security game $\mathsf{Game}^{\text{so-cca-real}}_{\mathsf{IBE},\mathcal{A},n,\mathcal{M},\mathsf{R}}(\kappa)$.

$\mathsf{Game}_k(1 \le k \le n\ell)$: It is the same as Game_0 except for two differences. The first difference is the way of creating the vector of challenge ciphertexts, where the first k bits sampled from $\mathcal{M}$ (possibly across many messages) are replaced by 1s. The second one is how the adversary $\mathcal{A}$'s corruption query is answered.

- When adversary $\mathcal{A}$ queries the encryption oracle for challenge ciphertexts, the challenger responds in this way:
  1. The challenger sets $(M^{(i)})_{i\in[n]} \leftarrow \mathcal{M}(\alpha)$. For each $i \in [n]$, let $M^{(i)} = \left(m_1^{(i)} \| \cdots \| m_\ell^{(i)}\right)$.
  2. Let $k = (\zeta - 1)\ell + \varrho$ with $\zeta \in [n]$ and $\varrho \in [\ell]$.
     For each $i \in [n]$ such that $i \neq \zeta$, the challenger sets
     $$CT^{(i)} = \begin{cases} \mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID}^{(i)}, \underbrace{1 \cdots 1}_{\ell}) & \text{if } 1 \le i < \zeta \\ \mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID}^{(i)}, M^{(i)}) & \text{if } \zeta < i \le n \end{cases}.$$
     For $i = \zeta$ and each $j \in [\ell]$, the challenger computes
     $$\begin{cases} (C_j^{(\zeta)}, K_j^{(\zeta)}) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, 1; R_j^{(\zeta)}) & \text{if } 1 \le j \le \varrho \\ (C_j^{(\zeta)}, K_j^{(\zeta)}) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, 1; R_j^{(\zeta)}) & \text{if } \varrho < j \le \ell \text{ and } m_j^{(\zeta)} = 1 \\ \begin{cases} C_j^{(\zeta)} \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, 0; R_j^{(\zeta)}), \\ K_j^{(\zeta)} \leftarrow \mathsf{Sample}'(\mathcal{K}; R_j^{(\zeta)K}) \end{cases} & \text{if } \varrho < j \le \ell \text{ and } m_j^{(\zeta)} = 0 \end{cases},$$
     $$K_{\ell+1}^{(\zeta)} = \left(K_a, \mathsf{H}(\mathsf{ID}^{(\zeta)}, C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)})\right), \ T^{(\zeta)} = \mathsf{XAuth}(K_1^{(\zeta)}, \ldots, K_{\ell+1}^{(\zeta)}),$$
     where $R_j^{(\zeta)} \leftarrow \mathcal{R}_{\mathsf{Encrypt}_{ex}}$ and $R_j^{(\zeta)K} \leftarrow \mathcal{R}_{\mathsf{Sample}'}$. Then, the challenger sets $CT^{(\zeta)} = (C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)}, T^{(\zeta)})$.
  3. Finally, the challenger returns $(CT^{(i)})_{i\in[n]}$ to the adversary $\mathcal{A}$ as its challenge ciphertexts. At the same time, for $i \in [n], j \in [\ell]$, the challenger also records all the random coins $R_j^{(i)}$ used to obtain $C_j^{(i)}$ and all the keys $K_j^{(i)}$. The random coins $R_j^{(i)K}$, which are used to sample $K_j^{(i)}$ when $m_j^{(i)} = 0$ and $(i-1)\ell + j > k$, are also recorded.
- When adversary $\mathcal{A}$ queries the corruption oracle with $I \subset [n]$, for each $i \in I$, the challenger responds with $\left(M^{(i)} = \left(m_1^{(i)} \| \cdots \| m_\ell^{(i)}\right), (\bar{R}_1^{(i)}, \ldots, \bar{R}_\ell^{(i)})\right)$. The random coins $\left(\bar{R}_j^{(i)}\right)_{j\in[\ell]}$ are prepared as follows.
  1. If $(i-1)\ell + j > k$, the challenger sets $\bar{R}_j^{(i)}$ as $R_j^{(i)}$ (in case of $m_j^{(i)} = 1$) or $\left(R_j^{(i)}, R_j^{(i)K}\right)$ (in case of $m_j^{(i)} = 0$). In fact, $\bar{R}_j^{(i)}$ are the original random coins that the challenger used to generate $(C_j^{(i)}, K_j^{(i)})$. This behaves just like that in Game_0.

2. Else (i.e., $(i-1)\ell + j \le k$), the challenger sets

$$\bar{R}_j^{(i)} = \begin{cases} R_j^{(i)} & \text{if } m_j^{(i)} = 1 \\ (\mathsf{POpen}(\mathsf{PK}, \mathsf{ID}, C_j^{(i)}),\ \mathsf{Sample}'^{-1}(\mathcal{K}, \hat{K}_j^{(i)})) & \text{if } m_j^{(i)} = 0 \end{cases}, \tag{7}$$

where $\hat{K}_j^{(i)} = \mathsf{ReSamp}((K_w^{(i)})_{1 \le w \le \ell+1, w \ne j}, T^{(i)})$. As soon as $\hat{K}_j^{(i)}$ is computed, reset $K_j^{(i)} := \hat{K}_j^{(i)}$. Recall that ReSamp is the resample algorithm of strengthened cross authentication code XAC.

We will prove that for $1 \le k \le n\ell$, Game_{k-1} and Game_k are computationally indistinguishable in Lemma 1. Then Game_0 and $\mathsf{Game}_{n\ell}$ are also computationally indistinguishable by hybrid argument. Observe that, in $\mathsf{Game}_{n\ell}$, when the adversary queries the encryption oracle for challenge ciphertexts, all messages from $\mathcal{M}$ are completely ignored and each challenge ciphertext is an encryption of message $\overbrace{1 \cdots 1}^{\ell}$. This results in a PPT-simulator $\mathcal{S}$ for $\mathcal{A}$ in $\mathsf{Game}_{n\ell}$.

Setup The simulator $\mathcal{S}$ first runs the algorithm Setup to generate the public parameter $\mathsf{PK} = (\mathsf{PK}_{ex}, \mathsf{H}, K_a)$ and the master secret key $\mathsf{MSK} = \mathsf{MSK}_{ex}$. Then, it sends PK to $\mathcal{A}$.

Query The adversary $\mathcal{A}$ adaptively issues key and decryption queries, and $\mathcal{S}$ answers the queries with the help of the master secret key MSK.

Challenge At some point, $\mathcal{A}$ queries the encryption oracle on $((\mathsf{ID}^{(i)})_{i \in [n]}, \alpha)$ for challenge ciphertexts. $\mathcal{S}$ forwards the query to its own oracle, receiving nothing in response. $\mathcal{S}$ then generates ciphertexts $(CT^{(i)})_{i \in [n]}$, where each ciphertext is an encryption of message $\overbrace{1 \cdots 1}^{\ell}$. Finally, $\mathcal{S}$ returns $(CT^{(i)})_{i \in [n]}$ to the adversary $\mathcal{A}$ as its challenge ciphertexts.

Corrupt $\mathcal{A}$ queries the corruption oracle on a set $I \subset [n]$. $\mathcal{S}$ queries its own corruption oracle on I and learns $(M^{(i)})_{i \in I}$. Then, for each index $i \in I$, $\mathcal{S}$ finds the coins $\bar{R}^{(i)} = \left(\bar{R}_1^{(i)}, \ldots, \bar{R}_\ell^{(i)}\right)$ that can open the ciphertext $CT^{(i)}$ to $M^{(i)}$ according to Eq.(7). (Note that, $CT^{(i)}$ is an encryption of message $\overbrace{1 \cdots 1}^{\ell}$. Since $\overbrace{1 \cdots 1}^{\ell}$-encryptions are equivocable, thus $CT^{(i)}$ can be opened to any message and $\bar{R}^{(i)}$ can be found by $\mathcal{S}$ with Eq.(7).) Finally, $\mathcal{S}$ sends $(M^{(i)}, \bar{R}^{(i)})_{i \in I}$ to $\mathcal{A}$.

Output The adversary $\mathcal{A}$ halts with output $out_{\mathcal{A}}$, and $\mathcal{S}$ halts with the same output.

Obviously, $\mathcal{S}$ can serves as the soa-simulator in $\mathsf{Game}^{\text{so-ideal}}_{\mathsf{IBE},\mathcal{S},n,\mathcal{M},\mathsf{R}}$, so we have that

$$\Pr\left[\mathsf{Game}_{n\ell} = 1\right] = \Pr\left[\mathsf{Game}^{\text{so-ideal}}_{\mathsf{IBE},\mathcal{S},n,\mathcal{M},\mathsf{R}}(\kappa) = 1\right].$$

To sum up, we get that $\Pr\left[\mathsf{Game}^{\text{so-cca-real}}_{\mathsf{IBE},\mathcal{A},n,\mathcal{M},\mathsf{R}} = 1\right] - \Pr\left[\mathsf{Game}^{\text{so-ideal}}_{\mathsf{IBE},\mathcal{S},n,\mathcal{M},\mathsf{R}} = 1\right]$ is negligible, which proves the theorem. □

Lemma 1 *If the extractable 1SPO-IBE scheme is IND-ID-CCA secure, the hash function H is collision-resistant and the strengthened $\ell+1$-cross-authentication code XAC is secure against substitution attacks, then for each $k \in [n\ell]$, Game_{k-1} and Game_k are computationally indistinguishable.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that can distinguish Game_k and Game_{k-1} with non-negligible advantage. We build an algorithm $\mathcal{B}$ that breaks $\mathsf{IND\text{-}ID\text{-}CCA}$ security of the extractable

1SPO-IBE scheme with non-negligible advantage. In the $\mathsf{IND\text{-}ID\text{-}CCA}$ security game of the extractable 1SPO-IBE scheme, $\mathcal{B}$ is given a public parameter, and also is provided with an encryption oracle for challenge ciphertext, a key generation oracle and a decryption oracle by its own challenger. Now $\mathcal{B}$ simulates an environment for $\mathcal{A}$.

Recall that $k \in [n\ell]$. Let $k = (\zeta - 1)\ell + \varrho$ with $\zeta \in [n]$ and $\varrho \in [\ell]$. When n messages are sampled from $\mathcal{M}$, we get totally $n\ell$ bits. In formula, $(M^{(i)})_{i\in[n]} \leftarrow \mathcal{M}(\alpha)$ with $M^{(i)} = \left(m_1^{(i)}\|\cdots\|m_\ell^{(i)}\right)$. Then the index k will locate bit $m_\varrho^{(\zeta)}$ among $(M^{(i)})_{i\in[n]}$. If $m_\varrho^{(\zeta)} = 1$, then Game_{k-1} and Game_k are identical. Thus, without loss of generality, we assume that $m_\varrho^{(\zeta)} = 0$.

Public Parameter. $\mathcal{B}$ gets a public parameter PK_{ex} of the extractable 1SPO-IBE scheme from its own challenger. Next, $\mathcal{B}$ chooses $K_a \leftarrow \mathcal{K}_a$, a collision-resistant hash function $\mathsf{H} : \mathcal{ID} \times \overbrace{\mathcal{C} \times \cdots \times \mathcal{C}}^{\ell} \rightarrow \mathcal{K}_b$, and sets the public parameter $\mathsf{PK} = (\mathsf{PK}_{ex}, \mathsf{H}, K_a)$. Then, it sends PK to $\mathcal{A}$.

Encryption Query. When $\mathcal{A}$ queries the encryption oracle on $((\mathsf{ID}^{(i)})_{i\in[n]}, \alpha)$ for challenge ciphertexts. $\mathcal{B}$ proceeds as follows.

1. Compute (ζ, ϱ) such that $k = (\zeta - 1)\ell + \varrho$ with $\zeta \in [n]$ and $\varrho \in [\ell]$. $\mathcal{B}$ queries its own encryption oracle with $\mathsf{ID}^{(\zeta)}$ for challenge ciphertext and session key, and let (C^*, K^*) be the response of $\mathcal{B}$'s encryption oracle.
2. $\mathcal{B}$ samples $(M^{(i)})_{i\in[n]} \leftarrow \mathcal{M}(\alpha)$. For each $i \in [n]$, let $M^{(i)} = \left(m_1^{(i)}\|\cdots\|m_\ell^{(i)}\right)$.
3. $\mathcal{B}$ prepares the vector of challenge ciphertexts for $\mathcal{A}$.
   (a) For $i \in [n]$ such that $i \neq \zeta$, $\mathcal{B}$ sets

$$CT^{(i)} = (C_1^{(i)}, \ldots, C_\ell^{(i)}, T^{(i)}) = \begin{cases} \mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID}^{(i)}, \underbrace{1\cdots1}_{\ell}) & \text{if } 1 \leq i < \zeta \\ \\ \mathsf{Encrypt}(\mathsf{PK}, \mathsf{ID}^{(i)}, M^{(i)}) & \text{if } \zeta < i \leq n \end{cases}.$$

   (b) For $i = \zeta$, $\mathcal{B}$ embeds its own challenge (C^*, K^*) in the creation of $CT^{(\zeta)} = (C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)}, T^{(\zeta)})$ by setting $C_\varrho^{(\zeta)} := C^*$ and $K_\varrho^{(\zeta)} := K^*$. More precisely, for each $j \in [\ell]$, $\mathcal{B}$ sets

$$\begin{cases} (C_j^{(\zeta)}, K_j^{(\zeta)}) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, 1; R_j^{(\zeta)}) & \text{if } 1 \leq j < \varrho \\ (C_\varrho^{(\zeta)}, K_\varrho^{(\zeta)}) = (C^*, K^*) & \text{if } j = \varrho \\ (C_j^{(\zeta)}, K_j^{(\zeta)}) \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, 1; R_j^{(\zeta)}) & \text{if } \varrho < j \leq \ell \text{ and } m_j^{(\zeta)} = 1 \\ \begin{cases} C_j^{(\zeta)} \leftarrow \mathsf{Encrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, 0; R_j^{(\zeta)}), \\ K_j^{(\zeta)} \leftarrow \mathsf{Sample}'(\mathcal{K}; R_j^{(\zeta)K}) \end{cases} & \text{if } \varrho < j \leq \ell \text{ and } m_j^{(\zeta)} = 0 \end{cases},$$

$$K_{\ell+1}^{(\zeta)} = \left(K_a, \mathsf{H}(\mathsf{ID}^{(\zeta)}, C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)})\right), \quad T^{(\zeta)} = \mathsf{XAuth}(K_1^{(\zeta)}, \ldots, K_{\ell+1}^{(\zeta)}),$$
$$CT^{(\zeta)} = (C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)}, T^{(\zeta)}),$$

   where $R_j^{(\zeta)} \leftarrow \mathcal{R}_{\mathsf{Encrypt}_{ex}}$ and $R_j^{(\zeta)K} \leftarrow \mathcal{R}_{\mathsf{Sample}'}$.
4. Finally, $\mathcal{B}$ returns $(CT^{(i)})_{i\in[n]}$ to $\mathcal{A}$ as its challenge ciphertexts. For $i \in [n], j \in [\ell]$, $\mathcal{B}$ records all the random coins $R_j^{(i)}$ used to obtain $C_j^{(i)}$ and all the keys $K_j^{(i)}$. $\mathcal{B}$ also records all the random coins $R_j^{(i)K}$ that are used to sample $K_j^{(i)}$ when $m_j^{(i)} = 0$ and $(i-1)\ell + j > k$.

Corruption Query. When $\mathcal{A}$ queries the corruption oracle on a set $I \subset [n]$, for each index $i \in I$, $\mathcal{B}$ responds with $(M^{(i)}, (\bar{R}_1^{(i)}, \ldots, \bar{R}_\ell^{(i)}))$. The random coins $\left(\bar{R}_j^{(i)}\right)_{j\in[\ell]}$ are prepared as follows. (Recall that, $(M^{(i)})_{i\in[n]} \leftarrow \mathcal{M}(\alpha)$, $M^{(i)} = \left(m_1^{(i)}\|\cdots\|m_\ell^{(i)}\right)$ and $m_\varrho^\zeta = 0$.)

1. If $(i-1)\ell + j > k$, $\mathcal{B}$ sets $\bar{R}_j^{(i)}$ as $R_j^{(i)}$ (in case of $m_j^{(i)} = 1$) or $\left(R_j^{(i)}, R_j^{(i)K}\right)$ (in case of $m_j^{(i)} = 0$). In fact, $\bar{R}_j^{(i)}$ are the original random coins that $\mathcal{B}$ used to generate $(C_j^{(i)}, K_j^{(i)})$.
2. Else (i.e., $(i-1)\ell + j \le k$), $\mathcal{B}$ sets

$$\bar{R}_j^{(i)} = \begin{cases} R_j^{(i)} & \text{if } m_j^{(i)} = 1 \\ (\mathsf{POpen}(\mathsf{PK}, \mathsf{ID}, C_j^{(i)}),\ \mathsf{Sample}'^{-1}(\mathcal{K}, \hat{K}_j^{(i)})) & \text{if } m_j^{(i)} = 0 \end{cases},$$

where $\hat{K}_j^{(i)} = \mathsf{ReSamp}((K_w^{(i)})_{1\le w\le \ell+1, w\ne j}, T^{(i)})$. As soon as $\hat{K}_j^{(i)}$ is computed, reset $K_j^{(i)} := \hat{K}_j^{(i)}$.

Private Key Query & Decryption Query. $\mathcal{B}$ answers the adversary $\mathcal{A}$'s private key and decryption queries as follows.

– $\mathcal{A}$'s private key query on $\widetilde{\mathsf{ID}}$. Since $\mathcal{A}$ is not allowed to query any $\mathsf{ID}^{(i)}$ $(1 \le i \le n)$, we have $\widetilde{\mathsf{ID}} \ne \mathsf{ID}^{(\zeta)}$. $\mathcal{B}$ can query its own key generation oracle on $\widetilde{\mathsf{ID}}$ to obtain the private key $\mathsf{SK}_{\widetilde{\mathsf{ID}}}$, and sends it to $\mathcal{A}$.

– $\mathcal{A}$'s decryption query on $\langle \widetilde{\mathsf{ID}}, \widetilde{CT} \rangle$. Since $\mathcal{A}$ is not allowed to query $\langle \mathsf{ID}^{(i)}, CT^{(i)} \rangle$ for $i = 1, \ldots, n$, we have $\langle \widetilde{\mathsf{ID}}, \widetilde{CT} \rangle \ne \langle \mathsf{ID}^{(\zeta)}, CT^{(\zeta)} \rangle$.
If $\widetilde{\mathsf{ID}} \ne \mathsf{ID}^{(\zeta)}$, $\mathcal{B}$ queries its own key generation oracle on $\langle \widetilde{\mathsf{ID}} \rangle$ to obtain the private key $\mathsf{SK}_{\widetilde{\mathsf{ID}}}$, decrypts $\widetilde{CT}$ with $\mathsf{SK}_{\widetilde{\mathsf{ID}}}$ and algorithm $\mathsf{Decrypt}$, and sends the result to $\mathcal{A}$.

Else (i.e., $\widetilde{\mathsf{ID}} = \mathsf{ID}^{(\zeta)}$), it follows that $\widetilde{CT} \ne CT^{(\zeta)}$. $\mathcal{B}$ proceeds as follows.

1. $\mathcal{B}$ parses $\widetilde{CT}$ as $(\widetilde{C}_1, \ldots, \widetilde{C}_\ell, \widetilde{T})$. Recall that $CT^{(\zeta)} = (C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)}, T^{(\zeta)})$ with $C_\varrho^{(\zeta)} = C^*$ and $K_\varrho^{(\zeta)} = K^*$.
2. $\mathcal{B}$ computes $\widetilde{K}_b = \mathsf{H}(\widetilde{\mathsf{ID}}, \widetilde{C}_1, \ldots, \widetilde{C}_\ell)$ and set $\widetilde{K}_{\ell+1} = (K_a, \widetilde{K}_b)$. It checks whether $\mathsf{XVer}(\widetilde{K}_{\ell+1}, \widetilde{T}) = 1$. If not, $\mathcal{B}$ returns the message $\widetilde{M}'' = \overbrace{0\cdots0}^{\ell}$ to $\mathcal{A}$.
If $\mathsf{XVer}(\widetilde{K}_{\ell+1}, \widetilde{T}) = 1$ holds, then for each $j \in [\ell]$, $\mathcal{B}$ does the following:
   - In case of $\widetilde{C}_j \ne C^*$, $\mathcal{B}$ queries its own decryption oracle with $\langle \widetilde{\mathsf{ID}} = \mathsf{ID}^{(\zeta)}, \widetilde{C}_j \rangle$ to obtain $(\widetilde{m}'_j, \widetilde{K}'_j) \leftarrow \mathsf{Decrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{SK}_{\widetilde{\mathsf{ID}}}, \widetilde{C}_j)$, and sets

$$\widetilde{m}''_j = \begin{cases} \mathsf{XVer}(\widetilde{K}'_j, \widetilde{T}) & \text{if } \widetilde{m}'_j = 1 \\ 0 & \text{if } \widetilde{m}'_j = 0 \end{cases}.$$

   - In case of $\widetilde{C}_j = C^*$ (Recall that $\mathcal{B}$ is not allowed to query $\langle \mathsf{ID}^{(\zeta)} = \widetilde{\mathsf{ID}}, C^* \rangle$ to its own decryption oracle), $\mathcal{B}$ sets

$$\widetilde{m}''_j = \begin{cases} 1 & \text{if } \mathsf{XVer}(K^*, \widetilde{T}) = 1 \\ 0 & \text{if } \mathsf{XVer}(K^*, \widetilde{T}) = 0 \end{cases}.$$

$\mathcal{B}$ sets $\widetilde{M}'' = (\widetilde{m}''_1\| \cdots \|\widetilde{m}''_\ell)$ and returns the message $\widetilde{M}''$ to $\mathcal{A}$.

If $\mathcal{B}$'s challenge ciphertext and session key (C^*, K^*), given by $\mathcal{B}$'s challenger, is an encryption of 1, then $\mathsf{Decrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{SK}_{\mathsf{ID}^{(\zeta)}}, C^*)$ will always outputs $(1, K^*)$; thus $\mathcal{B}$ has simulated Game_k properly. On the other hand, we will show in the following that if C^* is an encryption of 0 and K^* is uniformly distributed in $\mathcal{K}$, with overwhelming probability, $\mathcal{B}$ has simulated Game_{k-1} properly. Thus, if $\mathcal{A}$ can distinguish Game_{k-1} and Game_k with non-negligible advantage, algorithm $\mathcal{B}$ breaks IND-ID-CCA security of the extractable 1SPO-IBE scheme with non-negligible advantage.

Observe that, if C^* is an encryption of 0 and K^* is uniformly distributed in $\mathcal{K}$, the differences between the environment simulated by $\mathcal{B}$ and Game_{k-1} lie in:

1. Different way to open $(C_\varrho^{(\zeta)}, K_\varrho^{(\zeta)})$ in the corruption phase. Recall that, $C_\varrho^{(\zeta)}$ is an encryption of 0, and $K_\varrho^{(\zeta)}$ is a randomly distributed in $\mathcal{K}$.
   - In Game_{k-1}, $(C_\varrho^{(\zeta)}, K_\varrho^{(\zeta)})$ is opened with $(R_\varrho^{(\zeta)}, R_\varrho^{(\zeta)K})$, where $R_\varrho^{(\zeta)}$ is the random coins used to obtain $C_\varrho^{(\zeta)}$ and $R_\varrho^{(\zeta)K}$ is the random coins used to sample $K_\varrho^{(\zeta)}$ from $\mathcal{K}$.
   - In the environment simulated by $\mathcal{B}$, $C_\varrho^{(\zeta)} = C^*$ is an encryption of 0, and $K_\varrho^{(\zeta)} = K^*$ is a randomly distributed in $\mathcal{K}$, and $(C_\varrho^{(\zeta)}, K_\varrho^{(\zeta)})$ is opened with
   $$(\mathsf{POpen}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, C^*),\ \mathsf{Sample}'^{-1}(\mathcal{K}, \hat{K}_\varrho^{(\zeta)})),$$
   where $\hat{K}_\varrho^{(\zeta)} = \mathsf{ReSamp}((K_w^{(\zeta)})_{1\le w\le \ell+1, w\ne\varrho}, T^{(\zeta)})$.

   Given $C_\varrho^{(\zeta)}$, the random coins $R_\varrho^{(\zeta)}$ and the output of $\mathsf{POpen}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, C_\varrho^{(\zeta)} = C^*)$ share the same probability distribution, hence they are statistically indistinguishable, i.e.
   $$R_\varrho^{(\zeta)} \overset{s}{\approx} \mathsf{POpen}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, C^*).$$
   On the other hand, since $\hat{K}_\varrho^{(\zeta)} = \mathsf{ReSamp}((K_w^{(\zeta)})_{1\le w\le \ell+1, w\ne\varrho}, T^{(\zeta)})$, the resample algorithm of XAC guaranteed that conditioned on $T^{(\zeta)}$ and $(K_w^{(\zeta)})_{1\le w\le \ell+1, w\ne\varrho}$,
   $$\hat{K}_\varrho^{(\zeta)} \overset{s}{\approx} K_\varrho^{(\zeta)},\ \text{ hence } R_\varrho^{(\zeta)K} \overset{s}{\approx} \mathsf{Sample}'^{-1}(\mathcal{K}, \hat{K}_\varrho^{(\zeta)}).$$
   Consequently,
   $$(R_\varrho^{(\zeta)}, R_\varrho^{(\zeta)K}) \overset{s}{\approx} \Big(\mathsf{POpen}(\mathsf{PK}_{ex}, \mathsf{ID}^{(\zeta)}, C^*), \mathsf{Sample}'^{-1}(\mathcal{K}, \hat{K}_\varrho^{(\zeta)})\Big)$$
   conditioned on $CT^{(\zeta)}$ and other opened information.
2. Different way to compute bit $\widetilde{m}''_j$ for decryption query $\Big\langle \widetilde{\mathsf{ID}}, \widetilde{CT} = \Big(\widetilde{C}_1, \ldots, \widetilde{C}_\ell, \widetilde{T}\Big)\Big\rangle$ such that $\widetilde{\mathsf{ID}} = \mathsf{ID}^{(\zeta)}$, $\mathsf{XVer}((K_a, \mathsf{H}(\widetilde{\mathsf{ID}}, \widetilde{C}_1, \ldots, \widetilde{C}_\ell)), \widetilde{T}) = 1$ and $\widetilde{C}_j = C_\varrho^{(\zeta)} = C^*$ for some $j \in [\ell]$.
   - In Game_{k-1}, the challenger computes $(\widetilde{m}'_j, \widetilde{K}'_j) \leftarrow \mathsf{Decrypt}_{ex}(\mathsf{PK}_{ex}, \mathsf{SK}_{\widetilde{\mathsf{ID}}}, \widetilde{C}_j)$ and sets
   $$\widetilde{m}''_j := \begin{cases} \mathsf{XVer}(\widetilde{K}'_j, \widetilde{T}) & \text{if } \ \widetilde{m}'_j = 1 \\ 0 & \text{if } \ \widetilde{m}'_j = 0 \end{cases}.$$
   Recall that $\widetilde{C}_j = C_\varrho^{(\zeta)} = C^*$ is an encryption of 0, hence decryption of $\widetilde{C}_j$ will result in message bit $\widetilde{m}'_j = 0$ and a random key $\widetilde{K}'_j$, except with negligible probability. Consequently, $\widetilde{m}''_j = 0$ except with negligible probability.

– In the environment simulated by $\mathcal{B}$, $\widetilde{m}''_j$ is computed as

$$\widetilde{m}''_j = \begin{cases} 1 & \text{if } \mathsf{XVer}(K^*, \widetilde{T}) = 1 \\ 0 & \text{if } \mathsf{XVer}(K^*, \widetilde{T}) = 0 \end{cases}.$$

Next, we show that $\widetilde{m}''_j = 0$ except with negligible probability. We first show that if $\widetilde{\mathsf{ID}} = \mathsf{ID}^{(\zeta)}$ and $\widetilde{CT} \neq CT^{(\zeta)}$, then $\widetilde{T} = T^{(\zeta)}$ with negligible probability, due to the collision resistance of hash function H. The reason is as follows. Recall that, $\mathsf{XVer}((K_a, \mathsf{H}(\widetilde{\mathsf{ID}}, \widetilde{C}_1, \ldots, \widetilde{C}_\ell)), \widetilde{T}) = 1$. If $\widetilde{T} = T^{(\zeta)}$, then

$$\mathsf{XVer}((K_a, \mathsf{H}(\widetilde{\mathsf{ID}}, \widetilde{C}_1, \ldots, \widetilde{C}_\ell)), T^{(\zeta)}) = 1. \tag{8}$$

On the other hand, $K_{\ell+1}^{(\zeta)} = (K_a, K_b^{(\zeta)}) = (K_a, \mathsf{H}(\mathsf{ID}^{(\zeta)} = \widetilde{\mathsf{ID}}, C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)}))$ is used to generated $T^{(\zeta)}$ in the challenge ciphertext $CT^{(\zeta)}$, so

$$\mathsf{XVer}(K_{\ell+1}^{(\zeta)}, T^{(\zeta)}) = \mathsf{XVer}((K_a, \mathsf{H}(\widetilde{\mathsf{ID}}, C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)})), T^{(\zeta)}) = 1. \tag{9}$$

Since the $\ell + 1$-cross-authentication code XAC is *semi-unique*, it follows from Eqs.(8) and (9) that

$$\mathsf{H}(\widetilde{\mathsf{ID}}, \widetilde{C}_1, \ldots, \widetilde{C}_\ell) = \mathsf{H}(\widetilde{\mathsf{ID}}, C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)}). \tag{10}$$

The fact that $\widetilde{CT} \neq CT^{(\zeta)}$ means $(\widetilde{C}_1, \ldots, \widetilde{C}_\ell, \widetilde{T}) \neq (C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)}, T^{(\zeta)})$. Then $\widetilde{T} = T^{(\zeta)}$ implies $(\widetilde{C}_1, \ldots, \widetilde{C}_\ell) \neq (C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)})$. So

$$(\widetilde{\mathsf{ID}}, \widetilde{C}_1, \ldots, \widetilde{C}_\ell) \neq (\widetilde{\mathsf{ID}}, C_1^{(\zeta)}, \ldots, C_\ell^{(\zeta)}). \tag{11}$$

Eqs. (10) and (11) suggest a hash collision.
Now we assume $\widetilde{T} \neq T^{(\zeta)}$. Recall that $K_\varrho^{(\zeta)} = K^*$ is used in the generation of $T^{(\zeta)}$ in the challenge ciphertext $CT^{(\zeta)}$, so $\mathsf{XVer}(K^*, T^{(\zeta)}) = 1$. In the corruption phase, if the adversary asks to open $(C_\varrho^{(\zeta)}, K_\varrho^{(\zeta)})$ (i.e., (C^*, K^*)), it can obtain the information $(K_w^{(\zeta)})_{1 \le w \le \ell+1, w \neq \varrho}$ and $\hat{K}_\varrho^{(\zeta)}$, where $\hat{K}_\varrho^{(\zeta)} = \mathsf{ReSamp}((K_w^{(\zeta)})_{1 \le w \le \ell+1, w \neq \varrho}, T^{(\zeta)})$. Strongness of the XAC guarantees that $\hat{K}_\varrho^{(\zeta)}$ is statistically indistinguishable to $K_\varrho^{(\zeta)}$ (i.e., K^*), even given $(K_w^{(\zeta)})_{1 \le w \le \ell+1, w \neq \varrho}$ and $T^{(\zeta)}$. Observe that, the knowledge of $\hat{K}_\varrho^{(\zeta)}$ does not leak information about $K_\varrho^{(\zeta)}$ (i.e., K^*) since $\hat{K}_\varrho^{(\zeta)}$ is generated from $(K_w^{(\zeta)})_{1 \le w \le \ell+1, w \neq \varrho}$ and $T^{(\zeta)}$. Thus, we have

$$\Pr[m''_j = 1] = \Pr[\mathsf{XVer}(K^*, \widetilde{T}) = 1] \le \mathsf{Adv}_{\mathsf{XAC}}^{\mathsf{sub}}(\kappa).$$

To sum up, $\widetilde{m}''_j$ will be decrypted to 0, except with negligible probability, no matter in Game_{k-1} or the environment simulated by $\mathcal{B}$.

□

E Proof of Theorem 2

Proof. Following the approach by Lewko and Waters [25], we define two additional structures: *semi-functional* ciphertexts and *semi-functional* keys. These will not be used in the real system, but will be used in our proof.

Semi-functional Ciphertext Let g_2 denote a generator of subgroup $\mathbb{G}_{p_2}$. A semi-functional ciphertext is created as follows. We first use the encryption algorithm to obtain a normal ciphertext (c_0', c_1'). Then, we choose $t_2, t_2' \leftarrow \mathbb{Z}_N$, and set the semi-functional ciphertext to be

$$c_0 = c_0' g_2^{t_2},\ c_1 = c_1' g_2^{t_2 t_2'}.$$

Semi-functional Key A semi-functional key will take one of two forms. To create a semi-functional key, we first use the key generation algorithm to form a normal key $(\mathsf{ID}, D_0', D_1', D_2', \bar{D}_0', \bar{D}_1', \bar{D}_2')$. Then, we choose $r_2, r_2', r_2'', \bar{r}_2, \bar{r}_2', \bar{r}_2'' \leftarrow \mathbb{Z}_N$. The semi-functional key of type 1 is set as:

$$\mathsf{ID},\ D_0 = D_0' g_2^{r_2},\ D_1 = D_1' g_2^{r_2'},\ D_2 = D_2' g_2^{r_2''},$$
$$\bar{D}_0 = \bar{D}_0', \bar{D}_1 = \bar{D}_1',\ \bar{D}_2 = \bar{D}_2'.$$

The semi-functional key of type 2 is set as:

$$\mathsf{ID},\ D_0 = D_0' g_2^{r_2},\ D_1 = D_1' g_2^{r_2'},\ D_2 = D_2' g_2^{r_2''},$$
$$\bar{D}_0 = \bar{D}_0' g_2^{\bar{r}_2}, \bar{D}_1 = \bar{D}_1' g_2^{\bar{r}_2'},\ \bar{D}_2 = \bar{D}_2' g_2^{\bar{r}_2''}.$$

Let q denote the total number of key and decryption queries the adversary makes. We will prove the $\mathsf{IND\text{-}ID\text{-}CCA}$ security of our scheme using a hybrid argument over a sequence of games.

$\mathsf{Game}_{\mathsf{Real}}$ The real $\mathsf{IND\text{-}ID\text{-}CCA}$ security game.

$\mathsf{Game}_{\mathsf{Restricted}_1}$ This game is the same as $\mathsf{Game}_{\mathsf{Real}}$ except that the challenger outputs reject and halts if the adversary issues a key query $\langle \mathsf{ID} \rangle$ such that

$$\mathsf{ID} \not\equiv \mathsf{ID}^* \mod N, \quad \text{and} \quad \mathsf{ID} \equiv \mathsf{ID}^* \mod p_2,$$

where ID^* is the challenge identity.

$\mathsf{Game}_{\mathsf{Restricted}_2}$ This is like $\mathsf{Game}_{\mathsf{Restricted}_1}$ except for the way that the challenger answers the decryption queries made by the adversary.
Let $(C^* = (c_0^*, c_1^*), K^*)$ be the challenge ciphertext and session key. Recall that, C^* is a ciphertext encrypting δ under a challenge identity ID^*, where $\delta \leftarrow \{0,1\}$ is chosen by the challenger. When the adversary issues a decryption query $\langle \mathsf{ID}, C = (c_0, c_1) \rangle$, the challenger proceeds just like in $\mathsf{Game}_{\mathsf{Restricted}_1}$, except for the following cases.

1. If $\mathsf{ID} = \mathsf{ID}^*$, $c_0 = c_0^*$ and $c_1 \neq c_1^*$, then the challenger outputs $(0, K)$ with $K \leftarrow \mathbb{G}_T$.
2. Else if $\mathsf{ID} \not\equiv \mathsf{ID}^* \mod N$ and $\mathsf{ID} \equiv \mathsf{ID}^* \mod p_2$, then the challenger outputs reject and halts.
3. Else if $\mathsf{ID} = \mathsf{ID}^*$, $c_0 \neq c_0^*$ and $\mathsf{H}(\mathsf{ID}, c_0) = \mathsf{H}(\mathsf{ID}^*, c_0^*)$, then the challenger outputs reject and halts.
4. Else if $\mathsf{ID} = \mathsf{ID}^*$, $\mathsf{H}(\mathsf{ID}, c_0) \not\equiv \mathsf{H}(\mathsf{ID}^*, c_0^*) \mod N$ and $\mathsf{H}(\mathsf{ID}, c_0) \equiv \mathsf{H}(\mathsf{ID}^*, c_0^*) \mod p_2$, then the challenger outputs reject and halts.

$\mathsf{Game}_{\mathsf{ch}}$ This game is the same as $\mathsf{Game}_{\mathsf{Restricted}_2}$ except that the challenge ciphertext is replaced with a semi-functional ciphertext in case of $\delta = 1$.

$\mathsf{Game}_{k,1} (1 \leq k \leq q)$ This game is like $\mathsf{Game}_{\mathsf{ch}}$ except for the way that the challenger answers the adversary's queries. The challenger uses semi-functional keys of type 2 to answer the *first* $k-1$ queries made by the adversary. More precisely, for the i-th ($i \leq k-1$) query made by the adversary, if it is a key query on $\langle \mathsf{ID} \rangle$ then the challenger responds with a semi-functional key of type 2 for ID; if the i-th query is a decryption query on $\langle \mathsf{ID}, C\rangle$, the challenger first generates a semi-functional key of type 2 for ID, then calls $\mathsf{Decrypt}_{ex}$ with the semi-functional key of type 2 to decrypt C for the adversary.

To answer the k-th query made by the adversary, the challenger uses a semi-functional key of type 1 to respond. The challenger uses normal keys to answer the *remaining* queries made by the adversary.

$\mathsf{Game}_{k,2}(0 \le k \le q)$ This game is like $\mathsf{Game}_{\mathsf{ch}}$ except that the challenger uses a semi-functional key of type 2 to respond to the k-th query made by the adversary. Consequently, the challenger uses semi-functional keys of type 2 to answer the *first* k queries, and uses normal keys to answer the *remaining* queries made by the adversary.

$\mathsf{Game}_{\mathsf{Final}_0}$ This is like $\mathsf{Game}_{q,2}$ except that the challenge ciphertext $C^* = (c_0^*, c_1^*)$ for the challenge identity ID^* when $\delta = 1$ is given by

$$c_0^* = W_{14}^s g_4^{t_4} g_2^{t_2},\ c_1^* = (u^{\mathsf{ID}^*} v^{\mathsf{ID}^{*\prime}} h)^s g_4^{\mathsf{KDF}(X')} g_2^{t_2 t_2'},$$

where $s, t_4, t_2, t_2' \leftarrow \mathbb{Z}_N$, $X' \leftarrow \mathbb{G}_T$ and $\mathsf{ID}^{*\prime} = \mathsf{H}(\mathsf{ID}^*, c_0^*)$.

$\mathsf{Game}_{\mathsf{Final}_1}$ The game is the same as $\mathsf{Game}_{\mathsf{Final}_0}$ except that the challenge ciphertext $C^* = (c_0^*, c_1^*)$ under the challenge identity ID^* for $\delta = 1$ is determined by

$$c_0^* = W_{14}^s g_4^{t_4} g_2^{t_2},\ c_1^* = (u^{\mathsf{ID}^*} v^{\mathsf{ID}^{*\prime}} h)^s g_4^{t_4'} g_2^{t_2 t_2'},$$

where $s, t_4, t_4', t_2, t_2' \leftarrow \mathbb{Z}_N$ and $\mathsf{ID}^{*\prime} = \mathsf{H}(\mathsf{ID}^*, c_0^*)$.

$\mathsf{Game}_{\mathsf{Final}_2}$ This game is the same as $\mathsf{Game}_{\mathsf{Final}_1}$ except that the challenge session key K^* for $\delta = 1$ is given by $K^* \leftarrow \mathbb{G}_T$.

$\mathsf{Game}_{\mathsf{Final}_3}$ This game is the same as $\mathsf{Game}_{\mathsf{Final}_2}$ except that the challenge ciphertext $C^* = (c_0^*, c_1^*)$ for $\delta = 1$ is given by $c_0^*, c_1^* \leftarrow \mathbb{G}_{p_1 p_2 p_4}$.

$\mathsf{Game}_{\mathsf{Final}_4}$ This game is the same as $\mathsf{Game}_{\mathsf{Final}_3}$ except that the challenge ciphertext $C^* = (c_0^*, c_1^*)$ for $\delta = 1$ is given by $c_0^*, c_1^* \leftarrow \mathbb{G}$.

We prove these games are indistinguishable in the following lemmas. Note that $\mathsf{Game}_{\mathsf{ch}} = \mathsf{Game}_{0,2}$. In $\mathsf{Game}_{\mathsf{Final}_4}$, it is clear that the value of δ is information-theoretically hidden from the adversary. Hence the adversary has no advantage in $\mathsf{Game}_{\mathsf{Final}_4}$. Therefore, we conclude that the advantage of the adversary in $\mathsf{Game}_{\mathsf{Real}}$ is negligible. □

Lemma 2 *Suppose that $\mathcal{G}$ satisfies Assumption 1. Then $\mathsf{Game}_{\mathsf{Restricted}_1}$ and $\mathsf{Game}_{\mathsf{Real}}$ are computationally indistinguishable.*

Proof. Define event E_1: the adversary makes a key query of $\langle \mathsf{ID}\rangle$ such that $\mathsf{ID} \not\equiv \mathsf{ID}^* \mod N$ and $\mathsf{ID} \equiv \mathsf{ID}^* \mod p_2$.

If E_1 does not happen, $\mathsf{Game}_{\mathsf{Restricted}_1}$ is identical to $\mathsf{Game}_{\mathsf{Real}}$. All we have to do is to prove that E_1 happens with negligible probability.

If E_1 happens with non-negligible probability, we construct a PPT algorithm $\mathcal{B}$ that breaks Assumption 1 with non-negligible probability. Observe that, given $\mathbb{G}, \mathbb{G}_T, e, N, g, g_3, g_4, X_1X_2, Y_2Y_3, Z_2Z_4, T$, algorithm $\mathcal{B}$ can perfectly simulate $\mathsf{Game}_{\mathsf{Real}}$. During the simulation, for each key query $\langle \mathsf{ID}\rangle$, $\mathcal{B}$ computes $a = \gcd(\mathsf{ID}-\mathsf{ID}^*, N)$. $\mathcal{B}$ identifies the occurrence of E_1 with $e(X_1X_2, Y_2Y_3)^a = 1_{\mathbb{G}_T}$. Set $b = \frac{N}{a}$. There are three cases: 1. p_1 divides b; 2. p_3 divides b; 3. p_4 divides b.

$\mathcal{B}$ can determine if case 1 has occurred by testing if $e(X_1X_2, g)^b = 1_{\mathbb{G}_T}$. If this happens, $\mathcal{B}$ can then learn whether T has a $\mathbb{G}_{p_2}$ component or not by testing if $e(T, X_1X_2)^b = 1_{\mathbb{G}_T}$. If not, then T has a $\mathbb{G}_{p_2}$ component, i.e., $T \in \mathbb{G}_{p_1p_2p_3p_4}$; otherwise, $T \in \mathbb{G}_{p_1p_3p_4}$.

$\mathcal{B}$ can determine if case 2 has occurred by testing if $e(Y_2Y_3, g_3)^b = 1_{\mathbb{G}_T}$. If this happens, $\mathcal{B}$ can then learn whether T has a $\mathbb{G}_{p_2}$ component or not by testing if $e(T, Y_2Y_3)^b = 1_{\mathbb{G}_T}$. If not, then T has a $\mathbb{G}_{p_2}$ component, i.e., $T \in \mathbb{G}_{p_1p_2p_3p_4}$; otherwise, $T \in \mathbb{G}_{p_1p_3p_4}$.

$\mathcal{B}$ can determine if case 3 has occurred by testing if $e(Z_2Z_4, g_4)^b = 1_{\mathbb{G}_T}$. If this happens, $\mathcal{B}$ can then learn whether T has a $\mathbb{G}_{p_2}$ component or not by testing if $e(T, Z_2Z_4)^b = 1_{\mathbb{G}_T}$. If not, then T has a $\mathbb{G}_{p_2}$ component, i.e., $T \in \mathbb{G}_{p_1p_2p_3p_4}$; otherwise, $T \in \mathbb{G}_{p_1p_3p_4}$. □

Lemma 3 *Suppose that $\mathcal{G}$ satisfies CDH Assumption in $\mathbb{G}_{p_1}$, Assumption 1 and H is a collision-resistant hash function. Then $\mathsf{Game}_{\mathsf{Restricted}_2}$ and $\mathsf{Game}_{\mathsf{Restricted}_1}$ are computationally indistinguishable.*

Proof. Let $(C^* = (c_0^*, c_1^*), K^*)$ be the challenge ciphertext and session key. Recall that, C^* is the ciphertext encrypting δ under a challenge identity ID^* with $\delta \leftarrow \{0, 1\}$. We observe that $\mathsf{Game}_{\mathsf{Restricted}_1}$ and $\mathsf{Game}_{\mathsf{Restricted}_2}$ are the same unless the following events happen:

– Event E_2: the adversary makes a decryption query $\langle \mathsf{ID}, C\rangle$ such that $\mathsf{ID} = \mathsf{ID}^*, C = (c_0, c_1)$, $c_0 = c_0^*$ and $c_1 \neq c_1^*$, but the challenger responds with $(1, K)$ in $\mathsf{Game}_{\mathsf{Restricted}_1}$. Recall that in $\mathsf{Game}_{\mathsf{Restricted}_2}$, the challenger returns message 0 and a random session key for such a query, while in $\mathsf{Game}_{\mathsf{Restricted}_1}$, the challenge will employ decryption algorithm to answer the query. We will show that E_2 occurs with negligible probability in $\mathsf{Game}_{\mathsf{Restricted}_1}$.

  In $\mathsf{Game}_{\mathsf{Restricted}_1}$, if $\delta = 1$, $\mathsf{Decrypt}_{ex}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, C)$ always outputs bit 0 and a random session key, and E_2 never occurs in this case. Therefore, if E_2 happens, we must have $\delta = 0$ in $\mathsf{Game}_{\mathsf{Restricted}_1}$.

  We will construct a PPT algorithm $\mathcal{B}$ to solve the CDH problem over $\mathbb{G}_{p_1}$, if E_2 happens with non-negligible probability. $\mathcal{B}$ is given $(\mathbb{G}, \mathbb{G}_T, e, p_1, p_2, p_3, p_4,\ g, g_2, g_3, g_4, g^x, g^y)$ and going to compute g^{xy}. $\mathcal{B}$ simulates $\mathsf{Game}_{\mathsf{Restricted}_1}$ to the adversary as follows. Set $N = p_1p_2p_3p_4$ and choose $\alpha, \beta, \eta_u, \eta_v, \eta_h, \gamma_u, \gamma_v, \gamma_h,\ \gamma_w \in \mathbb{Z}_N$ uniformly at random, a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. $\mathcal{B}$ then computes $u = (g^x)^{\eta_u} g^{\gamma_u}, v = (g^x)^{\eta_v} g^{\gamma_v}, h = (g^x)^{\eta_h} g^{\gamma_h}, W_4 = g_4^{\gamma_w}$, and sends the adversary the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N),\ u, v, h, W_{14} = gW_4,\ g_4,\ e(g, g)^{\alpha},\ e(g, g)^{\beta},\ \mathsf{H}, \mathsf{KDF}).$$

  Note that $\mathcal{B}$ knows the master secret key $\mathsf{MSK} = (g, g_3, \alpha, \beta)$ associated with PK, thus is able to answer the key generation and decryption queries made by the adversary with the help of MSK. When the adversary submits the target identity ID^*, $\mathcal{B}$ sends $(C^* = (c_0^*, c_1^*), K^*)$ to the adversary, where

$$c_0^* = g^y \cdot g_2^{t_2} g_3^{t_3} g_4^{t_4},\ c_1^* \leftarrow \mathbb{G},\ K^* \leftarrow \mathbb{G}_T,$$

  and $t_2, t_3, t_4 \in \mathbb{Z}_N$ are chosen uniformly at random. Since $\delta = 0$, then from the adversary's point of view, the distribution of (C^*, K^*) is identical to that in $\mathsf{Game}_{\mathsf{Restricted}_1}$.

  Suppose E_2 happens during the simulation, i.e., the adversary makes a decryption query for $\langle \mathsf{ID} = \mathsf{ID}^*, C = (c_0 = c_0^*, c_1 \neq c_1^*)\rangle$ and $\mathcal{B}$ gets $(1, K)$ when decrypting C with the private key $\mathsf{SK}_{\mathsf{ID}^*}$. Let $\mathsf{SK}_{\mathsf{ID}^*} = (\mathsf{ID}^*, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$, then

$$e(c_1/g_4^{\mathsf{KDF}(X)}, W_{14}) = e(c_0^*,\ u^{\mathsf{ID}^*} v^{\mathsf{ID}^{*\prime}} h),$$

where $X = e(D_0 D_1^{\mathsf{ID}^{*\prime}}, c_0^*)/e(D_2, c_1)$ and $\mathsf{ID}^{*\prime} = \mathsf{H}(\mathsf{ID}^*, c_0^*)$. Observe that,

$$\begin{aligned} e(c_0^*,\ u^{\mathsf{ID}^*} v^{\mathsf{ID}^{*\prime}} h) &= e(g^y \cdot g_2^{t_2} g_3^{t_3} g_4^{t_4}, (g^x)^{\eta_u \mathsf{ID}^* + \eta_v \mathsf{ID}^{*\prime} + \eta_h} g^{\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h}) \\ &= e((g^{xy})^{\eta_u \mathsf{ID}^* + \eta_v \mathsf{ID}^{*\prime} + \eta_h} (g^y)^{\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h}, g). \end{aligned}$$

Hence, the $\mathbb{G}_{p_1}$ part of c_1 is $(g^{xy})^{\eta_u \mathsf{ID}^* + \eta_v \mathsf{ID}^{*\prime} + \eta_h} (g^y)^{\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h}$. $\mathcal{B}$ uses p_1, p_2, p_3, p_4 to extract g^{xy} from c_1,

$$g^{xy} = \left(\left(c_1 / (g^y)^{\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h} \right)^{p_2 p_3 p_4} \right)^{(p_2 p_3 p_4 (\eta_u \mathsf{ID}^* + \eta_v \mathsf{ID}^{*\prime} + \eta_h))^{-1} \mod p_1},$$

which is a solution to the CDH problem with respect to $(\mathbb{G}, \mathbb{G}_T, e, p_1, p_2, p_3, p_4, g, g_2, g_3, g_4, g^x, g^y)$. (Note that, since η_u, η_v, η_h are chosen uniformly at random in $\mathbb{Z}_N$ and are hidden by blinding factors $\gamma_u, \gamma_v, \gamma_h$, then with overwhelming probability, $\eta_u \mathsf{ID}^* + \eta_v \mathsf{ID}^{*\prime} + \eta_h$ is not equal to 0.)

To sum up, event E_2 happens with negligible probability if the CDH Assumption holds.

- Event E_3: the adversary makes a decryption query for $\langle \mathsf{ID}, C \rangle$ such that $\mathsf{ID} \not\equiv \mathsf{ID}^* \mod N$ and $\mathsf{ID} \equiv \mathsf{ID}^* \mod p_2$. Like the proof of Lemma 2, we can show that this event happens with negligible probability based on Assumption 1.
- Event E_4: the adversary makes a decryption query for $\langle \mathsf{ID}, C = (c_0, c_1) \rangle$ such that $\mathsf{ID} = \mathsf{ID}^*$, $\mathsf{H}(\mathsf{ID}, c_0) = \mathsf{H}(\mathsf{ID}^*, c_0^*)$ and $c_0 \neq c_0^*$. Clearly, this event happens with negligible probability, due to collision-resistance of H.
- Event E_5: the adversary makes a decryption query for $\langle \mathsf{ID}, C = (c_0, c_1) \rangle$ such that $\mathsf{ID} = \mathsf{ID}^*$, $\mathsf{H}(\mathsf{ID}, c_0) \not\equiv \mathsf{H}(\mathsf{ID}^*, c_0^*) \mod N$ and $\mathsf{H}(\mathsf{ID}, c_0) \equiv \mathsf{H}(\mathsf{ID}^*, c_0^*) \mod p_2$. Like the proof of Lemma 2, we can show that this event happens with negligible probability based on Assumption 1.

Thus, suppose that $\mathcal{G}$ satisfies CDH Assumption in $\mathbb{G}_{p_1}$, Assumption 1 and H is a collision-resistant hash function, $\mathsf{Game}_{\mathsf{Restricted}_1}$ and $\mathsf{Game}_{\mathsf{Restricted}_2}$ are computationally indistinguishable. □

Lemma 4 *Suppose that $\mathcal{G}$ satisfies Assumption 2. Then $\mathsf{Game}_{\mathsf{ch}}$ and $\mathsf{Game}_{\mathsf{Restricted}_2}$ are computationally indistinguishable.*

Proof. Suppose there exists a PPT algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{\mathsf{ch}}$ and $\mathsf{Game}_{\mathsf{Restricted}_2}$ with non-negligible advantage. Then we build a PPT algorithm $\mathcal{B}$ breaking Assumption 2 with non-negligible advantage. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N,\ g, g_3, g_4, T$ and going to tell whether $T \in \mathbb{G}_{p_1 p_2}$ or $T \in \mathbb{G}_{p_1}$. $\mathcal{B}$ will simulate $\mathsf{Game}_{\mathsf{ch}}$ or $\mathsf{Game}_{\mathsf{Restricted}_2}$ for $\mathcal{A}$. First $\mathcal{B}$ chooses $\alpha, \beta, \gamma_u, \gamma_v, \gamma_h,\ \gamma_w \in \mathbb{Z}_N$ uniformly at random, a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. It then sets $u = g^{\gamma_u}, v = g^{\gamma_v}, h = g^{\gamma_h}, W_4 = g_4^{\gamma_w}$, and sends $\mathcal{A}$ the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N),\ u, v, h, W_{14} = g W_4,\ g_4,\ e(g,g)^{\alpha},\ e(g,g)^{\beta},\ \mathsf{H}, \mathsf{KDF}).$$

Note that $\mathcal{B}$ knows the master secret key $\mathsf{MSK} = (g, g_3, \alpha, \beta)$ associated with PK, and $\mathcal{B}$ can answers the key and decryption queries of $\mathcal{A}$ with the help of MSK.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ a challenge identity ID^*. $\mathcal{B}$ chooses $\delta \leftarrow \{0,1\}$ and does the following. If $\delta = 0$, it chooses $c_0^*, c_1^* \leftarrow \mathbb{G}$ and $K^* \leftarrow \mathbb{G}_T$; otherwise, it chooses $t_4 \leftarrow \mathbb{Z}_N$ and sets

$$c_0^* = T \cdot g_4^{t_4},\ c_1^* = T^{\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h} g_4^{\mathsf{KDF}(e(T, g^{\alpha}))},\ K^* = e(T, g^{\beta}),$$

where $\mathsf{ID}^{*\prime} = \mathsf{H}(\mathsf{ID}^*, c_0^*)$. Finally, $\mathcal{B}$ sends the challenge ciphertext $C^* = (c_0^*, c_1^*)$ and session key K^* to $\mathcal{A}$.

If T is a random element of $\mathbb{G}_{p_1p_2}$, then $(\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h) \mod p_2$ is uniformly distributed over $\mathbb{Z}_{p_2}$ according to Chinese Remainder Theorem, and $C^* = (c_0^*, c_1^*)$ has the same distribution as semi-functional ciphertexts. Hence $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{ch}}$. If T is a random element of $\mathbb{G}_{p_1}$, then $C^* = (c_0^*, c_1^*)$ has the same distribution as normal ciphertexts. Hence $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{Restricted}_2}$. Consequently, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish $T \in \mathbb{G}_{p_1p_2}$ or $T \in \mathbb{G}_{p_1}$. Any non-negligible advantage of $\mathcal{A}$ is converted to a non-negligible advantage of $\mathcal{B}$. $\square$

Lemma 5 *Suppose that $\mathcal{G}$ satisfies Assumption 3. Then for each $k \in [q]$, $\mathsf{Game}_{k-1,2}$ and $\mathsf{Game}_{k,1}$ are computationally indistinguishable.*

Proof. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{k-1,2}$ and $\mathsf{Game}_{k,1}$. Then we can build an algorithm $\mathcal{B}$ breaking Assumption 3 with non-negligible advantage. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N, g, X_1X_2, Y_2Y_3, g_3, g_4, T$ and going to tell $T \in \mathbb{G}_{p_1p_2p_3}$ or $T \in \mathbb{G}_{p_1p_3}$. $\mathcal{B}$ will simulate $\mathsf{Game}_{k-1,2}$ or $\mathsf{Game}_{k,1}$ for $\mathcal{A}$. First $\mathcal{B}$ chooses $\alpha, \beta, \gamma_u, \gamma_v, \gamma_h, \gamma_w \in \mathbb{Z}_N$ uniformly at random, a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. It then sets $u = g^{\gamma_u}, v = g^{\gamma_v}, h = g^{\gamma_h}, W_4 = g_4^{\gamma_w}$, and sends $\mathcal{A}$ the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N),\ u, v, h, W_{14} = gW_4,\ g_4,\ e(g,g)^{\alpha},\ e(g,g)^{\beta},\ \mathsf{H}, \mathsf{KDF}).$$

Let us now explain how $\mathcal{B}$ answers the i-th query made by $\mathcal{A}$, which is a key query for $\langle \mathsf{ID} \rangle$ or a decryption query for $\langle \mathsf{ID}, C \rangle$. (Notice that, if the cases described in $\mathsf{Game}_{\mathsf{Restricted}_1}$ or $\mathsf{Game}_{\mathsf{Restricted}_2}$ happen when $\mathcal{A}$ makes a key or decryption query, $\mathcal{B}$ responds as in $\mathsf{Game}_{\mathsf{Restricted}_1}$ or $\mathsf{Game}_{\mathsf{Restricted}_2}$.)

1. If $i < k$, $\mathcal{B}$ first chooses $r, \bar{r}, r_3, r_3', r_3'', r_2, r_2', r_2'', \bar{r}_3, \bar{r}_3', \bar{r}_3'', \bar{r}_2, \bar{r}_2', \bar{r}_2'' \in \mathbb{Z}_N$ uniformly at random, and generates a semi-functional key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$ of type 2 for ID, where

$$D_0 = g^{\alpha}(u^{\mathsf{ID}}h)^r g_3^{r_3}(Y_2Y_3)^{r_2},\ D_1 = v^r g_3^{r_3'}(Y_2Y_3)^{r_2'},\ D_2 = g^r g_3^{r_3''}(Y_2Y_3)^{r_2''},$$
$$\bar{D}_0 = g^{\beta}(u^{\mathsf{ID}}h)^{\bar{r}} g_3^{\bar{r}_3}(Y_2Y_3)^{\bar{r}_2},\ \bar{D}_1 = v^{\bar{r}} g_3^{\bar{r}_3'}(Y_2Y_3)^{\bar{r}_2'},\ \bar{D}_2 = g^{\bar{r}} g_3^{\bar{r}_3''}(Y_2Y_3)^{\bar{r}_2''}.$$

   Then, $\mathcal{B}$ uses $\mathsf{SK}_{\mathsf{ID}}$ to respond $\mathcal{A}$'s query. That is, if the query is a key query, $\mathcal{B}$ sends $\mathsf{SK}_{\mathsf{ID}}$ to $\mathcal{A}$; otherwise (i.e., the query is a decryption query), $\mathcal{B}$ runs the algorithm $\mathsf{Decrypt}_{ex}$ with $\mathsf{SK}_{\mathsf{ID}}$ and returns the decryption results to $\mathcal{A}$.
   Note that $r_3, r_3', r_3'', \bar{r}_3, \bar{r}_3', \bar{r}_3'' \mod p_3$ are all randomly distributed in $\mathbb{Z}_{p_3}$ and $r_2, r_2', r_2'', \bar{r}_2, \bar{r}_2', \bar{r}_2''$ mod p_2 are all randomly distributed in $\mathbb{Z}_{p_2}$, according to Chinese Remainder Theorem. Thus, the private key $\mathsf{SK}_{\mathsf{ID}}$ is a properly distributed semi-functional key of type 2.
2. Else if $i > k$, $\mathcal{B}$ first generate a normal key $\mathsf{SK}_{\mathsf{ID}}$ for ID with the master secret key $\mathsf{MSK} = (g, g_3, \alpha, \beta)$. Then, $\mathcal{B}$ uses $\mathsf{SK}_{\mathsf{ID}}$ to respond $\mathcal{A}$'s query.
3. Else (i.e., $i = k$), $\mathcal{B}$ first chooses $\bar{r}, r_3, r_3', r_3'', \bar{r}_3, \bar{r}_3', \bar{r}_3'' \in \mathbb{Z}_N$ uniformly at random and sets $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$, where

$$D_0 = g^{\alpha} T^{\gamma_u \mathsf{ID} + \gamma_h} g_3^{r_3},\ D_1 = T^{\gamma_v} g_3^{r_3'},\ D_2 = T g_3^{r_3''},$$
$$\bar{D}_0 = g^{\beta}(u^{\mathsf{ID}}h)^{\bar{r}} g_3^{\bar{r}_3},\ \bar{D}_1 = v^{\bar{r}} g_3^{\bar{r}_3'},\ \bar{D}_2 = g^{\bar{r}} g_3^{\bar{r}_3''}.$$

   Then, $\mathcal{B}$ uses $\mathsf{SK}_{\mathsf{ID}}$ to respond to $\mathcal{A}$'s query.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ a target identity ID^*. $\mathcal{B}$ chooses $\delta \leftarrow \{0,1\}$ and does the following. If $\delta = 0$, it chooses $c_0^*, c_1^* \leftarrow \mathbb{G}$ and $K^* \leftarrow \mathbb{G}_T$; otherwise (i.e., $\delta = 1$), it chooses $t_4 \leftarrow \mathbb{Z}_N$ and sets

$$c_0^* = X_1X_2 \cdot g_4^{t_4},\ c_1^* = (X_1X_2)^{\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h} g_4^{\mathsf{KDF}(e(X_1X_2, g^\alpha))},\ K^* = e(X_1X_2, g^\beta),$$

where $\mathsf{ID}^{*\prime} = \mathsf{H}(\mathsf{ID}^*, c_0^*)$. Finally, $\mathcal{B}$ sends the challenge ciphertext $C^* = (c_0^*, c_1^*)$ and session key K^* to $\mathcal{A}$.

Next, we will show that the challenge ciphertext and $\mathsf{SK}_{\mathsf{ID}}$ appearing in the response of $\mathcal{A}$'s k-th query are properly distributed. The key point is to show that $\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h \mod p_2$ is randomly distributed in $\mathbb{Z}_{p_2}$.

1. If $T \leftarrow \mathbb{G}_{p_1p_3}$, say $T = g^r g_3^{\hat{r}_3}$, then the first three components of $\mathsf{SK}_{\mathsf{ID}}$ appearing in the response of $\mathcal{A}$'s k-th query are

$$D_0 = g^\alpha T^{\gamma_u \mathsf{ID} + \gamma_h} g_3^{r_3} = g^\alpha (u^{\mathsf{ID}} h)^r g_3^{\hat{r}_3(\gamma_u \mathsf{ID} + \gamma_h) + r_3},$$
$$D_1 = T^{\gamma_v} g_3^{r_3'} = v^r g_3^{\hat{r}_3 \gamma_v + r_3'},\ D_2 = T g_3^{r_3''} = g^r g_3^{\hat{r}_3 + r_3''}.$$

   Their distribution is exactly the same as that in the normal key.
   On the other hand, $\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h \mod p_2$ is also randomly distributed in $\mathbb{Z}_{p_2}$ according to Chinese Remainder Theorem, so the challenge ciphertext is distributed just like a semi-functional ciphertext.
   Therefore, $\mathcal{B}$ has perfectly simulated $\mathsf{Game}_{k-1,2}$.
2. If $T \leftarrow \mathbb{G}_{p_1p_2p_3}$, say $T = g^r g_2^{\hat{r}_2} g_3^{\hat{r}_3}$, we consider two cases.
   (a) The k-th query made by the adversary is a key query on $\langle \mathsf{ID} \rangle$. Since event E_1 that $\mathsf{ID} \equiv \mathsf{ID}^* \mod p_2$ has already been eliminated in $\mathsf{Game}_{\mathsf{Restricted}_1}$, then $\mathsf{ID} \not\equiv \mathsf{ID}^* \mod p_2$. Observe that, the first three components of $\mathsf{SK}_{\mathsf{ID}}$ are

$$D_0 = g^\alpha T^{\gamma_u \mathsf{ID} + \gamma_h} g_3^{r_3} = g^\alpha (u^{\mathsf{ID}} h)^r g_2^{\hat{r}_2(\gamma_u \mathsf{ID} + \gamma_h)} g_3^{\hat{r}_3(\gamma_u \mathsf{ID} + \gamma_h) + r_3},$$
$$D_1 = T^{\gamma_v} g_3^{r_3'} = v^r g_2^{\hat{r}_2 \gamma_v} g_3^{\hat{r}_3 \gamma_v + r_3'},\ D_2 = T g_3^{r_3''} = g^r g_2^{\hat{r}_2} g_3^{\hat{r}_3 + r_3''}.$$

   We want to prove that the private key $\mathsf{SK}_{\mathsf{ID}}$ is a properly distributed semi-functional key of type 1 and the challenge ciphertext is a properly distributed semi-functional ciphertext. It suffices to prove that $(\gamma_u \mathsf{ID} + \gamma_h)_{p_2}$, $(\gamma_v)_{p_2}$ and $(\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h)_{p_2}$ are all uniformly distributed over $\mathbb{Z}_{p_2}$, where $(x)_{p_2}$ denotes $x \mod p_2$. This is justified by the following facts.
   - Conditioned on $u = g^{\gamma_u}, v = g^{\gamma_v}, h = g^{\gamma_h}$, the random variables $(\gamma_u)_{p_2}, (\gamma_v)_{p_2}$ and $(\gamma_h)_{p_2}$ are uniformly distributed over $\mathbb{Z}_{p_2}$, due to Chinese Remainder Theorem.
   - The 3 by 3 matrix on the right side has full rank as long as $\mathsf{ID} \not\equiv \mathsf{ID}^* \mod p_2$.

$$\begin{pmatrix} (\gamma_u \mathsf{ID} + \gamma_h)_{p_2} \\ (\gamma_v)_{p_2} \\ (\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h)_{p_2} \end{pmatrix} = \begin{pmatrix} (\mathsf{ID})_{p_2} & 0 & 1 \\ 0 & 1 & 0 \\ (\mathsf{ID}^*)_{p_2} & (\mathsf{ID}^{*\prime})_{p_2} & 1 \end{pmatrix} \cdot \begin{pmatrix} (\gamma_u)_{p_2} \\ (\gamma_v)_{p_2} \\ (\gamma_h)_{p_2} \end{pmatrix}.$$

   (b) The k-th query is a decryption query of $\langle \mathsf{ID}, C = (c_0, c_1) \rangle$.
   - If $\mathsf{ID} = \mathsf{ID}^*$ and $c_0 = c_0^*$, $\mathcal{B}$ returns bit 0 and a random session key without using the private key $\mathsf{SK}_{\mathsf{ID}}$. This is consistent to the correct answer from decryption, except with negligible probability (Recall that we have proved in Lemma 3 that event E_2 happens with negligible probability). Hence, no information about $\mathsf{SK}_{\mathsf{ID}}$ is leaked and $(\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h)_{p_2}$ is uniformly distributed over $\mathbb{Z}_{p_2}$ according to Chinese Remainder Theorem.

– Else, let $\mathsf{ID}' = \mathsf{H}(\mathsf{ID}, c_0)$, $\mathcal{B}$ uses

$$(D_0 D_1^{\mathsf{ID}'}, D_2) = (g^{\alpha}(u^{\mathsf{ID}} v^{\mathsf{ID}'} h)^r g_2^{\hat{r}_2(\gamma_u \mathsf{ID}+\gamma_v \mathsf{ID}'+\gamma_h)} g_3^{\hat{r}_3(\gamma_u \mathsf{ID}+\gamma_v \mathsf{ID}'+\gamma_h)+r_3+r_3' \mathsf{ID}'}, g^r g_2^{\hat{r}_2} g_3^{\hat{r}_3+r_3''})$$

$$(\bar{D}_0 \bar{D}_1^{\mathsf{ID}'}, \bar{D}_2) = (g^{\alpha}(u^{\mathsf{ID}} v^{\mathsf{ID}'} h)^{\bar{r}} g_3^{\bar{r}_3+\bar{r}_3' \mathsf{ID}'}, g^{\bar{r}} g_3^{\bar{r}_3''})$$

to answer this decryption query according to Eq.(4), Eq.(5), Eq.(6).
Since event E_3, E_4, E_5 have been eliminated in $\mathsf{Game}_{\mathsf{Restricted}_2}$, then $(\mathsf{ID}, \mathsf{ID}') \not\equiv (\mathsf{ID}^*, \mathsf{ID}^{*\prime})$ mod p_2. Observe that,

$$\begin{pmatrix} (\gamma_u \mathsf{ID} + \gamma_v \mathsf{ID}' + \gamma_h)_{p_2} \\ (\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h)_{p_2} \end{pmatrix} = \begin{pmatrix} (\mathsf{ID})_{p_2} & (\mathsf{ID}')_{p_2} & 1 \\ (\mathsf{ID}^*)_{p_2} & (\mathsf{ID}^{*\prime})_{p_2} & 1 \end{pmatrix} \cdot \begin{pmatrix} (\gamma_u)_{p_2} \\ (\gamma_v)_{p_2} \\ (\gamma_h)_{p_2} \end{pmatrix}.$$

The 2 by 3 matrix on the right side of the above equation has rank 2 as long as $(\mathsf{ID}, \mathsf{ID}') \not\equiv (\mathsf{ID}^*, \mathsf{ID}^{*\prime}) \mod p_2$, so $(\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h)_{p_2}$ uniformly distributed over $\mathbb{Z}_{p_2}$, and the challenge ciphertext has the same distribution as the semi-functional ciphertext.

In both cases, $\mathcal{B}$ properly simulated $\mathsf{Game}_{k,1}$.

Hence, if T is a random element of $\mathbb{G}_{p_1 p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k-1,2}$. If T is a random element of $\mathbb{G}_{p_1 p_2 p_3}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{k,1}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish $T \in \mathbb{G}_{p_1 p_3}$ or $\mathbb{G}_{p_1 p_2 p_3}$. Any non-negligible advantage of $\mathcal{A}$ is converted to a non-negligible advantage of $\mathcal{B}$. □

Lemma 6 *Suppose that $\mathcal{G}$ satisfies Assumption 3. Then for each $k \in [q]$, $\mathsf{Game}_{k,1}$ and $\mathsf{Game}_{k,2}$ are computationally indistinguishable.*

Proof. This proof is very similar to the proof of the previous lemma. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{k,1}$ and $\mathsf{Game}_{k,2}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 3. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N, g, X_1X_2, Y_2Y_3, g_3, g_4, T$ and will simulate $\mathsf{Game}_{k,1}$ or $\mathsf{Game}_{k,2}$ with $\mathcal{A}$. First $\mathcal{B}$ chooses $\alpha, \beta, \gamma_u, \gamma_v, \gamma_h, \gamma_w \in \mathbb{Z}_N$ uniformly at random, a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. It then sets $u = g^{\gamma_u}, v = g^{\gamma_v}, h = g^{\gamma_h}, W_4 = g_4^{\gamma_w}$, and sends $\mathcal{A}$ the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N),\ u, v, h, W_{14} = gW_4,\ g_4,\ e(g,g)^{\alpha},\ e(g,g)^{\beta}, \mathsf{H}, \mathsf{KDF}).$$

Let us now explain how $\mathcal{B}$ answers the i-th query made by $\mathcal{A}$, which is a key query for $\langle \mathsf{ID} \rangle$ or a decryption query for $\langle \mathsf{ID}, C \rangle$.

1. If $i \neq k$, $\mathcal{B}$ responds as in the previous lemma.
2. Else (i.e., $i = k$), $\mathcal{B}$ first chooses $r, r_3, r_3', r_3'', r_2, r_2', r_2'', \bar{r}_3, \bar{r}_3', \bar{r}_3'' \in \mathbb{Z}_N$ uniformly at random and sets $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$, where

$$D_0 = g^{\alpha}(u^{\mathsf{ID}} h)^r g_3^{r_3} (Y_2Y_3)^{r_2},\ D_1 = v^r g_3^{r_3'} (Y_2Y_3)^{r_2'},\ D_2 = g^r g_3^{r_3''} (Y_2Y_3)^{r_2''},$$
$$\bar{D}_0 = g^{\beta} T^{\gamma_u \mathsf{ID}+\gamma_h} g_3^{\bar{r}_3},\ \bar{D}_1 = T^{\gamma_v} g_3^{\bar{r}_3'},\ \bar{D}_2 = T g_3^{\bar{r}_3''}.$$

Then, $\mathcal{B}$ uses $\mathsf{SK}_{\mathsf{ID}}$ to respond $\mathcal{A}$'s query.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ a challenge identity ID^*. $\mathcal{B}$ sends the challenge ciphertext $C^* = (c_0^*, c_1^*)$ and session key K^*, which are constructed exactly as in the previous lemma, to $\mathcal{A}$.

The rest of analysis is just like that in the previous lemma. $\square$

Lemma 7 *Suppose that $\mathcal{G}$ satisfies Assumption 4. Then $\mathsf{Game}_{\mathsf{Final}_0}$ and $\mathsf{Game}_{q,2}$ are computationally indistinguishable.*

Proof. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{\mathsf{Final}_0}$ and $\mathsf{Game}_{q,2}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 4. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N,\ g, g_2, g_3, g_4, g^a X_2,\ g^s Y_2, T$ and will simulate $\mathsf{Game}_{\mathsf{Final}_0}$ or $\mathsf{Game}_{q,2}$ with $\mathcal{A}$. First $\mathcal{B}$ chooses $\beta, \gamma_u, \gamma_v, \gamma_h, \gamma_w \in \mathbb{Z}_N$ uniformly at random, a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. It then sets $u = g^{\gamma_u}, v = g^{\gamma_v}, h = g^{\gamma_h}, W_4 = g_4^{\gamma_w}$, and sends $\mathcal{A}$ the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N),\ u, v, h, W_{14} = gW_4,\ g_4,\ e(g, g^a X_2) = e(g,g)^a,\ e(g,g)^\beta,\ \mathsf{H}, \mathsf{KDF}).$$

($\mathcal{B}$ sets $\alpha = a$ implicitly.) Now we show how $\mathcal{B}$ answers the query made by $\mathcal{A}$, which is a key query for $\langle \mathsf{ID} \rangle$ or a decryption query for $\langle \mathsf{ID}, C \rangle$. $\mathcal{B}$ first chooses $r, \bar{r}, r_3, r_3', r_3'', r_2, r_2', r_2'', \bar{r}_3, \bar{r}_3', \bar{r}_3'', \bar{r}_2, \bar{r}_2', \bar{r}_2'' \in \mathbb{Z}_N$ uniformly at random, and generates a semi-functional key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$ of type 2 for ID, where

$$\begin{aligned} D_0 &= g^a X_2 \cdot (u^{\mathsf{ID}} h)^r g_3^{r_3} g_2^{r_2},\ D_1 = v^r g_3^{r_3'} g_2^{r_2'},\ D_2 = g^r g_3^{r_3''} g_2^{r_2''}, \\ \bar{D}_0 &= g^\beta (u^{\mathsf{ID}} h)^{\bar{r}} g_3^{\bar{r}_3} g_2^{\bar{r}_2},\ \bar{D}_1 = v^{\bar{r}} g_3^{\bar{r}_3'} g_2^{\bar{r}_2'},\ \bar{D}_2 = g^{\bar{r}} g_3^{\bar{r}_3''} g_2^{\bar{r}_2''}. \end{aligned}$$

Then, $\mathcal{B}$ uses $\mathsf{SK}_{\mathsf{ID}}$ to respond $\mathcal{A}$'s query.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ a challenge identity ID^*. $\mathcal{B}$ chooses $\delta \leftarrow \{0,1\}$ and does the following. If $\delta = 0$, it chooses $c_0^*, c_1^* \leftarrow \mathbb{G}$ and $K^* \leftarrow \mathbb{G}_T$; otherwise (i.e., $\delta = 1$), it chooses $t_4, t_2, t_2' \leftarrow \mathbb{Z}_N$ and sets

$$c_0^* = g^s Y_2 \cdot g_4^{t_4} g_2^{t_2},\ c_1^* = (g^s Y_2)^{\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h} g_4^{\mathsf{KDF}(T)} g_2^{t_2'},\ K^* = e(g^\beta, g^s Y_2) = e(g,g)^{\beta s},$$

where $\mathsf{ID}^{*\prime} = \mathsf{H}(\mathsf{ID}^*, c_0^*)$. Finally, $\mathcal{B}$ sends the challenge ciphertext $C^* = (c_0^*, c_1^*)$ and session key K^* to $\mathcal{A}$.

It is clear that, if $T = e(g,g)^{as}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{q,2}$. If T is a random element of $\mathbb{G}_T$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{Final}_0}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two possibilities for T. $\square$

Lemma 8 *Suppose that KDF is a secure key derivation function. Then $\mathsf{Game}_{\mathsf{Final}_1}$ and $\mathsf{Game}_{\mathsf{Final}_0}$ are computationally indistinguishable.*

Proof. It is clear that the adversary distinguishes $\mathsf{Game}_{\mathsf{Final}_1}$ and $\mathsf{Game}_{\mathsf{Final}_0}$ with negligible probability, since in $\mathsf{Game}_{\mathsf{Final}_0}$, $X' \leftarrow \mathbb{G}_T$ and KDF is a secure key derivation function. $\square$

Lemma 9 *Suppose that $\mathcal{G}$ satisfies Assumption 4. Then $\mathsf{Game}_{\mathsf{Final}_2}$ and $\mathsf{Game}_{\mathsf{Final}_1}$ are computationally indistinguishable.*

Proof. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{\mathsf{Final}_2}$ and $\mathsf{Game}_{\mathsf{Final}_1}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 4. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N,\ g, g_2, g_3, g_4, g^a X_2,\ g^s Y_2, T$ and will simulate $\mathsf{Game}_{\mathsf{Final}_2}$ or $\mathsf{Game}_{\mathsf{Final}_1}$ with $\mathcal{A}$. First $\mathcal{B}$ chooses $\alpha, \gamma_u, \gamma_v, \gamma_h, \gamma_w \in \mathbb{Z}_N$ uniformly at random, a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. It then sets $u = g^{\gamma_u}, v = g^{\gamma_v}, h = g^{\gamma_h}, W_4 = g_4^{\gamma_w}$, and sends $\mathcal{A}$ the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N),\ u, v, h, W_{14} = gW_4,\ g_4,\ e(g,g)^\alpha,\ e(g^a X_2, g) = e(g,g)^a,\ \mathsf{H}, \mathsf{KDF}).$$

($\mathcal{B}$ sets $\beta = a$ implicitly.) Now we show how $\mathcal{B}$ answers the query made by $\mathcal{A}$, which is a key query for $\langle \mathsf{ID} \rangle$ or a decryption query for $\langle \mathsf{ID}, C \rangle$. $\mathcal{B}$ first chooses $r, \bar{r}, r_3, r_3', r_3'', r_2, r_2', r_2'', \bar{r}_3, \bar{r}_3', \bar{r}_3'', \bar{r}_2, \bar{r}_2', \bar{r}_2'' \in \mathbb{Z}_N$ uniformly at random, and generates a semi-functional key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$ of type 2 for ID, where

$$D_0 = g^\alpha (u^{\mathsf{ID}} h)^r g_3^{r_3} g_2^{r_2},\ D_1 = v^r g_3^{r_3'} g_2^{r_2'},\ D_2 = g^r g_3^{r_3''} g_2^{r_2''},$$
$$\bar{D}_0 = g^a X_2 \cdot (u^{\mathsf{ID}} h)^{\bar{r}} g_3^{\bar{r}_3} g_2^{\bar{r}_2},\ \bar{D}_1 = v^{\bar{r}} g_3^{\bar{r}_3'} g_2^{\bar{r}_2'},\ \bar{D}_2 = g^{\bar{r}} g_3^{\bar{r}_3''} g_2^{\bar{r}_2''}.$$

Then, $\mathcal{B}$ uses $\mathsf{SK}_{\mathsf{ID}}$ to respond $\mathcal{A}$'s query.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ a challenge identity ID^*. $\mathcal{B}$ chooses $\delta \leftarrow \{0,1\}$ and does the following. If $\delta = 0$, it chooses $c_0^*, c_1^* \leftarrow \mathbb{G}$ and $K^* \leftarrow \mathbb{G}_T$; otherwise (i.e., $\delta = 1$), it chooses $t_4, t_4', t_2, t_2' \leftarrow \mathbb{Z}_N$ and sets

$$c_0^* = g^s Y_2 \cdot g_4^{t_4} g_2^{t_2},\ c_1^* = (g^s Y_2)^{\gamma_u \mathsf{ID}^* + \gamma_v \mathsf{ID}^{*\prime} + \gamma_h} g_4^{t_4'} g_2^{t_2'},\ K^* = T,$$

where $\mathsf{ID}^{*\prime} = \mathsf{H}(\mathsf{ID}^*, c_0^*)$. Finally, $\mathcal{B}$ sends the challenge ciphertext $C^* = (c_0^*, c_1^*)$ and session key K^* to $\mathcal{A}$.

It is clear that, if $T = e(g,g)^{as}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{Final}_1}$. If T is a random element of $\mathbb{G}_T$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{Final}_2}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two possibilities for T. $\square$

Lemma 10 *Suppose that $\mathcal{G}$ satisfies Assumption 5. Then $\mathsf{Game}_{\mathsf{Final}_3}$ and $\mathsf{Game}_{\mathsf{Final}_2}$ are computationally indistinguishable.*

Proof. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game}_{\mathsf{Final}_3}$ and $\mathsf{Game}_{\mathsf{Final}_2}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 5. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N, gW_4, gA_2, u, u^s B_{24}, v, v^s X_{24}, h, h^s Y_{24}, g_2, g_3, g_4, T$ and will simulate $\mathsf{Game}_{\mathsf{Final}_3}$ or $\mathsf{Game}_{\mathsf{Final}_2}$ with $\mathcal{A}$. First $\mathcal{B}$ chooses $\alpha, \beta \in \mathbb{Z}_N$ uniformly at random, a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. It then sends $\mathcal{A}$ the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N), u, v, h, W_{14} = gW_4, g_4, e(gW_1, gA_2)^\alpha = e(g,g)^\alpha, e(gW_1, gA_2)^\beta = e(g,g)^\beta, \mathsf{H}, \mathsf{KDF}).$$

Now we show how $\mathcal{B}$ answers the query made by $\mathcal{A}$, which is a key query for $\langle \mathsf{ID} \rangle$ or a decryption query for $\langle \mathsf{ID}, C \rangle$. $\mathcal{B}$ first chooses $r, \bar{r}, r_3, r_3', r_3'', r_2, r_2', r_2'', \bar{r}_3, \bar{r}_3', \bar{r}_3'', \bar{r}_2, \bar{r}_2', \bar{r}_2'' \in \mathbb{Z}_N$ uniformly at random, and generates a semi-functional key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$ of type 2 for ID, where

$$D_0 = (gA_2)^\alpha \cdot (u^{\mathsf{ID}} h)^r g_3^{r_3} g_2^{r_2},\ D_1 = v^r g_3^{r_3'} g_2^{r_2'},\ D_2 = (gA_2)^r g_3^{r_3''} g_2^{r_2''},$$
$$\bar{D}_0 = (gA_2)^\beta \cdot (u^{\mathsf{ID}} h)^{\bar{r}} g_3^{\bar{r}_3} g_2^{\bar{r}_2},\ \bar{D}_1 = v^{\bar{r}} g_3^{\bar{r}_3'} g_2^{\bar{r}_2'},\ \bar{D}_2 = (gA_2)^{\bar{r}} g_3^{\bar{r}_3''} g_2^{\bar{r}_2''}.$$

Then, $\mathcal{B}$ uses $\mathsf{SK_{ID}}$ to respond $\mathcal{A}$'s query.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ a challenge identity ID^*. $\mathcal{B}$ first chooses $\delta \leftarrow \{0,1\}$ and a session key $K^* \leftarrow \mathbb{G}_T$. Then, if $\delta = 0$, $\mathcal{B}$ chooses $c_0^*, c_1^* \leftarrow \mathbb{G}$; otherwise (i.e., $\delta = 1$), it chooses $t_4, t_4', t_2, t_2' \leftarrow \mathbb{Z}_N$ and sets

$$c_0^* = T \cdot g_4^{t_4} g_2^{t_2},\ c_1^* = (u^s B_{24})^{\mathsf{ID}^*} (v^s X_{24})^{\mathsf{ID}^{*\prime}} (h^s Y_{24}) g_4^{t_4'} g_2^{t_2'},$$

where $\mathsf{ID}^{*\prime} = \mathsf{H}(\mathsf{ID}^*, c_0^*)$. Finally, $\mathcal{B}$ sends the challenge ciphertext $C^* = (c_0^*, c_1^*)$ and the session key K^* to $\mathcal{A}$.

It is clear that, if $T = g^s E_{24}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game_{Final_2}}$. If T is a random element of $\mathbb{G}_{p_1p_2p_4}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game_{Final_3}}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two possibilities for T. □

Lemma 11 *Suppose that $\mathcal{G}$ satisfies Assumption 6. Then $\mathsf{Game_{Final_4}}$ and $\mathsf{Game_{Final_3}}$ are computationally indistinguishable.*

Proof. Suppose there exists an algorithm $\mathcal{A}$ that distinguishes $\mathsf{Game_{Final_4}}$ and $\mathsf{Game_{Final_3}}$. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking Assumption 6. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, N,\ g, g_2, X_2X_3, g_4, T$ and will simulate $\mathsf{Game_{Final_4}}$ or $\mathsf{Game_{Final_3}}$ with $\mathcal{A}$. First $\mathcal{B}$ chooses $\alpha, \beta, \gamma_u, \gamma_v, \gamma_h,\ \gamma_w \in \mathbb{Z}_N$ uniformly at random, a collision-resistant hash function $\mathsf{H} : \mathbb{Z}_N \times \mathbb{G} \to \mathbb{Z}_N$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_N$. It then sets $u = g^{\gamma_u}, v = g^{\gamma_v}, h = g^{\gamma_h}, W_4 = g_4^{\gamma_w}$, and sends $\mathcal{A}$ the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, N),\ u, v, h, W_{14} = gW_4,\ g_4,\ e(g,g)^{\alpha},\ e(g,g)^{\beta},\ \mathsf{H}, \mathsf{KDF}).$$

Now we show how $\mathcal{B}$ answers the query made by $\mathcal{A}$, which is a key query for $\langle \mathsf{ID} \rangle$ or a decryption query for $\langle \mathsf{ID}, C \rangle$. $\mathcal{B}$ first chooses $r, \bar{r}, r_3, r_3', r_3'', r_2, r_2', r_2'', \bar{r}_3, \bar{r}_3', \bar{r}_3'', \bar{r}_2, \bar{r}_2', \bar{r}_2'' \in \mathbb{Z}_N$ uniformly at random, and generates a semi-functional key $\mathsf{SK_{ID}} = (\mathsf{ID}, D_0, D_1, D_2, \bar{D}_0, \bar{D}_1, \bar{D}_2)$ of type 2 for ID, where

$$D_0 = g^{\alpha}(u^{\mathsf{ID}}h)^r (X_2X_3)^{r_3} g_2^{r_2},\ D_1 = v^r (X_2X_3)^{r_3'} g_2^{r_2'},\ D_2 = g^r (X_2X_3)^{r_3''} g_2^{r_2''},$$
$$\bar{D}_0 = g^{\beta}(u^{\mathsf{ID}}h)^{\bar{r}} (X_2X_3)^{\bar{r}_3} g_2^{\bar{r}_2},\ \bar{D}_1 = v^{\bar{r}} (X_2X_3)^{\bar{r}_3'} g_2^{\bar{r}_2'},\ \bar{D}_2 = g^{\bar{r}} (X_2X_3)^{\bar{r}_3''} g_2^{\bar{r}_2''}.$$

Then, $\mathcal{B}$ uses $\mathsf{SK_{ID}}$ to respond $\mathcal{A}$'s query.

At some point, $\mathcal{A}$ sends $\mathcal{B}$ a challenge identity ID^*. $\mathcal{B}$ chooses $\delta \leftarrow \{0,1\}$ and a session key $K^* \leftarrow \mathbb{G}_T$. Then, if $\delta = 0$, it chooses $c_0^*, c_1^* \leftarrow \mathbb{G}$; otherwise (i.e., $\delta = 1$), it chooses $z \leftarrow \mathbb{Z}_N$ and sets $c_0^* = T$ and $c_1^* = T^z$. Finally, $\mathcal{B}$ sends the challenge ciphertext $C^* = (c_0^*, c_1^*)$ and the session key K^* to $\mathcal{A}$.

It is clear that, if T is a random element of $\mathbb{G}_{p_1p_2p_4}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game_{Final_3}}$. If T is a random element of $\mathbb{G}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game_{Final_4}}$. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two possibilities for T. □

F Extractable 1SPO-IBE Based on Boyen-Waters Anonymous HIBE

In this Appendix, we first show how to construct an extractable 1SPO-IBE from Boyen-Waters anonymous HIBE [9], which is based on a prime order bilinear group. Then, based on some mild complexity assumptions, we prove that the proposed extractable 1SPO-IBE scheme is IND-sID-CCA secure. One may modify it to achieve full security using the method proposed in Water's IBE scheme [29].

Specifically, the proposed scheme consists of the following algorithms:

$\mathsf{Setup}_{ex}(1^\kappa)$: Generate a bilinear group $(\mathbb{G}, \mathbb{G}_T, e, p)$ with the security parameter κ, where $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a non-degenerate bilinear map, and $\mathbb{G}, \mathbb{G}_T$ are cyclic groups of prime order p. Next choose $g, u, v, d \leftarrow \mathbb{G}$ and $\alpha, \beta, \{a_i, b_i, \theta_{i,j}\}_{0 \le i \le 3, 0 \le j \le 2} \leftarrow \mathbb{Z}_p$. Then, for each $0 \le i \le 3$ and $0 \le j \le 2$, set $g_{i,j} = g^{a_i \theta_{i,j}}$ and $h_{i,j} = g^{b_i \theta_{i,j}}$. Finally, choose two collision-resistant hash functions $\mathsf{H}_1 : \mathbb{Z}_p \times \mathbb{G} \to \mathbb{Z}_p$, $\mathsf{H}_2 : \mathbb{Z}_p \times \mathbb{G}^9 \to \mathbb{Z}_p$, and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_p$. The public parameter is

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, p),\ g, u, v, d,\ \{g_{i,j}, h_{i,j}\}_{0 \le i \le 3, 0 \le j \le 2},\ e(g,g)^\alpha,\ e(g,g)^\beta,\ \mathsf{H}_1, \mathsf{H}_2, \mathsf{KDF}).$$

The master secret key is $\mathsf{MSK} = (g^\alpha, g^\beta, \{g^{a_i}, g^{b_i}, g^{a_i b_i \theta_{i,j}}\}_{0 \le i \le 3, 0 \le j \le 2})$. We require the group $\mathbb{G}$ is an *efficiently samplable and explainable domain* associated with algorithms Sample and Sample^{-1}. Details on how to instantiate such groups are given in [3].

$\mathsf{KeyGen}_{ex}(\mathsf{PK}, \mathsf{MSK}, \mathsf{ID} \in \mathbb{Z}_p)$: Choose $\{r_i, \bar{r}_i\}_{0 \le i \le 3} \leftarrow \mathbb{Z}_p$. Output the private key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, k_0, \{k_{i,(a)}, k_{i,(b)}\}_{0 \le i \le 3}, w_0, \bar{k}_0, \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0 \le i \le 3}, \bar{w}_0)$, where

$$k_0 = g^\alpha \prod_{i=0}^{3} (g^{a_i b_i \theta_{i,0}} (g^{a_i b_i \theta_{i,1}})^{\mathsf{ID}})^{r_i},\ k_{i,(a)} = (g^{a_i})^{-r_i},\ k_{i,(b)} = (g^{b_i})^{-r_i},\ w_0 = \prod_{i=0}^{3} (g^{a_i b_i \theta_{i,2}})^{r_i},$$
$$\bar{k}_0 = g^\beta \prod_{i=0}^{3} (g^{a_i b_i \theta_{i,0}} (g^{a_i b_i \theta_{i,1}})^{\mathsf{ID}})^{\bar{r}_i},\ \bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i},\ \bar{w}_0 = \prod_{i=0}^{3} (g^{a_i b_i \theta_{i,2}})^{\bar{r}_i}.$$

$\mathsf{Encrypt}_{ex}(\mathsf{PK}, \mathsf{ID} \in \mathbb{Z}_p, m \in \{0,1\})$: If $m = 1$, choose $s, \{s_i\}_{0 \le i \le 3} \leftarrow \mathbb{Z}_p$ and compute

$$c_0 = g^s,\ \{c_{i,(a)} = (g_{i,0} g_{i,1}^{\mathsf{ID}} g_{i,2}^{\mathsf{ID}'})^{s_i},\ c_{i,(b)} = (h_{i,0} h_{i,1}^{\mathsf{ID}} h_{i,2}^{\mathsf{ID}'})^{s - s_i}\}_{0 \le i \le 3},\ c_2 = (u^t v^{\mathsf{KDF}(e(g,g)^{\alpha s})} d)^s,$$
$$K = e(g,g)^{\beta s},$$

where $\mathsf{ID}' = \mathsf{H}_1(\mathsf{ID}, c_0)$ and $t = \mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)})$, then output the ciphertext and the session key $(C, K) = ((c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}, c_2), K)$; otherwise (i.e., $m = 0$), choose c_0, $\{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}$, $c_2 \leftarrow \mathsf{Sample}(\mathbb{G})$, and output the ciphertext $C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}, c_2)$.

$\mathsf{Decrypt}_{ex}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, k_0, \{k_{i,(a)}, k_{i,(b)}\}_{0 \le i \le 3}, w_0, \bar{k}_0, \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0 \le i \le 3}, \bar{w}_0), C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}, c_2))$: Compute $\mathsf{ID}' = \mathsf{H}(\mathsf{ID}, c_0)$, $t = \mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)})$, and

$$k_0' = k_0 w_0^{\mathsf{ID}'},\ X = e(c_0, k_0') \prod_{i=0}^{3} (e(c_{i,(a)}, k_{i,(b)}) \cdot e(c_{i,(b)}, k_{i,(a)})).$$

(One can view $(k_0', \{k_{i,(a)}, k_{i,(b)}\}_{0 \le i \le 3})$ as a private key associated to the 2-level identity $\widetilde{\mathsf{ID}} = (\mathsf{ID}, \mathsf{ID}')$.) Then, check whether $e(c_2, g) = e(c_0,\ u^t v^{\mathsf{KDF}(X)} d)$. If not, set $m = 0$ and choose a session key $K \leftarrow \mathbb{G}_T$. Otherwise, set $m = 1$ and compute

$$\bar{k}_0' = \bar{k}_0 \bar{w}_0^{\mathsf{ID}'},\ K = e(c_0, \bar{k}_0') \prod_{i=0}^{3} (e(c_{i,(a)}, \bar{k}_{i,(b)}) \cdot e(c_{i,(b)}, \bar{k}_{i,(a)})).$$

Output (m, K).

Correctness. Observe that, $(k'_0, \{k_{i,(a)}, k_{i,(b)}\}_{0\le i\le 3})$ and $(\bar{k}'_0, \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0\le i\le 3})$ can be written as

$$k'_0 = g^{\alpha}\prod_{i=0}^{3}(g^{a_ib_i\theta_{i,0}}(g^{a_ib_i\theta_{i,1}})^{\mathsf{ID}}(g^{a_ib_i\theta_{i,2}})^{\mathsf{ID}'})^{r_i},\ k_{i,(a)} = (g^{a_i})^{-r_i},\ k_{i,(b)} = (g^{b_i})^{-r_i},$$
$$\bar{k}'_0 = g^{\beta}\prod_{i=0}^{3}(g^{a_ib_i\theta_{i,0}}(g^{a_ib_i\theta_{i,1}})^{\mathsf{ID}}(g^{a_ib_i\theta_{i,2}})^{\mathsf{ID}'})^{\bar{r}_i},\ \bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i}.$$

If $C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2)$ is an encryption of 1 under identity ID, then

$$X = e(c_0, k'_0)\prod_{i=0}^{3}(e(c_{i,(a)}, k_{i,(b)})\cdot e(c_{i,(b)}, k_{i,(a)})) = e(g,g)^{\alpha s},$$
$$e(c_2, g) = e((u^t v^{\mathsf{KDF}(e(g,g)^{\alpha s})} d)^s, g) = e(c_0,\ u^t v^{\mathsf{KDF}(X)} d),$$
$$K = e(c_0, \bar{k}'_0)\prod_{i=0}^{3}(e(c_{i,(a)}, \bar{k}_{i,(b)})\cdot e(c_{i,(b)}, \bar{k}_{i,(a)})) = e(g,g)^{\beta s},$$

so decryption always succeeds. On the other hand, if $C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2)$ is an encryption of 0 under identity ID, then $c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2 \in \mathbb{G}$ are chosen uniformly at random, thus $\mathsf{Pr}[e(c_2, g) = e(c_0,\ u^t v^{\mathsf{KDF}(X)} d)] \le \frac{1}{2^\kappa}$ where κ is the security parameter. So the completeness error is $\frac{1}{2^\kappa}$.

One-Sided Public Openability (1SPO). If $C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2)$ is an encryption of 0 under identity ID, then $c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2$ are randomly distributed in $\mathbb{G}$. Since the group $\mathbb{G}$ is an efficiently samplable and explainable domain associated with Sample and Sample^{-1}, $\mathsf{POpen}(\mathsf{PK}, \mathsf{ID}, C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2))$ can employ Sample^{-1} to open $(c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2)$. More precisely, $(R_0, \{R_{i,(a)}, R_{i,(b)}\}_{0\le i\le 3}, R_2) \leftarrow \mathsf{POpen}(\mathsf{PK}, \mathsf{ID}, (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2))$, where

$$R_0 \leftarrow \mathsf{Sample}^{-1}(\mathbb{G}, c_0),\ \{R_{i,(a)} \leftarrow \mathsf{Sample}^{-1}(\mathbb{G}, c_{i,(a)}), R_{i,(b)} \leftarrow \mathsf{Sample}^{-1}(\mathbb{G}, c_{i,(b)})\}_{0\le i\le 3},$$
$$R_2 \leftarrow \mathsf{Sample}^{-1}(\mathbb{G}, c_2).$$

Security. We first review some *mild* complexity assumptions in the bilinear group $(\mathbb{G}, \mathbb{G}_T, e, p)$.

Computational Diffie-Hellman (CDH) Problem. The CDH problem in $\mathbb{G}$ is defined as follows: Given a tuple $(\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y)$ as input, where $g \leftarrow \mathbb{G}$ and $x, y \leftarrow \mathbb{Z}_p$, output g^{xy}. The advantage of an algorithm $\mathcal{A}$ in solving the CDH problem is defined as $\mathsf{Pr}[\mathcal{A}(\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y) = g^{xy}]$, where the probability is over the random choices of $g \in \mathbb{G}$ and $x, y \in \mathbb{Z}_p$, and the random bits of $\mathcal{A}$. We say that the CDH assumption holds in $\mathbb{G}$ if all probabilistic polynomial time algorithms have at most a negligible advantage in solving the CDH problem in $\mathbb{G}$.

Decision Bilinear Diffie-Hellman (DBDH) Problem. The DBDH problem in $\mathbb{G}$ is defined as follows: Given a tuple $(\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y, g^z, e(g,g)^\omega)$ as input, output 1 if $\omega = xyz$ and 0 otherwise. The advantage of an algorithm $\mathcal{A}$ in solving the DBDH problem is defined as

$$|\mathsf{Pr}[\mathcal{A}(\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y, g^z, e(g,g)^\omega) = 1 : g \in \mathbb{G},\ x, y, z, \omega \leftarrow \mathbb{Z}_p]$$
$$-\mathsf{Pr}[\mathcal{A}(\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y, g^z, e(g,g)^{xyz}) = 1 : g \in \mathbb{G},\ x, y, z \leftarrow \mathbb{Z}_p]|,$$

where the probability is over the random choices of $g \in \mathbb{G}$ and $x, y, z, \omega \in \mathbb{Z}_p$, and the random bits of $\mathcal{A}$. We say that the DBDH assumption holds in $\mathbb{G}$ if all probabilistic polynomial time algorithms have at most a negligible advantage in solving the DBDH problem in $\mathbb{G}$.

Decision Linear (DLN) Problem. The DLN problem in $\mathbb{G}$ is defined as follows: Given a tuple $(\mathbb{G}, \mathbb{G}_T, e, p, g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, g^z)$ as input, output 1 if $z = z_3 + z_4$ and 0 otherwise. The advantage of an algorithm $\mathcal{A}$ in solving the DLN problem is defined as

$$\begin{aligned}&|\mathsf{Pr}[\mathcal{A}(\mathbb{G}, \mathbb{G}_T, e, p, g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, g^z) = 1 : g \in \mathbb{G},\ z_1, z_2, z_3, z_4, z \leftarrow \mathbb{Z}_p]\\ &-\mathsf{Pr}[\mathcal{A}(\mathbb{G}, \mathbb{G}_T, e, p, g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, g^{z_3+z_4}) = 1 : g \in \mathbb{G},\ z_1, z_2, z_3, z_4 \leftarrow \mathbb{Z}_p]|,\end{aligned}$$

where the probability is over the random choices of $g \in \mathbb{G}$ and $z_1, z_2, z_3, z_4, z \in \mathbb{Z}_p$, and the random bits of $\mathcal{A}$. We say that the DLN assumption holds in $\mathbb{G}$ if all probabilistic polynomial time algorithms have at most a negligible advantage in solving the DLN problem in $\mathbb{G}$.

We now state the security theorem of proposed extractable 1SPO-IBE scheme.

Theorem 3 *If DBDH, DLN and CDH Assumptions hold in $\mathbb{G}$, $\mathsf{H}_1, \mathsf{H}_2$ are two collision-resistant hash functions and KDF is a secure key derivation function, then the above extractable 1SPO-IBE scheme is $\mathsf{IND\text{-}sID\text{-}CCA}$ secure.*

Proof. To prove the $\mathsf{IND\text{-}sID\text{-}CCA}$ security of the above extractable 1SPO-IBE scheme, we consider the following games.

$\mathsf{Game}_{\mathsf{Real}}$: This is the real $\mathsf{IND\text{-}sID\text{-}CCA}$ security game.

$\mathsf{Game}_{\mathsf{Restricted}}$: This game is the same as $\mathsf{Game}_{\mathsf{Real}}$ except for the way that the challenger answers the decryption queries made by the adversary. Let $(C^* = (c_0^*, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0 \le i \le 3}, c_2^*), K^*)$ be the challenge ciphertext and session key. Recall that, C^* is a ciphertext encrypting δ under the challenge identity ID^*, where $\delta \leftarrow \{0, 1\}$ is chosen by the challenger. When the adversary issues a decryption query for $\langle \mathsf{ID}, C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}, c_2) \rangle$, the challenger proceeds just like in $\mathsf{Game}_{\mathsf{Real}}$, except for the following cases.

1. If $\mathsf{ID} = \mathsf{ID}^*$, $c_0 \neq c_0^*$ and $\mathsf{H}_1(\mathsf{ID}^*, c_0^*) = \mathsf{H}_1(\mathsf{ID}, c_0)$, then the challenger outputs reject and halts.
2. Else if $(\mathsf{ID}, c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}) = (\mathsf{ID}^*, c_0^*, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0 \le i \le 3})$ and $c_2 \neq c_2^*$, then the challenger outputs the message 0 and a random session key.
3. Else if $(\mathsf{ID}, c_0) = (\mathsf{ID}^*, c_0^*)$, $(\{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}) \neq (\{c_{i,(a)}^*, c_{i,(b)}^*\}_{0 \le i \le 3})$ and $\mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)}) = \mathsf{H}_2(\mathsf{ID}^*, c_0^*, c_{0,(a)}^*, c_{0,(b)}^*, \ldots, c_{3,(a)}^*, c_{3,(b)}^*)$, then the challenger outputs reject and halts.
4. Else if $(\mathsf{ID}, c_0) = (\mathsf{ID}^*, c_0^*)$ and $\mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)}) \neq \mathsf{H}_2(\mathsf{ID}^*, c_0^*, c_{0,(a)}^*, c_{0,(b)}^*, \ldots, c_{3,(a)}^*, c_{3,(b)}^*)$, then the challenger outputs the message 0 and a random session key.

Game_{-3} This is like $\mathsf{Game}_{\mathsf{Restricted}}$ except that the challenge ciphertext and session key $(C^* = (c_0^*, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0 \le i \le 3}, c_2^*), K^*)$ under the challenge identity ID^* for $\delta = 1$ is,

$$c_0^* = g^s,\ \{c_{i,(a)}^* = (g_{i,0} g_{i,1}^{\mathsf{ID}^*} g_{i,2}^{\mathsf{ID}^{*\prime}})^{s_i},\ c_{i,(b)}^* = (h_{i,0} h_{i,1}^{\mathsf{ID}^*} h_{i,2}^{\mathsf{ID}^{*\prime}})^{s - s_i}\}_{0 \le i \le 3},\ c_2^* = (u^t v^{\mathsf{KDF}(X')} d)^s,$$
$$K^* = e(g, g)^{\beta s},$$

where $s, \{s_i\}_{0 \le i \le 3} \leftarrow \mathbb{Z}_p$, $X' \leftarrow \mathbb{G}_T$, $\mathsf{ID}^{*\prime} = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$ and $t = \mathsf{H}_2(\mathsf{ID}^*, c_0^*, c_{0,(a)}^*, c_{0,(b)}^*, \ldots, c_{3,(a)}^*, c_{3,(b)}^*)$.

Game_{-2} The game is the same as Game_{-3} except that the challenge ciphertext and session key $(C^* = (c_0^*, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0\le i\le 3}, c_2^*), K^*)$ under the challenge identity ID^* for $\delta = 1$ is,

$$c_0^* = g^s,\ \{c_{i,(a)}^* = (g_{i,0}g_{i,1}^{\mathsf{ID}^*}g_{i,2}^{\mathsf{ID}^{*\prime}})^{s_i},\ c_{i,(b)}^* = (h_{i,0}h_{i,1}^{\mathsf{ID}^*}h_{i,2}^{\mathsf{ID}^{*\prime}})^{s-s_i}\}_{0\le i\le 3},\ c_2^* \leftarrow \mathbb{G}_T,$$
$$K^* = e(g,g)^{\beta s},$$

where $s, \{s_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$ and $\mathsf{ID}^{*\prime} = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$.

Game_{-1} This game is the same as Game_{-2} except that the challenge ciphertext and session key $(C^* = (c_0^*, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0\le i\le 3}, c_2^*), K^*)$ under the challenge identity ID^* for $\delta = 1$ is,

$$c_0^* = g^s,\ \{c_{i,(a)}^* = (g_{i,0}g_{i,1}^{\mathsf{ID}^*}g_{i,2}^{\mathsf{ID}^{*\prime}})^{s_i},\ c_{i,(b)}^* = (h_{i,0}h_{i,1}^{\mathsf{ID}^*}h_{i,2}^{\mathsf{ID}^{*\prime}})^{s-s_i}\}_{0\le i\le 3},\ c_2^* \leftarrow \mathbb{G},$$
$$K^* \leftarrow \mathbb{G}_T,$$

where $s, \{s_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$ and $\mathsf{ID}^{*\prime} = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$.

$\mathsf{Game}_k(0 \le k \le 3)$: This game is like Game_{-1} except that the challenge ciphertext and session key $(C^* = (c_0^*, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0\le i\le 3}, c_2^*), K^*)$ under the challenge identity ID^* for $\delta = 1$ is,

$$c_0^* = g^s,\ \{c_{i,(a)}^*, c_{i,(b)}^* \leftarrow \mathbb{G}\}_{0\le i\le k},\ \{c_{i,(a)}^* = (g_{i,0}g_{i,1}^{\mathsf{ID}^*}g_{i,2}^{\mathsf{ID}^{*\prime}})^{s_i},\ c_{i,(b)}^* = (h_{i,0}h_{i,1}^{\mathsf{ID}^*}h_{i,2}^{\mathsf{ID}^{*\prime}})^{s-s_i}\}_{k<i\le 3},$$
$$c_2^* \leftarrow \mathbb{G},\ K^* \leftarrow \mathbb{G}_T,$$

where $s, \{s_i\}_{k<i\le 3} \leftarrow \mathbb{Z}_p$ and $\mathsf{ID}^{*\prime} = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$.

We prove there games are indistinguishable in the following lemmas. In Game_3, it is clear that the value of δ is information-theoretically hidden from the adversary. Hence the adversary has no advantage in Game_3. Therefore, we conclude that the advantage of the adversary in $\mathsf{Game}_{\mathsf{Real}}$ is negligible. □

Lemma 12 *Suppose that CDH Assumption holds in $\mathbb{G}$ and $\mathsf{H}_1, \mathsf{H}_2$ are collision-resistant hash functions. Then $\mathsf{Game}_{\mathsf{Real}}$ and $\mathsf{Game}_{\mathsf{Restricted}}$ are computationally indistinguishable.*

Proof. Let $(C^* = (c_0^*, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0\le i\le 3}, c_2^*), K^*)$ be the challenge ciphertext and session key under the challenge identity ID^*. Recall that C^* is the ciphertext encrypting δ under ID^*, where $\delta \leftarrow \{0,1\}$ is chosen by the challenger. We observe that $\mathsf{Game}_{\mathsf{Real}}$ and $\mathsf{Game}_{\mathsf{Restricted}}$ behave equivalently unless the following events happen:

- event E_1: the adversary makes a decryption query for $\langle \mathsf{ID} = \mathsf{ID}^*, C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2)\rangle$ such that $c_0 \ne c_0^*$ and $\mathsf{H}_1(\mathsf{ID}, c_0) = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$. It is clear that, suppose that H_1 is a collision-resistant hash function, this event happens with negligible probability.
- event E_2: the adversary makes a decryption query for $\langle \mathsf{ID} = \mathsf{ID}^*, C = (c_0 = c_0^*, \{c_{i,(a)} = c_{i,(a)}^*, c_{i,(b)} = c_{i,(b)}^*\}_{0\le i\le 3}, c_2)\rangle$ such that $c_2 \ne c_2^*$ and the challenger responds with $(1, K)$ in $\mathsf{Game}_{\mathsf{Real}}$. Recall that in $\mathsf{Game}_{\mathsf{Restricted}}$, the challenger returns message 0 and a random session key for such a query, while in $\mathsf{Game}_{\mathsf{Real}}$, the challenger will employ decryption algorithm to answer the query. We will show that event E_2 occurs with negligible probability in $\mathsf{Game}_{\mathsf{Real}}$.

  In $\mathsf{Game}_{\mathsf{Real}}$, if $\delta = 1$, $\mathsf{Decrypt}_{ex}(\mathsf{PK}, \mathsf{SK}_{\mathsf{ID}}, C)$ always outputs bit 0 and a random session key and E_2 never occurs in this case. Therefore, if E_2 happens, we must have $\delta = 0$ in $\mathsf{Game}_{\mathsf{Real}}$. We show that if E_2 happens with non-negligible probability, we can construct a PPT algorithm $\mathcal{B}$

to solve the CDH problem over $\mathbb{G}$. $\mathcal{B}$ is given $(\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y)$ and going to compute g^{xy}. $\mathcal{B}$ simulates $\mathsf{Game}_{\mathsf{Real}}$ to the adversary as follows.

Initially, the adversary announces the identity ID^* it wants to be challenged upon. $\mathcal{B}$ chooses $\alpha, \beta, \{a_i, b_i, \theta_{i,j}\}_{0 \le i \le 3, 0 \le j \le 2}, \eta_u, \eta_v, \eta_d, \gamma_u, \gamma_v, \gamma_d \leftarrow \mathbb{Z}_p$ and sets

$$\{g_{i,j} = g^{a_i \theta_{i,j}}, h_{i,j} = g^{b_i \theta_{i,j}}\}_{0 \le i \le 3, 0 \le j \le 2}, u = (g^x)^{\eta_u} g^{\gamma_u}, v = (g^x)^{\eta_v} g^{\gamma_u}, d = (g^x)^{\eta_d} g^{\gamma_d}.$$

It also chooses two collision-resistant hash functions $\mathsf{H}_1 : \mathbb{Z}_p \times \mathbb{G} \to \mathbb{Z}_p$, $\mathsf{H}_2 : \mathbb{Z}_p \times \mathbb{G}^9 \to \mathbb{Z}_p$, and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_p$. Then, $\mathcal{B}$ sends the adversary the public parameter:

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, p),\ g, u, v, d,\ \{g_{i,j}, h_{i,j}\}_{0 \le i \le 3, 0 \le j \le 2},\ e(g,g)^\alpha,\ e(g,g)^\beta,\ \mathsf{H}_1, \mathsf{H}_2, \mathsf{KDF}).$$

Note that $\mathcal{B}$ knows the master secret key $\mathsf{MSK} = (g^\alpha, g^\beta, \{g^{a_i}, g^{b_i}, g^{a_i b_i \theta_{i,j}}\}_{0 \le i \le 3, 0 \le j \le 2})$ associated with PK, thus is able to answer the key generation and decryption queries made by the adversary with the help of MSK. When the adversary asks for the challenge ciphertext and session key under ID^*, $\mathcal{B}$ sends $(C^* = (c_0^* = g^y, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0 \le i \le 3}, c_2^*), K^*)$ to the adversary, where

$$\{c_{i,(a)}^*, c_{i,(b)}^*\}_{0 \le i \le 3}, c_2^* \leftarrow \mathbb{G}, K^* \leftarrow \mathbb{G}_T.$$

Since $\delta = 0$, from the adversary's point of view, the distribution of (C^*, K^*) is identical of that in $\mathsf{Game}_{\mathsf{Real}}$.

Suppose that event E_2 happens during the simulation, i.e., the adversary makes a decryption query for $\langle \mathsf{ID} = \mathsf{ID}^*, C = (c_0 = c_0^*, \{c_{i,(a)} = c_{i,(a)}^*, c_{i,(b)} = c_{i,(b)}^*\}_{0 \le i \le 3}, c_2) \rangle$ such that $c_2 \neq c_2^*$ and $\mathcal{B}$ gets $(1, K)$ when decrypting C with the private key $\mathsf{SK}_{\mathsf{ID}^*}$. Let $\mathsf{SK}_{\mathsf{ID}^*} = (\mathsf{ID}^*, k_0, \{k_{i,(a)}, k_{i,(b)}\}_{0 \le i \le 3}, w_0,\ \bar{k}_0,\ \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0 \le i \le 3},\ \bar{w}_0)$, then

$$e(c_2, g) = e(c_0^*,\ u^t v^r d),$$

where $t = \mathsf{H}_2(\mathsf{ID}^*, c_0^*, c_{0,(a)}^*, c_{0,(b)}^*, \ldots, c_{3,(a)}^*, c_{3,(b)}^*)$, $r = \mathsf{KDF}(X)$, $X = e(c_0^*, k_0 w_0^{\mathsf{ID}^{*\prime}}) \prod_{i=0}^{3} (e(c_{i,(a)}^*, k_{i,(b)}) \cdot e(c_{i,(b)}^*, k_{i,(a)}))$ and $\mathsf{ID}^{*\prime} = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$. Observe that,

$$e(c_0^*, u^t v^r d) = e(g, (u^t v^r d)^y) = e(g, (g^{xy})^{t\eta_u + r\eta_v + \eta_d} \cdot (g^y)^{t\gamma_u + r\gamma_v + \gamma_d}).$$

Since η_u, η_v, η_d are chosen uniformly at random in $\mathbb{Z}_p$ and are hidden by blinding factors $\gamma_u, \gamma_v, \gamma_d$, then with overwhelming probability, $t\eta_u + r\eta_v + \eta_d$ is not equal to 0 and $\mathcal{B}$ can compute $g^{xy} = \big(c_2/(g^y)^{t\gamma_u + r\gamma_v + \gamma_d}\big)^{1/(t\eta_u + r\eta_v + \eta_d)}$, which is a solution to the CDH problem with respect to $(\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y)$.

To sum up, event E_2 happens with negligible probability if the CDH Assumption holds.

- event E_3: the adversary makes a decryption query for $\langle \mathsf{ID} = \mathsf{ID}^*, C = (c_0 = c_0^*, \{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}, c_2) \rangle$ such that $(\{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}) \neq (\{c_{i,(a)}^*, c_{i,(b)^*}\}_{0 \le i \le 3})$ and $\mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)}) = \mathsf{H}_2(\mathsf{ID}^*, c_0^*, c_{0,(a)}^*, c_{0,(b)}^*, \ldots, c_{3,(a)}^*, c_{3,(b)}^*)$. It is clear that, suppose that H_2 is a collision-resistant hash function, this event happens with negligible probability.
- event E_4: the adversary makes a decryption query for $\langle \mathsf{ID} = \mathsf{ID}^*, C = (c_0 = c_0^*, \{c_{i,(a)}, c_{i,(b)}\}_{0 \le i \le 3}, c_2) \rangle$ such that $\mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)}) \neq \mathsf{H}_2(\mathsf{ID}^*, c_0^*, c_{0,(a)}^*, c_{0,(b)}^*, \ldots, c_{3,(a)}^*, c_{3,(b)}^*)$ and the challenger outputs $(1, K)$ in $\mathsf{Game}_{\mathsf{Real}}$. We show that, if this event happens with non-negligible probability, we can construct a PPT algorithm $\mathcal{B}$ to solve the CDH problem over $\mathbb{G}$. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y$ and will simulate $\mathsf{Game}_{\mathsf{Real}}$ with the adversary.

  Initially, the adversary announces the identity ID^* it wants to be challenged upon. $\mathcal{B}$ generates the system's public parameter and master secret key as follows.

1. Choose two collision-resistant hash functions $\mathsf{H}_1 : \mathbb{Z}_p \times \mathbb{G} \to \mathbb{Z}_p$, $\mathsf{H}_2 : \mathbb{Z}_p \times \mathbb{G}^9 \to \mathbb{Z}_p$ and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_p$.
2. Choose $\alpha, \beta, \{a_i, b_i, \theta_{i,j}\}_{0\le i\le 3, 0\le j\le 2} \leftarrow \mathbb{Z}_p$ and set $\{g_{i,j} = g^{a_i\theta_{i,j}},\ h_{i,j} = g^{b_i\theta_{i,j}}\}_{0\le i\le 3, 0\le j\le 2}$.
3. Choose $\{s_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$, and set $c_0^* = g^y$,

$$\{\tilde{c}_{i,(a)} = g^{s_i(a_i\theta_{i,0}+a_i\theta_{i,1}\mathsf{ID}^*+a_i\theta_{i,2}\mathsf{ID}^{*\prime})},\ \tilde{c}_{i,(b)} = (g^y g^{-s_i})^{b_i\theta_{i,0}+b_i\theta_{i,1}\mathsf{ID}^*+b_i\theta_{i,2}\mathsf{ID}^{*\prime}}\}_{0\le i\le 3},$$

   where $\mathsf{ID}^{*\prime} = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$. Observe that, for each $0 \le i \le 3$, $\tilde{c}_{i,(a)} = (g_{i,0} g_{i,1}^{\mathsf{ID}^*} g_{i,2}^{\mathsf{ID}^{*\prime}})^{s_i}$ and $\tilde{c}_{i,(b)} = (h_{i,0} h_{i,1}^{\mathsf{ID}^*} h_{i,2}^{\mathsf{ID}^{*\prime}})^{y-s_i}$.
4. Set $t^* = \mathsf{H}_2(\mathsf{ID}^*, c_0^*, \tilde{c}_{0,(a)}, \tilde{c}_{0,(b)}, \ldots, \tilde{c}_{3,(a)}, \tilde{c}_{3,(b)})$ and $r^* = \mathsf{KDF}(e(g^\alpha, g^y))$. Choose $\eta_u, \eta_v, \eta_d, \gamma_u, \gamma_v, \gamma_d \leftarrow \mathbb{Z}_p$, subject to the constraint that $\eta_d = -(\eta_u t^* + \eta_v r^*)$, and set

$$u = (g^x)^{\eta_u} g^{\gamma_u},\ v = (g^x)^{\eta_v} g^{\gamma_v},\ d = (g^x)^{\eta_d} g^{\gamma_d}.$$

5. Set the public parameter as

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, p),\ g, u, v, d,\ \{g_{i,j}, h_{i,j}\}_{0\le i\le 3, 0\le j\le 2},\ e(g,g)^\alpha,\ e(g,g)^\beta,\ \mathsf{H}_1, \mathsf{H}_2, \mathsf{KDF}),$$

   and the master secret key $\mathsf{MSK} = (g^\alpha, g^\beta, \{g^{a_i}, g^{b_i}, g^{a_i b_i \theta_{i,j}}\}_{0\le i\le 3, 0\le j\le 2})$. Obviously, from the perspective of the adversary the distribution of the public parameter is identical to the real construction.

$\mathcal{B}$ sends the adversary the public parameter PK. Since it knows the master secret key MSK associated with PK, then $\mathcal{B}$ is able to answer the key generation and decryption queries made by the adversary with the help of MSK. When the adversary asks for the challenge ciphertext and session key under ID^*, $\mathcal{B}$ chooses $\delta \leftarrow \{0,1\}$ and sends $(C^* = (c_0^*, \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0\le i\le 3}, c_2^*), K^*)$ to the adversary, where

$$\begin{cases} \{c_{i,(a)}^* = \tilde{c}_{i,(a)}, c_{i,(b)}^* = \tilde{c}_{i,(b)}\}_{0\le i\le 3}, c_2^* = (g^y)^{t^*\gamma_u + r^*\gamma_v + \gamma_d}, K^* = e(g^\beta, g^y) & \text{if } \delta = 1 \\ \{c_{i,(a)}^*, c_{i,(b)}^*\}_{0\le i\le 3}, c_2^* \leftarrow \mathbb{G}, K^* \leftarrow \mathbb{G}_T & \text{if } \delta = 0 \end{cases}.$$

Recall that $t^* = \mathsf{H}_2(\mathsf{ID}^*, c_0^*, \tilde{c}_{0,(a)}, \tilde{c}_{0,(b)}, \ldots, \tilde{c}_{3,(a)}, \tilde{c}_{3,(b)})$, $r^* = \mathsf{KDF}(e(g^\alpha, g^y))$ and $\eta_d = -(\eta_u t^* + \eta_v r^*)$, thus when $\delta = 1$, $c_2^* = (g^y)^{t^*\gamma_u + r^*\gamma_v + \gamma_d}$ can be written as $(u^{t^*} v^{r^*} d)^y$. Hence, whether $\delta = 1$ or $\delta = 0$, from the adversary's point of view, the distribution of (C^*, K^*) is identical of that in $\mathsf{Game}_{\mathsf{Real}}$.

Suppose that event E_4 happens during the simulation, i.e., the adversary makes a decryption query for $\langle \mathsf{ID} = \mathsf{ID}^*, C = (c_0 = c_0^*, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2)\rangle$ such that $\mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)}) \ne \mathsf{H}_2(\mathsf{ID}^*, c_0^*, c_{0,(a)}^*, c_{0,(b)}^*, \ldots, c_{3,(a)}^*, c_{3,(b)}^*)$ and $\mathcal{B}$ gets $(1, K)$ when decrypting C with the private key $\mathsf{SK}_{\mathsf{ID}^*}$. Let $\mathsf{SK}_{\mathsf{ID}^*} = (\mathsf{ID}^*, k_0, \{k_{i,(a)}, k_{i,(b)}\}_{0\le i\le 3}, w_0,\ \bar{k}_0,\ \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0\le i\le 3},\ \bar{w}_0)$, then

$$e(c_2, g) = e(c_0^*,\ u^t v^r d),$$

where $t = \mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)})$, $r = \mathsf{KDF}(X)$, $X = e(c_0^*, k_0 w_0^{\mathsf{ID}'}) \prod_{i=0}^{3} (e(c_{i,(a)}, k_{i,(b)}) \cdot e(c_{i,(b)}, k_{i,(a)}))$ and $\mathsf{ID}' = \mathsf{H}_1(\mathsf{ID}, c_0)$. Observe that,

$$e(c_0^*, u^t v^r d) = e(g, (u^t v^r d)^y) = e(g, (g^{xy})^{t\eta_u + r\eta_v + \eta_d} \cdot (g^y)^{t\gamma_u + r\gamma_v + \gamma_d}).$$

Since $t \neq t^*$ and $\eta_d = -(\eta_u t^* + \eta_v r^*)$, then with negligible probability, $t\eta_u + r\eta_v + \eta_d = (t-t^*)\eta_u + (r-r^*)\eta_v$ is equal to 0. (Note that, η_u, η_v are chosen uniformly at random in $\mathbb{Z}_p$ and are hidden by blinding factors $\gamma_u, \gamma_v, \gamma_d$.) Hence, with overwhelming probability, $\mathcal{B}$ can compute

$$g^{xy} = \left(\frac{c_2}{(g^y)^{t\gamma_u + r\gamma_v + \gamma_d}} \right)^{1/(t\eta_u + r\eta_v + \eta_d)},$$

which is a solution to the CDH problem with respect to $(\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y)$.

Hence, event E_4 happens with negligible probability if the CDH Assumption holds.

Thus, suppose that CDH Assumption holds in $\mathbb{G}$ and $\mathsf{H}_1, \mathsf{H}_2$ are collision-resistant hash functions, $\mathsf{Game}_{\mathsf{Real}}$ and $\mathsf{Game}_{\mathsf{Restricted}}$ are computationally indistinguishable. □

Lemma 13 *Suppose that DBDH Assumption holds in $\mathbb{G}$. Then $\mathsf{Game}_{\mathsf{Restricted}}$ and Game_{-3} are computationally indistinguishable.*

Proof. The proof is basically the same as the proof of confidentiality of Boyen-Waters anonymous HIBE [10] (i.e., Theorem 6 in [10]). Suppose there exists a PPT adversary $\mathcal{A}$ that distinguishes $\mathsf{Game}_{\mathsf{Restricted}}$ and Game_{-3}. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking DBDH Assumption. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, p, g, g^x, g^y, g^z, T$ and will simulate $\mathsf{Game}_{\mathsf{Restricted}}$ or Game_{-3} with $\mathcal{A}$.

Initially, the adversary $\mathcal{A}$ announces the identity ID^* it wants to be challenged upon. $\mathcal{B}$ first sets $c_0^* = g^z$ and $\mathsf{ID}^{*\prime} = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$. Next it chooses $\beta, \{a_i, b_i, \hat{\theta}_{i,j}, \tilde{\theta}_{i,j}\}_{0\le i\le 3, 0\le j\le 2}, \gamma_u, \gamma_v, \gamma_d \leftarrow \mathbb{Z}_p$, subject to the constraint that $\{\tilde{\theta}_{i,0} + \mathsf{ID}^*\tilde{\theta}_{i,1} + \mathsf{ID}^{*\prime}\tilde{\theta}_{i,2} = 0\}_{0\le i\le 3}$, and sets

$$\{g_{i,j} = \left(g^{\hat{\theta}_{i,j}} (g^x)^{\tilde{\theta}_{i,j}}\right)^{a_i}, h_{i,j} = \left(g^{\hat{\theta}_{i,j}} (g^x)^{\tilde{\theta}_{i,j}}\right)^{b_i}\}_{0\le i\le 3, 0\le j\le 2},\ u = g^{\gamma_u}, v = g^{\gamma_v}, d = g^{\gamma_d}.$$

Then, it chooses two collision-resistant hash functions $\mathsf{H}_1 : \mathbb{Z}_p \times \mathbb{G} \to \mathbb{Z}_p$, $\mathsf{H}_2 : \mathbb{Z}_p \times \mathbb{G}^9 \to \mathbb{Z}_p$, and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_p$. The adversary $\mathcal{A}$ is provided with the public parameter

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, p),\ g, u, v, d,\ \{g_{i,j}, h_{i,j}\}_{0\le i\le 3, 0\le j\le 2},\ e(g^x, g^y) = e(g,g)^{xy},\ e(g,g)^{\beta},\ \mathsf{H}_1, \mathsf{H}_2, \mathsf{KDF}).$$

Note that, it sets $\alpha = xy$ and $\{\theta_{i,j} = \hat{\theta}_{i,j} + x\tilde{\theta}_{i,j}\}_{0\le i\le 3, 0\le j\le 2}$ implicitly, which are unknown by $\mathcal{B}$.

Now we show that how $\mathcal{B}$ answers the query made by $\mathcal{A}$, which is a key query for $\langle \mathsf{ID} \rangle$ or a decryption query for $\langle \mathsf{ID}, C \rangle$.

- When $\mathcal{A}$ makes a key query for $\langle \mathsf{ID} \rangle$ such that $\mathsf{ID} \neq \mathsf{ID}^*$, $\mathcal{B}$ first defines $\{\hat{\vartheta}_i = \hat{\theta}_{i,0} + \mathsf{ID}\hat{\theta}_{i,1}, \tilde{\vartheta}_i = \tilde{\theta}_{i,0} + \mathsf{ID}\tilde{\theta}_{i,1}\}_{0\le i\le 3}$. Note that, with overwhelming probability, $\tilde{\vartheta}_i \neq 0$, since $\tilde{\theta}_{i,0}$ and $\tilde{\theta}_{i,1}$ are hidden by blinding factors $\hat{\theta}_{i,0}$ and $\hat{\theta}_{i,1}$, respectively. To proceed, $\mathcal{B}$ picks $\{\tilde{r}_i, \bar{r}_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$. It also selects $\{\chi_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$ in a manner to be specified later. Then, $\mathcal{B}$ computes

$$\begin{aligned}
k_0 &= (g^y)^{-\sum_{i=0}^{3}\chi_i\hat{\vartheta}_i/\tilde{\vartheta}_i}\prod_{i=0}^{3}\left(g^{a_ib_i\hat{\vartheta}_i}(g^x)^{a_ib_i\tilde{\vartheta}_i}\right)^{\tilde{r}_i},\\
k_{i,(a)} &= (g^{a_i})^{-\tilde{r}_i}(g^y)^{\chi_i/(b_i\tilde{\vartheta}_i)},\ k_{i,(b)} = (g^{b_i})^{-\tilde{r}_i}(g^y)^{\chi_i/(a_i\tilde{\vartheta}_i)},\\
w_0 &= (g^y)^{-\sum_{i=0}^{3}\chi_i\hat{\theta}_{i,2}/\tilde{\vartheta}_i}\prod_{i=0}^{3}\left(g^{a_ib_i\hat{\theta}_{i,2}}(g^x)^{a_ib_i\tilde{\theta}_{i,2}}\right)^{\tilde{r}_i},\\
\bar{k}_0 &= g^{\beta}\prod_{i=0}^{3}\left(g^{a_ib_i\hat{\vartheta}_i}(g^x)^{a_ib_i\tilde{\vartheta}_i}\right)^{\bar{r}_i},\ \bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i},\\
\bar{w}_0 &= \prod_{i=0}^{3}\left(g^{a_ib_i\hat{\theta}_{i,2}}(g^x)^{a_ib_i\tilde{\theta}_{i,2}}\right)^{\bar{r}_i}.
\end{aligned}$$

If we set $r_i = \tilde{r}_i - y\chi_i/(a_ib_i\tilde{\vartheta}_i)$, recall that $\alpha = xy$ and $\{\theta_{i,j} = \hat{\theta}_{i,j} + x\tilde{\theta}_{i,j}, \hat{\vartheta}_i = \hat{\theta}_{i,0} + \mathsf{ID}\hat{\theta}_{i,1}, \tilde{\vartheta}_i = \tilde{\theta}_{i,0} + \mathsf{ID}\tilde{\theta}_{i,1}\}_{0\le i\le 3, 0\le j\le 2}$, we have

$$\begin{aligned}
k_0 &= (g^{\alpha})^{\sum_{i=0}^{3}\chi_i}\prod_{i=0}^{3}\left(g^{a_ib_i\theta_{i,0}}(g^{a_ib_i\theta_{i,1}})^{\mathsf{ID}}\right)^{r_i},\\
k_{i,(a)} &= (g^{a_i})^{-r_i},\ k_{i,(b)} = (g^{b_i})^{-r_i},\\
w_0 &= (g^{\alpha})^{\sum_{i=0}^{3}\chi_i\tilde{\theta}_{i,2}/\tilde{\vartheta}_i}\prod_{i=0}^{3}(g^{a_ib_i\theta_{i,2}})^{r_i},\\
\bar{k}_0 &= g^{\beta}\prod_{i=0}^{3}\left(g^{a_ib_i\theta_{i,0}}(g^{a_ib_i\theta_{i,1}})^{\mathsf{ID}}\right)^{\bar{r}_i},\ \bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i},\ \bar{w}_0 = \prod_{i=0}^{3}(g^{a_ib_i\theta_{i,2}})^{\bar{r}_i}.
\end{aligned}$$

Observe that, if $\sum_{i=0}^{3}\chi_i = 1$ and $\sum_{i=0}^{3}\chi_i\tilde{\theta}_{i,2}/\tilde{\vartheta}_i = 0$, then the distribution of the private key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, k_0, \{k_{i,(a)},\ k_{i,(b)}\}_{0\le i\le 3}, w_0,\ \bar{k}_0, \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0\le i\le 3},\ \bar{w}_0)$ is the same as in the real scheme. As shown in [10] (i.e., Theorem 6 in [10]), $\sum_{i=0}^{3}\chi_i = 1$ and $\sum_{i=0}^{3}\chi_i\tilde{\theta}_{i,2}/\tilde{\vartheta}_i = 0$ constitute a linear system of 2 equations of 4 unknowns and admit a solution with overwhelming probability. Finally, $\mathcal{B}$ sends the private key $\mathsf{SK}_{\mathsf{ID}}$ to $\mathcal{A}$.

– When $\mathcal{A}$ makes a decryption query for $\langle \mathsf{ID}, C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2)\rangle$, let $\mathsf{ID}' = \mathsf{H}_1(\mathsf{ID}, c_0)$, $\mathcal{B}$ proceeds as follows.

  1. If $\mathsf{ID} \ne \mathsf{ID}^*$, $\mathcal{B}$ generates the private key $\mathsf{SK}_{\mathsf{ID}}$ as in the response of key query, and answers $\mathcal{A}$'s decryption query with the help of $\mathsf{SK}_{\mathsf{ID}}$.
  2. Else if $\mathsf{ID} = \mathsf{ID}^*$ and $\mathsf{ID}' = \mathsf{ID}^{*\prime}$, $\mathcal{B}$ responds as in $\mathsf{Game}_{\mathsf{Restricted}}$.
  3. Else (i.e., $\mathsf{ID} = \mathsf{ID}^*$ and $\mathsf{ID}' \ne \mathsf{ID}^{*\prime}$), $\mathcal{B}$ first defines $\{\hat{\vartheta}_i = \hat{\theta}_{i,0} + \mathsf{ID}\hat{\theta}_{i,1} + \mathsf{ID}'\hat{\theta}_{i,2}, \tilde{\vartheta}_i = \tilde{\theta}_{i,0} + \mathsf{ID}\tilde{\theta}_{i,1} + \mathsf{ID}'\tilde{\theta}_{i,2}\}_{0\le i\le 3}$. Since $\tilde{\theta}_{i,0} + \mathsf{ID}^*\tilde{\theta}_{i,1} + \mathsf{ID}^{*\prime}\tilde{\theta}_{i,2} = 0$ and $\mathsf{ID}' \ne \mathsf{ID}^{*\prime}$, then $\tilde{\vartheta}_i \ne 0$. To proceed, $\mathcal{B}$ picks $\{\tilde{r}_i, \bar{r}_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$. It also selects $\{\chi_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$ in a manner to be specified later.

Then, $\mathcal{B}$ computes

$$
\begin{aligned}
&k_0 = (g^y)^{-\sum_{i=0}^{3}\chi_i\hat{\vartheta}_i/\tilde{\vartheta}_i}\prod_{i=0}^{3}\left(g^{a_ib_i\hat{\vartheta}_i}(g^x)^{a_ib_i\tilde{\vartheta}_i}\right)^{\tilde{r}_i},\\
&k_{i,(a)} = (g^{a_i})^{-\tilde{r}_i}(g^y)^{\chi_i/(b_i\tilde{\vartheta}_i)},\ k_{i,(b)} = (g^{b_i})^{-\tilde{r}_i}(g^y)^{\chi_i/(a_i\tilde{\vartheta}_i)},\\
&\bar{k}_0 = g^\beta\prod_{i=0}^{3}\left(g^{a_ib_i\hat{\vartheta}_i}(g^x)^{a_ib_i\tilde{\vartheta}_i}\right)^{\bar{r}_i},\ \bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i},
\end{aligned}
$$

If we set $r_i = \tilde{r}_i - y\chi_i/(a_ib_i\tilde{\vartheta}_i)$, recall that $\alpha = xy$ and $\{\theta_{i,j} = \hat{\theta}_{i,j} + x\tilde{\theta}_{i,j}, \hat{\vartheta}_i = \hat{\theta}_{i,0} + \mathsf{ID}\hat{\theta}_{i,1} + \mathsf{ID}'\hat{\theta}_{i,2}, \tilde{\vartheta}_i = \tilde{\theta}_{i,0} + \mathsf{ID}\tilde{\theta}_{i,1} + \mathsf{ID}'\tilde{\theta}_{i,2}\}_{0\le i\le 3, 0\le j\le 2}$, we have

$$
\begin{aligned}
&k_0 = (g^\alpha)^{\sum_{i=0}^{3}\chi_i}\prod_{i=0}^{3}\left(g^{a_ib_i\theta_{i,0}}(g^{a_ib_i\theta_{i,1}})^{\mathsf{ID}}(g^{a_ib_i\theta_{i,2}})^{\mathsf{ID}'}\right)^{r_i},\\
&k_{i,(a)} = (g^{a_i})^{-r_i},\ k_{i,(b)} = (g^{b_i})^{-r_i},\\
&\bar{k}_0 = g^\beta\prod_{i=0}^{3}\left(g^{a_ib_i\theta_{i,0}}(g^{a_ib_i\theta_{i,1}})^{\mathsf{ID}}(g^{a_ib_i\theta_{i,2}})^{\mathsf{ID}'}\right)^{\bar{r}_i},\ \bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i}.
\end{aligned}
$$

Observe that, if $\sum_{i=0}^{3}\chi_i = 1$, then $(k_0, \{k_{i,(a)}, k_{i,(b)}\}_{0\le i\le 3},\ \bar{k}_0, \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0\le i\le 3})$ can be viewed as a private key for the 2-level identity $\widetilde{\mathsf{ID}} = (\mathsf{ID}, \mathsf{ID}')$. Similarly, $\sum_{i=0}^{3}\chi_i = 1$, which constitutes a linear system of 1 equation of 4 unknowns, has a solution with overwhelming probability. Next, $\mathcal{B}$ computes

$$
t = \mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)}),\ X = e(c_0, k_0)\prod_{i=0}^{3}(e(c_{i,(a)}, k_{i,(b)})\cdot e(c_{i,(b)}, k_{i,(a)})),
$$

and checks whether $e(c_2, g) = e(c_0,\ u^t v^{\mathsf{KDF}(X)} d)$. If not, $\mathcal{B}$ sets $m = 0$ and chooses a session key $K \leftarrow \mathbb{G}_T$. Otherwise, $\mathcal{B}$ sets $m = 1$ and computes

$$
K = e(c_0, \bar{k}_0)\prod_{i=0}^{3}(e(c_{i,(a)}, \bar{k}_{i,(b)})\cdot e(c_{i,(b)}, \bar{k}_{i,(a)})).
$$

Finally, $\mathcal{B}$ sends (m, K) to the adversary $\mathcal{A}$.

At some point, the adversary $\mathcal{A}$ asks for the challenge ciphertext and session key under ID^*. $\mathcal{B}$ chooses $\delta \leftarrow \{0,1\}$ and does the following. If $\delta = 0$, it chooses $\{c^*_{i,(a)}, c^*_{i,(b)}\}_{0\le i\le 3}, c^*_2 \leftarrow \mathbb{G}$ and $K^* \leftarrow \mathbb{G}_T$; otherwise (i.e., $\delta = 1$), it chooses $\{s_i\}_{0\le i\le 3}$ and sets

$$
\begin{gathered}
\{c^*_{i,(a)} = (g^{s_i})^{a_i(\hat{\theta}_{i,0}+\mathsf{ID}^*\hat{\theta}_{i,1}+\mathsf{ID}^{*\prime}\hat{\theta}_{i,2})}, c^*_{i,(b)} = (g^z g^{-s_i})^{b_i(\hat{\theta}_{i,0}+\mathsf{ID}^*\hat{\theta}_{i,1}+\mathsf{ID}^{*\prime}\hat{\theta}_{i,2})}\}_{0\le i\le 3},\\
t^* = \mathsf{H}_2(\mathsf{ID}^*, c^*_0, c^*_{0,(a)}, c^*_{0,(b)}, \ldots, c^*_{3,(a)}, c^*_{3,(b)}), r^* = \mathsf{KDF}(T), c^*_2 = (g^z)^{t^*\gamma_u + r^*\gamma_v + \gamma_d}, K^* = e(g^\beta, g^z).
\end{gathered}
$$

Finally, $\mathcal{B}$ sends the challenge ciphertext $C^* = (c^*_0, \{c^*_{i,(a)}, c^*_{i,(b)}\}_{0\le i\le 3}, c^*_2)$ and session key K^* to the adversary $\mathcal{A}$. Note that, since $\tilde{\theta}_{i,0} + \mathsf{ID}^*\tilde{\theta}_{i,1} + \mathsf{ID}^{*\prime}\tilde{\theta}_{i,2} = 0$, when $\delta = 1$, $c^*_{i,(a)}$ and $c^*_{i,(b)}$ can be written as $(g_{i,0}g_{i,1}^{\mathsf{ID}^*}g_{i,2}^{\mathsf{ID}^{*\prime}})^{s_i}$ and $(h_{i,0}h_{i,1}^{\mathsf{ID}^*}h_{i,2}^{\mathsf{ID}^{*\prime}})^{z-s_i}$,respectively.

It is clear that, if $T = e(g,g)^{xyz}$, then $\mathcal{B}$ has properly simulated $\mathsf{Game}_{\mathsf{Restricted}}$. If T is a random element of $\mathbb{G}_T$, then $\mathcal{B}$ has properly simulated Game_{-3}. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two possibilities for T. □

Lemma 14 *Suppose that* KDF *is a secure key derivation function. Then* Game_{-3} *and* Game_{-2} *are computationally indistinguishable.*

Proof. It is clear that the adversary distinguishes Game_{-3} and Game_{-2} with negligible probability, since in Game_{-3}, $X' \leftarrow \mathbb{G}_T$ and KDF is a secure key derivation function. □

Lemma 15 *Suppose that DBDH Assumption holds in* $\mathbb{G}$*. Then* Game_{-2} *and* Game_{-1} *are computationally indistinguishable.*

Proof. This argument follows almost identically to that of Lemma 13, except where the simulation is done over the parameter β in place of α. □

Lemma 16 *Suppose that DLN Assumption holds in* $\mathbb{G}$*. Then for each* $0 \le k \le 3$, Game_{k-1} *and* Game_k *are computationally indistinguishable.*

Proof. The proof is basically same as the proof of anonymity of Boyen-Waters anonymous HIBE [10] (i.e., Theorem 7 in [10]). For ease of description, without loss of generality, we assume $k = 0$. We will show that Game_{-1} and Game_0 are computationally indistinguishable. Suppose there exists a PPT adversary $\mathcal{A}$ that distinguishes Game_{-1} and Game_0. Then we can build an algorithm $\mathcal{B}$ with non-negligible advantage in breaking DLN Assumption. $\mathcal{B}$ is given $\mathbb{G}, \mathbb{G}_T, e, p, g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, T$ and will simulate Game_{-1} or Game_0 with $\mathcal{A}$.

Initially, the adversary $\mathcal{A}$ announces the identity ID^* it wants to be challenged upon. $\mathcal{B}$ first sets $c_0^* = T$ and $\mathsf{ID}^{*\prime} = \mathsf{H}_1(\mathsf{ID}^*, c_0^*)$. Next it chooses $\alpha, \beta, \{\theta_{0,j}\}_{0 \le j \le 2}, \{a_i, b_i, \hat{\theta}_{i,j}, \tilde{\theta}_{i,j}\}_{1 \le i \le 3, 0 \le j \le 2}, \gamma_u, \gamma_v, \gamma_d \leftarrow \mathbb{Z}_p$, subject to the constraint that $\{\tilde{\theta}_{i,0} + \mathsf{ID}^* \tilde{\theta}_{i,1} + \mathsf{ID}^{*\prime} \tilde{\theta}_{i,2} = 0\}_{1 \le i \le 3}$, and sets

$$u = g^{\gamma_u}, v = g^{\gamma_v}, d = g^{\gamma_d},$$
$$\{g_{0,j} = (g^{z_1})^{\theta_{0,j}}, h_{0,j} = (g^{z_2})^{\theta_{0,j}}\}_{0 \le j \le 2},$$
$$\{g_{i,j} = \left(g^{\hat{\theta}_{i,j}} (g^{z_1})^{\tilde{\theta}_{i,j}}\right)^{a_i}, h_{i,j} = \left(g^{\hat{\theta}_{i,j}} (g^{z_1})^{\tilde{\theta}_{i,j}}\right)^{b_i}\}_{1 \le i \le 3, 0 \le j \le 2}.$$

Then, it chooses two collision-resistant hash functions $\mathsf{H}_1 : \mathbb{Z}_p \times \mathbb{G} \to \mathbb{Z}_p$, $\mathsf{H}_2 : \mathbb{Z}_p \times \mathbb{G}^9 \to \mathbb{Z}_p$, and a key derivation function $\mathsf{KDF} : \mathbb{G}_T \to \mathbb{Z}_p$. The adversary $\mathcal{A}$ is provided with the public parameter

$$\mathsf{PK} = ((\mathbb{G}, \mathbb{G}_T, e, p),\ g, u, v, d,\ \{g_{i,j}, h_{i,j}\}_{0 \le i \le 3, 0 \le j \le 2},\ e(g,g)^{\alpha},\ e(g,g)^{\beta},\ \mathsf{H}_1, \mathsf{H}_2, \mathsf{KDF}).$$

Note that, it sets $a_0 = z_1, b_0 = z_2$ and $\{\theta_{i,j} = \hat{\theta}_{i,j} + z_1 \tilde{\theta}_{i,j}\}_{1 \le i \le 3, 0 \le j \le 2}$ implicitly, which are unknown by $\mathcal{B}$.

Now we show that how $\mathcal{B}$ answers the query made by $\mathcal{A}$, which is a key query for $\langle \mathsf{ID} \rangle$ or a decryption query for $\langle \mathsf{ID}, C \rangle$.

- When $\mathcal{A}$ makes a key query for $\langle \mathsf{ID} \rangle$ such that $\mathsf{ID} \ne \mathsf{ID}^*$, $\mathcal{B}$ first defines $\vartheta_0 = \theta_{0,0} + \mathsf{ID}\theta_{0,1}$ and $\{\hat{\vartheta}_i = \hat{\theta}_{i,0} + \mathsf{ID}\hat{\theta}_{i,1}, \tilde{\vartheta}_i = \tilde{\theta}_{i,0} + \mathsf{ID}\tilde{\theta}_{i,1}\}_{1 \le i \le 3}$. Note that, with overwhelming probability, $\tilde{\vartheta}_i \ne 0$, since $\tilde{\theta}_{i,0}$ and $\tilde{\theta}_{i,1}$ are hidden by blinding factors $\hat{\theta}_{i,0}$ and $\hat{\theta}_{i,1}$, respectively. To proceed, $\mathcal{B}$ picks

$\{r'_i, \bar{r}'_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$. It also selects $\{\chi_i, \chi'_i\}_{1\le i\le 3} \leftarrow \mathbb{Z}_p$ in a manner to be specified later. Then, $\mathcal{B}$ computes

$$
\begin{aligned}
&k_0 = g^{\alpha} \prod_{i=1}^{3} \left(\left((g^{z_2})^{-\hat{\vartheta}_i/\tilde{\vartheta}_i} \right)^{\vartheta_0 r'_0} \left(g^{a_i b_i \hat{\vartheta}_i} (g^{z_1})^{a_i b_i \tilde{\vartheta}_i} \right)^{r'_i} \right), \\
&k_{0,(a)} = (g^{z_1})^{-3r'_0}, k_{0,(b)} = (g^{z_2})^{-3r'_0}, \\
&\{k_{i,(a)} = (g^{a_i})^{-r'_i} (g^{z_2})^{\chi_i r'_0 \vartheta_0/(b_i \tilde{\vartheta}_i)},\ k_{i,(b)} = (g^{b_i})^{-r'_i} (g^{z_2})^{\chi_i r'_0 \vartheta_o/(a_i \tilde{\vartheta}_i)}\}_{1\le i\le 3}, \\
&w_0 = \left(\prod_{i=1}^{3} (g^{z_2})^{-\chi_i \hat{\theta}_{i,2} r'_0 \vartheta_0/\tilde{\vartheta}_i} \right) \prod_{i=1}^{3} \left(g^{a_i b_i \hat{\theta}_{i,2}} (g^{z_1})^{a_i b_i \tilde{\theta}_{i,2}} \right)^{r'_i}, \\
&\bar{k}_0 = g^{\alpha} \prod_{i=1}^{3} \left(\left((g^{z_2})^{-\hat{\vartheta}_i/\tilde{\vartheta}_i} \right)^{\vartheta_0 \bar{r}'_0} \left(g^{a_i b_i \hat{\vartheta}_i} (g^{z_1})^{a_i b_i \tilde{\vartheta}_i} \right)^{\bar{r}'_i} \right), \\
&\bar{k}_{0,(a)} = (g^{z_1})^{-3\bar{r}'_0}, \bar{k}_{0,(b)} = (g^{z_2})^{-3\bar{r}'_0}, \\
&\{\bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}'_i} (g^{z_2})^{\chi'_i \bar{r}'_0 \vartheta_0/(b_i \tilde{\vartheta}_i)},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}'_i} (g^{z_2})^{\chi'_i \bar{r}'_0 \vartheta_o/(a_i \tilde{\vartheta}_i)}\}_{1\le i\le 3}, \\
&\bar{w}_0 = \left(\prod_{i=1}^{3} (g^{z_2})^{-\chi'_i \hat{\theta}_{i,2} \bar{r}'_0 \vartheta_0/\tilde{\vartheta}_i} \right) \prod_{i=1}^{3} \left(g^{a_i b_i \hat{\theta}_{i,2}} (g^{z_1})^{a_i b_i \tilde{\theta}_{i,2}} \right)^{\bar{r}'_i},
\end{aligned}
$$

If we set $r_0 = 3r'_0$, $\{r_i = r'_i - z_2 \chi_i r'_0 \vartheta_0/(a_i b_i \tilde{\vartheta}_i)\}_{1\le i\le 3}$, $\bar{r}_0 = 3\bar{r}'_0$ and $\{\bar{r}_i = \bar{r}'_i - z_2 \chi'_i \bar{r}'_0 \vartheta_0/(a_i b_i \tilde{\vartheta}_i)$ $\}_{1\le i\le 3}$, recall that $a_0 = z_1$, $b_0 = z_2$, $\vartheta_0 = \theta_{0,0} + \mathsf{ID}\theta_{0,1}$ and $\{\theta_{i,j} = \hat{\theta}_{i,j} + z_1 \tilde{\theta}_{i,j}, \hat{\vartheta}_i = \hat{\theta}_{i,0} + \mathsf{ID}\hat{\theta}_{i,1}, \tilde{\vartheta}_i = \tilde{\theta}_{i,0} + \mathsf{ID}\tilde{\theta}_{i,1}\}_{1\le i\le 3, 0\le j\le 2}$, we have

$$
\begin{aligned}
&k_0 = g^{\alpha} \left((g^{z_2})^{r'_0 \vartheta_0} \right)^{\sum_{i=1}^{3} (\chi_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i} \left(\left(g^{a_0 b_0 \theta_{0,0}} (g^{a_0 b_0 \theta_{0,1}})^{\mathsf{ID}} \right)^{r_0} \right)^{\sum_{i=1}^{3} \chi_i/3} \prod_{i=1}^{3} \left(g^{a_i b_i \theta_{i,0}} (g^{a_i b_i \theta_{i,1}})^{\mathsf{ID}} \right)^{r_i}, \\
&k_{i,(a)} = (g^{a_i})^{-r_i},\ k_{i,(b)} = (g^{b_i})^{-r_i}, \\
&\bar{k}_0 = g^{\alpha} \left((g^{z_2})^{\bar{r}'_0 \vartheta_0} \right)^{\sum_{i=1}^{3} (\chi'_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i} \left(\left(g^{a_0 b_0 \theta_{0,0}} (g^{a_0 b_0 \theta_{0,1}})^{\mathsf{ID}} \right)^{\bar{r}_0} \right)^{\sum_{i=1}^{3} \chi'_i/3} \prod_{i=1}^{3} \left(g^{a_i b_i \theta_{i,0}} (g^{a_i b_i \theta_{i,1}})^{\mathsf{ID}} \right)^{\bar{r}_i}, \\
&\bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i}.
\end{aligned}
$$

Observe that, if $\sum_{i=1}^{3} (\chi_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i = 0$, $\sum_{i=1}^{3} (\chi_i - 1) = 0$, $\sum_{i=1}^{3} (\chi'_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i = 0$, $\sum_{i=1}^{3} (\chi'_i - 1) = 0$, $\sum_{i=1}^{3} \chi_i \tilde{\theta}_{i,2}/\tilde{\vartheta}_i = 3\theta_{0,2}/\vartheta_0$ and $\sum_{i=1}^{3} \chi'_i \tilde{\theta}_{i,2}/\tilde{\vartheta}_i = 3\theta_{0,2}/\vartheta_0$, then

$$k_0 = g^{\alpha}\prod_{i=0}^{3}(g^{a_ib_i\theta_{i,0}}(g^{a_ib_i\theta_{i,1}})^{\mathsf{ID}})^{r_i},\ k_{i,(a)} = (g^{a_i})^{-r_i},\ k_{i,(b)} = (g^{b_i})^{-r_i},$$

$$\begin{aligned}
w_0 &= \left(\prod_{i=1}^{3}(g^{z_2})^{-\chi_i\hat{\theta}_{i,2}r'_0\vartheta_0/\tilde{\vartheta}_i}\right)\prod_{i=1}^{3}\left(g^{a_ib_i\hat{\theta}_{i,2}}(g^{z_1})^{a_ib_i\tilde{\theta}_{i,2}}\right)^{r'_i}\\
&= (g^{z_1z_2r'_0})^{3\theta_{0,2}-\sum_{i=1}^{3}\chi_i\tilde{\theta}_{i,2}\vartheta_0/\tilde{\vartheta}_i}\left(\prod_{i=1}^{3}(g^{z_2})^{-\chi_i\hat{\theta}_{i,2}r'_0\vartheta_0/\tilde{\vartheta}_i}\right)\prod_{i=1}^{3}\left(g^{a_ib_i\hat{\theta}_{i,2}}(g^{z_1})^{a_ib_i\tilde{\theta}_{i,2}}\right)^{r'_i}\\
&= \prod_{i=0}^{3}(g^{a_ib_i\theta_{i,2}})^{r_i},
\end{aligned}$$

and $\bar{k}_0 = g^{\beta}\prod_{i=0}^{3}(g^{a_ib_i\theta_{i,0}}(g^{a_ib_i\theta_{i,1}})^{\mathsf{ID}})^{\bar{r}_i}$, $\bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i}$, $\bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i}$,

$$\begin{aligned}
\bar{w}_0 &= \left(\prod_{i=1}^{3}(g^{z_2})^{-\chi'_i\hat{\theta}_{i,2}\bar{r}'_0\vartheta_0/\tilde{\vartheta}_i}\right)\prod_{i=1}^{3}\left(g^{a_ib_i\hat{\theta}_{i,2}}(g^{z_1})^{a_ib_i\tilde{\theta}_{i,2}}\right)^{\bar{r}'_i}\\
&= (g^{z_1z_2\bar{r}'_0})^{3\theta_{0,2}-\sum_{i=1}^{3}\chi'_i\tilde{\theta}_{i,2}\vartheta_0/\tilde{\vartheta}_i}\left(\prod_{i=1}^{3}(g^{z_2})^{-\chi'_i\hat{\theta}_{i,2}\bar{r}'_0\vartheta_0/\tilde{\vartheta}_i}\right)\prod_{i=1}^{3}\left(g^{a_ib_i\hat{\theta}_{i,2}}(g^{z_1})^{a_ib_i\tilde{\theta}_{i,2}}\right)^{\bar{r}'_i}\\
&= \prod_{i=0}^{3}(g^{a_ib_i\theta_{i,2}})^{\bar{r}_i}.
\end{aligned}$$

Hence, the distribution of the private key $\mathsf{SK}_{\mathsf{ID}} = (\mathsf{ID}, k_0, \{k_{i,(a)},\ k_{i,(b)}\}_{0\le i\le 3}, w_0,\ \bar{k}_0, \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0\le i\le 3}, \bar{w}_0)$ is the same as in the real scheme. As shown in [10] (i.e., Theorem 7 in [10]), $\sum_{i=1}^{3}(\chi_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i = 0$, $\sum_{i=1}^{3}(\chi_i - 1) = 0$, $\sum_{i=1}^{3}(\chi'_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i = 0$, $\sum_{i=1}^{3}(\chi'_i - 1) = 0$, $\sum_{i=1}^{3}\chi_i\tilde{\theta}_{i,2}/\tilde{\vartheta}_i = 3\theta_{0,2}/\vartheta_0$ and $\sum_{i=1}^{3}\chi'_i\tilde{\theta}_{i,2}/\tilde{\vartheta}_i = 3\theta_{0,2}/\vartheta_0$ have a solution with overwhelming probability. Finally, $\mathcal{B}$ sends the private key $\mathsf{SK}_{\mathsf{ID}}$ to $\mathcal{A}$.

- When $\mathcal{A}$ makes a decryption query for $\langle \mathsf{ID}, C = (c_0, \{c_{i,(a)}, c_{i,(b)}\}_{0\le i\le 3}, c_2)\rangle$, let $\mathsf{ID}' = \mathsf{H}_1(\mathsf{ID}, c_0)$, $\mathcal{B}$ proceeds as follows.
  1. If $\mathsf{ID} \neq \mathsf{ID}^*$, $\mathcal{B}$ generates the private key $\mathsf{SK}_{\mathsf{ID}}$ as in the response of key query, and answers $\mathcal{A}$'s decryption query with the help of $\mathsf{SK}_{\mathsf{ID}}$.
  2. Else if $\mathsf{ID} = \mathsf{ID}^*$ and $\mathsf{ID}' = \mathsf{ID}^{*\prime}$, $\mathcal{B}$ responds as in $\mathsf{Game}_{\mathsf{Restricted}}$.
  3. Else (i.e., $\mathsf{ID} = \mathsf{ID}^*$ and $\mathsf{ID}' \neq \mathsf{ID}^{*\prime}$), $\mathcal{B}$ first defines $\vartheta_0 = \theta_{0,0} + \mathsf{ID}\theta_{0,1} + \mathsf{ID}'\theta_{0,2}$ and $\{\hat{\vartheta}_i = \hat{\theta}_{i,0} + \mathsf{ID}\hat{\theta}_{i,1} + \mathsf{ID}'\hat{\theta}_{i,2}, \tilde{\vartheta}_i = \tilde{\theta}_{i,0} + \mathsf{ID}\tilde{\theta}_{i,1} + \mathsf{ID}'\tilde{\theta}_{i,2}\}_{1\le i\le 3}$. Since $\{\tilde{\theta}_{i,0} + \mathsf{ID}^*\tilde{\theta}_{i,1} + \mathsf{ID}^{*\prime}\tilde{\theta}_{i,2} = 0\}_{1\le i\le 3}$ and $\mathsf{ID}' \neq \mathsf{ID}^{*\prime}$, then $\{\tilde{\vartheta}_i \neq 0\}_{1\le i\le 3}$. To proceed, $\mathcal{B}$ picks $\{r'_i, \bar{r}'_i\}_{0\le i\le 3} \leftarrow \mathbb{Z}_p$. It also selects $\{\chi_i, \chi'_i\}_{1\le i\le 3} \leftarrow \mathbb{Z}_p$ in a manner to be specified later. Then, $\mathcal{B}$ computes

$$\begin{aligned}
&k_0 = g^{\alpha}\prod_{i=1}^{3}\left(\left((g^{z_2})^{-\hat{\vartheta}_i/\tilde{\vartheta}_i}\right)^{\vartheta_0 r'_0}\left(g^{a_ib_i\hat{\vartheta}_i}(g^{z_1})^{a_ib_i\tilde{\vartheta}_i}\right)^{r'_i}\right),\\
&k_{0,(a)} = (g^{z_1})^{-3r'_0}, k_{0,(b)} = (g^{z_2})^{-3r'_0},\\
&\{k_{i,(a)} = (g^{a_i})^{-r'_i}(g^{z_2})^{\chi_i r'_0\vartheta_0/(b_i\tilde{\vartheta}_i)},\ k_{i,(b)} = (g^{b_i})^{-r'_i}(g^{z_2})^{\chi_i r'_0\vartheta_o/(a_i\tilde{\vartheta}_i)}\}_{1\le i\le 3},
\end{aligned}$$

and

$$\bar{k}_0 = g^{\alpha} \prod_{i=1}^{3} \left(\left((g^{z_2})^{-\hat{\vartheta}_i/\tilde{\vartheta}_i} \right)^{\vartheta_0 \bar{r}'_0} \left(g^{a_i b_i \hat{\vartheta}_i} (g^{z_1})^{a_i b_i \tilde{\vartheta}_i} \right)^{\bar{r}'_i} \right),$$
$$\bar{k}_{0,(a)} = (g^{z_1})^{-3\bar{r}'_0}, \bar{k}_{0,(b)} = (g^{z_2})^{-3\bar{r}'_0},$$
$$\{\bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}'_i} (g^{z_2})^{\chi'_i \bar{r}'_0 \vartheta_0/(b_i \tilde{\vartheta}_i)},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}'_i} (g^{z_2})^{\chi'_i \bar{r}'_0 \vartheta_o/(a_i \tilde{\vartheta}_i)}\}_{1 \le i \le 3}.$$

If we set $r_0 = 3r'_0$, $\{r_i = r'_i - z_2 \chi_i r'_0 \vartheta_0/(a_i b_i \tilde{\vartheta}_i)\}_{1\le i\le 3}$, $\bar{r}_0 = 3\bar{r}'_0$ and $\{\bar{r}_i = \bar{r}'_i - z_2 \chi'_i \bar{r}'_0 \vartheta_0/(a_i b_i \tilde{\vartheta}_i)$ $\}_{1\le i\le 3}$, recall that $a_0 = z_1$, $b_0 = z_2$, $\vartheta_0 = \theta_{0,0} + \mathsf{ID}\theta_{0,1} + \mathsf{ID}'\theta_{0,2}$ and $\{\theta_{i,j} = \hat{\theta}_{i,j} + z_1 \tilde{\theta}_{i,j}, \hat{\vartheta}_i = \hat{\theta}_{i,0} + \mathsf{ID}\hat{\theta}_{i,1} + \mathsf{ID}'\hat{\theta}_{i,2}, \tilde{\vartheta}_i = \tilde{\theta}_{i,0} + \mathsf{ID}\tilde{\theta}_{i,1} + \mathsf{ID}'\tilde{\theta}_{i,2}\}_{1\le i\le 3, 0\le j\le 2}$, we have

$$k_0 = g^{\alpha} \left((g^{z_2})^{r'_0 \vartheta_0} \right)^{\sum_{i=1}^{3} (\chi_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i}$$
$$\cdot \left(\left(g^{a_0 b_0 \theta_{0,0}} (g^{a_0 b_0 \theta_{0,1}})^{\mathsf{ID}} (g^{a_0 b_0 \theta_{0,2}})^{\mathsf{ID}'} \right)^{r_0} \right)^{\sum_{i=1}^{3} \chi_i/3} \prod_{i=1}^{3} \left(g^{a_i b_i \theta_{i,0}} (g^{a_i b_i \theta_{i,1}})^{\mathsf{ID}} (g^{a_i b_i \theta_{i,2}})^{\mathsf{ID}'} \right)^{r_i},$$
$$k_{i,(a)} = (g^{a_i})^{-r_i},\ k_{i,(b)} = (g^{b_i})^{-r_i},$$
$$\bar{k}_0 = g^{\alpha} \left((g^{z_2})^{\bar{r}'_0 \vartheta_0} \right)^{\sum_{i=1}^{3} (\chi'_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i}$$
$$\cdot \left(\left(g^{a_0 b_0 \theta_{0,0}} (g^{a_0 b_0 \theta_{0,1}})^{\mathsf{ID}} (g^{a_0 b_0 \theta_{0,2}})^{\mathsf{ID}'} \right)^{\bar{r}_0} \right)^{\sum_{i=1}^{3} \chi'_i/3} \prod_{i=1}^{3} \left(g^{a_i b_i \theta_{i,0}} (g^{a_i b_i \theta_{i,1}})^{\mathsf{ID}} (g^{a_i b_i \theta_{i,2}})^{\mathsf{ID}'} \right)^{\bar{r}_i},$$
$$\bar{k}_{i,(a)} = (g^{a_i})^{-\bar{r}_i},\ \bar{k}_{i,(b)} = (g^{b_i})^{-\bar{r}_i}.$$

Observe that, if if $\sum_{i=1}^{3}(\chi_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i = 0$, $\sum_{i=1}^{3}(\chi_i - 1) = 0$, $\sum_{i=1}^{3}(\chi'_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i = 0$ and $\sum_{i=1}^{3}(\chi'_i - 1) = 0$, then $(k_0, \{k_{i,(a)}, k_{i,(b)}\}_{0\le i\le 3}, \bar{k}_0, \{\bar{k}_{i,(a)}, \bar{k}_{i,(b)}\}_{0\le i\le 3})$ can be viewed as a private key for the 2-level identity $\widetilde{\mathsf{ID}} = (\mathsf{ID}, \mathsf{ID}')$. On the other hand, similarly, $\sum_{i=1}^{3}(\chi_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i = 0$, $\sum_{i=1}^{3}(\chi_i - 1) = 0$, $\sum_{i=1}^{3}(\chi'_i - 1)\hat{\vartheta}_i/\tilde{\vartheta}_i = 0$ and $\sum_{i=1}^{3}(\chi'_i - 1) = 0$ have a solution with overwhelming probability. Next, $\mathcal{B}$ computes

$$t = \mathsf{H}_2(\mathsf{ID}, c_0, c_{0,(a)}, c_{0,(b)}, \ldots, c_{3,(a)}, c_{3,(b)}),\ X = e(c_0, k_0) \prod_{i=0}^{3} (e(c_{i,(a)}, k_{i,(b)}) \cdot e(c_{i,(b)}, k_{i,(a)})),$$

and checks whether

$$e(c_2, g) = e(c_0,\ u^t v^{\mathsf{KDF}(X)} d).$$

If not, $\mathcal{B}$ sets $m = 0$ and chooses a session key $K \leftarrow \mathbb{G}_T$. Otherwise, $\mathcal{B}$ sets $m = 1$ and computes

$$K = e(c_0, \bar{k}_0) \prod_{i=0}^{3} (e(c_{i,(a)}, \bar{k}_{i,(b)}) \cdot e(c_{i,(b)}, \bar{k}_{i,(a)})).$$

Finally, $\mathcal{B}$ sends (m, K) to the adversary $\mathcal{A}$.

At some point, the adversary $\mathcal{A}$ asks for the challenge ciphertext and session key under ID^*. $\mathcal{B}$ chooses $\delta \leftarrow \{0,1\}$ and does the following. If $\delta = 0$, it chooses $\{c^*_{i,(a)}, c^*_{i,(b)}\}_{0\le i\le 3}, c^*_2 \leftarrow \mathbb{G}$ and $K^* \leftarrow \mathbb{G}_T$; otherwise (i.e., $\delta = 1$), it chooses $\{s_i\}_{1\le i\le 3}$ and sets

$$c^*_{0,(a)} = (g^{z_1 z_3})^{\theta_{0,0}+\mathsf{ID}^*\theta_{0,1}+\mathsf{ID}^{*\prime}\theta_{0,2}}, c^*_{0,(b)} = (g^{z_2 z_4})^{\theta_{0,0}+\mathsf{ID}^*\theta_{0,1}+\mathsf{ID}^{*\prime}\theta_{0,2}},$$
$$\{c^*_{i,(a)} = (g^{s_i})^{a_i(\hat\theta_{i,0}+\mathsf{ID}^*\hat\theta_{i,1}+\mathsf{ID}^{*\prime}\hat\theta_{i,2})}, c^*_{i,(b)} = (Tg^{-s_i})^{b_i(\hat\theta_{i,0}+\mathsf{ID}^*\hat\theta_{i,1}+\mathsf{ID}^{*\prime}\hat\theta_{i,2})}\}_{1\le i\le 3},$$
$$c^*_2 \leftarrow \mathbb{G},\ K^* \leftarrow \mathbb{G}_T.$$

Finally, $\mathcal{B}$ sends the challenge ciphertext $C^* = (c^*_0, \{c^*_{i,(a)}, c^*_{i,(b)}\}_{0\le i\le 3}, c^*_2)$ and session key K^* to the adversary $\mathcal{A}$. Observe that, if $T = g^{z_3+z_4}$, then $\{c^*_{i,(a)}, c^*_{i,(b)}\}_{0\le i\le 3}$ can be written as $\{(g_{i,0}g_{i,1}^{\mathsf{ID}^*}g_{i,2}^{\mathsf{ID}^{*\prime}})^{s_i}$, $(h_{i,0}h_{i,1}^{\mathsf{ID}^*}h_{i,2}^{\mathsf{ID}^{*\prime}})^{s-s_i}\}_{0\le i\le 3}$, where $s = z_3 + z_4$ and $s_0 = z_3$. (Recall that, $a_0 = z_1, b_0 = z_2$ and $\{\tilde\theta_{i,0} + \mathsf{ID}^*\tilde\theta_{i,1} + \mathsf{ID}^{*\prime}\tilde\theta_{i,2} = 0\}_{1\le i\le 3}$.) On the other hand, when T is a random element of $\mathbb{G}$, $c^*_{0,(a)}, c^*_{0,(b)}$ are distributed uniformly in $\mathbb{G}$, and $\{c^*_{i,(a)}, c^*_{i,(b)}\}_{1\le i\le 3}$ can be written as $\{(g_{i,0}g_{i,1}^{\mathsf{ID}^*}g_{i,2}^{\mathsf{ID}^{*\prime}})^{s_i}$, $(h_{i,0}h_{i,1}^{\mathsf{ID}^*}h_{i,2}^{\mathsf{ID}^{*\prime}})^{s-s_i}\}_{1\le i\le 3}$, where $s = \log_g T$.

To sum up, if $T = g^{z_3+z_4}$, then $\mathcal{B}$ has properly simulated Game_{-1}. If T is a random element of $\mathbb{G}$, then $\mathcal{B}$ has properly simulated Game_0. Hence, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish between two possibilities for T.

Note that, one can prove the computational indistinguishability of Game_{k-1} and Game_k for each $0 < k \le 3$ almost exactly as the above one, by exchanging the roles played by $\{g_{0,j}, h_{0,j}\}_{0\le j\le 2}$ with those played by $\{g_{k,j}, h_{k,j}\}_{0\le j\le 2}$ in the simulation, and taking case of the ramifications, *etc.* Specifically, a_0, b_0 will now be chosen by $\mathcal{B}$, whereas the given instance of the DLN problem will implicitly define $a_k = z_1$ and $b_k = z_2$. □