Identity-Based Matchmaking Encryption with Enhanced Privacy Against Chosen-Ciphertext Attacks

Sohto Chiku[1], Keitaro Hashimoto[2], Keisuke Hara[3], and Junji Shikata[1]

[1] Yokohama National University, Japan
Corresponding author: chiku-sohto-tw@ynu.jp
shikata-junji-rb@ynu.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST), Japan
keitaro.hashimoto@aist.go.jp
[3] The University of Osaka, Japan
hara-keisuke@ist.osaka-u.ac.jp

July 6, 2026

Abstract. Identity-based matchmaking encryption (IB-ME) proposed by Ateniese et al. [CRYPTO 2019; Journal of Cryptology 2021] enables users to communicate privately, anonymously, and authentically. Following the seminal work of Ateniese et al., considerable research has been conducted on the security and construction of IB-ME schemes, but no scheme with enhanced privacy against chosen-ciphertext attacks is known. In this paper, we propose two IB-ME schemes that simultaneously achieve enhanced privacy against chosen-ciphertext attacks. The first scheme is built on the bilinear Diffie–Hellman assumption in the random oracle model. Inspired by the scheme of Ateniese et al., our construction attains stronger security guarantees while providing more compact decryption keys and ciphertexts than the scheme of Ateniese et al. The second scheme is a generic construction built upon anonymous identity-based encryption, digital signatures, NIZK systems, and reusable extractors. This scheme is obtained by generalizing the specific scheme of Francati et al. [INDOCRYPT 2021] with several technique to solve CCA-related issues. It constitutes the first IB-ME scheme achieving the stronger security notions in the standard model. As an additional contribution, we classify existing authenticity notions for IB-ME into four categories—no message attacks (NMA) and chosen message attacks (CMA), each considered against both insiders and outsiders. This taxonomy enables a precise comparison among existing IB-ME schemes.

Keywords: Identity-Based Matchmaking Encryption · Chosen-Ciphertext Security · Pairing-Based Cryptography · Generic Construction.

Table of Contents

Identity-Based Matchmaking Encryption with Enhanced Privacy Against Chosen-Ciphertext Attacks 1
Sohto Chiku, Keitaro Hashimoto, Keisuke Hara, and Junji Shikata
1 Introduction . . . 3
1.1 Background . . . 3
1.2 Our Contributions . . . 3
1.3 Related Work . . . 5
1.4 Organization of This Paper . . . 6
2 Preliminaries . . . 6
2.1 Notation . . . 6
2.2 Asymmetric Bilinear Groups . . . 6
2.3 Identity-Based Encryption . . . 7
2.4 Non-Interactive Zero-Knowledge Proof Systems . . . 7
2.5 Reusable Extractors . . . 9
3 Identity-Based Matchmaking Encryption . . . 9
3.1 Syntax . . . 9
3.2 Security Notions . . . 10
4 Proposed IB-ME Scheme from BDH in the ROM . . . 12
4.1 Construction . . . 12
4.2 Security Proof . . . 13
4.3 Comparison . . . 25
5 New Generic Construction of IB-ME Scheme in the StdM . . . 25
5.1 Construction . . . 25
5.2 Security Proofs . . . 26

1 Introduction

1.1 Background

Identity-based matchmaking encryption (IB-ME), introduced by Ateniese et al. [3,4], is a novel identity-based cryptographic primitive that guarantees confidential and authenticated message delivery while anonymizing both the sender and the receiver. Similarly to conventional identity-based encryption (IBE) [6], a key generation center issues users' secret keys corresponding to their identities. In the IB-ME setting, both the sender and the receiver possess secret keys. Specifically, when a sender with identity σ transmits a message, it encrypts the message using its (secret) encryption key ek_σ together with the identity of the intended receiver rcv. Upon receiving the ciphertext, a receiver with identity ρ decrypts it using its secret decryption key dk_ρ while specifying the identity of the intended sender snd. Decryption succeeds if and only if the identities match, i.e., $\sigma = \mathsf{snd}$ and $\rho = \mathsf{rcv}$. If the identities do not match (i.e., $\sigma \neq \mathsf{snd}$ or $\rho \neq \mathsf{rcv}$), nothing is revealed beyond the fact that the identities are inconsistent. IB-ME has numerous practical applications, such as secret-handshake protocols and privacy-preserving bulletin boards.

Ateniese et al. [3, 4] defined *privacy* and *authenticity* as the two fundamental security requirements for IB-ME. Privacy ensures the confidentiality of messages against unintended receivers lacking the legitimate decryption key, while authenticity guarantees that only legitimate senders—those in possession of the correct encryption key—can generate valid ciphertexts, thereby preventing impersonation. Conceptually, privacy (resp. authenticity) corresponds to the semantic security of encryption schemes (resp. the unforgeability of signature schemes). Ateniese et al. also presented the first IB-ME scheme, based on the Boneh–Franklin IBE [6], which is secure under the bilinear Diffie–Hellman (BDH) assumption in the random oracle model (ROM). The initial work considered only a weaker form of privacy against chosen-plaintext attacks (CPA), so subsequent studies have explored stronger security notions. Chiku et al. [12] investigated privacy against chosen-ciphertext attacks (CCA), where the adversary has access to a decryption oracle on adaptively chosen ciphertexts. They proposed a hierarchical IB-ME scheme extending the IB-ME construction of Wang et al. [30], and then applied the Canetti–Halevi–Katz (CHK) transformation [9] to achieve CCA security. Lin et al. [23] concurrently considered CCA security in IB-ME and proposed another CCA-secure IB-ME scheme derived from a specific HIBE construction under the SXDH assumption, also using the CHK transformation.

Francati et al. [17] identified a gap between the privacy requirement IB-ME aims to achieve and Ateniese et al.'s definition of privacy. In particular, the original definition fails to ensure confidentiality when the receiver specifies an incorrect sender identity during decryption. That is, if $\mathsf{rcv} = \rho$ but $\mathsf{snd} \neq \sigma$, confidentiality is not formally guaranteed.[4] To address this limitation, Francati et al. introduced a new security notion, called *enhanced privacy*, which also accounts for such mismatched sender identities. They further proposed an IB-ME scheme achieving enhanced privacy based on Gentry's anonymous IBE [19], whose security relies on the non-standard q-augmented bilinear Diffie–Hellman exponent assumption. To achieve enhanced privacy from lattice-based assumptions, Boyen and Li [7] proposed an IB-ME scheme built from IBE, identity-based signatures (IBS), and reusable extractors. More recently, Belfiore et al. [5] proposed another generic construction of IB-ME with enhanced privacy, combining IBE with homomorphic signatures and reusable extractors.

Collectively, these works demonstrate that IB-ME can be constructed under various assumptions. However, existing schemes either provide only enhanced privacy against chosen-plaintext attacks or achieve CCA security but lack enhanced privacy. This observation naturally raises the following research question:

Q: Can we construct an IB-ME scheme that provides enhanced privacy against chosen-ciphertext attacks?

1.2 Our Contributions

We affirmatively answer the above research question. We propose two IB-ME schemes that achieve enhanced privacy against chosen-ciphertext attacks. The first scheme is a highly efficient IB-ME scheme from the

[4] As noted in [17], Ateniese et al. were aware of this issue and informally argued that their scheme preserves confidentiality in such cases.

Table 1: Comparison between our IB-ME schemes and the existing schemes. Here, CPA^+ and CCA^+ denote enhanced privacy under chosen-plaintext and chosen-ciphertext attacks, respectively. (Re)Ext stands for (reusable) randomness extractors, and HomSig stands for homomorphic signatures.

Schemes	Security properties		Assumptions	Model
	Privacy	Authenticity		
Ateniese et al. [3]	CPA	oNMA	BDH	ROM
Francati et al. [17]	CPA^+	iNMA	q-ABDHE, SIG, NIZK, ReExt	StdM
Chen et al. [11]	CPA	iNMA	SXDH	StdM
Wang et al. [30]	CPA	iCMA	Anon HIBE, IBS	StdM
Chiku et al. [12]	CCA	iCMA	Anon HIBE, HIBS, OTS	StdM
Lin et al. [23]	CCA	iNMA	SXDH	StdM
Boyen and Li [7]	CPA^+	iCMA	Anon IBE, IBS, ReExt, Ext	StdM
Belfiore et al. [5]	CPA^+	iNMA	Anon IBE, HomSig, ReExt	StdM
Ours (§ 4)	CCA^+	oCMA	BDH	ROM
Ours (§ 5)	CCA^+	iCMA	Anon IBE, SIG, NIZK, ReExt	StdM

BDH assumption in the ROM, and the second scheme is a new generic construction based on an anonymous identity-based encryption, digital signature scheme, NIZK system, and reusable extractors. A main technical difficulty in achieving CCA enhanced privacy is that the decryption oracle may be queried on the challenge ciphertext under mismatched sender or receiver identities. Such queries are not covered by ordinary privacy notions and can invalidate a direct CCA upgrade of CPA-enhanced-private constructions. In particular, reusable extractor security does not rule out degenerate or colliding behavior on fixed low-entropy identities, and hence a challenge ciphertext must not be reusable under a different claimed sender identity. Our constructions address this issue by ensuring that a ciphertext is bound to the intended sender-receiver pair: in the ROM construction this is achieved through the FO-style re-encryption check, while in the standard-model generic construction it is achieved by embedding the sender and receiver identities into the plaintext encrypted by the underlying IBE scheme and checking them during decryption.

The comparison of our schemes and the existing ones is summarized in Table 1.

New IB-ME scheme from BDH in the ROM. We construct an improved IB-ME scheme from the BDH assumption in the ROM. Our basic idea is to combine the Boneh-Franklin IBE scheme [6] and the Sakai-Ohgishi-Kasahara IB-NIKE scheme [28]. At a high level, a sender with identity σ has an IB-NIKE key $\mathsf{H}(\sigma)^{\mathsf{msk}}$ as its encryption key and a receiver with identity ρ has an IB-NIKE key $\mathsf{H}(\rho)^{\mathsf{msk}}$ and an IBE key $\mathsf{H}(\rho)^{\mathsf{msk}'}$ as its decryption key, where H is (appropriate) hash function, and msk (resp., msk') is a master secret key of the IB-NIKE scheme (resp., the IBE scheme). When the sender σ encrypts a message m to target a receiver rcv, it computes a ciphertext as $(g^r, \mathsf{m} \oplus \hat{\mathsf{H}}(e(X^r, \mathsf{H}(\mathsf{rcv})), e(\mathsf{H}(\sigma)^{\mathsf{msk}}, \mathsf{H}(\mathsf{rcv}))))$, where g is a generator (of the underlying group), $X = g^{\mathsf{msk}'}$ is a public parameter of the IBE scheme, and e is a symmetric pairing. To reduce the key size, we reuse the same master secret key for the IBE part and the IB-NIKE part. That is, we use the key $\mathsf{H}(\mathsf{id})^{\mathsf{msk}}$ for both the IBE scheme and the IB-NIKE scheme, where id is an identity for either the sender or the receiver. This reduces the size of a user's secret key, but weakens the security level since the compromise of a user leaks both encryption and decryption keys. To overcome this problem, we separate the domains of senders' and receivers' keys by employing asymmetric pairings. Using different hash functions H_1 and H_2, we compute the key of a sender σ as $\mathsf{H}_1(\sigma)^{\mathsf{msk}} \in \mathbb{G}_1$ and the key of a receiver ρ as $\mathsf{H}_2(\rho)^{\mathsf{msk}} \in \mathbb{G}_2$. This allows us to reduce the key size without weakening security. Intuitively, privacy is followed by the security of the IBE scheme, and authenticity is followed by the security of the IB-NIKE scheme[5]. To make the scheme secure against chosen-ciphertext attacks, we employ the Fujisaki-Okamoto (FO) transformation [18]. As a by-product, the re-encryption check introduced by the FO transformation also yields authenticity without additional cryptographic components. Moreover, we formally prove that our scheme also achieves enhanced privacy by utilizing a random oracle. As a result, we get a highly efficient

[5] Due to the bilinearity in the IB-NIKE part, the authenticity only holds when both sender and receiver are not compromised, i.e., authenticity only holds against outsiders. This is also the case in the work by Ateniese et al.

and strongly secure IB-ME scheme from the BDH assumption in the ROM. Both encryption and decryption keys contain only one group element, and the ciphertext contains one group element and a λ-bit string, both of which are smaller than those of the Ateniese et al. scheme. See Section 4 for more details.

New generic construction of IB-ME in the StdM. We propose a new generic construction of IB-ME from an anonymous IBE, a signature scheme (SIG), reusable extractors (Ext), and a non-interactive zero-knowledge proof system (NIZK). Toward our new construction, we first observe that Francati et al.'s construction [17] is based on Gentry's IBE [19]. They first construct an IB-ME scheme without authenticity from IBE and Ext. Roughly, in their construction, the sender first chooses a seed for Ext and masks (a part of) ciphertext of Gentry's IBE with Ext's random output. To add authenticity, the key generation center (KGC) issues a signature (i.e., certificate) for each sender's identity with the KGC's signature signing key, and the sender sends its certificate along with IBE ciphertexts. Here, to hide the sender's information to realize enhanced privacy, NIZK is used to show that the received ciphertext is from the intended sender without leaking any information. We observe that this idea is applicable to any IBE scheme, and NIZK is also useful for realizing CCA security. Although our construction follows the high-level paradigm of Francati et al. [17], a direct replacement of their NIZK proof system by a simulation-sound extractable one is not sufficient for CCA enhanced privacy. In the CCA setting, an adversary can submit the challenge ciphertext to the decryption oracle under different claimed identities. Since reusable extractor security only guarantees pseudorandomness for high-min-entropy inputs and does not prevent pointwise collisions on fixed identities, such identity-switching decryption queries require an additional binding mechanism. We therefore embed both the sender and receiver identities into the plaintext of the underlying IBE ciphertext and require Dec to check that the recovered identities match the identities supplied to the decryption algorithm. This identity binding, together with simulation-sound extractability of NIZK, is the key additional ingredient that enables our CCA proof. See Section 5 for more details.

It is worth noting that the first construction is not intended as an instantiation of the second construction. The two constructions realize the same high-level goal by different techniques and with different trade-offs. The ROM construction is optimized for compactness and relies on random oracles and pairing-based algebraic structure. The generic construction is a standard-model feasibility result relying on IBE, signatures, NIZK, and reusable extractors.

Reorganize authenticity notions of IB-ME. As a side contribution, we reorganize the authenticity notions in previous works. We notice that the existing definitions can be classified along two points: one is whether an adversary has access to the encryption oracle, and the other is whether it can compromise the target receiver's secret key. For the former point, we refer to the respective attacks as chosen message attacks (CMA) and no message attacks (NMA) based on the presence or absence of access to the encryption oracle. For the latter point, we call the adversary who compromises the target receiver *insiders* and otherwise *outsiders* since we can regard the adversary, who knows the receiver's key, as inside the communication.[6] As a result, we define four authenticity notions oNMA, iNMA, oCMA, and iCMA[7] (Table 1 shows their correspondence with the previous works). We believe this taxonomy enables a precise comparison among existing IB-ME schemes.

1.3 Related Work

Identity-based encryption. Identity-based encryption, proposed by Shamir [29], is an encryption scheme that allows users to use arbitrary strings (e.g., e-mail addresses) as their public keys. After quite a long time, Boneh and Franklin constructed the first IBE scheme [6] using bilinear pairings, and then many IBE schemes have been proposed from various assumptions [1, 14, 19, 21, 22, 31, 32]. In IBE, the sender specifies only the receiver's identity, but in IB-ME, the sender specifies not only the receiver's identity but also the sender's identity.

Identity-based signcryption. Signcryption [33] is a cryptographic primitive that offers private and authenticated delivery of messages. The motivation for signcryption is to provide equivalent functionality more

[6] Here, we employ the naming used in a similar situation in signcryption [25]

[7] The prefix o (resp. i) indicates the adversary is an outsider (resp. insider).

efficiently than a simple combination of encryption and signature schemes. The notion of identity-based signcryption (IB-SC) was proposed by Malone-Lee [24]. The difference between IB-ME and IB-SC is that the former ensures the anonymity of communicating users and the confidentiality of messages when ciphertexts are decrypted with mismatched sender identities. Therefore, IB-ME provides better security properties than IB-SC.

(General) Matchmaking encryption. Ateniese et al. proposed matchmaking encryption [3,4]. In ME setting, the sender and the receiver have their own attribute, and they can specify access policies the other party must satisfy. Ateniese et al. also gave generic constructions of ME based on functional encryption, signature scheme, and non-interactive zero-knowledge. Francati et al. [15,16] proposed a simple ME scheme based on two-key predicate encryption. Note that IB-ME is an ME supporting the policy of identity equivalence.

1.4 Organization of This Paper

The remaining part of this paper is organized as follows. In Section 2, we introduce notations and definitions of the cryptographic primitives that will be used in this paper. Then, in Section 3, we give the relevant definitions including syntax and security definitions of IB-ME. Section 4 shows an efficient and strongly secure IB-ME scheme based on BDH assumption in the ROM, and presents a comparison between our IB-ME schemes and the existing schemes. Finally, in Section 5, we provide a new generic construction of IB-ME based on IBE, signature scheme, and NIZK in the StdM.

2 Preliminaries

In this section, we first define some notations used in this work. Then we recall asymmetric bilinear groups, identity-based encryption, identity-based signature, and reusable computational extractors.

2.1 Notation

$\mathbb{N}$ denotes the set of positive integers. $\emptyset$ denotes the empty set. $\hat{e}$ denotes the base of the natural logarithm. PPT stands for probabilistic polynomial time. For $n \in \mathbb{N}$, we denote $[n] \coloneqq \{1, 2, \ldots, n\}$. $x \coloneqq y$ denotes that x is defined by y. $y \leftarrow \mathcal{A}(x; r)$ denotes that a PPT algorithm $\mathcal{A}$ outputs y on input x and randomness r. We simply denote $y \leftarrow \mathcal{A}(x)$ when $\mathcal{A}$ uses uniform randomness. $\mathcal{A}^{\mathcal{O}}$ means $\mathcal{A}$ has oracle access to a function $\mathcal{O}(\cdot)$. $\mathsf{poly}(\lambda)$ denotes a polynomial in λ. We say that a function $f(\lambda)$ is negligible in λ if $f(\lambda) = o(1/\lambda^c)$ for every $c \in \mathbb{Z}$, and we write $\mathsf{negl}(\lambda)$ to denote a negligible function in λ. $x \leftarrow\!\!\$\ \mathcal{X}$ denotes that an element x is sampled uniformly at random from a finite set $\mathcal{X}$. Let X be a distribution over $\mathcal{X}$. The min-entropy of X is defined as $\mathrm{H}_\infty(X) \coloneqq -\log \max_{x \in \mathcal{X}} \Pr[X = x]$. We call a distribution with min-entropy κ κ-distribution. $x \leftarrow\!\!\$\ X$ denotes an element $x \in \mathcal{X}$ is sampled following the distribution X.

2.2 Asymmetric Bilinear Groups

We recall (asymmetric) bilinear groups[8] and the bilinear Diffie-Hellman (BDH) assumption from [8]. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be groups of prime order p. Let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be respective generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be an efficiently computable function that satisfies (1) for any $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $\alpha, \beta \in \mathbb{Z}_p$, $e(u^\alpha, v^\beta) = e(u, v)^{\alpha\beta}$ (i.e., bilinearity) and (2) $e(g_1, g_2) \neq 1$, where 1 is the unit element in $\mathbb{G}_T$ (i.e., non-degeneracy). This function e is called a *bilinear map* or *pairing*. We call $G \coloneqq (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ a bilinear group. We define bilinear group generators that generate a bilinear group corresponding to the input security parameter.

Definition 1 (Bilinear Group Generator). *A* bilinear group generator $\mathcal{G}$ *is a* PPT *algorithm that, on input* 1^λ*, outputs the description of a bilinear group* $G = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$.

[8] This work only uses asymmetric bilinear groups. So, we omit the term "asymmetric".

We define the BDH assumption for $\mathcal{G}$.

Definition 2 (Bilinear Diffie-Hellman (BDH) Assumption [8]). *Let $\mathcal{G}$ be a bilinear group generator. We say that* BDH assumption holds for $\mathcal{G}$ *if for all* PPT *adversaries $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{A},\mathcal{G}}(\lambda) \coloneqq \Pr\left[D = e(g_1,g_2)^{\alpha\beta\gamma} \;\middle|\; \begin{array}{c} G \coloneqq (p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,g_1,g_2,e) \leftarrow \mathcal{G}(1^\lambda), \\ \alpha,\beta,\gamma \leftarrow\!\!\$\, \mathbb{Z}_p, \\ D \leftarrow \mathcal{A}(G, g_1^\alpha, g_2^\alpha, g_2^\beta, g_1^\gamma) \end{array}\right] = \mathsf{negl}(\lambda).$$

2.3 Identity-Based Encryption

Syntax. An IBE scheme IBE consists of the following four algorithms.

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**:** The setup algorithm takes the security parameter 1^λ, and outputs a public parameter mpk and a master secret key msk. mpk defines the identity space $\mathcal{ID}$, the message space $\mathcal{M}$, and the ciphertext space $\mathcal{CT}$.

$\mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}) \to \mathsf{sk}_{\mathsf{id}}$**:** The key generation algorithm takes mpk, msk and an identity $\mathsf{id} \in \mathcal{ID}$ as input and outputs a secret key $\mathsf{sk}_{\mathsf{id}}$.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mathsf{m}) \to \mathsf{ct}$**:** The encryption algorithm takes mpk, $\mathsf{id} \in \mathcal{ID}$, and a plaintext $\mathsf{m} \in \mathcal{M}$ as input, and outputs a ciphertext $\mathsf{ct} \in \mathcal{CT}$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}) \to \mathsf{m}$ **or** $\perp$**:** The decryption algorithm takes mpk, $\mathsf{sk}_{\mathsf{id}}$, and ct as input, and outputs $\mathsf{m} \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$.

Correctness. We say that an IBE scheme IBE is *correct* if for all $\lambda \in \mathbb{N}$, $\mathsf{id} \in \mathcal{ID}$ and $\mathsf{m} \in \mathcal{M}$, it holds that

$$\Pr\left[\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct}) = \mathsf{m} \;\middle|\; \begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda), \\ \mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}), \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}, \mathsf{m}) \end{array}\right] = 1.$$

Security. We recall adaptive-identity anonymity against chosen-plaintext attacks ($\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}$ security) for IBE.

Definition 3 ($\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}$ Security of IBE). *We say that an IBE scheme* IBE *is* $\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}$ secure *if for all* PPT *adversaries $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{A},\mathsf{IBE}}(\lambda) \coloneqq \left|\Pr\left[\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}^{\mathcal{A}}_{\mathsf{IBE}}(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{negl}(\lambda),$$

where the security game $\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}^{\mathcal{A}}_{\mathsf{IBE}}(\lambda)$ *is depicted in Fig. 1.*

2.4 Non-Interactive Zero-Knowledge Proof Systems

Syntax. A non-interactive zero-knowledge (NIZK) proof system for a corresponding relation R consists of the following algorithms:

$\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$**:** The setup algorithm takes the security parameter 1^λ, and outputs a common reference string crs.

$\mathsf{Prove}(\mathsf{crs}, x, w) \to \pi$**:** The proving algorithm takes crs, an instance x, and a witness w, and outputs a proof π.

$\mathsf{Verify}(\mathsf{crs}, x, \pi) \to 0$ **or** 1**:** The verification algorithm takes crs, x and π, and outputs a bit $b \in \{0,1\}$.

$\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}^{\mathcal{A}}_{\mathsf{IBE}}(\lambda)$	Oracle $\mathcal{O}_{SK}(\mathsf{id})$
$\mathsf{id}^* \coloneqq \bot; \mathcal{L}_{SK} \coloneqq \emptyset$	**if** $\mathsf{id} = \mathsf{id}^*$ **then**
$\mathsf{coin} \leftarrow\!\!\$\ \{0,1\}$	**return** $\bot$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$
$(\mathsf{id}^*, \mathsf{m}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{SK}}(\mathsf{mpk})$	$\mathcal{L}_{SK} \leftarrow \mathcal{L}_{SK} \cup \{\mathsf{id}\}$
if $\mathsf{id}^* \in \mathcal{L}_{SK}$ **then**	**return** $\mathsf{sk}_{\mathsf{id}}$
return coin	
$\mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*, \mathsf{m}^*)$	
$\mathsf{ct}_1 \leftarrow \mathcal{CT}$	
$\widehat{\mathsf{coin}} \leftarrow \mathcal{A}^{\mathcal{O}_{SK}}(\mathsf{ct}_{\mathsf{coin}})$	
if $\mathsf{coin} = \widehat{\mathsf{coin}}$ **then**	
return 1	
else	
return 0	

Fig. 1: The security game for IBE.

Correctness. We say that an NIZK proof system NIZK is correct if for all $\lambda \in \mathbb{N}$, and x, w such that $R(x, w)$, it holds that

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1 \;\middle|\; \begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), \\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w) \end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

Security. We recall the zero-knowledge property and the knowledge extractability.

Definition 4 (Zero-Knowledge of NIZK). *We say that an NIZK proof system* NIZK *for a relation* R *satisfies* zero-knowledge *if there exists efficient simulator* $\mathcal{S} \coloneqq (\mathcal{S}_1, \mathcal{S}_2)$ *such that for all* PPT *adversaries* $\mathcal{A}$*,*

$$\begin{aligned} &\mathsf{Adv}^{\mathsf{zk}}_{\mathcal{A},\mathsf{NIZK}}(\lambda) \coloneqq \\ &\left|\Pr\left[\mathcal{A}^{\mathcal{O}(\mathsf{crs},\cdot,\cdot)}(\mathsf{crs}) \;\middle|\; \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)\right] - \Pr\left[\mathcal{A}^{\mathcal{O}'(\mathsf{crs},\tau,\cdot,\cdot)}(\mathsf{crs}) \;\middle|\; (\mathsf{crs}, \tau) \leftarrow \mathcal{S}_1(1^\lambda)\right]\right| \\ &= \mathsf{negl}(\lambda), \end{aligned}$$

where the oracles $\mathcal{O}$ *and* $\mathcal{O}'$ *are defined as follows:*

- $\mathcal{O}(\mathsf{crs}, \cdot, \cdot)$*: On input* (x, w)*, it outputs* $\mathsf{Prove}(\mathsf{crs}, x, w)$ *if* $(x, w) \in R$*, and* $\bot$ *otherwise.*
- $\mathcal{O}'(\mathsf{crs}, \tau, \cdot, \cdot)$*: On input* (x, w)*,* $\mathcal{O}'$ *outputs* $\mathcal{S}_2(\mathsf{crs}, \tau, x)$ *if* $(x, w) \in R$*, and* $\bot$ *otherwise.*

Definition 5 (Simulation-Sound Extractability of NIZK). *We say that an NIZK proof system* NIZK *for a relation* R *satisfies* simulation-sound extractability *if there exists an extraction algorithm* $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ *such that the following holds:*

- *The output of* $\mathcal{E}_1(1^\lambda)$ *is identically distributed as* $\mathcal{S}_1(1^\lambda)$ *when restricted to the first two components* (crs, τ) *.*
- *For all* PPT *adversaries* $\mathcal{A}$*,*

$$\begin{aligned} \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{A},\mathsf{NIZK}}(\lambda) &\coloneqq \Pr\left[(x, \pi) \notin \mathcal{L} \wedge (x, w) \notin R \wedge \mathsf{Verify}(\mathsf{crs}, x, \pi) = 1 \;\middle|\; \begin{array}{l} (\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\lambda), \\ (x, \pi) \leftarrow \mathcal{A}^{\mathcal{O}'(\mathsf{crs},\tau,\cdot,\cdot)}(\mathsf{crs}), \\ w \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, x, \pi) \end{array}\right] \\ &= \mathsf{negl}(\lambda), \end{aligned}$$

where $\mathcal{L}$ *is the set containing the queries* $\mathcal{A}$ *makes to* $\mathcal{O}'$ *and their responses, in the form of query response pairs.*

2.5 Reusable Extractors

Let $\mathsf{seedLen} = \mathsf{poly}(\lambda)$ be an integer and $\mathsf{Ext} : \{0,1\}^{\mathsf{seedLen}} \times \mathcal{X} \to \mathcal{Y}$ be an efficient computable function that on input a seed $s \in \{0,1\}^{\mathsf{seedLen}}$ and a value $x \in \mathcal{X}$ outputs $y \in \mathcal{Y}$. Intuitively, we say that Ext is an extractor if $y = \mathsf{Ext}(s,x)$ is pseudorandom when s is sampled uniformly at random from $\{0,1\}^{\mathsf{seedLen}}$ and x is sampled from a k-distribution X (defined over $\mathcal{X}$) for appropriate k, even if the seed s is made public. We consider *reusable* extractors [13], which produce pseudorandom outputs even when the same input is evaluated multiple times with different seeds. The formal definition is provided below.

Definition 6 (Reusable Extractors). *We say that* $\mathsf{Ext} : \{0,1\}^{\mathsf{seedLen}} \times \mathcal{X} \to \mathcal{Y}$ *is a* (κ, n)-reusable extractor *if for any distribution* X *over* $\mathcal{X}$ *such that* $\mathrm{H}_\infty(X) \geq \kappa$*, for all* PPT *adversaries* $\mathcal{A}$*, it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ext}}_{\mathcal{A},\mathsf{Ext}}(\lambda) \coloneqq \quad & \Big|\Pr\Big[1 \leftarrow \mathcal{A}\left(1^\lambda, \{(s_i, \mathsf{Ext}(s_i,x)\}_{i\in[n]}\right) \;\Big|\; s_i \leftarrow\!\$\; \{0,1\}^{\mathsf{seedLen}}, x \leftarrow\!\$\; X\Big] \\ & - \Pr\Big[1 \leftarrow \mathcal{A}\left(1^\lambda, \mathsf{aux}, \{(s_i, y_i)\}_{i\in[n]}\right) \;\Big|\; s_i \leftarrow\!\$\; \{0,1\}^{\mathsf{seedLen}}, y_i \leftarrow\!\$\; \mathcal{Y}\Big]\Big| \\ = \; & \mathsf{negl}(\lambda).\end{aligned}$$

Note that reusable extractors can be constructed in the standard model [2, 13, 26]).

3 Identity-Based Matchmaking Encryption

In this section, we first recall the syntax and security definition of identity-based matchmaking encryption (IB-ME) defined by Ateniese et al. [3, 4] and enhanced by Francati et al. [17].

3.1 Syntax

An IB-ME scheme $\mathsf{IB\text{-}ME}$ consists of the following five algorithms.

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$**:** The setup algorithm takes the security parameter 1^λ, and outputs a public parameter mpk and master secret key msk. mpk defines the identity space $\mathcal{ID}$, the message space $\mathcal{M}$ and the ciphertext space $\mathcal{CT}$.

$\mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma) \to \mathsf{ek}_\sigma$**:** The sender key generation algorithm takes mpk, msk, and a sender's identity $\sigma \in \mathcal{ID}$ as input, and outputs an encryption key ek_σ.

$\mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho) \to \mathsf{dk}_\rho$**:** The receiver key generation algorithm takes mpk, msk, and a receiver's identity $\rho \in \mathcal{ID}$ as input and outputs a decryption key dk_ρ.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma, \mathsf{rcv}, \mathsf{m}) \to \mathsf{ct}$**:** The encryption algorithm takes mpk, ek_σ, a receiver's identity rcv, and a plaintext $\mathsf{m} \in \mathcal{M}$ as input and outputs a ciphertext $\mathsf{ct} \in \mathcal{CT}$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct}) \to \mathsf{m}$ **or** $\perp$**:** The decryption algorithm takes mpk, dk_ρ, a sender's identity snd, and ct as input and outputs $\mathsf{m} \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$.

Correctness. We say that an IB-ME scheme $\mathsf{IB\text{-}ME}$ is *correct* if for all $\lambda \in \mathbb{N}, \sigma, \rho, \mathsf{snd}, \mathsf{rcv} \in \mathcal{ID}$ such that $\mathsf{snd} = \sigma$ and $\mathsf{rcv} = \rho$, and $\mathsf{m} \in \mathcal{M}$, it holds that

$$\Pr\left[\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct}) = \mathsf{m} \;\middle|\; \begin{array}{l}(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda), \\ \mathsf{ek}_\sigma \leftarrow \mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma), \\ \mathsf{dk}_\rho \leftarrow \mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho), \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma, \mathsf{rcv}, \mathsf{m})\end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

We say that an IB-ME scheme is *perfectly correct* if the above probability is equal to 1 (i.e., no error occurs).

3.2 Security Notions

IB-ME schemes must satisfy two primary security properties: *privacy* and *authenticity*. In essence, privacy ensures that nothing is disclosed to unintended recipients who do not adhere to the sender's policy, while authenticity guarantees that it is impossible to impersonate the sender without possessing the sender's secret key.

Privacy. Ateniese et al. [3] introduced the notion of *privacy against chosen-plaintext attacks* (CPA privacy), which guarantees message confidentiality as long as the adversary does not possess the receiver's decryption key. However, Francati et al. [17] observed that this definition is insufficient for IB-ME schemes and proposed a stronger notion, called *enhanced privacy*. This notion guarantees confidentiality even in scenarios where not only the receiver may be compromised, but also the sender's identity is unknown to the adversary. Since then, enhanced privacy has been regarded as a desirable security requirement for IB-ME. We now recall the formal definition of enhanced privacy against chosen-plaintext attacks (CPA enhanced privacy), as introduced by Francati et al.

Definition 7 (Priv^+-CPA Security of IB-ME). *We say that an IB-ME scheme* $\mathsf{IB\text{-}ME}$ *is* Priv^+-CPA secure *if for all admissible* PPT *adversaries* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cpa}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) := \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right| = \mathsf{negl}(\lambda),$$

where the security game $\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$ *is depicted in Fig. 2. An adversary* $\mathcal{A}$ *is admissible if for all* $\rho \in \mathcal{L}_R$*, it satisfies at least one of the following invariants*

$$\begin{array}{ll} \textit{(Case 1)} & \rho \neq \mathsf{rcv}^*_0 \wedge \rho \neq \mathsf{rcv}^*_1, \\ \textit{(Case 2)} & H_\infty(\Sigma^*_0) \geq \kappa \wedge H_\infty(\Sigma^*_1) \geq \kappa, \\ \textit{(Case 3)} & \rho \neq \mathsf{rcv}^*_0 \wedge H_\infty(\Sigma^*_1) \geq \kappa, \\ \textit{(Case 4)} & \rho \neq \mathsf{rcv}^*_1 \wedge H_\infty(\Sigma^*_0) \geq \kappa. \end{array}$$

In this work, we consider *enhanced privacy against chosen-ciphertext attacks* (CCA enhanced privacy). To this end, we extend Francati et al.'s Priv^+-CPA security definition by allowing the adversary to access a decryption oracle, thereby formulating the notion of Priv^+-CCA security. It is worth noting that, in Case 2, CPA enhanced privacy and CCA enhanced privacy coincide: since the adversary may already possess any receiver's decryption key, access to a decryption oracle does not provide additional advantage.

Definition 8 (Priv^+-CCA Security of IB-ME). *We say that an IB-ME scheme* $\mathsf{IB\text{-}ME}$ *is* Priv^+-CCA secure *if for all admissible* PPT *adversaries* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) := \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda).$$

where the security game $\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$ *is depicted in Fig. 2. An adversary* $\mathcal{A}$ *is admissible if for all* $\rho \in \mathcal{L}_R$*, it satisfies at least one of the following invariants*

$$\begin{array}{ll} \textit{(Case 1)} & \rho \neq \mathsf{rcv}^*_0 \wedge \rho \neq \mathsf{rcv}^*_1, \\ \textit{(Case 2)} & H_\infty(\Sigma^*_0) \geq \kappa \wedge H_\infty(\Sigma^*_1) \geq \kappa, \\ \textit{(Case 3)} & \rho \neq \mathsf{rcv}^*_0 \wedge H_\infty(\Sigma^*_1) \geq \kappa, \\ \textit{(Case 4)} & \rho \neq \mathsf{rcv}^*_1 \wedge H_\infty(\Sigma^*_0) \geq \kappa. \end{array}$$

Remark 1. While the enhanced privacy considers four cases (Case 1 ,2, 3 and 4), we can prove that if an IB-ME scheme is secure in Case 3, then it is also secure in Case 4. Therefore, we do not need to prove the security in Case 4 in general. The detail is as follows. Let $\mathcal{A}_3$ be an adversary that breaks the security of the given IB-ME scheme in Case 3, we can construct an another adversary $\mathcal{A}_4$ that breaks the security in Case 4 by using $\mathcal{A}_3$. Given mpk, $\mathcal{A}_4$ runs $\mathcal{A}_3$ on input mpk. Then, $\mathcal{A}_4$ answers oracle queries from $\mathcal{A}_3$ with its given oracles.

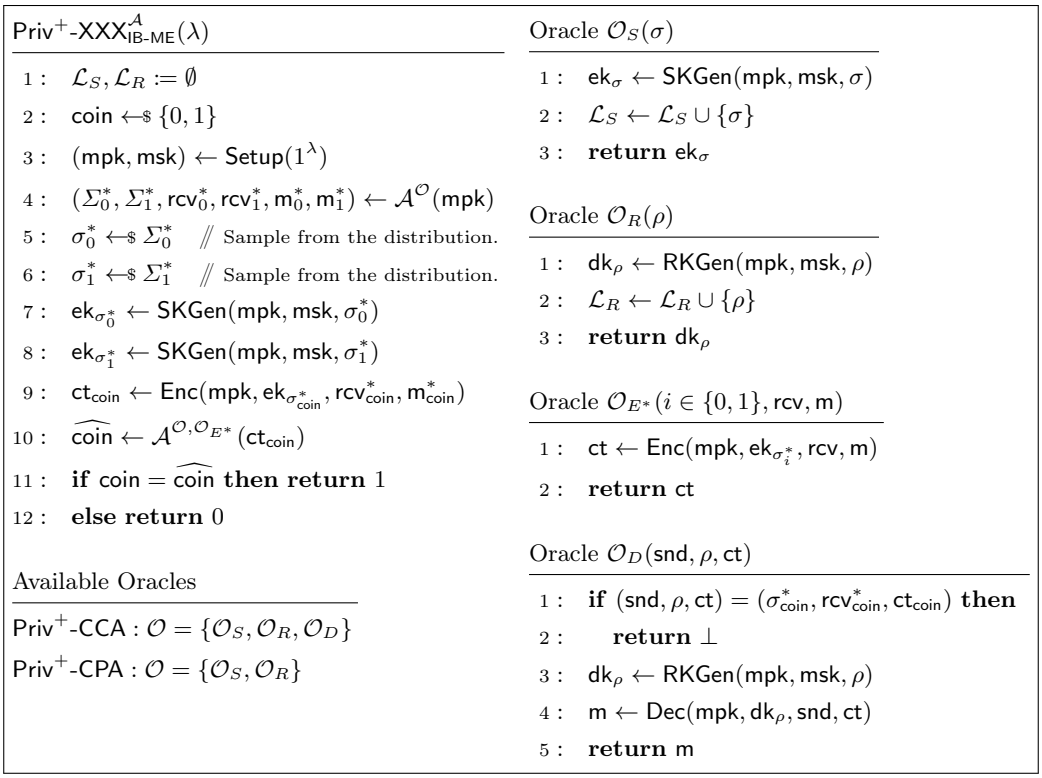


Fig. 2: The enhanced privacy game for IB-ME schemes.

Once $\mathcal{A}_3$ outputs the challenge $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$, $\mathcal{A}_4$ outputs $(\Sigma_1^*, \Sigma_0^*, \mathsf{rcv}_1^*, \mathsf{rcv}_0^*, \mathsf{m}_1^*, \mathsf{m}_0^*)$ as its own challenge. Then, $\mathcal{A}_4$ continues to answer oracle queries from $\mathcal{A}_3$. Note that when $\mathcal{O}_{E^*}(i \in \{0,1\}, \mathsf{rcv}, \mathsf{m})$ query is made, $\mathcal{A}_4$ sends $\mathcal{O}_{E^*}(i \oplus 1, \mathsf{rcv}, \mathsf{m})$ to its given $\mathcal{O}_{E^*}$. Finally, when $\mathcal{A}_3$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{A}_4$ outputs it as its own final output. It is clear that if $\mathcal{A}_3$ breaks the enhanced privacy in Case 3, $\mathcal{A}_4$ breaks the enhanced privacy in Case 4 with the same advantage.

Authenticity. Ateniese et al. [3] introduced a relatively weak notion of authenticity, which we refer to as *authenticity against no-message attacks from outsiders* (Auth-oNMA). Here, the term "outsiders" indicates that neither the target sender nor the target receiver is compromised. This notion was later strengthened to allow adversaries that may compromise the target receiver [11, 17]. Since such adversaries possess the receiver's secret key, we call them *insiders*, and the corresponding property *authenticity against no-message attacks from insiders* (Auth-iNMA). In this work, we further introduce *authenticity against chosen-message attacks from outsiders or insiders* (Auth-oCMA and Auth-iCMA, respectively). In the Auth-xCMA game, the adversary is granted access to an encryption oracle and may obtain ciphertexts corresponding to messages of its choice, analogous to the signing oracle in the unforgeability game for standard digital signatures. We remark that Auth-iCMA coincides with the notion of "strong authenticity" defined by Wang et al. [30], whereas Auth-oCMA is newly introduced in this paper. The formal definitions of these authenticity notions are given through the security games illustrated in Fig. 3.

Definition 9 (Auth-$\{\mathsf{o},\mathsf{i}\}$NMA **Security of IB-ME).** *Let* $\mathsf{x} \in \{\mathsf{o},\mathsf{i}\}$. *We say that an IB-ME scheme* IB-ME *is* Auth-xNMA secure *if for all* PPT *adversaries* $\mathcal{A}$, *it holds that*

$$\mathsf{Adv}^{\mathsf{auth\text{-}xnma}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) := \Pr\left[\mathsf{Auth\text{-}xNMA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] = \mathsf{negl}(\lambda),$$

where the security game $\mathsf{Auth\text{-}xNMA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$ *is depicted in Fig. 3.*

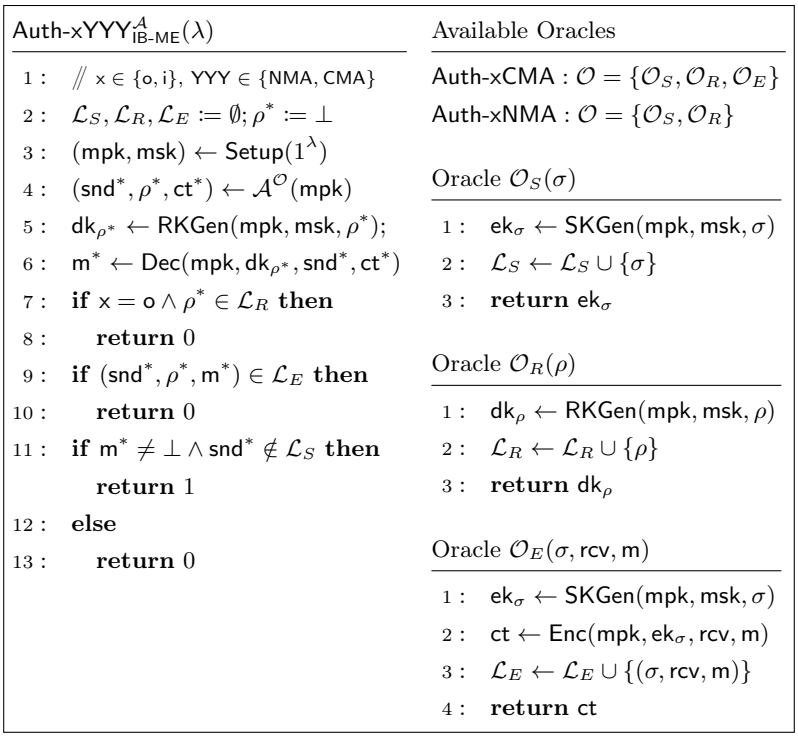


Fig. 3: The authenticity game for IB-ME schemes.

Definition 10 (Auth-{o, i}CMA Security of IB-ME). *Let* $\mathsf{x} \in \{\mathsf{o}, \mathsf{i}\}$. *We say that an IB-ME scheme* IB-ME *is* Auth-xCMA secure *if for all* PPT *adversaries* $\mathcal{A}$, *it holds that*

$$\mathsf{Adv}^{\mathsf{auth\text{-}xcma}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) := \Pr\left[\mathsf{Auth\text{-}xCMA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] = \mathsf{negl}(\lambda),$$

where the security game $\mathsf{Auth\text{-}xCMA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$ *is depicted in Fig. 3.*

Remark 2. It is worth noting that this distinction between insider and outsider adversaries is a well-established concept in the context of signcryption [25].

4 Proposed IB-ME Scheme from BDH in the ROM

This section shows an IB-ME scheme from the BDH assumption in the ROM, which satisfies CCA-enhanced privacy. Our idea is to combine the Boneh-Franklin IBE scheme [6] and the Sakai-Ohgishi-Kasahara IB-NIKE scheme [28]. We also introduce several optimizations to reduce secret key and ciphertext sizes. To achieve stronger security, we employ the FO transformation [18]. Interestingly, the FO transformation allows us to achieve not only CCA enhanced privacy at minimal cost but also Auth-oCMA security for free, since the re-encryption check of g^r acts as message authentication. As a result, we obtain a highly efficient and strongly secure IB-ME scheme compared to the scheme of Ateniese et al.

4.1 Construction

The proposed IB-ME scheme $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ is as follows. Its identity and message spaces are $\mathcal{ID} = \{0,1\}^*$ and $\mathcal{M} = \{0,1\}^{\mathsf{msgLen}}$, respectively.

$\mathsf{Setup}(1^\lambda)$**:** It first generates a bilinear group $G \coloneqq (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda)$ and selects hash functions $\mathsf{H}_1 : \{0,1\}^* \to \mathbb{G}_1$, $\mathsf{H}_2 : \{0,1\}^* \to \mathbb{G}_2$, $\hat{\mathsf{H}} : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_T \times \mathbb{G}_T \to \{0,1\}^{\mathsf{msgLen}+\lambda}$, and $\mathsf{G} : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^{\mathsf{msgLen}} \times \{0,1\}^\lambda \to \mathbb{Z}_p$. Then, it samples $x \leftarrow\!\!\$\ \mathbb{Z}_p$ and sets $X \coloneqq g_1^x$. Finally, it outputs $\mathsf{mpk} \coloneqq (G, \mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}, \mathsf{G}, X)$ and $\mathsf{msk} \coloneqq x$.

$\mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma)$**:** It computes $\mathsf{u}_\sigma \coloneqq \mathsf{H}_1(\sigma)$ and $\mathsf{sk}_\sigma \coloneqq \mathsf{u}_\sigma^x$. It outputs $\mathsf{ek}_\sigma \coloneqq (\sigma, \mathsf{sk}_\sigma)$.

$\mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho)$**:** It computes $\mathsf{u}_\rho \coloneqq \mathsf{H}_2(\rho)$ and $\mathsf{sk}_\rho \coloneqq \mathsf{u}_\rho^x$. It outputs $\mathsf{dk}_\rho \coloneqq (\rho, \mathsf{sk}_\rho)$.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma, \mathsf{rcv}, \mathsf{m})$**:** It picks $\mathsf{k} \leftarrow\!\!\$\ \{0,1\}^\lambda$ and computes $r \coloneqq \mathsf{G}(\sigma, \mathsf{rcv}, \mathsf{m}, \mathsf{k})$. Then, it computes $\mathsf{u}_{\mathsf{rcv}} \coloneqq \mathsf{H}_2(\mathsf{rcv})$ and

$$R \coloneqq g_1^r, \qquad \mathsf{ctxt} \coloneqq \mathsf{m}||\mathsf{k} \oplus \hat{\mathsf{H}}(\sigma, \mathsf{rcv}, R, e(X^r, \mathsf{u}_{\mathsf{rcv}}), e(\mathsf{sk}_\sigma, \mathsf{u}_{\mathsf{rcv}})).$$

Finally, it outputs $\mathsf{ct} \coloneqq (R, \mathsf{ctxt})$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct} = (R, \mathsf{ctxt}))$**:** It computes $\mathsf{u}_{\mathsf{snd}} \coloneqq \mathsf{H}_1(\mathsf{snd})$ and

$$\mathsf{m}||\mathsf{k} \coloneqq \mathsf{ctxt} \oplus \hat{\mathsf{H}}(\mathsf{snd}, \rho, R, e(R, \mathsf{sk}_\rho), e(\mathsf{u}_{\mathsf{snd}}, \mathsf{sk}_\rho)).$$

It then computes $r \coloneqq \mathsf{G}(\mathsf{snd}, \rho, \mathsf{m}, \mathsf{k})$ and checks if $R = g_1^r$. If so, it outputs m. Otherwise, it outputs $\perp$.

Correctness. We can verify that $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ is perfectly correct. For any $\lambda \in \mathbb{N}$, $(\mathsf{mpk}, \mathsf{msk}) \in \mathsf{Setup}(1^\lambda)$ and any $\sigma, \rho, \mathsf{snd}, \mathsf{rcv} \in \{0,1\}^*$ such that $\sigma = \mathsf{snd}$ and $\rho = \mathsf{rcv}$, we have

$$e(X^r, \mathsf{u}_{\mathsf{rcv}}) = e((g_1^x)^r, \mathsf{H}_2(\mathsf{rcv})) = e(g_1^r, \mathsf{H}_2(\rho)^x) = e(R, \mathsf{sk}_\rho),$$
$$e(\mathsf{sk}_\sigma, \mathsf{u}_{\mathsf{rcv}}) = e(\mathsf{H}_1(\sigma)^x, \mathsf{H}_2(\mathsf{rcv})) = e(\mathsf{H}_1(\mathsf{snd}), \mathsf{H}_2(\rho)^x) = e(\mathsf{u}_{\mathsf{snd}}, \mathsf{sk}_\rho).$$

That is, it holds that

$$\hat{\mathsf{H}}(\sigma, \mathsf{rcv}, R, e(X, \mathsf{u}_{\mathsf{rcv}})^r, e(\mathsf{sk}_\sigma, \mathsf{u}_{\mathsf{rcv}})) = \hat{\mathsf{H}}(\mathsf{snd}, \rho, R, e(R, \mathsf{sk}_\rho), e(\mathsf{u}_{\mathsf{snd}}, \mathsf{sk}_\rho)),$$

and thus the receiver recovers $\mathsf{m}||\mathsf{k}$ that the sender $\sigma = \mathsf{snd}$ encrypts. Thus, the receiver can recompute $r \coloneqq \mathsf{G}(\mathsf{snd}, \rho, \mathsf{m}, \mathsf{k})$ that satisfies $R = g_1^r$.

4.2 Security Proof

We can show that $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ is $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ and $\mathsf{Auth\text{-}oCMA}$ secure in the ROM.

Theorem 1. *Suppose the hash functions* H_1, H_2, $\hat{\mathsf{H}}$, *and* G *are random oracles. If there exists an adversary* $\mathcal{A}$ *that breaks the* $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ *security of* $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$*, there exists an adversary* $\mathcal{B}$ *that breaks the* BDH *assumption for* $\mathcal{G}$ *such that*

$$\mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cca}}_{\mathcal{A}, \mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda)$$
$$\leq \begin{cases} 3\hat{e}^2(1+q_R)^2 q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda) + \dfrac{q_D}{p} + \dfrac{3q_{\mathsf{G}}}{2^\lambda} & (\textit{Case 1}) \\ \dfrac{q_{\hat{\mathsf{H}}} + q_{\mathsf{G}}}{2^\kappa} & (\textit{Case 2}) \\ (2\hat{e}(1+q_R)+1)\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_1,\mathcal{G}}(\lambda) + \dfrac{\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}}{2^\kappa} + \dfrac{q_{\mathsf{G}}}{2^\kappa} + \dfrac{q_D}{p} + \dfrac{3q_{\mathsf{G}}}{2^\lambda} & (\textit{Case 3}) \\ (2\hat{e}(1+q_R)+1)\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_1,\mathcal{G}}(\lambda) + \dfrac{\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}}{2^\kappa} + \dfrac{q_{\mathsf{G}}}{2^\kappa} + \dfrac{q_D}{p} + \dfrac{3q_{\mathsf{G}}}{2^\lambda} & (\textit{Case 4}) \end{cases}$$

where p *is the order of the underlying bilinear group and* q_R, q_D, $q_{\hat{\mathsf{H}}}$, *and* q_{G} *are the maximum number of queries* $\mathcal{A}$ *sends to* $\mathcal{O}_R$, $\mathcal{O}_D$, $\hat{\mathsf{H}}$ *and* G *oracles, respectively.*

We will prove IB-ME$^{\mathsf{BDH}}$ is Priv$^+$-CCA secure in each Case 1, Case 2, and Case 3.[9] To prove that IB-ME$^{\mathsf{BDH}}$ is Priv$^+$-CCA secure in Case 1 and Case 3, we will consider an IB-ME scheme IB-ME$^{\mathsf{Basic}}$ that is a simple version of IB-ME$^{\mathsf{BDH}}$ and show that IB-ME$^{\mathsf{Basic}}$ is Priv$^+$-CPA secure in Case 1 and Case 3. Then, we show that IB-ME$^{\mathsf{BDH}}$ is Priv$^+$-CCA in Case 1 and Case 3 assuming that IB-ME$^{\mathsf{Basic}}$ is Priv$^+$-CPA in Case 1 and Case 3, respectively. Also, we show that IB-ME$^{\mathsf{BDH}}$ is Priv$^+$-CCA in Case 2 in the random oracle model.

Basic IB-ME scheme. The IB-ME scheme IB-ME$^{\mathsf{Basic}}$ is as follows. The differences between IB-ME$^{\mathsf{Basic}}$ and IB-ME$^{\mathsf{BDH}}$ are that IB-ME$^{\mathsf{Basic}}$.Enc samples uniform randomness r instead of generating it with a hash function G, and IB-ME$^{\mathsf{Basic}}$.Dec does not perform the ciphertext validity check (i.e., does not check if $R = g_1^r$ holds). Its identity and message spaces are $\mathcal{ID} = \{0,1\}^*$ and $\mathcal{M} = \{0,1\}^{\mathsf{msgLen}+\lambda}$, respectively.

$\mathsf{Setup}(1^\lambda)$**:** It is identical to IB-ME$^{\mathsf{BDH}}$.Setup except that G is not chosen.
$\mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma)$**:** It is identical to IB-ME$^{\mathsf{BDH}}$.SKGen.
$\mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho)$**:** It is identical to IB-ME$^{\mathsf{BDH}}$.RKGen.
$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma, \mathsf{rcv}, \mathsf{m})$**:** It chooses $r \leftarrow\!\!\$\ \mathbb{Z}_p$ and computes $\mathsf{u}_{\mathsf{rcv}} := \mathsf{H}_2(\mathsf{rcv})$ and

$$R := g_1^r, \qquad \mathsf{ctxt} := \mathsf{m} \oplus \hat{\mathsf{H}}(\sigma, \mathsf{rcv}, R, e(X^r, \mathsf{u}_{\mathsf{rcv}}), e(\mathsf{sk}_\sigma, \mathsf{u}_{\mathsf{rcv}})).$$

It outputs $\mathsf{ct} := (R, \mathsf{ctxt})$.
$\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct} = (R, \mathsf{ctxt}))$**:** It computes $\mathsf{u}_{\mathsf{snd}} := \mathsf{H}_1(\mathsf{snd})$ and

$$\mathsf{m} := \mathsf{ctxt} \oplus \hat{\mathsf{H}}(\mathsf{snd}, \rho, R, e(R, \mathsf{sk}_\rho), e(\mathsf{u}_{\mathsf{snd}}, \mathsf{sk}_\rho)).$$

Finally, it outputs m.

We can easily verify that IB-ME$^{\mathsf{Basic}}$ is correct. We now show that IB-ME$^{\mathsf{Basic}}$ is Priv$^+$-CPA secure in Case 1 and Case 3.

Claim 1. *Suppose that the hash functions $\mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}$ are random oracles. If there exists an adversary $\mathcal{A}$ that breaks the* Priv$^+$-CPA *security of* IB-ME$^{\mathsf{Basic}}$ *in Case 1, there exists an adversary $\mathcal{B}$ that breaks the BDH assumption for $\mathcal{G}$ such that*

$$\left| \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{Basic}}} \;\middle|\; \textit{Case 1} \right] - \frac{1}{2} \right| \le (\hat{e}(1+q_R))^2 q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda),$$

where q_R and $q_{\hat{\mathsf{H}}}$ are the maximum number of queries $\mathcal{A}$ sends to $\mathcal{O}_R$ and $\hat{\mathsf{H}}$ oracles, respectively. The running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Proof. To prove the theorem, we consider the following sequence of games. We define the advantage of $\mathcal{A}$ in Game_i as

$$\epsilon_i := \left| \Pr\left[\mathsf{Game}_i^{\mathcal{A}}(\lambda) \Rightarrow 1 \right] - \frac{1}{2} \right|.$$

Game_0**.** This is the original security game. By definition, we have

$$\epsilon_0 = \left| \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case 1} \right] - \frac{1}{2} \right|.$$

$\mathsf{Game}_{1,0}$**.** In this game, we add abort conditions. We guess the challenge identity ρ_0^* that is not sent to $\mathcal{O}_R$ oracle. If the guess fails, the game aborts and sets a random coin as $\mathcal{A}$'s output. To do so, we change the challenger's procedures as follows. (The other procedures are worked as in the previous game.)

– When $\mathcal{A}$ sends ρ to H_2 oracle, it flips a coin d which yields 0 with probability $1-\delta$. Then, it samples $b \leftarrow\!\!\$\ \mathbb{Z}_p$, computes $\mathsf{u}_\rho := g_2^b$ and updates $\mathcal{L}_{\mathsf{H}_2} \leftarrow \mathcal{L}_{\mathsf{H}_2} \cup \{(\rho, \mathsf{u}_\rho, b, d)\}$. Then it returns u_ρ to $\mathcal{A}$.

[9] As noted in Remark 1, we do not need to prove in Case 4 because its security is implied by the security in Case 3.

– When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, it searches an entry $(\rho, \mathsf{u}_\rho, b, d) \in \mathcal{L}_{\mathsf{H}_2}$[10]. If $d = 0$, the game aborts. Otherwise (i.e., $d = 1$), it computes $\mathsf{sk}_\rho := (g_2^x)^b$ and returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
– When $\mathcal{A}$ outputs $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, it searches $(\mathsf{rcv}_{\mathsf{coin}}, \mathsf{u}_{\mathsf{rcv}_{\mathsf{coin}}}, b, d)$ from $\mathcal{L}_{\mathsf{H}_2}$. If $d = 1$, the game aborts. Otherwise (i.e., $d = 0$), it works as in Game_0.

The advantage of $\mathcal{A}$ in $\mathsf{Game}_{1,0}$ is equal to the advantage of $\mathcal{A}$ in Game_0 conditioning on the game does not abort. Therefore, we have

$$\epsilon_{1,0} = \epsilon_0 \cdot \Pr[\neg\mathsf{abort}].$$

Let us estimate the probability $\Pr[\neg\mathsf{abort}]$. The probability that the game does not abort in $\mathcal{O}_R$ oracle is δ^{q_R}. The probability that the game does not abort when $\mathcal{A}$ requests a challenge ciphertext is $1 - \delta$. Hence, the overall non-aborting probability is $\delta^{q_R}(1 - \delta)$. This value is maximum when $\hat{\delta} = \frac{q_R}{1+q_R}$, and thus we have $\Pr[\neg\mathsf{abort}] = \delta^{q_R}(1 - \delta) \leq \frac{1}{\hat{e}(1+q_R)}$ for large q_R. Therefore, we have

$$\epsilon_0 \leq \hat{e}(1 + q_R) \cdot \epsilon_{1,0}.$$

$\mathsf{Game}_{1,1}$**.** In this game, we add abort conditions. We guess the challenge identity ρ_1^* that is not sent to $\mathcal{O}_R$ oracle. If the guess fails, the game aborts and sets a random coin as $\mathcal{A}$'s output. Due to a similar argument above, we can prove

$$\epsilon_{1,0} \leq \hat{e}(1 + q_R) \cdot \epsilon_{1,1}.$$

Game_2**.** In this game, the challenge ciphertext $\mathsf{ct}_{\mathsf{coin}} := (R^*, \mathsf{ctxt}^*)$ is computed as

$$r^* \leftarrow\$ \mathbb{Z}_p, \quad Z \leftarrow\$ \{0,1\}^{\mathsf{msgLen}+\lambda}, \quad R^* := g_1^{r^*}, \quad \mathsf{ctxt}^* \leftarrow \mathsf{m}_{\mathsf{coin}}^* \oplus Z.$$

Let BadQ be the event that $\mathcal{A}$ queries $(\cdot, \mathsf{rcv}_{\mathsf{coin}}^*, R^*, U^*, \cdot)$ to the oracle $\hat{\mathsf{H}}$ where $U^* := e(R^*, \mathsf{sk}_{\mathsf{rcv}_{\mathsf{coin}}^*})$. Since Z is chosen independently at random from random oracles, $\mathcal{A}$ can distinguish the two games if BadQ occurs, and otherwise, they proceed identically. Thus, we have

$$|\epsilon_2 - \epsilon_{1,1}| \leq \Pr[\mathsf{BadQ}].$$

To estimate $\Pr[\mathsf{BadQ}]$, we show that if $\mathcal{A}$ triggers BadQ, we can construct an adversary $\mathcal{B}$ that solves the BDH problem.The construction of $\mathcal{B}$ is as follows.

1. Upon receiving $(G = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), g_1^\alpha, g_2^\alpha, g_2^\beta, g_1^\gamma)$, $\mathcal{B}$ sets $X := g_1^\alpha$ (i.e., msk is implicitly set α) and prepares three random oracles $\mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}$ (i.e., initialize the lists $\mathcal{L}_{\mathsf{H}_1}, \mathcal{L}_{\mathsf{H}_2}, \mathcal{L}_{\hat{\mathsf{H}}}$). Also, $\mathcal{B}$ flips a coin $\mathsf{coin} \leftarrow\$ \{0,1\}$. Then, $\mathcal{B}$ executes $\mathcal{A}$ on input $\mathsf{mpk} := (G, \mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}, X)$.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}$ answers them as follows:
   (a) When $\mathcal{A}$ sends σ to H_1 oracle, $\mathcal{B}$ samples $b \leftarrow\$ \mathbb{Z}_p$ and computes $\mathsf{u}_\sigma := g_1^b$. Then, $\mathcal{B}$ updates $\mathcal{L}_{\mathsf{H}_1} \leftarrow \mathcal{L}_{\mathsf{H}_1} \cup \{(\sigma, \mathsf{u}_\sigma, b)\}$ and returns u_σ to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to H_2 oracle, $\mathcal{B}$ samples $b \leftarrow\$ \mathbb{Z}_p$. With probability $1 - \delta$, $\mathcal{B}$ computes $\mathsf{u}_\rho := (g_2^\beta)^b$ and updates $\mathcal{L}_{\mathsf{H}_2} \leftarrow \mathcal{L}_{\mathsf{H}_2} \cup \{(\rho, \mathsf{u}_\rho, b, 0)\}$. Otherwise, $\mathcal{B}$ computes $\mathsf{u}_\rho := g_2^b$ and updates $\mathcal{L}_{\mathsf{H}_2} \leftarrow \mathcal{L}_{\mathsf{H}_2} \cup \{(\rho, \mathsf{u}_\rho, b, 1)\}$. Then, $\mathcal{B}$ returns u_ρ to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends (σ, ρ, R, U, V) to $\hat{\mathsf{H}}$ oracle, $\mathcal{B}$ samples $Z \leftarrow\$ \{0,1\}^{\mathsf{msgLen}+\lambda}$ and updates $\mathcal{L}_{\hat{\mathsf{H}}} \leftarrow \mathcal{L}_{\hat{\mathsf{H}}} \cup \{(\sigma, \rho, R, U, V, Z)\}$. Then, $\mathcal{B}$ returns Z to $\mathcal{A}$.
   (d) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}$ searches $(\sigma, \mathsf{u}_\sigma, b) \in \mathcal{L}_{\mathsf{H}_1}$ and computes $\mathsf{sk}_\sigma := (g_1^\alpha)^b$. Then, $\mathcal{B}$ returns $\mathsf{ek}_\sigma := (\sigma, \mathsf{sk}_\sigma)$ to $\mathcal{A}$.
   (e) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}$ searches $(\rho, \mathsf{u}_\rho, b, d) \in \mathcal{L}_{\mathsf{H}_2}$. If $d = 0$, $\mathcal{B}$ aborts the game. Otherwise (i.e., $d = 1$), $\mathcal{B}$ computes $\mathsf{sk}_\rho := (g_2^\alpha)^b$. Then, $\mathcal{B}$ returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.

[10] If no entry exists, $\mathsf{H}_2(\rho)$ is internally queried and flips a coin d. (In the rest of this paper, when we have a similar situation, we also deal with it in the same manner.)

(f) When $\mathcal{A}$ outputs $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}$ searches $(\mathsf{rcv}_{\mathsf{coin}}^*, \mathsf{u}_{\mathsf{rcv}_{\mathsf{coin}}^*}, b^*, d^*) \in \mathcal{L}_{\mathsf{H}_2}$. If $d^* = 1$, $\mathcal{B}$ aborts the game. Otherwise, $\mathcal{B}$ sets $R^* := g_1^\gamma$ and computes $\mathsf{ctxt}^* := \mathsf{m}^* \oplus Z$ where $Z \leftarrow\!\$\ \{0,1\}^{\mathsf{msgLen}+\lambda}$. Then $\mathcal{B}$ sets $\mathsf{ct}_0 := (R^*, \mathsf{ctxt}^*)$ and computes ct_1 as the protocol description, and returns $\mathsf{ct}_{\mathsf{coin}}$ to $\mathcal{A}$.

3. Finally, $\mathcal{A}$ outputs a guess $\widehat{\mathsf{coin}}$. Then, $\mathcal{B}$ picks an entry $(\cdot, \mathsf{rcv}_{\mathsf{coin}}^*, R^*, U^*, \cdot) \in \mathcal{L}_{\hat{\mathsf{H}}}$ at random and outputs $D := (U^*)^{\frac{1}{b^*}}$ as the solution of the BDH problem.

We can see that $\mathcal{B}$ perfectly simulates the Priv^+-CPA game against $\mathcal{A}$ if $\mathcal{B}$ does not abort. Moreover, we know that $\mathsf{sk}_{\mathsf{rcv}_{\mathsf{coin}}^*} = (\mathsf{u}_{\mathsf{rcv}_{\mathsf{coin}}^*})^\alpha = (g_2^{\alpha\beta})^{b^*}$ and $R^* = g_1^\gamma$, and thus

$$U^* = e(R^*, \mathsf{sk}_{\mathsf{rcv}_{\mathsf{coin}}^*}) = e(g_1^\gamma, g_2^{\alpha\beta b^*}) = (e(g_1, g_2)^{\alpha\beta\gamma})^{b^*}.$$

If $\mathcal{A}$ distinguishes the two games, $\mathcal{A}$ has queried $\hat{\mathsf{H}}(\cdot, \mathsf{rcv}_{\mathsf{coin}}^*, R^*, U^*, \cdot)$, and thus with probability at least $\frac{1}{q_{\hat{\mathsf{H}}}}$, $\mathcal{B}$ can solve the BDH problem correctly. Thus we have

$$|\epsilon_2 - \epsilon_{1,1}| \le \Pr[\mathsf{BadQ}] \le q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{G},\mathcal{B}}(\lambda).$$

In Game_2, ct_0 and ct_1 are random elements over the ciphertext space and it is independent of the challenge bit coin. Thus, coin is information-theoretically hidden from $\mathcal{A}$, so we have $\epsilon_2 = 0$.

Putting everything together, we obtain

$$\left| \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 1 \right] - \frac{1}{2} \right| \le \hat{e}^2 (1 + q_R)^2 q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda).$$

□

Claim 2. *Suppose that the hash functions $\mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}$ are random oracles. If there exists an adversary $\mathcal{A}$ that breaks the Priv^+-CPA security of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ in Case 3, there exists an adversary $\mathcal{B}$ that breaks the BDH assumption for $\mathcal{G}$ such that*

$$\left| \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{Basic}}} \;\middle|\; \textit{Case 3} \right] - \frac{1}{2} \right| \le \hat{e}(1 + q_R) q_{\hat{\mathsf{H}}} \left(\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{G},\mathcal{B}}(\lambda) + \frac{1}{2^\kappa} \right),$$

where q_R and $q_{\hat{\mathsf{H}}}$ are the maximum number of queries $\mathcal{A}$ sends to $\mathcal{O}_R$ and $\hat{\mathsf{H}}$ oracles, respectively. The running time of $\mathcal{B}$ is about that of $\mathcal{A}$.

Proof. To prove the theorem, we consider the following sequence of games. We define the advantage of $\mathcal{A}$ in Game_i as

$$\epsilon_i := \left| \Pr\left[\mathsf{Game}_i^{\mathcal{A}}(\lambda) \Rightarrow 1 \right] - \frac{1}{2} \right|.$$

Game_0. This is the original security game. By definition, we have

$$\epsilon_0 = \left| \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{Basic}}} \;\middle|\; \text{Case } 3 \right] - \frac{1}{2} \right|.$$

Game_1. In this game, we add abort conditions. We guess the challenge identity ρ_0^* that is not sent to $\mathcal{O}_R$ oracle. If the guess fails, the game aborts and sets a random coin as $\mathcal{A}$'s output. By the same argument to hop from Game_0 to $\mathsf{Game}_{1,0}$ in the proof of Claim 1, we can prove

$$\epsilon_0 \le \hat{e}(1 + q_R) \cdot \epsilon_1.$$

Game_2. In this game, the challenge ciphertext for $\mathsf{coin} = 0$ $\mathsf{ct}_0 := (R_0^*, \mathsf{ctxt}_0^*)$ is computed as

$$r^* \leftarrow\!\$\ \mathbb{Z}_p, \quad Z \leftarrow\!\$\ \{0,1\}^{\mathsf{msgLen}+\lambda}, \quad R_0^* := g_1^{r^*}, \quad \mathsf{ctxt}_0^* \leftarrow \mathsf{m}_0^* \oplus Z.$$

By the same argument to hop from $\mathsf{Game}_{1,1}$ to Game_2 in the proof of Claim 1, we have

$$|\epsilon_2 - \epsilon_1| \leq q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{G},\mathcal{B}}(\lambda).$$

Game_3**.** In this game, the challenge ciphertext for $\mathsf{coin} = 1$ $\mathsf{ct}_1 := (R_1^*, \mathsf{ctxt}_1^*)$ is computed as

$$r^* \leftarrow\!\$\, \mathbb{Z}_p, \quad Z \leftarrow\!\$\, \{0,1\}^{\mathsf{msgLen}+\lambda}, \quad R_1^* := g_1^{r^*}, \quad \mathsf{ctxt}_1^* \leftarrow \mathsf{m}_1^* \oplus Z.$$

$\mathcal{A}$ may notice this change when it sends σ_1^* to $\hat{\mathsf{H}}$ oracle. Since σ_1^* is chosen independently at random from κ-distribution, we have

$$|\epsilon_3 - \epsilon_2| \leq \frac{q_{\hat{\mathsf{H}}}}{2^\kappa}.$$

In Game_3, ct_0 and ct_1 are random elements over the ciphertext space and thus they are independent of the challenge bit coin. Thus, coin is information-theoretically hidden from $\mathcal{A}$, so we have

$$\epsilon_3 = 0.$$

Putting everything together, we obtain

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{Basic}}} \;\middle|\; \text{Case 3}\right] - \frac{1}{2}\right| \leq \hat{e}(1+q_R)q_{\hat{\mathsf{H}}}\left(\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{G},\mathcal{B}}(\lambda) + \frac{1}{2^\kappa}\right).$$

□

Now, we will show $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ is Priv^+-CCA secure, which is proved by Lemmata 1 and 2. First, we show that in Cases 1 and 3, we can upgrade the CPA enhanced privacy of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ into CCA one via FO transformation.

Lemma 1. *Suppose the hash function* G *is a random oracle. If there exists an adversary* $\mathcal{A}$ *that breaks the* Priv^+-CCA *security of* $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ *in Case 1, there exists an adversary* $\mathcal{B}$ *that breaks the* BDH *assumption for* $\mathcal{G}$ *such that*

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{BDH}}} \;\middle|\; \textit{Case 1}\right] - \frac{1}{2}\right| \leq 3\hat{e}^2(1+q_R)^2 q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda) + \frac{q_D}{p} + \frac{3q_{\mathsf{G}}}{2^\lambda},$$

where p *is the order of the underlying bilinear group and* q_R*,* q_D*,* $q_{\hat{\mathsf{H}}}$*, and* q_{G} *are the maximum number of queries* $\mathcal{A}$ *sends to* $\mathcal{O}_R$*,* $\mathcal{O}_D$*,* $\hat{\mathsf{H}}$ *and* G *oracles, respectively.*

Proof. We now prove the Priv^+-CCA security of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ assuming the Priv^+-CPA security of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$. The proof is similar to the proof of the FO transformation for PKE schemes [18]. To prove the theorem, we consider the following sequence of games Game_i. Define the advantage of $\mathcal{A}$ in Game_i as

$$\epsilon_i := \left|\Pr\left[\mathsf{Game}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right|.$$

Game_0**.** This is the original security game. By definition, we have

$$\epsilon_0 = \mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda).$$

Game_1**.** In this game, the randomness $\mathsf{k}^* \in \{0,1\}^\lambda$ used to generate the challenge ciphertext is chosen in the setup phase instead of the challenge phase. Since there is no difference in $\mathcal{A}$'s view, we have

$$\epsilon_1 = \epsilon_0.$$

Game_2**.** In this game, we change the behavior of G oracle. When $\mathcal{A}$ sends a tuple $(\sigma, \rho, \mathsf{m}, \mathsf{k})$ to G, the challenger picks $r \leftarrow\!\!\$\ \mathbb{Z}_p$, and computes

$$\mathsf{sk}_\sigma := \mathsf{H}_1(\sigma)^x, \qquad \mathsf{ct} \leftarrow \mathsf{IB\text{-}ME}^{\mathsf{Basic}}.\mathsf{Enc}(\mathsf{mpk}^{11}, \mathsf{sk}_\sigma, \rho, \mathsf{m}||\mathsf{k}; r).$$

Then, it updates $\mathcal{L}_\mathsf{G} \leftarrow \mathcal{L}_\mathsf{G} \cup \{((\sigma, \rho, \mathsf{m}, \mathsf{k}), r, \mathsf{ct})\}$ and returns r to $\mathcal{A}$.

Since there is no difference in the behaviors of oracles from $\mathcal{A}$'s viewpoint, we have

$$\epsilon_2 = \epsilon_1.$$

We remark that sk_σ is unique for each identity σ, and thus the ciphertext computed as above can be uniquely determined by $(\sigma, \rho, \mathsf{m}, \mathsf{k})$.

Game_3**.** In this game, we change the behavior of $\mathcal{O}_D$ oracle. When $\mathcal{A}$ sends $(\mathsf{snd}, \rho, \mathsf{ct})$ to $\mathcal{O}_D$, it finds an entry $((\mathsf{snd}, \rho, \mathsf{m}, \mathsf{k}), r, \mathsf{ct}) \in \mathcal{L}_\mathsf{G}$. If such a tuple exists, $\mathsf{m}||\mathsf{k}$ is returned to $\mathcal{A}$. Otherwise, $\perp$ is returned to $\mathcal{A}$.

Let BadD be the event that $\mathcal{A}$ submits a decryption query on $(\mathsf{snd}, \rho, \mathsf{ct})$ such that $((\mathsf{snd}, \rho, \mathsf{m}, \mathsf{k}), r, \mathsf{ct}) \notin \mathcal{L}_\mathsf{G}$ but it is not rejected in the previous game. Due to the perfect correctness of the scheme, the two games proceed identically unless BadD occurs. Thus, we have

$$|\epsilon_3 - \epsilon_2| \leq \Pr[\mathsf{BadD}].$$

We now estimate $\Pr[\mathsf{BadD}]$. In the previous game, if $((\mathsf{snd}, \rho, \mathsf{m}, \mathsf{k}), r, \mathsf{ct}) \notin \mathcal{L}_\mathsf{G}$ when $(\mathsf{snd}, \rho, \mathsf{ct})$ is sent to $\mathcal{O}_D$, $\mathsf{G}(\mathsf{snd}, \rho, \mathsf{m}, \mathsf{k})$ is queried internally and $r \leftarrow\!\!\$\ \mathbb{Z}_q$ is sampled. Then, $\mathcal{O}_D$ checks whether $R = g_1^r$ holds. For any $R \in \mathbb{G}_1$, the probability that $R = g_1^r$ holds for randomly chosen $r \in \mathbb{Z}_p$ is $1/p$. Since $\mathcal{A}$ queries $\mathcal{O}_D$ at most q_D, we have

$$|\epsilon_3 - \epsilon_2| \leq \Pr[\mathsf{BadD}] \leq \frac{q_D}{p}.$$

After this game, the decryption oracle is simulated without any decryption keys.

Game_4**.** In this game, we add an abort condition into G oracle. If $\mathcal{A}$ sends a tuple $(\cdot, \cdot, \cdot, \mathsf{k})$ such that $\mathsf{k} = \mathsf{k}^*$ before the challenge phase, the game aborts. Since $\mathsf{k}^* \in \{0,1\}^\lambda$ is chosen at random and information-theoretically hidden from $\mathcal{A}$ before the challenge phase, we have

$$|\epsilon_4 - \epsilon_3| \leq \frac{q_\mathsf{G}}{2^\lambda}.$$

Game_5**.** In this game, we change how to generate the challenge ciphertext ct^*. To generate ct^*, the challenger chooses $r^* \leftarrow\!\!\$\ \mathbb{Z}_p$ and computes

$$\mathsf{sk}_{\sigma^*_{\mathsf{coin}}} := \mathsf{H}_1(\sigma^*_{\mathsf{coin}})^x, \mathsf{ct}_{\mathsf{coin}} \leftarrow \mathsf{IB\text{-}ME}^{\mathsf{Basic}}.\mathsf{Enc}(\mathsf{mpk}, \mathsf{sk}_{\sigma^*_{\mathsf{coin}}}, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{m}^*_{\mathsf{coin}}||\mathsf{k}^*; r^*).$$

Now, the randomness r^* is chosen independently from G. Let BadQ be the event that $\mathcal{A}$ sends $(\cdot, \cdot, \cdot, \mathsf{k}^*)$ to G oracle after it receives the challenge ciphertext. Since $\mathcal{A}$'s view is identical unless BadQ occurs, we have

$$|\epsilon_5 - \epsilon_4| \leq \Pr[\mathsf{BadQ}].$$

To estimate $\Pr[\mathsf{BadQ}]$, we show that if $\mathcal{A}$ can trigger the event BadQ, there exists an adversary $\mathcal{B}_1$ that breaks the $\mathsf{Priv}^+\text{-}\mathsf{CPA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ (in Case 1).

The construction of $\mathcal{B}_1$ is as follows. Upon receiving mpk (of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$), $\mathcal{B}_1$ samples $\mathsf{k}^* \leftarrow\!\!\$\ \{0,1\}^\lambda$, prepares mpk of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$, and executes $\mathcal{A}$ on input it. Then, $\mathcal{B}_1$ simulates the $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ game against $\mathcal{A}$ as in Game_5. When a query is sent to $\mathcal{O}_S$ or $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ uses its oracles to generate encryption or decryption keys. When $\mathcal{A}$ requests a challenge ciphertext on $(\Sigma^*_0, \Sigma^*_1, \mathsf{rcv}^*_0, \mathsf{rcv}^*_1, \mathsf{m}^*_0, \mathsf{m}^*_1)$, $\mathcal{B}_1$ samples $\sigma^*_{\mathsf{coin}} \leftarrow\!\!\$\ \Sigma^*_{\mathsf{coin}}$ and another independent randomness $\mathsf{k}' \leftarrow\!\!\$\ \{0,1\}^\lambda$, sets $D := \{\sigma^*_{\mathsf{coin}}\}$ is a distribution that outputs σ^*_{coin}

[11] For simplicity, we use the same symbol mpk for $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ and $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ since mpk of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ covers that of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$.

with probability 1 and sends $(D, D, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{m}^*_{\mathsf{coin}}||\mathsf{k}^*, \mathsf{m}^*_{\mathsf{coin}}||\mathsf{k}')$ to its challenger, receiving the challenge ciphertext $\mathsf{ct}_{\mathsf{coin}}$. $\mathcal{B}_1$ forwards it to $\mathcal{A}$. When $\mathcal{A}$ triggers the event BadQ, $\mathcal{B}_1$ outputs $\widehat{\mathsf{coin}} := 0$ to its challenger as its guess of coin. If $\mathcal{A}$ does not trigger the event BadQ, $\mathcal{B}_1$ outputs a randomly chosen $\widehat{\mathsf{coin}} \leftarrow\!\!\$\ \{0,1\}$ to its challenger.

Now, we evaluate the $\mathcal{B}_1$'s advantage. Let Fail be the event that BadQ occurs when $\widehat{\mathsf{coin}} = 1$ (i.e., $\mathsf{ct}_{\mathsf{coin}}$ does not contain k^*). Since k^* is uniformly distributed and independent from $\mathcal{B}_1$'s view in this case, $\Pr[\mathsf{Fail}] \leq q_{\mathsf{G}}/2^{\lambda}$. Assume Fail did not happen, i.e., BadQ occurs only when $\widehat{\mathsf{coin}} = 0$. Since $\mathcal{B}_1$ always outputs 0 when BadQ occurs, $\Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}}\right] = 1$. If BadQ did not occur, $\mathcal{B}_1$ outputs a random coin and thus $\Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}}\right] = 1/2$. Thus, we have

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{B}_1}_{\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case 1}\right] - \frac{1}{2}\right| + \frac{q_{\mathsf{G}}}{2^{\lambda}} \\
&\geq \left|\Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}}\right] - \frac{1}{2}\right| \\
&= \left|\Pr[\mathsf{BadQ}] + \frac{1}{2}\Pr[\neg\mathsf{BadQ}] - \frac{1}{2}\right| = \frac{1}{2}\Pr[\mathsf{BadQ}].
\end{aligned}$$

Therefore, we have

$$|\epsilon_5 - \epsilon_4| \leq \Pr[\mathsf{BadQ}] \leq 2\hat{e}^2(1+q_R)^2 q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_1,\mathcal{G}}(\lambda) + \frac{2q_{\mathsf{G}}}{2^{\lambda}}.$$

We finally bound ϵ_5. If $\mathcal{A}$ can break the Priv^+-CCA security in Game_5 in Case 1, there exists an adversary $\mathcal{B}_2$ that breaks the Priv^+-CPA security of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ in Case 1 such that

$$\epsilon_5 = \mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cpa}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda).$$

The proof is straightforward because $\mathcal{B}_2$ can simulate $\mathcal{O}_D$ without any decryption keys and the challenge ciphertext is generated with independent randomness r^*.

Putting everything together, we obtain

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{BDH}}} \;\middle|\; \text{Case 1}\right] - \frac{1}{2}\right| \leq 3\hat{e}^2(1+q_R)^2 q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda) + \frac{q_D}{p} + \frac{3q_{\mathsf{G}}}{2^{\lambda}}.$$

□

Lemma 2. *An adversary $\mathcal{A}$ attacking the Priv^+-CCA security in Case 2 of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ has advantage*

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{BDH}}} \;\middle|\; \textit{Case 2}\right] - \frac{1}{2}\right| \leq \frac{q_{\hat{\mathsf{H}}} + q_{\mathsf{G}}}{2^{\kappa}}.$$

where $q_{\hat{\mathsf{H}}}$ and q_{G} are the maximum number of queries $\mathcal{A}$ makes to the $\hat{\mathsf{H}}$ and G oracles, respectively.

Proof. To prove the theorem, we consider the following sequence of games Game_i. Define the advantage of $\mathcal{A}$ in Game_i as

$$\epsilon_i := \left|\Pr\left[\mathsf{Game}^{\mathcal{A}}_i(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right|.$$

Game_0. This is the original security game. By definition, we have

$$\epsilon_0 = \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{BDH}}} \;\middle|\; \text{Case 2}\right] - \frac{1}{2}\right|.$$

Game_1**.** In this game, the challenge ciphertext $\mathsf{ct}_{\mathsf{coin}}$ is computed as $\mathsf{ct}_{\mathsf{coin}} \leftarrow (g_1^r, (\mathsf{m}^*_{\mathsf{coin}}||\mathsf{k}) \oplus Z)$ for random $r \leftarrow\!\!\$\ \mathbb{Z}_p$ and $Z \leftarrow\!\!\$\ \{0,1\}^{\mathsf{msgLen}+\lambda}$. $\mathcal{A}$ may notice this change when it sends σ_0^* or σ_1^* to $\hat{\mathsf{H}}$ or G oracle. Since σ_0^* and σ_1^* are chosen independently at random from κ-distribution, we have

$$|\epsilon_1 - \epsilon_0| \leq \frac{q_{\hat{\mathsf{H}}} + q_{\mathsf{G}}}{2^\kappa}.$$

In Game_1, ct_0 and ct_1 are distributed uniformly at random. This means that coin is information-theoretically hidden from $\mathcal{A}$, so we have

$$\epsilon_1 = 0.$$

Putting everything together, we obtain

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{BDH}}} \;\middle|\; \text{Case 2}\right] - \frac{1}{2}\right| \leq \frac{q_{\hat{\mathsf{H}}} + q_{\mathsf{G}}}{2^\kappa}.$$

□

Lemma 3. *Suppose the hash function* G *is a random oracle. If there exists an adversary* $\mathcal{A}$ *that breaks the* $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ *security of* $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ *in Case 3, there exists an adversary* $\mathcal{B}$ *that breaks the* BDH *assumption for* $\mathcal{G}$ *such that*

$$\begin{aligned}&\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{BDH}}} \;\middle|\; \textit{Case 3}\right] - \frac{1}{2}\right| \\ &\leq (2\hat{e}(1+q_R)+1)\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_1,\mathcal{G}}(\lambda) + \frac{\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}}{2^\kappa} + \frac{q_D}{p} + \frac{3q_{\mathsf{G}}}{2^\lambda},\end{aligned}$$

where p *is the order of the underlying bilinear group and* q_R*,* q_D*,* $q_{\hat{\mathsf{H}}}$*, and* q_{G} *are the maximum number of queries* $\mathcal{A}$ *sends to* $\mathcal{O}_R$*,* $\mathcal{O}_D$*,* $\hat{\mathsf{H}}$ *and* G *oracles, respectively.*

Proof. The proof is almost identical to the proof of Lemma 1 except that FO transformation is only applied for when the challenge coin is 0. Formally, we will consider the following sequence of hybrids.

Game_0**.** This is the original security game. By definition, we have

$$\epsilon_0 = \mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda).$$

Game_1**-**Game_4**.** They are identical to the same game in the proof of Lemma 1. Note that the prepared randomness is only used for the challenge ciphertext when $\mathsf{coin} = 0$.

Game_5**.** In this game, we change how to generate the challenge ciphertext ct^* when $\mathsf{coin} = 0$. To generate ct_0, the challenger chooses $r^* \leftarrow\!\!\$\ \mathbb{Z}_p$ and computes

$$\mathsf{ct}_0 \leftarrow \mathsf{IB\text{-}ME}^{\mathsf{BDH}}.\mathsf{Enc}(\mathsf{mpk}, \mathsf{sk}_{\sigma^*}, \mathsf{rcv}^*, \mathsf{m}^*||\mathsf{k}^*; r^*).$$

Now, the randomness r^* is chosen independently from G. Let BadQ be the event that $\mathcal{A}$ sends $(\cdot,\cdot,\cdot,\mathsf{k}^*)$ to G oracle after it receives the challenge ciphertext. Since $\mathcal{A}$'s view is identical unless BadQ occurs, we have

$$|\epsilon_5 - \epsilon_4| \leq \Pr[\mathsf{BadQ}].$$

To estimate $\Pr[\mathsf{BadQ}]$, we show that if $\mathcal{A}$ can trigger the event BadQ, there exists an adversary $\mathcal{B}_1$ that breaks the $\mathsf{Priv}^+\text{-}\mathsf{CPA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ (in Case 1).

The construction of $\mathcal{B}_1$ is as follows. Upon receiving mpk (of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$), $\mathcal{B}_1$ samples its challenge coin $b \leftarrow\!\!\$\ \{0,1\}$, $\mathsf{k}^* \leftarrow\!\!\$\ \{0,1\}^\lambda$, prepares mpk of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$, and executes $\mathcal{A}$ on input it. Then, $\mathcal{B}_1$ simulates the $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ game against $\mathcal{A}$ as in Game_5. When a query is sent to $\mathcal{O}_S$ or $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ uses its oracles to generate encryption or decryption keys. When $\mathcal{A}$ requests a challenge ciphertext on $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$,

if $b = 0$, $\mathcal{B}_1$ samples $\sigma_0^* \leftarrow\$ \Sigma_0^*$ and another independent randomness $\mathsf{k}' \leftarrow\$ \{0,1\}^\lambda$, sets $D := \{\sigma_0^*\}$ is a distribution that outputs σ_0^* with probability 1 and sends $(D, D, \mathsf{rcv}_0^*, \mathsf{rcv}_0^*, \mathsf{m}_0^*||\mathsf{k}^*, \mathsf{m}_0^*||\mathsf{k}')$ to its challenger, receiving the challenge ciphertext $\mathsf{ct}_{\mathsf{coin}}$. $\mathcal{B}_1$ forwards it to $\mathcal{A}$. Otherwise, if $b = 1$, $\mathcal{B}_1$ generates $\mathsf{ct}_{\mathsf{coin}}$ by itself. When $\mathcal{A}$ triggers the event BadQ, $\mathcal{B}_1$ outputs $\widehat{0} := 0$ to its challenger as its guess of coin. If $\mathcal{A}$ does not trigger the event BadQ, $\mathcal{B}_1$ outputs a randomly chosen $\widehat{\mathsf{coin}} \leftarrow\$ \{0,1\}$ to its challenger.

Now, we evaluate the $\mathcal{B}_1$'s advantage. Let Fail be the event that BadQ occurs when $\widehat{\mathsf{coin}} = 1$ (i.e., $\mathsf{ct}_{\mathsf{coin}}$ does not contain k^*). Since k^* is uniformly distributed and independent from $\mathcal{B}_1$'s view in this case, $\Pr[\mathsf{Fail}] \leq q_{\mathsf{G}}/2^\lambda$. Assume Fail did not happen, i.e., BadQ occurs only when $\widehat{\mathsf{coin}} = 0$. Since $\mathcal{B}_1$ always outputs 0 when BadQ occurs, $\Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}}\right] = 1$. If BadQ did not occur, $\mathcal{B}_1$ outputs a random coin and thus $\Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}}\right] = 1/2$. Thus, we have

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{B}_1}_{\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 1\right] - \frac{1}{2}\right| + \frac{q_{\mathsf{G}}}{2^\lambda} \\
&\geq \left|\Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}}\right] - \frac{1}{2}\right| \\
&= \left|\Pr[\mathsf{BadQ}] + \frac{1}{2}\Pr[\neg\mathsf{BadQ}] - \frac{1}{2}\right| = \frac{1}{2}\Pr[\mathsf{BadQ}].
\end{aligned}$$

Therefore, we have

$$|\epsilon_5 - \epsilon_4| \leq \Pr[\mathsf{BadQ}] \leq 2\hat{e}^2(1+q_R)^2 q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_1,\mathcal{G}}(\lambda) + \frac{2q_{\mathsf{G}}}{2^\lambda}.$$

Game_6**.** In this game, when $\mathsf{coin} = 1$, g_1^r in the challenge ciphertext ct_1 is computed with random $r \leftarrow\$ \mathbb{Z}_p$ instead of r computed with G. $\mathcal{A}$ may notice this change when it sends σ_1^* to G oracle. Since σ_1^* is chosen independently at random from κ-distribution, we have

$$|\epsilon_6 - \epsilon_5| \leq \frac{q_{\mathsf{G}}}{2^\kappa}.$$

We finally bound ϵ_6. If $\mathcal{A}$ can break the Priv^+-CCA security in Game_5 in Case 3, there exists an adversary $\mathcal{B}_2$ that breaks the Priv^+-CPA security of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ in Case 3 such that

$$\epsilon_6 = \mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cpa}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda).$$

The proof is straightforward because $\mathcal{B}_2$ can simulate $\mathcal{O}_D$ without any decryption keys and the challenge ciphertext is generated with independent randomness r^*.

Putting everything together, we obtain

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{BDH}}} \;\middle|\; \text{Case } 3\right] - \frac{1}{2}\right| \\
&\leq (2\hat{e}(1+q_R)+1)\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_1,\mathcal{G}}(\lambda) + \frac{\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}}{2^\kappa} + \frac{q_{\mathsf{G}}}{2^\kappa} + \frac{q_D}{p} + \frac{3q_{\mathsf{G}}}{2^\lambda}.
\end{aligned}$$

□

From Lemma 3 and the observation in Remark 1, we obtain the following lemma.

Lemma 4. *Suppose the hash function* G *is a random oracle. If there exists an adversary* $\mathcal{A}$ *that breaks the* Priv^+-CCA *security of* $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ *in Case 4, there exists an adversary* $\mathcal{B}$ *that breaks the* BDH *assumption for* $\mathcal{G}$ *such that*

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{BDH}}} \;\middle|\; \textit{Case } 4\right] - \frac{1}{2}\right| \\
&\leq (2\hat{e}(1+q_R)+1)\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}\mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_1,\mathcal{G}}(\lambda) + \frac{\hat{e}(1+q_R)q_{\hat{\mathsf{H}}}}{2^\kappa} + \frac{q_{\mathsf{G}}}{2^\kappa} + \frac{q_D}{p} + \frac{3q_{\mathsf{G}}}{2^\lambda}.
\end{aligned}$$

where p is the order of the underlying bilinear group and q_R, q_D, $q_{\hat{\mathsf{H}}}$, and q_{G} are the maximum number of queries $\mathcal{A}$ sends to $\mathcal{O}_R$, $\mathcal{O}_D$, $\hat{\mathsf{H}}$ and G oracles, respectively.

Theorem 2. *Suppose the hash functions H_1, H_2, $\hat{\mathsf{H}}$, and G are random oracles. Under the BDH assumption, $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ is $\mathsf{Auth\text{-}oCMA}$ secure in the ROM. Formally, if there exists an adversary $\mathcal{A}$ that breaks the $\mathsf{Auth\text{-}oCMA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$, there exists an adversary $\mathcal{B}$ that breaks the BDH assumption such that*

$$\mathsf{Adv}^{\mathsf{auth\text{-}ocma}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda) \leq \frac{\hat{e}^2(q_S+q_R)^2 q_{\hat{\mathsf{H}}}}{4} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda) + \frac{q_{\mathsf{G}}}{2^{\mathsf{msgLen}+\lambda}} + \frac{1}{p},$$

where p is the order of the underlying bilinear group and q_S, q_R, $q_{\hat{\mathsf{H}}}$, and q_{G} are the maximum number of queries $\mathcal{A}$ makes to the $\mathcal{O}_S$, $\mathcal{O}_R$, $\hat{\mathsf{H}}$, and G oracles, respectively.

Proof. Let $\mathcal{A}$ be an adversary that breaks $\mathsf{Auth\text{-}oCMA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$. To prove the theorem, we consider the following sequence of games Game_i for $i \in \{0, \cdots, 3\}$. Define the advantage of $\mathcal{A}$ in Game_i as

$$\epsilon_i := \Pr\Big[\mathsf{Game}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\Big].$$

Game_0. This is the original $\mathsf{Auth\text{-}oCMA}$ game. By definition, we have

$$\epsilon_0 = \mathsf{Adv}^{\mathsf{auth\text{-}ocma}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda).$$

Game_1. In this game, we change the behavior of $\mathcal{O}_S$, $\mathcal{O}_R$, and $\mathcal{O}_E$ as follows.

- When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, it computes $\mathsf{sk}_\sigma := \mathsf{H}_1(\sigma)^x$. Then, it searches entries $(\mathsf{snd}, \mathsf{rcv}, \mathsf{m}||\mathsf{k}, \mathsf{ctxt}) \in \mathcal{L}_E$ such that $\mathsf{snd} = \sigma$. If such entries exist, it works as follows for each such entry. Let $\mathsf{u}_{\mathsf{rcv}} := \mathsf{H}_2(\mathsf{rcv})$ and $r := \mathsf{G}(\sigma, \mathsf{rcv}, \mathsf{m}, \mathsf{k})$.
  - If there exists an entry $(\mathsf{snd}, \mathsf{rcv}, g_1^r, e(X^r, \mathsf{u}_{\mathsf{rcv}}), e(\mathsf{sk}_\sigma, \mathsf{u}_{\mathsf{rcv}}), *) \in \mathcal{L}_{\hat{\mathsf{H}}}$, it aborts the game. (In this case, it cannot program the random oracle.)
  - Else, it updates

$$\mathcal{L}_{\hat{\mathsf{H}}} \leftarrow \mathcal{L}_{\hat{\mathsf{H}}} \cup \{(\mathsf{snd}, \mathsf{rcv}, g_1^r, e(X^r, \mathsf{u}_{\mathsf{rcv}}), e(\mathsf{sk}_\sigma, \mathsf{u}_{\mathsf{rcv}}), \mathsf{ctxt} \oplus (\mathsf{m}||\mathsf{k}))\}.$$

  After that, it removes the programmed entries from $\mathcal{L}_E$.

  Finally, it returns $\mathsf{ek}_\sigma := (\sigma, \mathsf{sk}_\sigma)$ to $\mathcal{A}$.
- When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, it computes $\mathsf{sk}_\rho := \mathsf{H}_2(\rho)^x$. Then, it searches entries $(\mathsf{snd}, \mathsf{rcv}, \mathsf{m}||\mathsf{k}, \mathsf{ctxt}) \in \mathcal{L}_E$ such that $\mathsf{rcv} = \rho$. If such entries exist, it works as follows for each such entry. Let $\mathsf{u}_{\mathsf{snd}} := \mathsf{H}_1(\mathsf{snd})$ and $r := \mathsf{G}(\mathsf{snd}, \rho, \mathsf{m}, \mathsf{k})$.
  - If there exists an entry $(\mathsf{snd}, \mathsf{rcv}, g_1^r, e(g_1^r, \mathsf{sk}_\rho), e(\mathsf{H}_1(\mathsf{snd}), \mathsf{sk}_\rho), *) \in \mathcal{L}_{\hat{\mathsf{H}}}$, it aborts the game.
  - Else, for each entry, it updates

$$\mathcal{L}_{\hat{\mathsf{H}}} \leftarrow \mathcal{L}_{\hat{\mathsf{H}}} \cup \big\{(\mathsf{snd}, \mathsf{rcv}, g_1^r, e(g_1^r, \mathsf{sk}_\rho), e(\mathsf{u}_{\mathsf{snd}}, \mathsf{sk}_\rho), \mathsf{ctxt} \oplus (\mathsf{m}||\mathsf{k}))\big\}.$$

  Finally, it returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
- When $\mathcal{A}$ sends a tuple $(\sigma, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_E$ oracle, it samples $\mathsf{k} \leftarrow\!\!\$\ \{0,1\}^\lambda$ and computes $r := \mathsf{G}(\sigma, \mathsf{rcv}, \mathsf{m}, \mathsf{k})$ and $R := g_1^r$. Then, it computes ctxt as follows.
  1. If $\sigma \in \mathcal{L}_S$, it retrieves sk_σ[12] and computes $\mathsf{u}_{\mathsf{rcv}} := \mathsf{H}_2(\mathsf{rcv})$ and

$$\mathsf{ctxt} := (\mathsf{m}||\mathsf{k}) \oplus \hat{\mathsf{H}}(\sigma, \mathsf{rcv}, R, e(X^r, \mathsf{u}_{\mathsf{rcv}}), e(\mathsf{sk}_\sigma, \mathsf{u}_{\mathsf{rcv}})).$$

  2. If $\sigma \notin \mathcal{L}_S$ and $\mathsf{rcv} \in \mathcal{L}_R$, it retrieves $\mathsf{sk}_{\mathsf{rcv}}$[13] and computes $\mathsf{u}_{\mathsf{snd}} := \mathsf{H}_1(\mathsf{snd})$ and

$$\mathsf{ctxt} := (\mathsf{m}||\mathsf{k}) \oplus \hat{\mathsf{H}}(\sigma, \mathsf{rcv}, R, e(R, \mathsf{sk}_{\mathsf{rcv}}), e(\mathsf{u}_{\mathsf{snd}}, \mathsf{sk}_{\mathsf{rcv}})).$$

[12] Since $\sigma \in \mathcal{L}_S$, the challenger already has computed sk_σ.

[13] Since $\mathsf{rcv} \in \mathcal{L}_R$, the challenger already has computed $\mathsf{sk}_{\mathsf{rcv}}$.

3. If $\sigma \notin \mathcal{L}_S$ and $\mathsf{rcv} \notin \mathcal{L}_R$, it samples $\mathsf{ctxt} \leftarrow\!\$\ \{0,1\}^{\mathsf{msgLen}+\lambda}$ and updates $\mathcal{L}_E \leftarrow \mathcal{L}_E \cup \{(\sigma, \mathsf{rcv}, \mathsf{m}||\mathsf{k}, \mathsf{ctxt})\}$.

Let Fail be the event that Game_1 aborts if $(\mathsf{snd}, \mathsf{rcv}, g_1^r, e(X^r, \mathsf{u}_{\mathsf{rcv}}), e(\mathsf{sk}_\sigma, \mathsf{u}_{\mathsf{rcv}}), *) \in \mathcal{L}_{\hat{\mathsf{H}}}$ exists. Game_0 and Game_1 are identical unless Fail occurs. Therefore, we have

$$|\epsilon_1 - \epsilon_0| \leq \Pr[\mathsf{Fail}].$$

To estimate $\Pr[\mathsf{Fail}]$, we show that if $\mathcal{A}$ can trigger Fail, we can construct an adversary $\mathcal{B}_1$ that solves the BDH problem. The construction of $\mathcal{B}_1$ is as follows.

1. Upon receiving $(G = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), g_1^\alpha, g_2^\alpha, g_2^\beta, g_1^\gamma)$, $\mathcal{B}_1$ sets $X \coloneqq g_1^\alpha$ (i.e., msk is implicitly set α) and prepares the random oracles H_1, H_2, $\hat{\mathsf{H}}$, and G (i.e., initialize the lists $\mathcal{L}_{\mathsf{H}_1}$, $\mathcal{L}_{\mathsf{H}_2}$, $\mathcal{L}_{\hat{\mathsf{H}}}$, and $\mathcal{L}_{\mathsf{G}}$). Then, $\mathcal{B}_1$ samples $I \leftarrow\!\$\ [q_{\hat{\mathsf{H}}}]$ and executes $\mathcal{A}$ on input $\mathsf{mpk} \coloneqq (G, \mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}, \mathsf{G}, X)$.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}_1$ answers them as follows:
   (a) When $\mathcal{A}$ sends σ to H_1 oracle, $\mathcal{B}_1$ samples $b \leftarrow\!\$\ \mathbb{Z}_p$. With probability $1-\delta$, $\mathcal{B}_1$ computes $\mathsf{u}_\sigma = (g_1^\gamma)^b$, and updates $\mathcal{L}_{\mathsf{H}_1} \leftarrow \mathcal{L}_{\mathsf{H}_1} \cup \{(\sigma, \mathsf{u}_\sigma, b, 0)\}$. Otherwise, $\mathcal{B}_1$ computes $\mathsf{u}_\sigma \coloneqq g_1^b$ and updates $\mathcal{L}_{\mathsf{H}_1} \leftarrow \mathcal{L}_{\mathsf{H}_1} \cup \{(\sigma, \mathsf{u}_\sigma, b, 1)\}$. Then, $\mathcal{B}_1$ returns u_σ to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to H_2 oracle, $\mathcal{B}_1$ samples $\hat{b} \leftarrow\!\$\ \mathbb{Z}_p$. With probability $1-\delta$, $\mathcal{B}_1$ computes $\mathsf{u}_\rho \coloneqq (g_2^\beta)^{\hat{b}}$, and updates $\mathcal{L}_{\mathsf{H}_2} \leftarrow \mathcal{L}_{\mathsf{H}_2} \cup \{(\rho, \mathsf{u}_\rho, \hat{b}, 0)\}$. Otherwise, $\mathcal{B}_1$ computes $\mathsf{u}_\rho = g_2^{\hat{b}}$ and updates $\mathcal{L}_{\mathsf{H}_2} \leftarrow \mathcal{L}_{\mathsf{H}_2} \cup \{(\rho, \mathsf{u}_\rho, \hat{b}, 1)\}$. Then, $\mathcal{B}_1$ returns u_ρ to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends (σ, ρ, R, U, V) to $\hat{\mathsf{H}}$ oracle, if this is the I-th query to $\hat{\mathsf{H}}$, $\mathcal{B}_1$ checks if both $(\sigma, \mathsf{u}_\sigma, b, d) \in \mathcal{L}_{\mathsf{H}_1}$ and $(\rho, \mathsf{u}_\rho, \hat{b}, \hat{d}) \in \mathcal{L}_{\mathsf{H}_2}$ has coin $d = 0$ and $\hat{d} = 0$. If not, $\mathcal{B}_1$ aborts the game. Otherwise $(d = \hat{d} = 0)$, $\mathcal{B}_1$ outputs $D \coloneqq V^{\frac{1}{b\hat{b}}}$ as the solution of the BDH problem. If this is not the I-th query, $\mathcal{B}_1$ samples $Z \leftarrow\!\$\ \{0,1\}^{\mathsf{msgLen}}$ and updates $\mathcal{L}_{\hat{\mathsf{H}}} \leftarrow \mathcal{L}_{\hat{\mathsf{H}}} \cup \{(\sigma, \rho, R, U, V, Z)\}$. $\mathcal{B}_1$ returns Z to $\mathcal{A}$.
   (d) When $\mathcal{A}$ sends $(\sigma, \rho, \mathsf{m}, \mathsf{k})$ to G oracle, $\mathcal{B}_1$ samples $r \leftarrow\!\$\ \mathbb{Z}_p$ and updates $\mathcal{L}_{\mathsf{G}} \leftarrow \mathcal{L}_{\mathsf{G}} \cup \{(\sigma, \rho, \mathsf{m}, \mathsf{k}, r)\}$. Then, $\mathcal{B}_1$ returns r to $\mathcal{A}$.
   (e) When $\mathcal{A}$ sends $(\sigma, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_E$ oracle, it answers as in Game_1.
   (f) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ extracts $(\sigma, \mathsf{u}_\sigma, b, d)$ from $\mathcal{L}_{\mathsf{H}_1}$. If $d = 0$, $\mathcal{B}_1$ aborts the game. Otherwise, if $d = 1$, $\mathcal{B}_1$ computes $\mathsf{sk}_\sigma = (g_1^\alpha)^b$ and works as in Game_1.
   (g) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ extracts $(\rho, \mathsf{u}_\rho, \hat{b}, d)$ from $\mathcal{L}_{\mathsf{H}_2}$. If $d = 0$, $\mathcal{B}_1$ aborts the game. Otherwise, if $d = 1$, $\mathcal{B}_1$ computes $\mathsf{sk}_\rho = (g_2^\alpha)^{\hat{b}}$ and works as in Game_1.

Roughly, $\mathcal{B}_1$ guesses the identities and the $\hat{\mathsf{H}}$ query that causes the event Fail, and if $\mathcal{B}_1$ succeeds to guess, it perfectly simulates the $\mathsf{Auth\text{-}oCMA}$ game against $\mathcal{A}$. Let us estimate the probability that $\mathcal{B}_1$ succeeds to guess. The probability Fail occurs at the I-th $\hat{\mathsf{H}}$ query is $\frac{1}{q_{\hat{\mathsf{H}}}}$. The probability $\mathcal{O}_S$ and $\mathcal{O}_R$ do not abort is $\delta^{q_S+q_R}$. The probability the game does not abort when $\mathcal{A}$ sends the I-th $\hat{\mathsf{H}}$ query is $(1-\delta)^2$. Hence, the overall probability that $\mathcal{B}_1$ succeeds to guess is $\frac{1}{q_{\hat{\mathsf{H}}}} \cdot \delta^{q_S+q_R}(1-\delta)^2$. This value is maximum when $\hat{\delta} = 1 - \frac{2}{q_S+q_R+2}$, and thus the probability is at most $\frac{4}{\hat{e}^2(q_S+q_R)^2 q_{\hat{\mathsf{H}}}}$ for large $q_S + q_R$. Moreover, if $\mathcal{B}_1$ succeeds to guess, we know that $\mathsf{u}_\sigma = (g_1^\gamma)^b$ and $\mathsf{u}_\rho = (g_2^\beta)^{\hat{b}}$ if $\sigma \notin \mathcal{L}_S$ and $\rho \notin \mathcal{L}_R$, and thus

$$V = e(\mathsf{u}_\sigma, \mathsf{u}_\rho)^\alpha = e(g_1^{\gamma b}, g_2^{\beta\hat{b}})^\alpha = (e(g_1, g_2)^{\alpha\beta\gamma})^{b\hat{b}}.$$

$\mathcal{B}_1$ can solve the BDH problem correctly when it does not abort. Thus, we have

$$|\epsilon_1 - \epsilon_0| \leq \Pr[\mathsf{Fail}] \leq \frac{\hat{e}^2(q_S+q_R)^2 q_{\hat{\mathsf{H}}}}{4} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_1, \mathcal{G}}(\lambda).$$

Game_2. In this game, the challenger decrypts ctxt^* with a random $Z^* \leftarrow\!\$\ \{0,1\}^{\mathsf{msgLen}+\lambda}$ instead of $Z^* \coloneqq \hat{\mathsf{H}}(\mathsf{snd}^*, \rho^*, R^*, e(R^*, \mathsf{sk}_{\rho^*}), e(\mathsf{H}_1(\mathsf{snd}^*), \mathsf{sk}_{\rho^*}))$.

Let BadQ be the event that $\mathcal{A}$ makes a query $(\sigma^*, \rho^*, \cdot, \cdot, V^*)$ to the oracle $\hat{\mathsf{H}}$ where $V^* \coloneqq e(\mathsf{u}_{\sigma^*}, \mathsf{u}_{\rho^*})^x$. Since Z^* is now chosen independently from random oracles, $\mathcal{A}$ notices the difference between the two games if BadQ occurs and otherwise the two games proceed identically. Thus, we have

To estimate $\Pr[\mathsf{BadQ}]$, we show that if $\mathcal{A}$ triggers BadQ, we can construct an adversary $\mathcal{B}_2$ that solves the BDH problem. The construction of $\mathcal{B}_2$ is as follows.

1. Upon receiving $(G = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), g_1^\alpha, g_2^\alpha, g_2^\beta, g_1^\gamma)$, $\mathcal{B}_2$ sets $X \coloneqq g_1^\alpha$ (i.e., msk is implicitly set α) and prepares three random oracles H_1, H_2, $\hat{\mathsf{H}}$, and G (i.e., initialize the lists $\mathcal{L}_{\mathsf{H}_1}$, $\mathcal{L}_{\mathsf{H}_2}$, $\mathcal{L}_{\hat{\mathsf{H}}}$, and $\mathcal{L}_{\mathsf{G}}$). Then, $\mathcal{B}_2$ executes $\mathcal{A}$ on input $\mathsf{mpk} \coloneqq (G, \mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}, \mathsf{G}, X)$.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}_2$ answers them as follows:
   (a) When $\mathcal{A}$ sends σ to H_1 oracle, $\mathcal{B}_2$ samples $b \leftarrow\!\$\ \mathbb{Z}_p$. With probability $1-\delta$, $\mathcal{B}_2$ computes $\mathsf{u}_\sigma = (g_1^\gamma)^b$ and updates $\mathcal{L}_{\mathsf{H}_1} \leftarrow \mathcal{L}_{\mathsf{H}_1} \cup \{(\sigma, \mathsf{u}_\sigma, b, 0)\}$. Otherwise, $\mathcal{B}_2$ computes $\mathsf{u}_\sigma \coloneqq g_1^b$ and updates $\mathcal{L}_{\mathsf{H}_1} \leftarrow \mathcal{L}_{\mathsf{H}_1} \cup \{(\sigma, \mathsf{u}_\sigma, b, 1)\}$. Then, $\mathcal{B}_2$ returns u_σ to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to H_2 oracle, $\mathcal{B}_2$ samples $\hat{b} \leftarrow\!\$\ \mathbb{Z}_p$. With probability $1-\delta$, $\mathcal{B}_2$ computes $\mathsf{u}_\rho \coloneqq (g_2^\beta)^{\hat{b}}$ and updates $\mathcal{L}_{\mathsf{H}_2} \leftarrow \mathcal{L}_{\mathsf{H}_2} \cup \{(\rho, \mathsf{u}_\rho, \hat{b}, 0)\}$. Otherwise, $\mathcal{B}_2$ computes $\mathsf{u}_\rho = g_2^{\hat{b}}$ and updates $\mathcal{L}_{\mathsf{H}_2} \leftarrow \mathcal{L}_{\mathsf{H}_2} \cup \{(\rho, \mathsf{u}_\rho, \hat{b}, 1)\}$. Then, $\mathcal{B}_2$ returns u_ρ to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends (σ, ρ, R, U, V) to $\hat{\mathsf{H}}$ oracle, $\mathcal{B}_2$ samples $Z \leftarrow\!\$\ \{0,1\}^{\mathsf{msgLen}}$ and updates $\mathcal{L}_{\hat{\mathsf{H}}} \leftarrow \mathcal{L}_{\hat{\mathsf{H}}} \cup \{(\sigma, \rho, R, U, V, Z)\}$. Then, $\mathcal{B}_2$ returns Z to $\mathcal{A}$.
   (d) When $\mathcal{A}$ sends $(\sigma, \rho, \mathsf{m}, \mathsf{k})$ to G oracle, $\mathcal{B}_2$ samples $r \leftarrow\!\$\ \mathbb{Z}_p$ and updates $\mathcal{L}_{\mathsf{G}} \leftarrow \mathcal{L}_{\mathsf{G}} \cup \{(\sigma, \rho, \mathsf{m}, \mathsf{k}, r)\}$. Then, $\mathcal{B}_2$ returns r to $\mathcal{A}$.
   (e) When $\mathcal{A}$ sends $(\sigma, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_E$ oracle, it answers as in the previous game.
   (f) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ extracts $(\sigma, \mathsf{u}_\sigma, b, d)$ from $\mathcal{L}_{\mathsf{H}_1}$. If $d = 0$, $\mathcal{B}_2$ aborts the game. Otherwise (that is, $d = 1$), $\mathcal{B}_2$ computes $\mathsf{sk}_\sigma = (g_1^\alpha)^b$ and returns $\mathsf{ek}_\sigma \coloneqq (\sigma, \mathsf{sk}_\sigma)$ to $\mathcal{A}$.
   (g) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ extracts $(\rho, \mathsf{u}_\rho, \hat{b}, d)$ from $\mathcal{L}_{\mathsf{H}_2}$. If $d = 0$, $\mathcal{B}_2$ aborts the game. Otherwise (that is, $d = 1$), $\mathcal{B}_2$ computes $\mathsf{sk}_\rho = (g_2^\alpha)^{\hat{b}}$ and return $\mathsf{dk}_\rho \coloneqq (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
3. $\mathcal{A}$ outputs $(\mathsf{snd}^*, \rho^*, \mathsf{ct}^* \coloneqq (R^*, \mathsf{ctxt}^*))$. $\mathcal{B}_2$ sets $\sigma^* \coloneqq \mathsf{snd}^*$. If both $(\sigma^*, \mathsf{u}_{\sigma^*}, b^*, d^*) \in \mathcal{L}_{\mathsf{H}_1}$ and $(\rho^*, \mathsf{u}_{\rho^*}, \hat{b}^*, \hat{d}^*) \in \mathcal{L}_{\mathsf{H}_2}$ do not have coins $d^* = 0$ and $\hat{d}^* = 0$, $\mathcal{B}_2$ aborts the game. Otherwise, $\mathcal{B}_2$ picks an entry $(\sigma^*, \rho^*, R^*, U, V, \hat{h}) \in \mathcal{L}_{\hat{\mathsf{H}}}$ at random, and outputs $D \coloneqq V^{\frac{1}{b^*\hat{b}^*}}$ as the solution of the BDH problem.

We can see that $\mathcal{B}_2$ perfectly simulates the $\mathsf{Auth\text{-}oCMA}$ game if $\mathcal{B}_2$ does not abort. Let us estimate the probability $\Pr[\neg\mathsf{abort}]$. The probability $\mathcal{O}_S$ and $\mathcal{O}_R$ do not abort is $\delta^{q_S+q_R}$. The probability the game does not abort when $\mathcal{A}$ outputs a forgery is $(1-\delta)^2$. Hence, the overall non-aborting probability is $\delta^{q_S+q_R}(1-\delta)^2$. This value is maximum when $\hat{\delta} = \frac{q_S+q_R}{q_S+q_R+2}$, and thus $\Pr[\neg\mathsf{abort}] \leq \frac{4}{\hat{e}^2(q_S+q_R)^2}$ for large $q_S + q_R$. Moreover, we know that $\mathsf{u}_{\sigma^*} = (g_1^\gamma)^{b^*}$, $\mathsf{u}_{\rho^*} = (g_2^\beta)^{\hat{b}^*}$, and thus

$$V^* = e(\mathsf{u}_{\sigma^*}, \mathsf{u}_{\rho^*})^\alpha = e(g_1^{\gamma b^*}, g_2^{\beta\hat{b}^*})^\alpha = (e(g_1, g_2)^{\alpha\beta\gamma})^{b^*\hat{b}^*}.$$

If $\mathcal{A}$ can distinguish the two games, $\mathcal{A}$ has queried $\hat{\mathsf{H}}(\sigma^*, \mathsf{rcv}^*, \cdot, \cdot, V^*)$, and thus $\mathcal{B}_2$ can solve the BDH problem correctly with probability at least $\frac{1}{q_{\hat{\mathsf{H}}}}$. Therefore,

$$|\epsilon_2 - \epsilon_1| \leq \Pr[\mathsf{BadQ}] \leq \frac{\hat{e}^2(q_S+q_R)^2 q_{\hat{\mathsf{H}}}}{4} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B}_2,\mathcal{G}}(\lambda).$$

Game_3. In this game, the challenger checks if $\mathsf{G}(\mathsf{snd}^*, \rho^*, \mathsf{m}^*, \mathsf{k}^*)$ has been queried, and if so, it aborts the game. Otherwise, it samples $r^* \leftarrow\!\$\ \mathbb{Z}_p$ at random instead of generating it with G. Since $\mathsf{m}^*||\mathsf{k}^*$ is chosen independently at random, the probability $\mathsf{G}(\mathsf{m}^*, \mathsf{k}^*, \mathsf{snd}^*, \rho^*)$ was queried is $\frac{q_\mathsf{G}}{2^{\mathsf{msgLen}+\lambda}}$, and thus we have

$$|\epsilon_3 - \epsilon_2| \leq \frac{q_\mathsf{G}}{2^{\mathsf{msgLen}+\lambda}}.$$

We finally evaluate ϵ_3. In Game_3, $\mathcal{A}$ breaks the $\mathsf{Auth\text{-}oCMA}$ security if $R^* = g_1^{r^*}$ holds for randomly chosen $r^* \in \mathbb{Z}_p$. Since for any $R \in \mathbb{G}_1$ the probability that $R^* = g_1^{r^*}$ holds for a randomly chosen $r^* \in \mathbb{Z}_p$ is $\frac{1}{p}$, we have

$$\epsilon_3 = \frac{1}{p}.$$

Putting everything together and folding both adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ into one adversary $\mathcal{B}$, we obtain

$$\mathsf{Adv}^{\mathsf{auth\text{-}ocma}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda) \leq \frac{\hat{e}^2(q_S+q_R)^2 q_{\hat{\mathsf{H}}}}{4} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda) + \frac{q_\mathsf{G}}{2^{\mathsf{msgLen}+\lambda}} + \frac{1}{p}.$$

□

Table 2: Comparison of the IB-ME schemes from the BDH assumption in the ROM. The column "Ciphertext" indicates the difference between the length of ciphertext and that of plaintext. $|\mathbb{G}_1|$, $|\mathbb{G}_2|$ and $|\mathbb{G}_T|$ denotes the size of respective group elements.

Schemes	Security		Space complexity		
	Priv	Auth	Enc. key	Dec. key	Ciphertext
Ateniese et al. [3]	CPA	oNMA	$\lvert\mathbb{G}_1\rvert$	$3\lvert\mathbb{G}_2\rvert$	$2\lvert\mathbb{G}_1\rvert + \lambda$
Boyen and Li [7] (IBE [6]+IBS [10])	CPA$^+$	iCMA	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_2\rvert$	$3\lvert\mathbb{G}_1\rvert + 3\lambda$
IB-ME$^{\mathsf{BDH}}$	CCA$^+$	oCMA	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_2\rvert$	$\lvert\mathbb{G}_1\rvert + \lambda$

4.3 Comparison

We compare IB-ME$^{\mathsf{BDH}}$ with the Ateniese et al. scheme and the Boyen and Li scheme. Table 2 summarizes their properties. Among them, IB-ME$^{\mathsf{BDH}}$ is the best in terms of key and ciphertext sizes, as they are only one group element. In addition, it achieves stronger Priv$^+$-CCA and Auth-oCMA security. In this sense, IB-ME$^{\mathsf{BDH}}$ improves upon the Ateniese et al. scheme in both security guarantees and space complexity within the ROM under the BDH assumption. Compared with the Boyen and Li scheme, IB-ME$^{\mathsf{BDH}}$ offers better space complexity. Regarding security properties, IB-ME$^{\mathsf{BDH}}$ has stronger Priv$^+$-CCA security, but weaker oCMA authenticity. The authenticity guarantee of IB-ME$^{\mathsf{BDH}}$ is limited to outsiders; an adversary holding the target receiver's decryption key can generate valid ciphertexts for arbitrary claimed senders. We conclude that this is a trade-off between security and efficiency.

5 New Generic Construction of IB-ME Scheme in the StdM

This section shows a new generic construction of IB-ME based on IBE, signature scheme (SIG), NIZK proof system and reusable extractors, which we call IB-ME$^{\mathsf{IBE+SIG}}$. To achieve enhanced privacy, we use anonymous IBE, mask IBE ciphertexts with outputs of a reusable extractor, and hide the sender's information using NIZK proofs as Francati et al. [17] does. As a result, IBE ciphertext looks random if the adversary does not know either sender's identity or the receiver's decryption key.

5.1 Construction

To construct an IB-ME scheme with identity space $\mathcal{ID} = \{0,1\}^*$ and message space $\mathcal{M} = \{0,1\}^{\mathsf{msgLen}}$, we use the following building blocks.

- An IBE scheme $\mathsf{IBE} = (\mathsf{IBE.Setup}, \mathsf{IBE.KGen}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ with $\mathcal{ID}_{\mathsf{IBE}} = \{0,1\}^*$ and its ciphertext space is $\mathcal{CT}_{\mathsf{IBE}}$.
- A signature scheme $\mathsf{SIG} = (\mathsf{SIG.KGen}, \mathsf{SIG.Sign}, \mathsf{SIG.Ver})$.
- A reusable extractor $\mathsf{Ext} : \{0,1\}^{\mathsf{seedLen}} \times \mathcal{ID} \to \{0,1\}^{\mathsf{ctLen}}$. We assume that IBE ciphertexts in $\mathcal{CT}_{\mathsf{IBE}}$ are encoded as fixed-length bit strings in $\{0,1\}^{\mathsf{ctLen}}$, and Ext outputs elements in $\{0,1\}^{\mathsf{ctLen}}$.
- An NIZK argument $\mathsf{NIZK} = (\mathsf{NIZK.Setup}, \mathsf{NIZK.Prove}, \mathsf{NIZK.Verify})$ for the following NP relation:

$$R = \left\{ \begin{array}{c} ((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \\ (\mathsf{m}, r, \mathsf{snd}, \mathsf{rcv}, \mathsf{sig}_{\mathsf{snd}})) \end{array} : \begin{array}{c} \hat{\mathsf{ct}} = \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m} \| \mathsf{snd} \| \mathsf{rcv}; r) \oplus \mathsf{Ext}(s, \mathsf{snd}) \\ \wedge \mathsf{SIG.Ver}(\mathsf{mpk}_{\mathsf{SIG}}, \mathsf{snd}, \mathsf{sig}_{\mathsf{snd}}) = 1 \end{array} \right\}$$

We require the following compatibility conditions among the building blocks. First, IBE ciphertexts have fixed-length encodings as bit strings in $\{0,1\}^{\mathsf{ctLen}}$. Second, $\mathsf{IBE.Dec}$ is defined on all ctLen-bit strings and returns either a message in the prescribed message space or $\perp$. Third, the reusable extractor Ext outputs ctLen-bit strings, so that the masking operation $\hat{\mathsf{ct}} = \mathsf{ct}_{\mathsf{IBE}} \oplus \mathsf{Ext}(s, \mathsf{snd})$ is well-defined over fixed-length encodings. Fourth, the algorithms $\mathsf{IBE.Enc}$, Ext, and $\mathsf{SIG.Verify}$ are efficiently computable and can be represented inside the NP relation handled by the NIZK system. Our generic construction should be understood under these compatibility requirements.

The proposed IB-ME scheme $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ is as follows. Its identity and message spaces are $\{0,1\}^*$ and $\{0,1\}^{\mathsf{msgLen}}$, respectively.

$\mathsf{Setup}(1^\lambda)$**:** It computes $(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$, $(\mathsf{mpk}_{\mathsf{SIG}}, \mathsf{msk}_{\mathsf{SIG}}) \leftarrow \mathsf{SIG.KGen}(1^\lambda)$ and $\mathsf{crs} \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$, and outputs $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{SIG}}, \mathsf{crs})$ and $\mathsf{msk} := (\mathsf{msk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{SIG}})$.
$\mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma)$**:** It computes $\mathsf{sig}_\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma)$ and outputs $\mathsf{ek}_\sigma = (\sigma, \mathsf{sig}_\sigma)$.
$\mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho)$**:** It computes $\mathsf{sk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho)$ and outputs $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$.
$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma = (\sigma, \mathsf{sig}_\sigma), \mathsf{rcv}, \mathsf{m})$**:** It samples $s \leftarrow\!\!\$\ \{0,1\}^{\mathsf{seedLen}}$ and computes $Z := \mathsf{Ext}(s, \sigma)$, $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m} \| \sigma \| \mathsf{rcv}; r)$, $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$, $\pi \leftarrow \mathsf{NIZK.Prove}(\mathsf{crs}, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), (\mathsf{m}, r, \sigma, \mathsf{rcv}, \mathsf{sig}_\sigma))$. It outputs $\mathsf{ct} := (s, \hat{\mathsf{ct}}, \pi)$.
$\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho = (\rho, \mathsf{sk}_\rho), \mathsf{snd}, \mathsf{ct} = (s, \hat{\mathsf{ct}}, \pi))$**:** If $\mathsf{NIZK.Verify}(\mathsf{crs}, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi) = 0$, it outputs $\perp$. Otherwise, it computes $\mathsf{ct}_{\mathsf{IBE}} := \hat{\mathsf{ct}} \oplus \mathsf{Ext}(s, \mathsf{snd})$ and $\mathsf{m}' \| \mathsf{snd}' \| \mathsf{rcv}' \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{sk}_\rho, \mathsf{ct}_{\mathsf{IBE}})$. If $(\mathsf{snd}, \rho) = (\mathsf{snd}', \mathsf{rcv}')$, it outputs m'; otherwise, it outputs $\perp$.

The plaintext encrypted by the underlying IBE scheme contains both the sender and receiver identities. This binding is crucial for CCA security. Even if the reusable extractor has pointwise collisions on fixed identities, a challenge ciphertext submitted under a different claimed sender identity will not decrypt successfully. Indeed, after unmasking and IBE decryption, the Dec algorithm outputs a message only if the identities recovered from the IBE plaintext match the identities supplied to Dec. Thus, a ciphertext generated for a pair (σ, rcv) cannot be reinterpreted as a ciphertext for another pair (σ', ρ), even when $\mathsf{Ext}(s, \sigma) = \mathsf{Ext}(s, \sigma')$ for some public seed s.

Correctness. We can verify that $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ is correct with negligible correctness errors. First, the receiver accepts a correctly generated NIZK proof. Then, under condition $\mathsf{snd} = \sigma \wedge \mathsf{rcv} = \rho$ and the correctness of the IBE scheme, for any messages and seeds, the decrypted value satisfies $(\mathsf{m}', \mathsf{snd}', \mathsf{rcv}') = (\mathsf{m}, \sigma, \mathsf{rcv})$. Therefore, the decryption algorithm finally outputs the encrypted message m with a probability of all but negligible.

5.2 Security Proofs

We can show that $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ is $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ and Auth-iCMA secure.

Theorem 3. *If there exists an adversary $\mathcal{A}$ that breaks the* Priv^+*-CCA security of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$*, there exists an adversary $\mathcal{B}_1$ that breaks the zero-knowledge property of* NIZK*, an adversary $\mathcal{B}_2$ that breaks the simulation-sound extractability of* NIZK*, an adversary $\mathcal{B}_3$ that breaks the* ANO-IND-ID-CPA *security of* IBE *and an adversary $\mathcal{B}_4$ that breaks the security of* Ext *such that*

$$\mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) \le \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda) + \begin{cases} q_D \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3,\mathsf{IBE}}(\lambda) & \textit{(Case 1)} \\ 2\mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_4,\mathsf{Ext}}(\lambda) & \textit{(Case 2)} \\ q_D \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3,\mathsf{IBE}}(\lambda) + \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_4,\mathsf{Ext}}(\lambda). & \textit{(Case 3)} \\ q_D \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3,\mathsf{IBE}}(\lambda) + \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_4,\mathsf{Ext}}(\lambda) & \textit{(Case 4)} \end{cases}$$

where q_D is the maximum number of queries $\mathcal{A}$ makes to the decryption oracle.

The $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ (Theorem 3) is shown by Lemmata 5 to 8.

Lemma 5. *If there exists an adversary $\mathcal{A}$ that breaks the* Priv^+*-CCA security of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ *in Case 1, there exists an adversary $\mathcal{B}_1$ that breaks the zero-knowledge property of* NIZK*, an adversary $\mathcal{B}_2$ that breaks the simulation-sound extractability of* NIZK*, and an adversary $\mathcal{B}_3$ that breaks the* ANO-IND-ID-CPA *security of* IBE *such that*

$$\left| \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 1} \right] - \frac{1}{2} \right| \le \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda) + q_D \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3,\mathsf{IBE}}(\lambda),$$

where q_D is the maximum number of queries $\mathcal{A}$ makes to the decryption oracle.

Proof. Let $\mathcal{A}$ be an adversary that breaks the Priv^+-CCA security of $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$. To prove the theorem, we consider the following sequence of games Game_i. Define the advantage of $\mathcal{A}$ in Game_i as

$$\epsilon_i := \left|\Pr\left[\mathsf{Game}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right|.$$

Game_0 : This is the original security game in Case 1. Thus, we have

$$\epsilon_0 = \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case 1}\right] - \frac{1}{2}\right|.$$

Game_1 : In this game, we change how to generate the NIZK proof in the challenge ciphertext. In Game_0, the challenger uses Prove algorithm to generates π with $\mathsf{crs} \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$, but in this game, the challenger executes the simulator $\mathcal{S}_2$ of NIZK and generates π using the trapdoor τ generated by $\mathcal{S}_1$.

To show Game_0 and Game_1 are indistinguishable, we construct $\mathcal{B}_1$ that breaks the zero-knowledge of NIZK using $\mathcal{A}$. The description of $\mathcal{B}_1$ as follows:

1. Upon receiving crs, $\mathcal{B}_1$ computes $(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and $(\mathsf{mpk}_{\mathsf{SIG}}, \mathsf{msk}_{\mathsf{SIG}}) \leftarrow \mathsf{SIG.KGen}(1^\lambda)$. Then, $\mathcal{B}_1$ executes $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{SIG}}, \mathsf{crs})$. In addition, $\mathcal{B}_1$ chooses $\mathsf{coin} \leftarrow\!\!\$\ \{0,1\}$ and keeps it secret.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}_1$ answers as follows:
   - When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ computes $\mathsf{sig}_\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma)$ and returns $\mathsf{ek}_\sigma = (\sigma, \mathsf{sig}_\sigma)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ computes $\mathsf{sk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho)$, and returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(\mathsf{snd}, \rho, \mathsf{ct})$ to $\mathcal{O}_D$ oracle, if $\mathsf{NIZK.Verify}(\mathsf{crs}, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi) = 0$, $\mathcal{B}_1$ outputs $\perp$. Otherwise, $\mathcal{B}_1$ computes $\mathsf{sk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho)$ and $\mathsf{ct}_{\mathsf{IBE}} := \hat{\mathsf{ct}} \oplus \mathsf{Ext}(s, \mathsf{snd})$ and $\mathsf{m}' \| \mathsf{snd}' \| \mathsf{rcv}' \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{sk}_\rho, \mathsf{ct}_{\mathsf{IBE}})$. If $(\mathsf{snd}', \mathsf{rcv}') = (\mathsf{snd}, \rho)$, $\mathcal{B}_1$ outputs m'; otherwise, it outputs $\perp$.
   - When $\mathcal{A}$ outputs $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_1$ samples $\sigma_i^* \leftarrow\!\!\$\ \Sigma_i^*$ for each $i \in \{0,1\}$ and computes $\mathsf{sig}_{\sigma_0^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_0^*)$ and $\mathsf{sig}_{\sigma_1^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_1^*)$. $\mathcal{B}_1$ samples $s \leftarrow\!\!\$\ \{0,1\}^{\mathsf{seedLen}}$ and $r \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{IBE}}$, and computes $Z := \mathsf{Ext}(s, \sigma^*_{\mathsf{coin}})$. Next, $\mathcal{B}_1$ computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{m}^*_{\mathsf{coin}} \| \sigma^*_{\mathsf{coin}} \| \mathsf{rcv}^*_{\mathsf{coin}}; r)$ and $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$. Then, $\mathcal{B}_1$ sends $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), (\mathsf{m}^*_{\mathsf{coin}}, r, \sigma^*_{\mathsf{coin}}, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{sig}_{\sigma^*_{\mathsf{coin}}}))$ to its challenger and receives a NIZK proof π. Finally, $\mathcal{B}_1$ returns $\mathsf{ct}_{\mathsf{coin}} := (s, \hat{\mathsf{ct}}, \pi)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$ oracle, $\mathcal{B}_1$ samples $s \leftarrow\!\!\$\ \{0,1\}^{\mathsf{seedLen}}$ and $r \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{IBE}}$, and computes $Z := \mathsf{Ext}(s, \sigma_i^*)$. Next, $\mathcal{B}_1$ computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m} \| \sigma_i^* \| \mathsf{rcv}; r)$ and $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$. Then, $\mathcal{B}_1$ sends $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), (\mathsf{m}, r, \sigma_i^*, \mathsf{rcv}, \mathsf{sig}_{\sigma_i^*}))$ to its challenger and receives a NIZK proof π. Finally, $\mathcal{B}_1$ returns $\mathsf{ct}^* := (s, \hat{\mathsf{ct}}, \pi)$ to $\mathcal{A}$.
3. When $\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0,1\}$, $\mathcal{B}_1$ outputs 1 (if $\mathsf{coin} = \widehat{\mathsf{coin}}$) or 0 (otherwise) to its challenger.

It is clear that if π is generated with $\mathsf{NIZK.Prove}$, then $\mathcal{B}_1$ perfectly simulates Game_0 for $\mathcal{A}$. On the other hand, if π is generated with $\mathcal{S}_2$, then $\mathcal{B}_1$ perfectly simulates Game_1. Therefore, we have

$$|\epsilon_0 - \epsilon_1| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1, \mathsf{NIZK}}(\lambda).$$

Game_2. In this game, we change how to generate CRS in the setup phase. Specifically, the challenger executes $\mathcal{E}_1$ to generate a CRS of NIZK instead of $\mathcal{S}_1$.

From the simulation-sound extractability of NIZK, the output of $\mathcal{E}_1(1^\lambda)$ is identically distributed to that of $\mathcal{S}_1(1^\lambda)$ on the first two components (crs, τ). Therefore, Game_1 and Game_2 are identical, and it holds that

$$\epsilon_1 = \epsilon_2.$$

Game_3**.** In this game, we change how the challenger responds to the decryption queries. In Game_2, the challenger computes $\mathsf{m}'\|\mathsf{snd}'\|\mathsf{rcv}' \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{sk}_\rho, \mathsf{ct}_{\mathsf{IBE}})$, but in this game, the challenger runs $(\mathsf{m}, r, \mathsf{snd}', \mathsf{rcv}', \mathsf{sig}_\sigma) \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi)$. Further, the challenger maintains a list $\mathcal{L}_{\mathsf{NIZK}}$ consisting of tuples $(x, \pi, \mathsf{m}, \mathsf{snd}, \mathsf{rcv})$ where $x = (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}})$, for ciphertexts whose proofs were generated by the NIZK simulator.

Let Forge be an event that $\mathcal{A}$ makes a decryption query that satisfies $\mathsf{NIZK.Verify}(\mathsf{crs}, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi) = 1$ but the statement $x = (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}})$ and the witness $w = (\mathsf{m}, r, \mathsf{snd}', \mathsf{rcv}', \mathsf{sig}_\sigma)$ does not satisfy $(x, w) \in R$. If Forge does not occur, then any query that fails verification is rejected in both games. Further, any query that passes the check satisfies $\hat{ct} = \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}', \mathsf{m}\|\mathsf{snd}'\|\mathsf{rcv}'; r) \oplus \mathsf{Ext}(s, \mathsf{snd}')$, and thus the decryption oracle returns non-$\perp$ only when $(\mathsf{snd}, \rho) = (\mathsf{snd}', \mathsf{rcv}')$ holds. Therefore, because IBE is correct, Game_2 and Game_3 return the same result and proceed identically with negligible probability. Thus, we have,

$$|\epsilon_2 - \epsilon_3| \leq \Pr[\mathsf{Forge}].$$

To estimate $\Pr[\mathsf{Forge}]$, we construct $\mathcal{B}_2$ that breaks the simulation-sound extractability of NIZK using $\mathcal{A}$. The description of $\mathcal{B}_2$ as follows:

1. Upon receiving crs, $\mathcal{B}_2$ computes $(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and $(\mathsf{mpk}_{\mathsf{SIG}}, \mathsf{msk}_{\mathsf{SIG}}) \leftarrow \mathsf{SIG.KGen}(1^\lambda)$. Next, $\mathcal{B}_2$ prepares $\mathcal{L}_{\mathsf{Dec}} := \emptyset$ and $\mathcal{L}_{\mathsf{NIZK}} := \emptyset$. Then, $\mathcal{B}_2$ executes $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{SIG}}, \mathsf{crs})$.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}_2$ answers as follows:
   - When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ computes $\mathsf{sig}_\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma)$ and returns $\mathsf{ek}_\sigma = (\sigma, \mathsf{sig}_\sigma)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ computes $\mathsf{sk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho)$, and returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(\mathsf{snd}, \rho, \mathsf{ct})$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_2$ outputs $\perp$ if $\mathsf{NIZK.Verify}(\mathsf{crs}, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi) = 0$. Otherwise, if $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi, (\mathsf{m}', \mathsf{snd}', \mathsf{rcv}')) \in \mathcal{L}_{\mathsf{NIZK}}$, if $(\mathsf{snd}, \rho) = (\mathsf{snd}', \mathsf{rcv}')$, it returns m'; otherwise it returns $\perp$. Else, $\mathcal{B}_2$ computes $\mathsf{sk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho)$ and $\mathsf{ct}_{\mathsf{IBE}} := \hat{\mathsf{ct}} \oplus \mathsf{Ext}(s, \mathsf{snd})$ and $\mathsf{m}'\|\mathsf{snd}'\|\mathsf{rcv}' \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{sk}_\rho, \mathsf{ct}_{\mathsf{IBE}})$. If $(\mathsf{snd}', \mathsf{rcv}') \neq (\mathsf{snd}, \rho)$, $\mathcal{B}_2$ returns $\perp$; otherwise, $\mathcal{B}_2$ returns m' to $\mathcal{A}$ and updates $\mathcal{L}_{\mathsf{Dec}} := \mathcal{L}_{\mathsf{Dec}} \cup \{((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi)\}$.
   - When $\mathcal{A}$ outputs $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_2$ samples $\sigma_i^* \leftarrow\!\!\$\ \Sigma_i^*$ for each $i \in \{0, 1\}$ and computes $\mathsf{sig}_{\sigma_0^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_0^*)$ and $\mathsf{sig}_{\sigma_1^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_1^*)$. $\mathcal{B}_2$ samples $s \leftarrow\!\!\$\ \{0, 1\}^{\mathsf{seedLen}}$ and $r \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{IBE}}$, and computes $Z := \mathsf{Ext}(s, \sigma^*_{\mathsf{coin}})$. Next, $\mathcal{B}_2$ computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{m}^*_{\mathsf{coin}}\|\sigma^*_{\mathsf{coin}}\|\mathsf{rcv}^*_{\mathsf{coin}}; r)$ and $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$. Then, $\mathcal{B}_2$ sends $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), (\mathsf{m}^*_{\mathsf{coin}}, r, \sigma^*_{\mathsf{coin}}, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{sig}_{\sigma^*_{\mathsf{coin}}}))$ to its challenger and receives a NIZK proof π. $\mathcal{B}_2$ stores $x = (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}})$, π, and $(\mathsf{m}^*_{\mathsf{coin}}, \sigma^*_{\mathsf{coin}}, \mathsf{rcv}^*_{\mathsf{coin}})$ into the list $\mathcal{L}_{\mathsf{NIZK}}$ and returns $\mathsf{ct}_{\mathsf{coin}} := (s, \hat{\mathsf{ct}}, \pi)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$ oracle, $\mathcal{B}_2$ samples $s \leftarrow\!\!\$\ \{0, 1\}^{\mathsf{seedLen}}$ and $r \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{IBE}}$, and computes $Z := \mathsf{Ext}(s, \sigma_i^*)$. Next, $\mathcal{B}_2$ computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m}\|\sigma_i^*\|\mathsf{rcv}; r)$ and $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$. Then, $\mathcal{B}_2$ sends $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), (\mathsf{m}, r, \sigma_i^*, \mathsf{rcv}, \mathsf{sig}_{\sigma_i^*}))$ to its challenger and receives a NIZK proof π. $\mathcal{B}_2$ stores $x = (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}})$, π, and $(\mathsf{m}, \sigma_i^*, \mathsf{rcv})$ into the list $\mathcal{L}_{\mathsf{NIZK}}$ and returns $\mathsf{ct}^* := (s, \hat{\mathsf{ct}}, \pi)$ to $\mathcal{A}$.
3. When $\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0, 1\}$, $\mathcal{B}_2$ chooses an element $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi)$ from $\mathcal{L}_{\mathsf{Dec}}$ at random and outputs it to its challenger.

In the reduction, $\mathcal{B}_2$ does not know in advance which decryption query of $\mathcal{A}$ will correspond to a witness outside the language. So, it stores all valid pairs $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi)$ submitted to the decryption oracle in the list $\mathcal{L}_{\mathsf{Dec}}$ and, at the end, outputs an element chosen uniformly at random from this list. Consequently, when $\mathcal{A}$ triggers Forge, $\mathcal{B}_2$ outputs $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi)$ that satisfies $\mathsf{NIZK.Verify}(\mathsf{crs}, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi) = 1$, but the statement and the extracted witness do not satisfy the relation with probability $\frac{1}{q_D}$ where q_D is the maximum number of queries $\mathcal{A}$ makes to the decryption oracle. Note that, since queries whose proof was generated by the simulation oracle are handled using $\mathcal{L}_{\mathsf{NIZK}}$ and are never passed to the extractor, any

proof output by $\mathcal{B}_2$ satisfies the condition $(x, \pi) \notin \mathcal{L}$ required by the simulation-sound extractability game. Therefore, we have

$$|\epsilon_2 - \epsilon_3| \leq \Pr[\mathsf{Forge}] \leq q_D \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda).$$

We now justify the equivalence between the real decryption oracle and the extraction-based simulated oracle. Let $\mathsf{ct} = (s, \hat{\mathsf{ct}}, \pi)$ be a decryption query with claimed identities (snd, ρ), and let $x = (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}})$. If $\mathsf{NIZK.Verify}(\mathsf{crs}, x, \pi) = 0$, both oracles return $\perp$. If (x, π) was generated by the simulator, the modified oracle answers using the stored tuple in $\mathcal{L}_{\mathsf{NIZK}}$, and this is exactly the message and identities used when the simulated proof was created. Otherwise, if the extractor outputs $(\mathsf{m}', r', \mathsf{snd}', \mathsf{rcv}', \mathsf{sig}')$ such that $(x, (\mathsf{m}', r', \mathsf{snd}', \mathsf{rcv}', \mathsf{sig}')) \in R$, then $\hat{\mathsf{ct}} = \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}', \mathsf{m}' \| \mathsf{snd}' \| \mathsf{rcv}'; r') \oplus \mathsf{Ext}(s, \mathsf{snd}')$. Therefore, when $(\mathsf{snd}, \rho) = (\mathsf{snd}', \mathsf{rcv}')$, the real decryption algorithm unmasks the same IBE ciphertext and returns m' by correctness of IBE. When $(\mathsf{snd}, \rho) \neq (\mathsf{snd}', \mathsf{rcv}')$, the identity check in Dec rejects. Hence the real decryption oracle and the extraction-based simulated oracle differ only if an accepting proof outside the simulated-proof table yields an extracted witness not satisfying R, which is exactly the Forge event.

Game_4. In this game, we modify the way the challenge ciphertext is generated. In Game_4, the challenge ciphertext is computed as follows: When $\mathcal{A}$ outputs $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, the challenger samples $\sigma_0^* \leftarrow\!\$\ \Sigma_0^*$ and $\sigma_1^* \leftarrow\!\$\ \Sigma_1^*$, and computes $\mathsf{sig}_{\sigma_0^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_0^*)$ and $\mathsf{sig}_{\sigma_1^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_1^*)$. Then it samples $s^* \leftarrow\!\$\ \{0,1\}^{\mathsf{seedLen}}$ and computes $Z^* := \mathsf{Ext}(s^*, \sigma^*_{\mathsf{coin}})$. Next, the challenger samples $\mathsf{ct}_{\mathsf{IBE}} \leftarrow\!\$\ \mathcal{CT}_{\mathsf{IBE}}$ and computes $\hat{\mathsf{ct}}^* := \mathsf{ct}^*_{\mathsf{IBE}} \oplus Z^*$ and $\pi^* \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}}))$. Finally, $\mathsf{ct}_{\mathsf{coin}} := (s^*, \hat{\mathsf{ct}}^*, \pi^*)$ is returned to $\mathcal{A}$.

To show the difference between Game_3 and Game_4 are negligible, we construct $\mathcal{B}_3$ that breaks ANO-IND-ID-CPA security of IBE by using $\mathcal{A}$. The description of $\mathcal{B}_3$ as follows:

1. Upon receiving $\mathsf{mpk}_{\mathsf{IBE}}$, $\mathcal{B}_3$ sample $\mathsf{coin} \leftarrow\!\$\ \{0,1\}$ and computes $(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and $(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\lambda)$. Next, $\mathcal{B}_3$ prepares $\mathcal{L}_{\mathsf{NIZK}} := \emptyset$ Then, $\mathcal{B}_3$ executes $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{SIG}}, \mathsf{crs})$.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}_3$ answers as follows:
   - When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_3$ computes $\mathsf{sig}_\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma)$ and returns $\mathsf{ek}_\sigma = (\sigma, \mathsf{sig}_\sigma)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_3$ computes sends ρ to its challenger and receives sk_ρ. Then, $\mathcal{B}_3$ returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(\mathsf{snd}, \rho, \mathsf{ct} = (s, \hat{\mathsf{ct}}, \pi))$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_3$ outputs $\perp$ if $\mathsf{NIZK.Verify}(\mathsf{crs}, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi) = 0$. Otherwise, if $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi, (\mathsf{m}', \mathsf{snd}', \mathsf{rcv}')) \in \mathcal{L}_{\mathsf{NIZK}}$, if $(\mathsf{snd}, \rho) = (\mathsf{snd}', \mathsf{rcv}')$, it returns m'; otherwise it returns $\perp$. Otherwise, $\mathcal{B}_3$ computes $(\mathsf{m}, r, \mathsf{snd}', \mathsf{rcv}', \mathsf{sig}_\sigma) \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi)$. If $(\mathsf{snd}', \mathsf{rcv}') = (\mathsf{snd}, \rho)$, $\mathcal{B}_3$ returns m; otherwise it returns $\perp$.
   - When $\mathcal{A}$ outputs $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_3$ samples $\sigma_0^* \leftarrow\!\$\ \Sigma_0^*$ and $\sigma_1^* \leftarrow\!\$\ \Sigma_1^*$, and computes $\mathsf{sig}_{\sigma_0^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_0^*)$ and $\mathsf{sig}_{\sigma_1^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_1^*)$. $\mathcal{B}_3$ samples $s^* \leftarrow\!\$\ \{0,1\}^{\mathsf{seedLen}}$, and computes $Z^* := \mathsf{Ext}(s^*, \sigma^*_{\mathsf{coin}})$. Next, $\mathcal{B}_3$ sends $(\mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{m}^*_{\mathsf{coin}} \| \mathsf{snd}^*_{\mathsf{coin}} \| \mathsf{rcv}^*_{\mathsf{coin}})$ to its challenger and receives $\mathsf{ct}^*_{\mathsf{IBE}}$. Then, $\mathcal{B}_3$ computes $\hat{\mathsf{ct}}^* := \mathsf{ct}^*_{\mathsf{IBE}} \oplus Z^*$ and $\pi^* \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}}))$. $\mathcal{B}_3$ stores $x = (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}})$, π^*, and $(\mathsf{m}^*_{\mathsf{coin}}, \sigma^*_{\mathsf{coin}}, \mathsf{rcv}^*_{\mathsf{coin}})$ into the list $\mathcal{L}_{\mathsf{NIZK}}$ and returns $\mathsf{ct}_{\mathsf{coin}} := (s^*, \hat{\mathsf{ct}}^*, \pi^*)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$ oracle, $\mathcal{B}_3$ samples $s \leftarrow\!\$\ \{0,1\}^{\mathsf{seedLen}}$ and $r \leftarrow\!\$\ \mathcal{R}_{\mathsf{IBE}}$, and computes $Z := \mathsf{Ext}(s, \sigma_i^*)$. Next, $\mathcal{B}_3$ computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m} \| \sigma_i^* \| \mathsf{rcv}; r)$ and $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$. Then, $\mathcal{B}_3$ computes $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}))$. $\mathcal{B}_3$ stores $x = (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}})$, π, and $(\mathsf{m}, \sigma_i^*, \mathsf{rcv})$ into the list $\mathcal{L}_{\mathsf{NIZK}}$ and returns $\mathsf{ct}^* := (s, \hat{\mathsf{ct}}, \pi)$ to $\mathcal{A}$.
3. When $\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0,1\}$, $\mathcal{B}_3$ outputs 1 (if $\mathsf{coin} = \widehat{\mathsf{coin}}$) or 0 (otherwise) to its challenger.

It is clear that if the IBE challenger uses $\mathsf{coin}_{\mathsf{IBE}} = 0$ (that is $\mathsf{ct}^*_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{m}^*_{\mathsf{coin}} \| \sigma^*_{\mathsf{coin}} \| \mathsf{rcv}^*_{\mathsf{coin}})$), then $\mathcal{B}_3$ perfectly simulates Game_3 for $\mathcal{A}$. On the other hand, if $\mathsf{coin}_{\mathsf{IBE}} = 1$ (that is $\mathsf{ct}^*_{\mathsf{IBE}} \leftarrow\!\$\ \mathcal{CT}_{\mathsf{IBE}}$), then $\mathcal{B}_3$ perfectly simulates Game_4 for $\mathcal{A}$. Therefore, we have

$$|\epsilon_3 - \epsilon_4| \leq 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3,\mathsf{IBE}}(\lambda).$$

In Game_4, the challenge ciphertext is independent of the challenge coin. Therefore, we have $\epsilon_4 = 0$.

Putting everything together, we have

$$\mathsf{Adv}^{\mathsf{priv}^+\text{-}\mathsf{cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda) + q_D\mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3,\mathsf{IBE}}(\lambda).$$

□

Lemma 6. *If there exists an adversary $\mathcal{A}$ that breaks the* Priv^+-CCA *security in Case 2 of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$*, there exists an adversary $\mathcal{B}_1$ that breaks the zero-knowledge property of* NIZK*, and $\mathcal{B}_2$ against the security of* Ext *such that*

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 2}\right] - \frac{1}{2}\right| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{Ext}}(\lambda).$$

In Case 2, since the adversary may obtain receiver keys for all identities, the decryption oracle can be simulated using the receiver-key oracle and the public decryption algorithm; hence, it does not require an additional extraction-based simulation in contrast to the proof in Case 1 (and Case 3).

Proof. Let $\mathcal{A}$ be an adversary that breaks the Priv^+-CCA security of $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ in Case 2. To prove the theorem, we consider the following sequence of games Game_i. Define the advantage of $\mathcal{A}$ in Game_i as

$$\epsilon_i := \left|\Pr\left[\mathsf{Game}^{\mathcal{A}}_i(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right|.$$

Game_0 : This is the original security game in Case 2. Thus we have

$$\epsilon_0 = \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case 2}\right] - \frac{1}{2}\right|.$$

Game_1 : In this game, we change how the challenger generates the challenge ciphertext. In Game_0, the challenger uses Prove algorithm to generates π, but in this game, the challenger executes the simulator $\mathcal{S}_2$ of NIZK and generates π using the trapdoor τ generated by $\mathcal{S}_1$.

By the same argument in the proof of Lemma 5 to transit from Game_0 to Game_1, there exists $\mathcal{B}_1$ such that

$$|\epsilon_0 - \epsilon_1| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda).$$

Game_2*:* In this game, we change how to generate ciphertexts on σ^*_0. When $\mathcal{A}$ sends $(0, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$ and requests the challenge ciphertext ct_0, ciphertexts are generated with $Z \leftarrow\!\$\ \{0,1\}^{\mathsf{ctLen}}$ instead of $Z := \mathsf{Ext}(s, \sigma^*_0)$.

Now, NIZK proofs given to the adversary hide the sender's information due to the zero-knowledge property of NIZK. Therefore, the challenged σ^*_0 has sufficient min-entropy for the adversary. Also, $\mathcal{A}$ requests ciphertexts on σ^*_0 at most $q_E + 1$ times. Therefore, $(\kappa, q_E + 1)$-reusable computational extractor Ext ensures that the extracted randomness Z looks random for $\mathcal{A}$. Hence, Game_1 and Game_2 are indistinguishable, and there exists an adversary $\mathcal{B}_2$ such that

$$|\epsilon_1 - \epsilon_2| \leq \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{Ext}}(\lambda).$$

Game_3*:* In this game, we change how to generate ciphertexts on σ^*_1. When $\mathcal{A}$ sends $(1, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$ and requests the challenge ciphertext ct_1, ciphertexts are generated with $Z \leftarrow\!\$\ \{0,1\}^{\mathsf{ctLen}}$ instead of $Z := \mathsf{Ext}(s, \sigma^*_1)$.

Now, NIZK proofs given to the adversary hide the sender's information due to the zero-knowledge property of NIZK. Therefore, the challenged σ^*_1 has sufficient min-entropy for the adversary. Also, $\mathcal{A}$ requests ciphertexts on σ^*_1 at most $q_E + 1$ times. Therefore, $(\kappa, q_E + 1)$-reusable computational extractor Ext ensures that the extracted randomness Z looks random for $\mathcal{A}$. Hence, Game_2 and Game_3 are indistinguishable, and there exists an adversary $\mathcal{B}_3$ such that

$$|\epsilon_2 - \epsilon_3| \leq \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_3,\mathsf{Ext}}(\lambda).$$

In Game_3, the challenge ciphertext is generated with a uniformly random value $Z \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ctLen}}$ instead of $Z = \mathsf{Ext}(s, \sigma^*_{\mathsf{coin}})$. Therefore, the masked ciphertext $\hat{\mathsf{ct}} = \mathsf{ct}_{\mathsf{IBE}} \oplus Z$ is distributed independently of the challenge coin coin. Thus, we have $\epsilon_3 = 0$.

Putting everything we have

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case 2}\right] - \frac{1}{2}\right| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{Ext}}(\lambda).$$

□

Lemma 7. *If there exists an adversary $\mathcal{A}$ that breaks the* $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ *security of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ *in Case 3, there exists an adversary $\mathcal{B}_1$ that breaks the zero-knowledge property of* NIZK*, $\mathcal{B}_2$ against the security of* Ext*, $\mathcal{B}_3$ that breaks the* $\mathsf{ANO\text{-}IND\text{-}ID\text{-}CPA}$ *security of* IBE*, and $\mathcal{B}_4$ that breaks the simulation-sound extractability of* NIZK *such that*

$$\begin{aligned}&\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 3}\right] - \frac{1}{2}\right| \\ &\qquad \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda) + q_D\mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3,\mathsf{IBE}}(\lambda) + \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_4,\mathsf{Ext}}(\lambda),\end{aligned}$$

where q_D is the maximum number of queries $\mathcal{A}$ makes to the decryption oracle.

Proof. Let $\mathcal{A}$ be an adversary that breaks the $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ in Case 3. To prove the theorem, we consider the following sequence of games. Define the probability that $\mathcal{A}$ wins in Game_i as

$$\epsilon_i := \left|\Pr\left[\mathsf{Game}^{\mathcal{A}}_i(\lambda) \Rightarrow 1\right] - \frac{1}{2}\right|.$$

Game_0. This is the original security game in Case 3. By definition, we have

$$\epsilon_0 = \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case 3}\right] - \frac{1}{2}\right|.$$

Game_1. In this game, we change how the challenger generates the challenge ciphertext. In Game_0, the challenger uses Prove algorithm to generates π, but in this game, the challenger executes the simulator $\mathcal{S}_2$ of NIZK and generates π using the trapdoor τ generated by $\mathcal{S}_1$.

By the same argument in the proof of Lemma 5 to transit from Game_0 to Game_1, there exists $\mathcal{B}_1$ such that

$$|\epsilon_0 - \epsilon_1| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda).$$

Game_2. In this game, we change how the challenger generates the CRS in the setup phase. Specifically, the challenger executes $\mathcal{E}_1$ to generate a CRS of NIZK.

From the simulation-sound extractability of NIZK, the output of $\mathcal{E}_1(1^\lambda)$ is identically distributed to that of $\mathcal{S}_1(1^\lambda)$ on the first two components (crs, τ). Therefore, Game_1 and Game_2 are identical experiments, and it holds that

$$\epsilon_1 = \epsilon_2.$$

Game_3. In this game, we change how the challenger responds to the decryption queries. In Game_2, the challenger computes $\mathsf{m}'\|\mathsf{snd}'\|\mathsf{rcv}' \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{sk}_\rho, \mathsf{ct}_{\mathsf{IBE}})$, but in this game, the challenger runs $(\mathsf{m}, r, \mathsf{snd}', \mathsf{rcv}', \mathsf{sig}_\sigma) \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi)$. Then, it returns m' only if $(\mathsf{snd}, \rho) = (\mathsf{snd}', \mathsf{rcv}')$; otherwise it returns $\perp$.

By the same argument in the proof of Lemma 5 to transit from Game_2 to Game_3, there exists $\mathcal{B}_2$ such that

$$|\epsilon_2 - \epsilon_3| \leq q_D\mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda).$$

Game_4**.** In this game, we modify how to generate the challenge ciphertext for $\mathsf{coin} = 0$. In Game_3, the challenger computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}_0^*, \mathsf{m}_0^* \| \sigma_0^* \| \mathsf{rcv}_0^*)$ whereas in Game_4 the challenger instead samples $\mathsf{ct}_{\mathsf{IBE}} \leftarrow\!\!\$\ \mathcal{CT}_{\mathsf{IBE}}$.

To show the difference between Game_3 and Game_4 are negligible, we construct $\mathcal{B}_3$ that breaks ANO-IND-ID-CPA security of IBE by using $\mathcal{A}$. The description of $\mathcal{B}_3$ as follows:

1. Upon receiving $\mathsf{mpk}_{\mathsf{IBE}}$, $\mathcal{B}_3$ samples $\mathsf{coin} \leftarrow\!\!\$\ \{0,1\}$ and computes $(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and $(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\lambda)$. Next, $\mathcal{B}_3$ prepares a list $\mathcal{L}_{\mathsf{NIZK}}$. Then, $\mathcal{B}_3$ executes $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{SIG}}, \mathsf{crs})$.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}_3$ answers as follows:
   - When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_3$ computes $\mathsf{sig}_\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma)$ and returns $\mathsf{ek}_\sigma = (\sigma, \mathsf{sig}_\sigma)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_3$ sends ρ to its challenger and receives sk_ρ. Then, $\mathcal{B}_3$ returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(\mathsf{snd}, \rho, \mathsf{ct} = (s, \hat{\mathsf{ct}}, \pi))$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_3$ outputs $\perp$ if $\mathsf{NIZK.Verify}(\mathsf{crs}, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi) = 0$. Otherwise, if $((s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi, (\mathsf{m}', \mathsf{snd}', \mathsf{rcv}')) \in \mathcal{L}_{\mathsf{NIZK}}$, if $(\mathsf{snd}, \rho) = (\mathsf{snd}', \mathsf{rcv}')$, it returns m'; otherwise it returns $\perp$. Otherwise, $\mathcal{B}_3$ computes $(\mathsf{m}, r, \mathsf{snd}', \mathsf{rcv}', \mathsf{sig}_\sigma) \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}), \pi)$. If $(\mathsf{snd}', \mathsf{rcv}') = (\mathsf{snd}, \rho)$, $\mathcal{B}_3$ returns m; otherwise it returns $\perp$.
   - When $\mathcal{A}$ outputs $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_3$ samples $\sigma_0^* \leftarrow\!\!\$\ \Sigma_0^*$ and $\sigma_1^* \leftarrow\!\!\$\ \Sigma_1^*$, and computes $\mathsf{sig}_{\sigma_0^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_0^*)$ and $\mathsf{sig}_{\sigma_1^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma_1^*)$. $\mathcal{B}_3$ samples $s^* \leftarrow\!\!\$\ \{0,1\}^{\mathsf{seedLen}}$, and computes $Z^* := \mathsf{Ext}(s^*, \sigma_{\mathsf{coin}}^*)$. Next, if $\mathsf{coin} = 0$, $\mathcal{B}_3$ sends $(\mathsf{rcv}_{\mathsf{coin}}^*, \mathsf{m}_{\mathsf{coin}}^* \| \sigma_{\mathsf{coin}}^* \| \mathsf{rcv}_{\mathsf{coin}}^*)$ to its challenger and receives $\mathsf{ct}_{\mathsf{IBE}}^*$. Then, $\mathcal{B}_3$ computes $\hat{\mathsf{ct}}^* := \mathsf{ct}_{\mathsf{IBE}}^* \oplus Z^*$. Otherwise, if $\mathsf{coin} = 1$, $\mathcal{B}_3$ computes $\hat{\mathsf{ct}}^*$ by itself. Then, it computes $\pi^* \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}}))$. $\mathcal{B}_3$ stores $x = (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}})$, π^* and $(\mathsf{m}_{\mathsf{coin}}^*, \sigma_{\mathsf{coin}}^*, \mathsf{rcv}_{\mathsf{coin}}^*)$ into the list $\mathcal{L}_{\mathsf{NIZK}}$ and returns $\mathsf{ct}_{\mathsf{coin}} := (s^*, \hat{\mathsf{ct}}^*, \pi^*)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$ oracle, $\mathcal{B}_3$ samples $s \leftarrow\!\!\$\ \{0,1\}^{\mathsf{seedLen}}$ and $r \leftarrow\!\!\$\ \mathcal{R}_{\mathsf{IBE}}$, and computes $Z := \mathsf{Ext}(s, \sigma_i^*)$. Next, $\mathcal{B}_3$ computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m} \| \sigma_i^* \| \mathsf{rcv}; r)$ and $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$. Then, $\mathcal{B}_3$ computes $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}))$. $\mathcal{B}_3$ stores $x = (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}})$, π, and $(\mathsf{m}, \sigma_i^*, \mathsf{rcv})$ into the list $\mathcal{L}_{\mathsf{NIZK}}$ and returns $\mathsf{ct}^* := (s, \hat{\mathsf{ct}}, \pi)$ to $\mathcal{A}$.
3. When $\mathcal{A}$ outputs $\widehat{\mathsf{coin}} \in \{0,1\}$, $\mathcal{B}_3$ outputs 1 (if $\mathsf{coin} = \widehat{\mathsf{coin}}$) or 0 (otherwise) to its challenger.

It is clear that if the IBE challenger uses $\mathsf{coin}_{\mathsf{IBE}} = 0$ (that is $\mathsf{ct}_{\mathsf{IBE}}^* \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}_{\mathsf{coin}}^*, \mathsf{m}_{\mathsf{coin}}^* \| \sigma_{\mathsf{coin}}^* \| \mathsf{rcv}_{\mathsf{coin}}^*)$), then $\mathcal{B}_3$ perfectly simulates Game_3 for $\mathcal{A}$. On the other hand, if $\mathsf{coin}_{\mathsf{IBE}} = 1$ (that is $\mathsf{ct}_{\mathsf{IBE}}^* \leftarrow\!\!\$\ \mathcal{CT}_{\mathsf{IBE}}$), then $\mathcal{B}_3$ perfectly simulates Game_4 for $\mathcal{A}$. Note that if $\mathsf{coin} = 1$, Game_3 and Game_4 are identical. Therefore, we have

$$|\epsilon_3 - \epsilon_4| \leq 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3, \mathsf{IBE}}(\lambda).$$

Game_5**.** In this game, we change how to generate ciphertexts on σ_1^*. When $\mathcal{A}$ sends $(1, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$ and requests the challenge ciphertext, ciphertexts are generated with $Z \leftarrow\!\!\$\ \{0,1\}^{\mathsf{ctLen}}$ instead of $Z := \mathsf{Ext}(s, \sigma_1^*)$.

By the same argument in the proof of Lemma 6 (transit from Game_1 to Game_2), Game_4 and Game_5 here are indistinguishable, and there exists an adversary $\mathcal{B}_4$ such that

$$|\epsilon_4 - \epsilon_5| \leq \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_4, \mathsf{Ext}}(\lambda).$$

In Game_5, the challenge ciphertext is independent of coin. More specifically, when $\mathsf{coin} = 0$, the IBE ciphertext $\mathsf{ct}_{\mathsf{IBE}}$ is sampled uniformly at random from the ciphertext space. When $\mathsf{coin} = 1$, Z is chosen uniformly at random from the ciphertext space. In either case, the masked ciphertext $\hat{\mathsf{ct}} = \mathsf{ct}_{\mathsf{IBE}} \oplus Z$ is independent of the challenge coin coin. Hence the adversary's view is independent of coin and thus $\epsilon_5 = 0$.

Putting everything together, we have

$$\left| \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 3 \right] - \frac{1}{2} \right|$$
$$\leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1, \mathsf{NIZK}}(\lambda) + q_D \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2, \mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3, \mathsf{IBE}}(\lambda) + \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_4, \mathsf{Ext}}(\lambda).$$

□

From Lemma 7 and the observation in Remark 1, we obtain the following lemma.

Lemma 8. *If there exists an adversary $\mathcal{A}$ that breaks the* Priv^+-CCA *security of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$ *in Case 3, there exists an adversary $\mathcal{B}_1$ that breaks the zero-knowledge property of* NIZK*, $\mathcal{B}_2$ against the security of* Ext*, $\mathcal{B}_3$ that breaks the* ANO-IND-ID-CPA *security of* IBE*, and $\mathcal{B}_4$ that breaks the simulation-sound extractability of* NIZK *such that*

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 4}\right] - \frac{1}{2}\right| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda) + q_D\mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda) + 2\mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cpa}}_{\mathcal{B}_3,\mathsf{IBE}}(\lambda) + \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_4,\mathsf{Ext}}(\lambda),$$

where q_D is the maximum number of queries $\mathcal{A}$ makes to the decryption oracle.

Theorem 4. *If there exists an adversary $\mathcal{A}$ that breaks the* Auth-iCMA *security of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$*, there exists an adversary $\mathcal{B}_1$ that breaks the zero-knowledge property of* NIZK*, an adversary $\mathcal{B}_2$ that breaks the simulation-sound extractability of* NIZK *and an adversary $\mathcal{B}_3$ that breaks the* EUF-CMA *security of* SIG *such that*

$$\mathsf{Adv}^{\mathsf{auth\text{-}icma}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda) + \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2,\mathsf{NIZK}}(\lambda) + \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{B}_3,\mathsf{SIG}}(\lambda).$$

Proof. Let $\mathcal{A}$ be an adversary that breaks the Auth-iCMA security of $\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}$. To prove the theorem, we consider the following sequence of games. Define the advantage of $\mathcal{A}$ in Game_i as

$$\epsilon_i := \Pr\left[\mathsf{Game}_i^{\mathcal{A}}(\lambda) \Rightarrow 1\right].$$

Game_0 : This is the original security game. Thus we have

$$\epsilon_0 = \mathsf{Adv}^{\mathsf{auth\text{-}icma}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda).$$

Game_1 : In this game, we change how the challenger generates ciphertexts generated in $\mathcal{O}_E$ oracle. In Game_0, the challenger uses Prove algorithm to generate the NIZK proof π, but in this game, the challenger executes the simulator $\mathcal{S}_2$ of NIZK and generates π using the trapdoor τ generated by $\mathcal{S}_1$.

As the proof of Lemma 5, there exists an adversary $\mathcal{B}_1$ such that

$$|\epsilon_0 - \epsilon_1| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1,\mathsf{NIZK}}(\lambda).$$

Game_2*:* In this game, we change how the challenger generates the CRS in the setup phase. Specifically, the challenger executes $\mathcal{E}_1$ to generate the CRS of NIZK.

From the simulation-sound extractability of NIZK, the output of $\mathcal{E}_1(1^\lambda)$ is identically distributed to that of $\mathcal{S}_1(1^\lambda)$ on the first two components (crs, τ). Therefore, Game_1 and Game_2 are identical, and it holds that

$$\epsilon_1 = \epsilon_2.$$

Game_3*:* In this game, we define the following event. Let $x^* = (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}})$ and $w^* = (\mathsf{m}^*, r^*, \sigma^*, \mathsf{rcv}^*, \mathsf{sig}^*_\sigma) \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi^*, (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}}), \pi^*)$. We define Sound as the event that $\mathsf{NIZK.Verify}(\mathsf{crs}, x^*, \pi^*) = 1$ but $(x^*, w^*) \notin R$. If the event Sound occurs, Game_3 aborts. Since Game_3 and Game_2 are identical unless Sound occurs, we have

$$|\epsilon_2 - \epsilon_3| \leq \Pr[\mathsf{Sound}].$$

We next estimate $\Pr[\mathsf{Sound}]$. To do so, we construct $\mathcal{B}_2$ that breaks the simulation-sound extractability of NIZK by using $\mathcal{A}$. The description of $\mathcal{B}_2$ is as follows.

1. Upon receiving crs, $\mathcal{B}_2$ computes $(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and $(\mathsf{mpk}_{\mathsf{SIG}}, \mathsf{msk}_{\mathsf{SIG}}) \leftarrow \mathsf{SIG.KGen}(1^\lambda)$. Then, $\mathcal{B}_2$ executes $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{SIG}}, \mathsf{crs})$.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}_2$ answers as follows:

- When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ computes $\mathsf{sig}_\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{msk}_{\mathsf{SIG}}, \sigma)$ and returns $\mathsf{ek}_\sigma = (\sigma, \mathsf{sig}_\sigma)$ to $\mathcal{A}$.
- When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ computes $\mathsf{sk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho)$, and returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
- When $\mathcal{A}$ sends $(\sigma, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_E$ oracle, $\mathcal{B}_2$ samples $s \leftarrow\!\$\ \{0,1\}^{\mathsf{seedLen}}$ and $r \leftarrow\!\$\ \mathcal{R}_{\mathsf{IBE}}$, and computes $Z := \mathsf{Ext}(s, \sigma)$. Next, $\mathcal{B}_2$ computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m}\|\sigma\|\mathsf{rcv}; r)$ and $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$. Then, $\mathcal{B}_2$ sends the statement $x = (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}})$ and the witness $w = (\mathsf{m}, r, \sigma, \mathsf{rcv}, \mathsf{sig}_\sigma)$ to its challenger and receives π. Finally, $\mathcal{B}_2$ returns $\mathsf{ct} := (s, \hat{\mathsf{ct}}, \pi)$ to $\mathcal{A}$.

3. When $\mathcal{A}$ outputs $(\mathsf{snd}^*, \rho^*, \mathsf{ct}^*)$, $\mathcal{B}_2$ outputs $((s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}}), \pi^*)$ to its challenger.

If the event Sound occurs, then $\mathcal{A}$ has produced a proof π^* such that

$$\mathsf{NIZK.Verify}(\mathsf{crs}, (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}}), \pi^*) = 1$$

while the corresponding extracted witness $(\sigma^*, \mathsf{sig}^*_\sigma)$ obtained by the extractor does not satisfy $\mathsf{SIG.Verify}(\mathsf{mpk}_{\mathsf{SIG}}, \sigma, \mathsf{sig}_\sigma) = 1$. This means that π^* is an accepting proof for a false statement. Therefore, whenever Sound happens, $\mathcal{B}_2$ outputs such a proof to its challenger and thus breaks the simulation-sound extractability of NIZK. Therefore, we have

$$|\epsilon_2 - \epsilon_3| \leq \Pr[\mathsf{Sound}] = \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2, \mathsf{NIZK}}(\lambda).$$

Finally, we will bound ϵ_3. To this end, we define the following event. Let $(\mathsf{m}^*, r^*, \sigma^*, \mathsf{rcv}^*, \mathsf{sig}^*_\sigma) \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi^*, (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}}), \pi^*)$:

- Forge: $\mathcal{A}$ outputs π^* such that the extracted $(\sigma^*, \mathsf{sig}^*_\sigma)$ satisfies $\mathsf{SIG.Verify}(\mathsf{mpk}_{\mathsf{SIG}}, \sigma^*, \mathsf{sig}_{\sigma^*}) = 1$.

Since Sound does not occur in Game_3, if $\mathcal{A}$ breaks the security, Forge occurs. Hence, we have

$$\epsilon_3 = \Pr[\mathsf{Forge}].$$

We now estimate $\Pr[\mathsf{Forge}]$. To do so, we construct $\mathcal{B}_3$ that breaks $\mathsf{EUF\text{-}CMA}$ security of SIG by using $\mathcal{A}$. The description of $\mathcal{B}_3$ is as follows.

1. Upon receiving $\mathsf{mpk}_{\mathsf{SIG}}$, $\mathcal{B}_3$ computes $(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and $(\mathsf{crs}, \tau, \xi) \leftarrow \mathcal{E}_1(1^\lambda)$. Then, $\mathcal{B}_3$ executes $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{SIG}}, \mathsf{crs})$.
2. When $\mathcal{A}$ makes oracle queries, $\mathcal{B}_3$ answers as follows:
   - When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_3$ sends σ to its signing oracle and receives sig_σ. Then, $\mathcal{B}_3$ returns $\mathsf{ek}_\sigma = (\sigma, \mathsf{sig}_\sigma)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_3$ computes $\mathsf{sk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho)$, and returns $\mathsf{dk}_\rho := (\rho, \mathsf{sk}_\rho)$ to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(\sigma, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_E$ oracle, $\mathcal{B}_3$ samples $s \leftarrow\!\$\ \{0,1\}^{\mathsf{seedLen}}$ and $r \leftarrow\!\$\ \mathcal{R}_{\mathsf{IBE}}$, and computes $Z := \mathsf{Ext}(s, \sigma)$. Next, $\mathcal{B}_3$ computes $\mathsf{ct}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m}\|\sigma\|\mathsf{rcv}; r)$, $\hat{\mathsf{ct}} := \mathsf{ct}_{\mathsf{IBE}} \oplus Z$ and $\pi \leftarrow \mathcal{S}_2(\mathsf{crs}, \tau, (s, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}, \mathsf{mpk}_{\mathsf{SIG}}))$. Finally, $\mathcal{B}_3$ returns $\mathsf{ct} := (s, \hat{\mathsf{ct}}, \pi)$ to $\mathcal{A}$.
3. When $\mathcal{A}$ outputs $(\mathsf{snd}^*, \rho^*, \mathsf{ct}^*)$, $\mathcal{B}_3$ computes $\mathsf{sk}_{\rho^*} \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho^*)$, sets $\mathsf{dk}_{\rho^*} := (\rho^*, \mathsf{sk}_{\rho^*})$, and checks whether $\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_{\rho^*}, \mathsf{snd}^*, \mathsf{ct}^*) \neq \bot$. Then, it runs $(\mathsf{m}^*, r^*, \sigma^*, \mathsf{rcv}^*, \mathsf{sig}^*) \leftarrow \mathcal{E}_2(\mathsf{crs}, \xi, (s^*, \mathsf{mpk}_{\mathsf{IBE}}, \hat{\mathsf{ct}}^*, \mathsf{mpk}_{\mathsf{SIG}}), \pi^*)$. If $\mathsf{SIG.Verify}(\mathsf{mpk}_{\mathsf{SIG}}, \sigma^*, \mathsf{sig}^*) = 1$, $\mathcal{B}_3$ outputs $(\sigma^*, \mathsf{sig}^*)$ to its challenger. Otherwise, $\mathcal{B}_3$ halts.

If Dec accepts, then $(\sigma^*, \mathsf{rcv}^*) = (\mathsf{snd}^*, \rho^*)$. Moreover, $\mathsf{SIG.Verify}(\mathsf{mpk}_{\mathsf{SIG}}, \sigma^*, \mathsf{sig}^*) = 1$ if $\mathcal{A}$ breaks the authenticity. Since $\mathcal{A}$ succeeds only when $(\sigma^*, \cdot) \notin \mathcal{L}_S$, $\sigma^* = \mathsf{snd}^*$ was never queried to the signing oracle. Thus $(\sigma^*, \mathsf{sig}^*)$ is an EUF-CMA forgery. That indicates $\mathcal{B}_3$ breaks the $\mathsf{EUF\text{-}CMA}$ security of SIG if Forge occurs. Thus, we have

$$\epsilon_3 = \Pr[\mathsf{Forge}] = \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{B}_3, \mathsf{SIG}}(\lambda).$$

Putting everything together, we obtain

$$\mathsf{Adv}^{\mathsf{auth\text{-}icma}}_{\mathcal{A}, \mathsf{IB\text{-}ME}^{\mathsf{IBE}+\mathsf{SIG}}}(\lambda) \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathcal{B}_1, \mathsf{NIZK}}(\lambda) + \mathsf{Adv}^{\mathsf{ext}}_{\mathcal{B}_2, \mathsf{NIZK}}(\lambda) + \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{B}_3, \mathsf{SIG}}(\lambda).$$

□

The generic construction is intended as a feasibility result in the standard model. Its efficiency depends on the concrete NIZK system used to prove the relation, which may be general unless the underlying IBE and signature schemes have suitable algebraic structure. Providing efficient instantiations for specific standard-model IBE schemes is an interesting direction for future work.

Acknowledgements The authors thank the anonymous reviewers for their constructive comments and suggestions. The authors used Microsoft Copilot to identify editorial issues; all technical content and final decisions were verified by the authors. This work was in part supported by JST K Program Grant Number JPMJKP24U2, Japan. Keitaro Hashimoto and Keisuke Hara were partially supported by JST CREST JPMJCR22M1, Japan. Also, Keisuke Hara was partially supported by JST-AIP JPMJCR22U5, Japan. Junji Shikata was partially supported by JSPS KAKENHI Grant Numbers JP23K24846, JP26K02910, Japan.

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Berlin, Heidelberg, May / June 2010.
2. Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 57–74. Springer, Berlin, Heidelberg, August 2013.
3. Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. Match me if you can: Matchmaking encryption and its applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 701–731. Springer, Cham, August 2019.
4. Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. Match me if you can: Matchmaking encryption and its applications. *Journal of Cryptology*, 34(3):16, July 2021.
5. Roberta Cimorelli Belfiore, Andrea De Cosmo, and Anna Lisa Ferrara. Identity-based matchmaking encryption from standard lattice assumptions. In Christina Pöpper and Lejla Batina, editors, *ACNS 24: 22nd International Conference on Applied Cryptography and Network Security, Part II*, volume 14584 of *Lecture Notes in Computer Science*, pages 163–188, Abu Dhabi, UAE, March 5–8, 2024. Springer, Cham, Switzerland.
6. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Berlin, Heidelberg, August 2001.
7. Xavier Boyen and Qinyi Li. Identity-based matchmaking encryption with enhanced privacy — a generic construction with practical instantiations. In *ESORICS 2023*, Heidelberg, September 2023. Springer.
8. Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In Vijayalakshmi Atluri, Catherine Meadows, and Ari Juels, editors, *ACM CCS 2005*, pages 320–329. ACM Press, November 2005.
9. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, Berlin, Heidelberg, May 2004.
10. Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap Diffie-Hellman groups. In Yvo Desmedt, editor, *PKC 2003: 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30, Miami, FL, USA, January 6–8, 2003. Springer, Berlin, Heidelberg, Germany.
11. Jie Chen, Yu Li, Jinming Wen, and Jian Weng. Identity-based matchmaking encryption from standard assumptions. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part III*, volume 13793 of *LNCS*, pages 394–422. Springer, Cham, December 2022.
12. Sohto Chiku, Keisuke Hara, and Junji Shikata. Hierarchical identity-based matchmaking encryption. In Mark Manulis, editor, *Applied Cryptography and Network Security Workshops*, pages 274–295, Cham, 2026. Springer Nature Switzerland.
13. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 621–630. ACM Press, May / June 2009.
14. Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 22–41. Springer, Berlin, Heidelberg, December 2014.

15. Danilo Francati, Daniele Friolo, Giulio Malavolta, and Daniele Venturi. Multi-key and multi-input predicate encryption from learning with errors. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 573–604. Springer, Cham, April 2023.
16. Danilo Francati, Daniele Friolo, Giulio Malavolta, and Daniele Venturi. Multi-key and multi-input predicate encryption (for conjunctions) from learning with errors. *Journal of Cryptology*, 37(3):24, July 2024.
17. Danilo Francati, Alessio Guidi, Luigi Russo, and Daniele Venturi. Identity-based matchmaking encryption without random oracles. In Avishek Adhikari, Ralf Küsters, and Bart Preneel, editors, *INDOCRYPT 2021*, volume 13143 of *LNCS*, pages 415–435, Heidelberg, December 2021. Springer.
18. Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In Hideki Imai and Yuliang Zheng, editors, *PKC'99*, volume 1560 of *LNCS*, pages 53–68. Springer, Berlin, Heidelberg, March 1999.
19. Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 445–464. Springer, Berlin, Heidelberg, May / June 2006.
20. Shai Halevi, editor. *CRYPTO 2009*, volume 5677 of *LNCS*. Springer, Berlin, Heidelberg, August 2009.
21. Dennis Hofheinz, Dingding Jia, and Jiaxin Pan. Identity-based encryption tightly secure under chosen-ciphertext attacks. In Peyrin and Galbraith [27], pages 190–220.
22. Shuichi Katsumata, Shota Yamada, and Takashi Yamakawa. Tighter security proofs for GPV-IBE in the quantum random oracle model. In Peyrin and Galbraith [27], pages 253–282.
23. Shen Lin, Yu Li, and Jie Chen. Cca-secure identity-based matchmaking encryption from standard assumptions. In Chunpeng Ge and Moti Yung, editors, *Information Security and Cryptology*, pages 253–273, Singapore, 2024. Springer Nature Singapore.
24. John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002.
25. Takahiro Matsuda, Kanta Matsuura, and Jacob C. N. Schuldt. Efficient constructions of signcryption schemes and signcryption composability. In Bimal K. Roy and Nicolas Sendrier, editors, *INDOCRYPT 2009*, volume 5922 of *LNCS*, pages 321–342. Springer, Berlin, Heidelberg, December 2009.
26. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Halevi [20], pages 18–35.
27. Thomas Peyrin and Steven Galbraith, editors. *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*. Springer, Cham, December 2018.
28. Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *The 2000 Symposium on Cryptography and Information Security*, January 2000.
29. Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 47–53. Springer, Berlin, Heidelberg, August 1984.
30. Yuejun Wang, Baocang Wang, Qiqi Lai, and Yu Zhan. Identity-based matchmaking encryption with stronger security and instantiation on lattices. *Theoretical Computer Science*, 1029:115048, 2025.
31. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Halevi [20], pages 619–636.
32. Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, Berlin, Heidelberg, May 2005.
33. Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In Burton S. Kaliski, Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 165–179. Springer, Berlin, Heidelberg, August 1997.