Identity-Based Matchmaking Encryption, Revisited
Efficient Constructions with Strong Security

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

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

March 26, 2025

Abstract. Identity-based matchmaking encryption (IB-ME) [Ateniese et al., Crypto 2019] allows users to communicate privately, anonymously, and authentically. After the seminal paper by Ateniese et al., much work has been done on the security and construction of IB-ME. In this work, we revisit the security definitions of IB-ME and provide improved constructions. First, we classify the existing security notions of IB-ME, systematically categorizing privacy into three categories (CPA, CCA, and privacy in the case of mismatch) and authenticity into four categories (NMA and CMA, both against insiders and outsiders). In particular, we reconsider privacy when the sender's identity is mismatched during decryption and provide a new simple security game called mismatch security, capturing its essence. Second, we propose efficient and strongly secure IB-ME schemes from the bilinear Diffie-Hellman assumption in the random oracle model and from anonymous identity-based encryption and identity-based signature in the quantum random oracle model. The first scheme is based on Boneh-Franklin IBE, similar to the Ateniese et al. scheme, but ours achieves a more compact decryption key and ciphertext and stronger CCA-privacy, CMA-authenticity, and mismatch security. The second scheme is an improved generic construction, which achieves not only stronger security but also the shortest ciphertext among existing generic constructions. This generic construction provides a practical scheme from lattices in the quantum random oracle model.

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

Table of Contents

1 Introduction

1.1 Background

Identity-based matchmaking encryption (IB-ME), proposed by Ateniese et al. [2, 3], is a new identity-based cryptographic primitive designed to guarantee confidential and authenticated message delivery while anonymizing both sender and receiver. Similarly to conventional identity-based encryption (IBE) [5], a key generation center generates secret keys of users corresponding to their identity, and in the IB-ME setting, both sender and receiver possess their secret keys. When a sender with identity σ sends a message, it encrypts the message with its (secret) encryption key ek_σ and the identity of the target receiver rcv. When a receiver with identity ρ decrypts the ciphertext, it uses its secret decryption key dk_ρ and specifies the identity of the target sender snd. The decryption process is successful only if the identities match, i.e., $\sigma = \mathsf{snd}$ and $\rho = \mathsf{rcv}$ hold. In case the identities do not match (i.e., $\sigma \neq \mathsf{snd}$ or $\rho \neq \mathsf{rcv}$), nothing is leaked except the fact that the identities are mismatched. IB-ME has many practical applications e.g., secret handshake protocols, privacy-preserving bulletin boards [2,3], etc.

Security notions for IB-ME. Ateniese et al. [2,3] defined *privacy* and *authenticity* as the security requirements for IB-ME. In essence, privacy guarantees the confidentiality of messages against unintentional receivers who do not have the legitimate decryption key; Authenticity guarantees the legitimacy of senders, preventing impersonation without knowing their encryption key. We can see that privacy (resp., authenticity) is similar to the semantic security of encryption schemes (resp., unforgeability of signature schemes).

Following the pioneering work by Ateniese et al., a lot of works have explored more desirable security notions. Regarding authenticity, Francati et al. [18] and Chen et al. [11] defined a new notion of authenticity that allows an adversary to compromise receiver secret keys freely, in contrast to the definition of Ateniese et al.[3]. Wang et al. [32] proposed an extended version of the notions of authenticity, which they call "strong authenticity", allowing the adversary to access an encryption oracle that computes a ciphertext of adversarially chosen messages[4]. For stronger privacy guarantees, Chiku et al. [13] and Lin et al. [25] considered privacy against chosen-ciphertext attacks (CCA), where an adversary can access a decryption oracle that computes plaintexts of adversarially chosen ciphertexts. Furthermore, Francati et al. [18] highlighted a deficiency in the original definition of privacy by Ateniese et al. They pointed out that it does not account for privacy in the case where the target identity snd chosen by a receiver mismatches with the actual sender's identity σ. That is, the original definition does not guarantee the confidentiality of messages in the case $\mathsf{rcv} = \rho$ but $\mathsf{snd} \neq \sigma$ occurs during decryption[5]. This gap led them to introduce a new privacy concept called "enhanced privacy", which captures privacy in cases involving mismatched sender identities used during decryption.

As explained above, many security definitions for IB-ME have been considered. In particular, existing works compared the efficiency of each scheme, ignoring the differences in the security properties. In other words, their comparisons are inaccurate. Moreover, the definition of enhanced privacy proposed by Francati et al. [18] is comprehensive and well-defined; however, their definition requires detailed case-by-case distinctions, which in turn incurs complicated proofs. From such a situation, we realize the first question:

Q1: What are the proper security definitions of IB-ME for accurate comparisons?

Constructions of IB-ME. Ateniese et al. introduced the initial IB-ME scheme from the bilinear Diffie-Hellman (BDH) assumption in the random oracle model (ROM) [2, 3], based on the Boneh-Franklin IBE scheme [5]. A drawback of the scheme is long decryption keys and ciphertexts: they include three resp. two group elements. This raises the following second question.

Q2: Can we construct a more efficient and strongly secure IB-ME scheme from the BDH assumption in the ROM?

[3] The difference was not explained explicitly in [11,18] In particular, despite this difference, Francati et al. cited the original work, which misleads the reader into thinking that the two definitions are the same.

[4] The attack scenario can be seen as ordinary chosen message attacks (CMA), but they did not explain it as such.

[5] As mentioned in [18], Ateniese et al. noticed this gap and informally argued that their IB-ME scheme ensures the confidentiality of messages in such a case.

Table 1: Comparison between our IB-ME schemes and the existing schemes. (Re)Ext stands for (reusable) randomness extractors, and Hom Sig stands for holomorphic signatures.

Schemes	Security properties			Assumptions	Model
	Privacy	Authenticity	Mismatch		
Ateniese et al. [2]	CPA	oNMA		BDH	ROM
Francati et al. [18]	CPA	iNMA	$\surd$	q-ABDHE+SIG+NIZK+ReExt	StdM
Chen et al. [11]	CPA	iNMA		SXDH	StdM
Wang et al. [32]	CPA	iCMA		Anon HIBE+IBS	StdM
Chiku et al. [13]	CCA	iCMA		Anon HIBE + HIBS + OTS	StdM
Boyen and Li [6]	CPA	iCMA	$\surd$	Anon IBE+IBS+ReExt+Ext	StdM
Lin et al. [25]	CCA	iNMA		SXDH	StdM
Belfiore et al. [4]	CPA	iNMA	$\surd$	Anon IBE + Hom Sig + ReExt	StdM
Ours (§ 4)	CCA	oCMA	$\surd$	BDH	ROM
Ours (§ 5)	CCA	iCMA	$\surd$	Anon IBE+IBS	ROM

Following the initial work by Ateniese et al., several works have made efforts to develop improved IB-ME schemes, with a particular focus on the standard model (StdM) [6, 11, 18, 32]. Francati et al. [18] proposed an IB-ME scheme in the StdM based on Gentry's anonymous IBE scheme [20]. Although their scheme is secure in the StdM, it relies on a non-standard q-augmented bilinear Diffie-Hellman exponent assumption. To remove the reliance on nonstandard assumptions, Chen et al. [11] constructed an IB-ME scheme based on an anonymous IBE scheme by Chen et al. [12], whose security relies on the symmetric external Diffie-Hellman (SXDH) assumption in the StdM, and Wang et al. [32] proposed a generic construction of IB-ME from anonymous 2-level hierarchical IBE (HIBE) and identity-based signature (IBS) to realize lattice-based IB-ME schemes. However, they do not consider the stronger notion of security, especially enhanced privacy. To realize an IB-ME scheme with enhanced privacy from lattices, Belfiore et al. [4] proposed another generic construction of IB-ME from IBE and homomorphic signatures. Boyen and Li [6] showed that an IB-ME scheme with enhanced privacy can be constructed from IBE, IBS, and (reusable) extractors. Chiku et al. [13] proposed a hieratical IB-ME scheme by extending Wang et al.'s construction. They convert it into Priv-CCA secure IB-ME via Canetti-Halevi-Katz (CHK) transformation [8]. Concurrently, Lin et al. [25] proposed another Priv-CCA secure IB-ME scheme based on a specific HIBE scheme from SXDH assumption and CHK transformation.

These works allow us to obtain various IB-ME schemes from both classical and post-quantum assumptions. However, all of them only provide weaker CPA privacy, which is insufficient for real-world applications. Another issue is their ciphertext sizes are long since they use heavily primitives (e.g., HIBE) or include many seeds for extractors in a ciphertext. This fact gives us the third question:

Q3: Can we generically construct a more efficient and strongly secure IB-ME scheme?

1.2 Our Contributions

We revisit the concept of IB-ME and answer the above three research questions. First, we reformalize the security notions for IB-ME. Then we present a highly efficient and strongly secure IB-ME scheme from the BDH assumption in the ROM. Finally, we proposed a new generic construction from IBE and IBS in the QROM. The comparison of our schemes and the existing ones is summarized in Table 1. See Section 6 for a detailed comparison, especially of their efficiency.

A1: Re-formalizing security notions of IB-ME. We sort out the differences in security notions for IB-ME. At first, 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 name the respective attacks as chosen message attacks (CMA) and no message attacks (NMA) according to 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).

For privacy, we rename the original definition by Ateniese et al. as CPA privacy since the adversary cannot access the decryption oracle, and define CCA privacy as in [13, 25].[8] Then, we redefine the security game for "enhanced privacy" which captures privacy in the case of mismatch during decryption. Francati et al. [18] defined a single definition that includes both the privacy originally considered (CPA privacy) and privacy in mismatch cases, which complicates understanding the definition and security proofs. Thus, we extract the essence of privacy in the case of mismatch and give a new simple security definition, called Priv-MisMatch security. Roughly, it captures the confidentiality of messages in the case the adversary knows the target receiver's secret key but does not know the sender's identity. As a sanity check, we show that our CPA/CCA privacy and Priv-MisMatch security implies Francati's CPA/CCA enhanced privacy[9]. As a result, we can separate security proofs for CPA/CCA privacy and privacy in the case of mismatch. See Section 3 for more details.

A2: An efficient and strongly secure 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 combining the Boneh-Franklin IBE scheme [5] and the Sakai-Ohgishi-Kasahara IB-NIKE scheme [30]. 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 sender or 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[10]. To achieve the stronger CCA security, we employ the Fujisaki-Okamoto (FO) transformation [19]. Quite surprisingly, the FO transformation allows us to achieve oCMA security for free. Moreover, we formally prove that our scheme also achieves Priv-MisMatch security. As a result, we get a highly efficient 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.

A3: An efficient and strongly secure generic construction of IB-ME in the QROM. We propose a generic construction of IB-ME from anonymous IBE and IBS in the ROM. Toward our new construction, we first observe that Boyen and Li's construction [6], which follows the "Encrypt-then-Sign" paradigm, is unsuitable for realizing CCA privacy. Roughly, in their construction, the ciphertext consists of the IBE ciphertext, two seeds for randomness extractors, and the IBS signature for them. They showed that the

[6] Here, we employ the naming used in a similar situation in signcryption [27]

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

[8] Since all existing schemes, including ours, achieve CPA security against insiders who know sender's secret keys, we do not consider privacy against weaker outsiders explicitly. Therefore, we simply use CPA to refer to security against insiders.

[9] Originally, Francati considered enhanced privacy in the CPA setting. We extend it to the CCA setting and prove the relationship.

[10] 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.

scheme realizes CPA privacy if IBE is CPA secure and IBS is CMA secure. Some readers might imagine that Boyen and Li's IB-ME scheme is CCA secure if IBE is CCA secure. However, this is not the case because other elements in the IB-ME ciphertext may be malleable. Especially if IBS is only CMA secure, the CCA adversary can create another valid signature for the challenge ciphertext without knowing the signing key.[11] To prevent this attack, IBS must be *strong* CMA secure. This means that, to realize CCA security in their IB-ME scheme, both IBE and IBS require stronger security notions.

To obtain a CCA secure IB-ME scheme without assuming strong security for IBS, we construct an IB-ME scheme from IBE and IBS with another approach, which is seen as the "Sign-then-Encrypt" paradigm. In our construction, a sender σ holds an IBS's user key ek_σ, and a receiver ρ holds an IBE's user key. The sender σ encrypts a message m to a receiver rcv as $\mathsf{ct} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m}||\mathsf{sig})$, where $\mathsf{mpk}_{\mathsf{IBE}}$ (resp., $\mathsf{mpk}_{\mathsf{IBS}}$) is a public parameter of the IBE (resp., IBS) scheme and $\mathsf{sig} \leftarrow \mathsf{IBS.Sign}(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{ek}_\sigma, \mathsf{m}||\rho)$. We can show that this simple construction achieves the CCA security and the iCMA security from the CCA security of the IBE scheme and the CMA security of the IBS scheme, respectively. However, it is not $\mathsf{Priv\text{-}MisMatch}$ secure because an adversary who knows the receiver's keys can decrypt the IBE ciphertexts and thus obtain the encrypted messages *without knowing the sender's identity*. To hide ciphertext even in the case of mismatch (i.e., $\mathsf{snd} \neq \sigma$), we employ a random oracle inspired by our BDH-based IBME scheme. Now, the IBE ciphertext ct, which encrypts the message and signature, is masked by the randomness $Z \coloneqq \mathsf{H}(\sigma)$ where H is a random oracle. That is, $\mathsf{ct} \oplus Z$ is the actual ciphertext of our IB-ME scheme. This prevents an adversary from distinguishing the masked ciphertext from a random one without knowing the sender's identity, thanks to the unpredictability of RO. As a result, we can formally show the $\mathsf{Priv\text{-}MisMatch}$ security of our generic construction. We obtain an efficient and strongly secure generic construction of IB-ME in the random oracle model without assuming strong EUF-CMA security for IBS. See Section 5 for more details.

1.3 Related Work

Identity-based encryption. Identity-based encryption, proposed by Shamir [31], 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 [5] using bilinear pairings, and then a lot of IBE schemes have been proposed from various assumptions [1, 14, 20, 22, 23, 33, 34]. 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 [36] is a cryptographic primitive that offers private and authenticated delivery of messages. The motivation for signcryption is to provide equivalent functionality more efficiently than a simple combination of encryption and signature schemes. The notion of identity-based signcryption (IB-SC) was proposed by Malone-Lee [26]. 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 [2, 3]. 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. Recently, Francati et al. [16, 17] 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

[11] In more detail, the IBS signature is one-time padded with the one-time key extracted by the randomness extractor. We note that this does not affect our argument because a one-time pad is malleable.

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. In Section 5, we provide a new generic construction of IB-ME based on IBE and IBS in the QROM. Finally, Section 6 presents a comparison between our IB-ME schemes and the existing schemes.

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 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[12] and the bilinear Diffie-Hellman (BDH) assumption from [7]. 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* $G = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$.

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

Definition 2 (Bilinear Diffie-Hellman (BDH) Assumption [7]). *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*

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

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 $\mathcal{CT}$.

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

$\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}^{\mathcal{A}}_{\mathsf{IBE}}(\lambda)$

$\mathcal{L}_{SK} := \emptyset$
$\mathsf{coin} \leftarrow\$ \{0,1\}$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{id}^*, \mathsf{m}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{SK}, \mathcal{O}_D}(\mathsf{mpk})$
if $\mathsf{id}^* \in \mathcal{L}_{SK}$ **then**
    return coin
$\mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{id}^*, \mathsf{m}^*)$
$\mathsf{ct}_1 \leftarrow \mathcal{CT}$
$\widehat{\mathsf{coin}} \leftarrow \mathcal{A}^{\mathcal{O}_{SK}, \mathcal{O}_D}(\mathsf{ct}_{\mathsf{coin}})$
if $\mathsf{coin} = \widehat{\mathsf{coin}}$ **then**
    return 1
else
    return 0

Oracle $\mathcal{O}_{SK}(\mathsf{id})$

if $\mathsf{id} = \mathsf{id}^*$ **then**
    return $\perp$
$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$
$\mathcal{L}_{SK} \leftarrow \mathcal{L}_{SK} \cup \{\mathsf{id}\}$
return $\mathsf{sk}_{\mathsf{id}}$

Oracle $\mathcal{O}_D(\mathsf{id}, \mathsf{ct})$

if $(\mathsf{id}, \mathsf{ct}) = (\mathsf{id}^*, \mathsf{ct}_{\mathsf{coin}})$ **then**
    return $\perp$
$\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$
$\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathsf{id}}, \mathsf{ct})$
return m

Fig. 1: The security game for IBE.

$\mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}) \to \mathsf{sk}_{\mathsf{id}}$**:** 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

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

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

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

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

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

2.4 Identity-Based Signature

Syntax. An IBS scheme IBS 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 the secret master key msk. mpk defines the identity space $\mathcal{ID}$, message space $\mathcal{M}$, and signature bit length sigLen.

$\mathsf{EUF\text{-}ID\text{-}CMA}^{\mathcal{A}}_{\mathsf{IBS}}(\lambda)$	Oracle $\mathcal{O}_{SK}(\mathsf{id})$
$\mathcal{L}_{SK}, \mathcal{L}_{SIG} := \emptyset$	$\mathsf{sk_{id}} \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	$\mathcal{L}_{SK} \leftarrow \mathcal{L}_{SK} \cup \{\mathsf{id}\}$
$(\mathsf{id}^*, \mathsf{m}^*, \mathsf{sig}^*) \leftarrow \mathcal{A}^{\mathcal{O}_{SK}, \mathcal{O}_{SIG}}(\mathsf{mpk})$	**return** $\mathsf{sk_{id}}$
if $\mathsf{id}^* \in \mathcal{L}_{SK} \vee (\mathsf{id}^*, \mathsf{m}^*) \in \mathcal{L}_{SIG}$ **then**	Oracle $\mathcal{O}_{SIG}(\mathsf{id}, \mathsf{m})$
return 0	$\mathsf{sk_{id}} \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id})$
if $\mathsf{Ver}(\mathsf{mpk}, \mathsf{id}^*, \mathsf{m}^*, \mathsf{sig}^*) = 1$ **then**	$\mathsf{sig} \leftarrow \mathsf{Sign}(\mathsf{mpk}, \mathsf{sk_{id}}, \mathsf{m})$
return 1	$\mathcal{L}_{SIG} \leftarrow \mathcal{L}_{SIG} \cup \{(\mathsf{id}, \mathsf{m})\}$
else	**return** sig
return 0	

Fig. 2: The security game for IBS.

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

$\mathsf{Sign}(\mathsf{mpk}, \mathsf{sk_{id}}, \mathsf{m}) \rightarrow \mathsf{sig}$**:** The signing algorithm takes mpk, $\mathsf{sk_{id}}$, and a message $\mathsf{m} \in \mathcal{M}$ as input and outputs a signature sig.

$\mathsf{Ver}(\mathsf{mpk}, \mathsf{id}, \mathsf{m}, \mathsf{sig}) \rightarrow 0$ **or** 1**:** The verification algorithm takes mpk, $\mathsf{id} \in \mathcal{ID}$, m and sig as input, and outputs a bit $b \in \{0, 1\}$.

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

$$\Pr\left[\mathsf{Ver}(\mathsf{mpk}, \mathsf{id}, \mathsf{m}, \mathsf{sig}) = 1 \;\middle|\; \begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda), \\ \mathsf{sk_{id}} \leftarrow \mathsf{KGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{id}), \\ \mathsf{sig} \leftarrow \mathsf{Sign}(\mathsf{mpk}, \mathsf{sk_{id}}, \mathsf{m}) \end{array}\right] = 1 - \mathsf{negl}(\lambda).$$

Security. We recall adaptive-identity unforgeability against chosen message attacks ($\mathsf{EUF\text{-}ID\text{-}CMA}$ security) [24].

Definition 4 ($\mathsf{EUF\text{-}ID\text{-}CMA}$ Security of IBS). *We say that an IBS scheme* IBS *is* $\mathsf{EUF\text{-}ID\text{-}CMA}$ *secure if for all* PPT *adversaries* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{euf\text{-}id\text{-}cma}}_{\mathcal{A},\mathsf{IBS}}(\lambda) := \Pr\left[\mathsf{EUF\text{-}ID\text{-}CMA}^{\mathcal{A}}_{\mathsf{IBS}}(\lambda) \Rightarrow 1\right] = \mathsf{negl}(\lambda),$$

where the security game $\mathsf{EUF\text{-}ID\text{-}CMA}^{\mathcal{A}}_{\mathsf{IBS}}(\lambda)$ *is depicted in Fig. 2.*

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. [2]. Then, we introduce stronger security notions of them and reformulate privacy in the case of mismatch during decryption introduced by Francati et al. [18].

3.1 Syntax

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

$\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{mpk}, \mathsf{msk})$**:** 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, Reconsidered

Standard 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. We revisit the definitions of privacy and authenticity outlined by Ateniese et al. [2]. To clarify, we rename their definitions *privacy against chosen plaintext attacks* ($\mathsf{Priv\text{-}CPA}$) and *authenticity against no-message attacks from outsiders* ($\mathsf{Auth\text{-}oNMA}$). The term "outsiders" indicates that neither the target sender nor the target receiver is compromised. Subsequently, an authenticity notion is improved in which adversaries can compromise the target receiver [11, 18]. Since the adversary knows the target receiver's key, we call such adversary insiders and call the corresponding authenticity notion *authenticity against no-message attacks from insiders* ($\mathsf{Auth\text{-}iNMA}$). It is worth noting that this distinction between insider and outsider adversaries is a well-established concept in the context of signcryption [27].

The security games are depicted in Fig. 3. We remark that we employ a "real-or-random" style $\mathsf{Priv\text{-}CPA}$ game instead of the Ateniese et al.'s "left-or-right" style game. In greater detail, to account for sender and receiver anonymity, Ateniese et al. designed the security game where the adversary outputs $\{(\mathsf{snd}_i, \mathsf{rcv}_i, \mathsf{m}_i)\}_{i \in \{0,1\}}$ and presents a challenge ciphertext generated with one of them depending on the challenge bit $\mathsf{coin} \in \{0, 1\}$. On the contrary, we define the game in a way that the adversary outputs $(\mathsf{snd}, \mathsf{rcv}, \mathsf{m})$ and is provided with either a real ciphertext generated using this information or a random ciphertext sampled from the ciphertext space $\mathcal{CT}$ similar to the anonymity in IBE (cf. Section 2.3). In essence, our definition asserts that ciphertexts convey no information beyond what is derived from the master public keys. Note that our definition immediately encompasses the Ateniese et al.'s definition.

Definition 5 ($\mathsf{Priv\text{-}CPA}$ **Security of IB-ME).** *We say that an IB-ME scheme* $\mathsf{IB\text{-}ME}$ *is* $\mathsf{Priv\text{-}CPA}$ secure *if for all* PPT *adversaries* $\mathcal{A}$*, it holds that*

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

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

Definition 6 ($\mathsf{Auth\text{-}\{o,i\}NMA}$ **Security of IB-ME).** *Let* $\mathsf{x} \in \{\mathsf{o}, \mathsf{i}\}$*. We say that an IB-ME scheme* $\mathsf{IB\text{-}ME}$ *is* $\mathsf{Auth\text{-}xNMA}$ 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.*

$\mathsf{Priv\text{-}XXX}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$

$/\!\!/ \ \mathsf{XXX} \in \{\mathsf{CPA}, \mathsf{CCA}\}$
$\mathcal{L}_S, \mathcal{L}_R := \emptyset; \mathsf{rcv}^* := \bot$
$\mathsf{coin} \leftarrow\!\!\$\ \{0, 1\}$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
$(\sigma^*, \mathsf{rcv}^*, \mathsf{m}^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{mpk})$
$\textbf{if } \mathsf{rcv}^* \in \mathcal{L}_R \textbf{ then}$
$\quad \textbf{return } \mathsf{coin}$
$\mathsf{ek}_{\sigma^*} \leftarrow \mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma^*)$
$\mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma^*}, \mathsf{rcv}^*, \mathsf{m}^*)$
$\mathsf{ct}_1 \leftarrow\!\!\$\ \mathcal{CT}$
$\widehat{\mathsf{coin}} \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{ct}_{\mathsf{coin}})$
$\textbf{if } \mathsf{coin} = \widehat{\mathsf{coin}} \textbf{ then}$
$\quad \textbf{return } 1$
$\textbf{else}$
$\quad \textbf{return } 0$

$\mathsf{Auth\text{-}xYYY}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$

$/\!\!/ \ \mathsf{x} \in \{\mathsf{o}, \mathsf{i}\},\ \mathsf{YYY} \in \{\mathsf{NMA}, \mathsf{CMA}\}$
$\mathcal{L}_S, \mathcal{L}_R, \mathcal{L}_E := \emptyset; \rho^* := \bot$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathsf{snd}^*, \rho^*, \mathsf{ct}^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{mpk})$
$\mathsf{dk}_{\rho^*} \leftarrow \mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho^*);$
$\mathsf{m}^* \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_{\rho^*}, \mathsf{snd}^*, \mathsf{ct}^*)$
$\textbf{if } \mathsf{x} = \mathsf{o} \wedge \rho^* \in \mathcal{L}_R \textbf{ then}$
$\quad \textbf{return } 0$
$\textbf{if } \mathsf{YYY} = \mathsf{CMA}$
$\qquad \wedge (\mathsf{snd}^*, \rho^*, \mathsf{m}^*) \in \mathcal{L}_E \textbf{ then}$
$\quad \textbf{return } 0$
$\textbf{if } \mathsf{m}^* \neq \bot \wedge \mathsf{snd}^* \notin \mathcal{L}_S \textbf{ then}$
$\quad \textbf{return } 1$
$\textbf{else}$
$\quad \textbf{return } 0$

Available Oracles

$\mathsf{Priv\text{-}CCA} : \mathcal{O} = \{\mathcal{O}_S, \mathcal{O}_R, \mathcal{O}_D\}$
$\mathsf{Auth\text{-}xCMA} : \mathcal{O} = \{\mathcal{O}_S, \mathcal{O}_R, \mathcal{O}_E\}$
Others : $\mathcal{O} = \{\mathcal{O}_S, \mathcal{O}_R\}$

Oracle $\mathcal{O}_S(\sigma)$

$\mathsf{ek}_\sigma \leftarrow \mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma)$
$\mathcal{L}_S \leftarrow \mathcal{L}_S \cup \{\sigma\}$
$\textbf{return } \mathsf{ek}_\sigma$

Oracle $\mathcal{O}_R(\rho)$

$\textbf{if } \rho = \mathsf{rcv}^* \textbf{ then}$
$\quad \textbf{return } \bot$
$\mathsf{dk}_\rho \leftarrow \mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho)$
$\mathcal{L}_R \leftarrow \mathcal{L}_R \cup \{\rho\}$
$\textbf{return } \mathsf{dk}_\rho$

Oracle $\mathcal{O}_E(\sigma, \mathsf{rcv}, \mathsf{m})$

$\mathsf{ek}_\sigma \leftarrow \mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma)$
$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma, \mathsf{rcv}, \mathsf{m})$
$\mathcal{L}_E \leftarrow \mathcal{L}_E \cup \{(\sigma, \mathsf{rcv}, \mathsf{m})\}$
$\textbf{return } \mathsf{ct}$

Oracle $\mathcal{O}_D(\mathsf{snd}, \rho, \mathsf{ct})$

$\textbf{if } (\mathsf{snd}, \rho, \mathsf{ct}) = (\sigma^*, \mathsf{rcv}^*, \mathsf{ct}_{\mathsf{coin}}) \textbf{ then}$
$\quad \textbf{return } \bot$
$\mathsf{dk}_\rho \leftarrow \mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho)$
$\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{snd}, \mathsf{dk}_\rho, \mathsf{ct})$
$\textbf{return } \mathsf{m}$

Fig. 3: The privacy and authenticity games for IB-ME schemes.

$\mathsf{Priv\text{-}MisMatch}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$	Oracle $\mathcal{O}_{E^*}(\mathsf{rcv}, \mathsf{m})$
$\mathcal{L}_S, \mathcal{L}_R := \emptyset$	$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma^*}, \mathsf{rcv}, \mathsf{m})$
$\mathsf{coin} \leftarrow\!\!\$\ \{0,1\}$	**return** ct
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	
$(\Sigma^*, \mathsf{rcv}^*, \mathsf{m}^*) \leftarrow \mathcal{A}^{\mathcal{O}_S, \mathcal{O}_R}(\mathsf{mpk})$	
$\mathsf{dk}_{\mathsf{rcv}^*} \leftarrow \mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \mathsf{rcv}^*)$	
$\sigma^* \leftarrow\!\!\$\ \Sigma^*$ // Sample from the distribution.	
$\mathsf{ek}_{\sigma^*} \leftarrow \mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma^*)$	
$\mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma^*}, \mathsf{rcv}^*, \mathsf{m}^*)$	
$\mathsf{ct}_1 \leftarrow\!\!\$\ \mathcal{CT}$	
$\widehat{\mathsf{coin}} \leftarrow \mathcal{A}^{\mathcal{O}_S, \mathcal{O}_R, \mathcal{O}_{E^*}}(\mathsf{dk}_{\mathsf{rcv}^*}, \mathsf{ct}_{\mathsf{coin}})$	
if $\mathsf{coin} = \widehat{\mathsf{coin}}$ **then return** 1	
else return 0	

Fig. 4: The privacy game in the case of mismatch for IB-ME schemes. The oracles $\mathcal{O}_S$ and $\mathcal{O}_R$ are defined in Fig. 3.

Stronger security notions. In this work, we define stronger security notions for IB-ME. We consider *privacy against chosen-ciphertext attacks* (Priv-CCA) and *authenticity against chosen-message attacks from outsiders or insiders* (Auth-oCMA or Auth-iCMA). In the Priv-CCA game, an adversary can access the decryption oracle, similar to the standard CCA attack scenario. In the Auth-xCMA game, an adversary can access the encryption oracle and receive a ciphertext for a message of its choice, as with the signing oracle in the unforgeability game for (standard) digital signature. These notions Priv-CCA and Auth-xCMA are the desired security properties in practice. We note that Priv-CCA security was first defined in [13, 25] and Auth-iCMA is the same as "strong authenticity" by Wang et al. [32] while Auth-oCMA is newly introduced in this paper.

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

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

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

Definition 8 **(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.*

Privacy in the case of mismatch during decryption. We additionally consider the case where ciphertexts are decrypted with the valid receiver's key but mismatched sender's identities. Intuitively, IB-ME must ensure the privacy of messages in this case from the design concept of IB-ME. This guarantees that an adversary who compromises a receiver but has no knowledge about the sender cannot decrypt ciphertexts. This is a crucial security property of IB-ME, but the original work did not consider it explicitly[13]. Subsequently,

[13] Ateniese et al. informally argued that their IB-ME scheme hides the message and the sender's identity in the case of mismatch, but they did not provide a formal model or a formal proof.

$\mathsf{Priv}^{+}\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)/\boxed{\mathsf{Priv}^{+}\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)}$

$\mathcal{L}_S, \mathcal{L}_R := \emptyset$
$\mathsf{coin} \leftarrow\!\!\$ \{0,1\}$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^{\lambda})$
$(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1, \mathsf{m}_0^*, \mathsf{m}_1^*) \leftarrow \mathcal{A}^{\mathcal{O}_S, \mathcal{O}_R, \boxed{\mathcal{O}_D}}(\mathsf{mpk})$
$\sigma_0^* \leftarrow\!\!\$ \Sigma_0^*$ // Sample from the distribution.
$\sigma_1^* \leftarrow\!\!\$ \Sigma_1^*$ // Sample from the distribution.
$\mathsf{ek}_{\sigma_0^*} \leftarrow \mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma_0^*)$
$\mathsf{ek}_{\sigma_1^*} \leftarrow \mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma_1^*)$
$\mathsf{ct}_{\mathsf{coin}} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma^*_{\mathsf{coin}}}, \mathsf{rcv}^*_{\mathsf{coin}}, \mathsf{m}^*)$
$\widehat{\mathsf{coin}} \leftarrow \mathcal{A}^{\mathcal{O}_S, \mathcal{O}_R, \mathcal{O}_{E^*}, \boxed{\mathcal{O}_D}}(\mathsf{ct}_{\mathsf{coin}})$
if $\mathsf{coin} = \widehat{\mathsf{coin}}$ **then return** 1
else return 0

Oracle $\mathcal{O}_R(\mathsf{rcv})$

$\mathsf{dk}_{\rho} \leftarrow \mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho)$
$\mathcal{L}_R \leftarrow \mathcal{L}_R \cup \{\rho\}$
return dk_{ρ}

Oracle $\mathcal{O}_{E^*}(i \in \{0,1\}, \mathsf{rcv}, \mathsf{m})$

$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_i^*}, \mathsf{rcv}, \mathsf{m})$
return ct

Fig. 5: The enhanced privacy game for IB-ME schemes. The boxed codes are only for $\mathsf{Priv}^{+}\text{-}\mathsf{CCA}$ game. The oracles $\mathcal{O}_S, \mathcal{O}_D$ is defined in Fig. 3.

Francati et al. [18] defined a new privacy notion called "enhanced privacy" that captures privacy in this case. To model an adversary that does not know who the sender is, Francati et al. assumed that the target senders' identities are chosen from the corresponding high min-entropy distributions. Their definition effectively captures this intuition, but they used a single game that includes both conventional privacy and privacy in the case of mismatch, complicating the understanding of the definition and security proofs. In addition, their original definition does not capture the so-called "offline guessing attack", where an adversary who knows the decryption key can try to guess the sender's identity locally after it gets a ciphertext. Therefore, in this work, we redefine the above intuition as another simple security game, which we call Priv-MisMatch security.

The new security game Priv-MisMatch is shown in Fig. 4. The difference from the definition by Francati et al. is that (1) the adversary specifies one target receiver and is given the secret key of the target receiver explicitly, and (2) the adversary tries to distinguish whether the challenge ciphertext is real or random as Priv-CPA/Priv-CCA games. This represents the intuition that, even if the adversary knows the the target receiver's key, it is difficult for the adversary to guess the sender's identity and the privacy of messages is guaranteed (i.e., a ciphertext does not leak any information about the sender, receiver, and the encrypted message). Also, we explicitly consider the advantage due to the offline guessing attack. The formal definition is as follows.

Definition 9 **(Priv-MisMatch Security of IB-ME).** *We say that an IB-ME scheme* IB-ME *is* Priv-MisMatch secure *if for all κ-admissible* PPT *adversaries $\mathcal{A}$, it holds that*

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

where the security game $\mathsf{Priv\text{-}MisMatch}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$ is depicted in Fig. 4 and $\mathbf{T}_{\mathcal{A}} = \mathsf{poly}(\lambda)$ denotes the running time of $\mathcal{A}$. Note that we say that the adversary $\mathcal{A}$ is κ-admissible if its outputs Σ^* is κ-distributions.

The term $\mathbf{T}_{\mathcal{A}}/2^{\kappa}$ represents the advantage of the adversary's offline guessing attacks. Since the adversary knows the receiver's decryption key, it can perform an exhaustive search offline and find the correct sender's identity. If the sender's identities are chosen from a distribution with sufficiently large entropy, such a guess is infeasible for PPT adversaries. Therefore, for a reasonable Priv-MisMatch security, $\kappa \geq \omega(\log \lambda)$ would be assumed [18]. In this case, we have $\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) = \mathsf{negl}(\lambda)$.

3.3 Relationship between Our Definition and Francati's One

In the following, we show that $\mathsf{Priv\text{-}CPA}/\mathsf{Priv\text{-}CCA}$ and $\mathsf{Priv\text{-}MisMatch}$ cover all the security notions considered in the enhanced privacy in [18]. The enhanced privacy [18] defined by Francati et al. considered the following three cases:

- (1) the adversary cannot decrypt the challenge (i.e., it does not hold a valid decryption key),
- (2) the sender's attributes do not match with the receiver's policy on the challenge (i.e., the challenge has a policy that is not satisfied by the sender),
- (3) the adversary receives two challenges. For the first challenge, the adversary cannot decrypt it, and for the second challenge, the sender's attributes do not match with the receiver's policy. In a nutshell, this condition is a hybrid of (1) and (2).

Formally, they are defined as follows. Originally, Francati et al. defined CPA security, but here we also consider CCA security.

Definition 10 **(Priv^+-CPA Security of IB-ME [18]).** *We say that an IB-ME scheme* $\mathsf{IB\text{-}ME}$ *is* Priv^+-CPA secure *if for all κ-admissible* PPT *adversaries $\mathcal{A}$, it holds that*

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

where the security game $\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda)$ *is depicted in Fig. 5. An adversary $\mathcal{A}$ is admissible if for all $\rho \in \mathcal{L}_R$, it satisfies at least one of the following invariants*

$$\begin{array}{ll} \textit{(Case 1)} & \rho \neq \mathsf{rcv}_0^* \wedge \rho \neq \mathsf{rcv}_1^*, \\ \textit{(Case 2)} & H_\infty(\Sigma_0^*) \geq \kappa \wedge H_\infty(\Sigma_1^*) \geq \kappa, \\ \textit{(Case 3)} & \rho \neq \mathsf{rcv}_0^* \wedge H_\infty(\Sigma_1^*) \geq \kappa, \\ \textit{(Case 4)} & \rho \neq \mathsf{rcv}_1^* \wedge H_\infty(\Sigma_0^*) \geq \kappa. \end{array}$$

Definition 11 **(Priv^+-CCA Security of IB-ME).** *We say that an IB-ME scheme* $\mathsf{IB\text{-}ME}$ *is* Priv^+-CCA secure *if for all κ-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. 5. An adversary $\mathcal{A}$ is admissible if for all $\rho \in \mathcal{L}_R$, it satisfies at least one of the following invariants*

$$\begin{array}{ll} \textit{(Case 1)} & \rho \neq \mathsf{rcv}_0^* \wedge \rho \neq \mathsf{rcv}_1^*, \\ \textit{(Case 2)} & H_\infty(\Sigma_0^*) \geq \kappa \wedge H_\infty(\Sigma_1^*) \geq \kappa, \\ \textit{(Case 3)} & \rho \neq \mathsf{rcv}_0^* \wedge H_\infty(\Sigma_1^*) \geq \kappa, \\ \textit{(Case 4)} & \rho \neq \mathsf{rcv}_1^* \wedge H_\infty(\Sigma_0^*) \geq \kappa. \end{array}$$

The following theorems show that $\mathsf{Priv\text{-}CPA}/\mathsf{Priv\text{-}CCA}$ and $\mathsf{Priv\text{-}MisMatch}$ cover all the security notions considered by Francati et al. [18].

Theorem 1. *If an IB-ME scheme* $\mathsf{IB\text{-}ME}$ *satisfies* $\mathsf{Priv\text{-}CPA}$ *(resp.,* $\mathsf{Priv\text{-}CCA}$*) security and* $\mathsf{Priv\text{-}MisMatch}$ *security, then it satisfies* Priv^+-CPA *(resp.,* Priv^+-CCA*) security.*

Proof. We first prove **Case 1 (that is $\rho \neq \mathsf{rcv}_0^* \wedge \rho \neq \mathsf{rcv}_1^*$).**

Lemma 1. *If there exists an adversary $\mathcal{A}$ that breaks the* Priv^+-CPA *(resp.,* Priv^+-CCA*) security for case 1, there exists an adversary $\mathcal{B}$ that breaks* $\mathsf{Priv\text{-}CPA}$ *(resp.,* $\mathsf{Priv\text{-}CCA}$*) security such that*

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 1}\right] - \frac{1}{2}\right| \leq 2\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda)$$

$$\left(\textit{resp.}, \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 1}\right] - \frac{1}{2}\right| \leq 2\mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda)\right).$$

Proof. To prove the lemma, we consider the following sequence of games Game_i for $i \in \{0,1,2\}$. We 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 security game conditioned on $\mathsf{coin} = 0$.

Game_1**.** In this game, we replace $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_0^*}, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$ with $\mathsf{ct}^* \leftarrow\$\ \mathcal{CT}$.

To show the difference between Game_0 and Game_1 is negligible, we construct $\mathcal{B}_1$ that breaks the $\mathsf{Priv\text{-}CPA}$ (resp., $\mathsf{Priv\text{-}CCA}$) security of $\mathsf{IB\text{-}ME}$ using $\mathcal{A}$. The boxed descriptions are only used for the reduction of $\mathsf{Priv\text{-}CCA}$.

1. Upon receiving the master public key mpk, $\mathcal{B}_1$ executes $\mathcal{A}$ on input mpk.
2. $\mathcal{B}_1$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ sends σ to its $\mathcal{O}_S$ oracle and receives ek_σ. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_1$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to its $\mathcal{O}_D$ oracle and receives m. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.
3. When $\mathcal{A}$ sends $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_1$ picks $\sigma_0^* \leftarrow\$\ \Sigma_0^*$ and $\sigma_1^* \leftarrow\$\ \Sigma_1^*$, sends $(\sigma_0^*, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$ to its challenger, and receives the challenge ciphertext ct^*. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$. Moreover, $\mathcal{B}_1$ sends σ_0^* and σ_1^* to its $\mathcal{O}_S$ oracle and receives $\mathsf{ek}_{\sigma_0^*}$ and $\mathsf{ek}_{\sigma_1^*}$.
4. $\mathcal{B}_1$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$, $\mathcal{B}_1$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_i^*}, \mathsf{rcv}, \mathsf{m})$ and returns ct to $\mathcal{A}$.
   (d) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_1$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to its $\mathcal{O}_D$ oracle and receives m. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.
5. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}_1$ sends it to the challenger as its guess.

Let $\mathsf{coin}' \in \{0,1\}$ be the challenge bit for $\mathcal{B}_1$. It is clear that if $\mathsf{coin}' = 0$ (that is $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\mathsf{ek}_{\sigma_0^*}}, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$), then $\mathcal{B}_1$ perfectly simulates Game_0 for $\mathcal{A}$. On the other hand, if $\mathsf{coin}' = 1$ (that is $\mathsf{ct}^* \leftarrow\$\ \mathcal{CT}$), then $\mathcal{B}_1$ perfectly simulates Game_1 for $\mathcal{A}$. Therefore, we have $\epsilon_0 = \Pr\Big[\mathsf{coin}' = \widehat{\mathsf{coin}} \,\Big|\, \mathsf{coin}' = 0\Big]$ and $\epsilon_1 = \Pr\Big[\mathsf{coin}' = \widehat{\mathsf{coin}} \,\Big|\, \mathsf{coin}' = 1\Big]$. Thus, we have

$$\begin{aligned}
\Pr\Big[\mathsf{Priv\text{-}CPA}_{\mathsf{IB\text{-}ME}}^{\mathcal{B}_1}(\lambda) \Rightarrow 1\Big] &= \frac{1}{2}\left(\Pr\Big[\mathsf{coin}' = \widehat{\mathsf{coin}} \,\Big|\, \mathsf{coin}' = 1\Big] + \Pr\Big[\mathsf{coin}' = \widehat{\mathsf{coin}} \,\Big|\, \mathsf{coin}' = 0\Big]\right) \\
&= \frac{1}{2}\left(1 - \Pr\Big[\widehat{\mathsf{coin}} = 0 \,\Big|\, \mathsf{coin}' = 1\Big] + \Pr\Big[\widehat{\mathsf{coin}} = 0 \,\Big|\, \mathsf{coin}' = 0\Big]\right) \\
&= \frac{1}{2}\left(1 - \epsilon_1 + \epsilon_0\right),
\end{aligned}$$

$$\left(\text{resp., } \Pr\Big[\mathsf{Priv\text{-}CCA}_{\mathsf{IB\text{-}ME}}^{\mathcal{B}_1}(\lambda) \Rightarrow 1\Big] = \frac{1}{2}(1 - \epsilon_1 + \epsilon_0)\right),$$

which in turn implies

$$|\epsilon_0 - \epsilon_1| = 2\mathsf{Adv}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}^{\mathsf{priv\text{-}cpa}}(\lambda) \quad \left(\text{resp., } |\epsilon_0 - \epsilon_1| = 2\mathsf{Adv}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}^{\mathsf{priv\text{-}cca}}(\lambda)\right).$$

Game_2. In this game, we replace $\mathsf{ct}^* \leftarrow\!\$\ \mathcal{CT}$ with $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_1^*}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$. By doing this, Game_2 is the original security game conditioned on $\mathsf{coin} = 1$.

To show the difference between Game_1 and Game_2 is negligible, we construct $\mathcal{B}_2$ that breaks Priv-CPA (resp., Priv-CCA) security of IB-ME using $\mathcal{A}$. The boxed descriptions are only used for the reduction of Priv-CCA.

1. Upon receiving the master public key mpk, $\mathcal{B}_2$ executes $\mathcal{A}$ on input mpk.
2. $\mathcal{B}_2$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ sends σ to its $\mathcal{O}_S$ oracle and receives ek_σ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_2$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to its $\mathcal{O}_D$ oracle and receives m. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
3. When $\mathcal{A}$ sends $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_2$ picks $\sigma_0^* \leftarrow\!\$\ \Sigma_0^*$ and $\sigma_1^* \leftarrow\!\$\ \Sigma_1^*$, sends $(\sigma_1^*, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$ to its challenger, and receives the challenge ciphertext ct^*. Then $\mathcal{B}_2$ returns it to $\mathcal{A}$. Moreover, $\mathcal{B}_2$ sends σ_0^* and σ_1^* to its challenger and receives $\mathsf{ek}_{\sigma_0^*}$ and $\mathsf{ek}_{\sigma_1^*}$.
4. $\mathcal{B}_2$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ sends σ to its $\mathcal{O}_S$ oracle and receives ek_σ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$, $\mathcal{B}_2$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_i^*}, \mathsf{rcv}, \mathsf{m})$ and returns ct to $\mathcal{A}$.
   (d) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_2$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to its $\mathcal{O}_D$ oracle and receives m. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
5. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}_2$ sends it to the challenger as its guess.

Let $\mathsf{coin}' \in \{0,1\}$ be the challenge bit for $\mathcal{B}_2$. It is clear that if $\mathsf{coin}' = 0$ (that is $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\mathsf{ek}_{\sigma_1^*}}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$), then $\mathcal{B}_2$ perfectly simulates Game_2 for $\mathcal{A}$. On the other hand, if $\mathsf{coin}' = 1$ (that is $\mathsf{ct}^* \leftarrow\!\$\ \mathcal{CT}$), then $\mathcal{B}_2$ perfectly simulates Game_1 for $\mathcal{A}$. Therefore, we have $\epsilon_1 = \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 1\right]$ and $\epsilon_2 = \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 0\right]$. Thus, we have

$$\begin{aligned}
\Pr\left[\mathsf{Priv\text{-}CPA}^{\mathcal{B}_2}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] &= \frac{1}{2}\left(\Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 0\right] + \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 1\right]\right) \\
&= \frac{1}{2}\left(1 - \Pr\left[\widehat{\mathsf{coin}} = 0 \;\middle|\; \mathsf{coin}' = 0\right] + \Pr\left[\widehat{\mathsf{coin}} = 0 \;\middle|\; \mathsf{coin}' = 1\right]\right) \\
&= \frac{1}{2}\left(1 - \epsilon_2 + \epsilon_1\right),
\end{aligned}$$

$$\left(\text{resp., } \Pr\left[\mathsf{Priv\text{-}CCA}^{\mathcal{B}_1}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] = \frac{1}{2}(1 - \epsilon_1 + \epsilon_0)\right),$$

which in turn implies

$$|\epsilon_1 - \epsilon_2| = 2\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda) \quad \left(\text{resp., } |\epsilon_1 - \epsilon_2| = 2\mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)\right).$$

From the above arguments, we have

$$\left|\Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 1\right] - \frac{1}{2}\right|$$
$$\leq \frac{1}{2}\left|\Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 1 \land \mathsf{coin} = 0\right] - \Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 1 \land \mathsf{coin} = 1\right]\right|$$
$$= \frac{1}{2}|\epsilon_0 - \epsilon_2| \leq \frac{1}{2}(|\epsilon_0 - \epsilon_1| + |\epsilon_1 - \epsilon_2|) = 2\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda),$$

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

□

Next, we prove **Case 2 (that is $H_\infty(\Sigma_0^*) \geq \kappa \land H_\infty(\Sigma_1^*) \geq \kappa$).**

Lemma 2. *If there exists an adversary $\mathcal{A}$ that breaks the* $\mathsf{Priv}^{+}\text{-}\mathsf{CPA}$ *(resp.,* $\mathsf{Priv}^{+}\text{-}\mathsf{CCA}$*) security for case 2, then there exists an adversary $\mathcal{B}$ that breaks* $\mathsf{Priv\text{-}MisMatch}$ *security such that*

$$\left|\Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 2}\right] - \frac{1}{2}\right| \leq 2\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda)$$
$$\left(\textit{resp., } \left|\Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 2}\right] - \frac{1}{2}\right| \leq 2\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda)\right).$$

Proof. To prove the lemma, we consider the following sequence of games Game_i for $i \in \{0, 1, 2\}$. We 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 conditioned on $\mathsf{coin} = 0$.

Game_1**.** In this game, we replace $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_0^*}, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$ with $\mathsf{ct}^* \leftarrow\$ \mathcal{CT}$.

To show the difference between Game_0 and Game_1 is negligible, we construct $\mathcal{B}_1$ that breaks $\mathsf{Priv\text{-}MisMatch}$ security of $\mathsf{IB\text{-}ME}$ using $\mathcal{A}$. The boxed descriptions are only used to simulate the $\mathsf{Priv}^{+}\text{-}\mathsf{CCA}$ game.

1. Upon receiving the master public key mpk, $\mathcal{B}_1$ executes $\mathcal{A}$ on input mpk.
2. $\mathcal{B}_1$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ sends σ to its $\mathcal{O}_S$ oracle and receives ek_σ. Then $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.

(c) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$, $\mathcal{B}_1$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_1$ computes $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$ and returns it to $\mathcal{A}$.

3. When $\mathcal{A}$ sends $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_1$ picks $\sigma_1^* \leftarrow\$ \Sigma_1^*$, sends $(\Sigma_0^*, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$ to its challenger and receives the target receiver key and challenge ciphertext $(\mathsf{dk}_{\mathsf{rcv}_0^*}, \mathsf{ct}^*)$. Then, $\mathcal{B}_1$ returns ct^* to $\mathcal{A}$. Moreover, $\mathcal{B}_1$ sends σ_1^* to its $\mathcal{O}_S$ oracle and receives $\mathsf{ek}_{\sigma_1^*}$.
4. $\mathcal{B}_1$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ sends σ to its $\mathcal{O}_S$ oracle and receives ek_σ. Then $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ returns $\mathsf{dk}_{\mathsf{rcv}_0^*}$ to $\mathcal{A}$ if $\rho = \mathsf{rcv}_0^*$. Otherwise, $\mathcal{B}_1$ sends ρ to its $\mathcal{O}_R$ oracle, receives dk_ρ, and returns it to $\mathcal{A}$.

(c) When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$, if $i = 0$ holds, $\mathcal{B}_1$ sends $(\mathsf{rcv}, \mathsf{m})$ to its $\mathcal{O}_{E^*}$ oracle and receives ct. Otherwise, $\mathcal{B}_1$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_1^*}, \mathsf{rcv}, \mathsf{m})$. Then $\mathcal{B}_1$ returns ct to $\mathcal{A}$.

(d) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$, $\mathcal{B}_1$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_1$ computes $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$ and returns it to $\mathcal{A}$.

5. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}_1$ sends it to the challenger as its guess.

Let $\mathsf{coin}' \in \{0,1\}$ be the challenge bit for $\mathcal{B}_1$. It is clear that if $\mathsf{coin}' = 0$ (that is $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\mathsf{ek}_{\sigma_0^*}}, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$), then $\mathcal{B}_1$ perfectly simulates Game_0 for $\mathcal{A}$. On the other hand, if $\mathsf{coin}' = 1$ (that is $\mathsf{ct}^* \leftarrow\!\!\$\ \mathcal{CT}$), then $\mathcal{B}_1$ perfectly simulates Game_1 for $\mathcal{A}$. Therefore, we have $\epsilon_0 = \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \,\middle|\, \mathsf{coin}' = 0\right]$ and $\epsilon_1 = \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \,\middle|\, \mathsf{coin}' = 1\right]$. Thus, we have

$$\begin{aligned}\Pr\left[\mathsf{Priv\text{-}MisMatch}^{\mathcal{B}_1}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] &= \frac{1}{2}\left(\Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \,\middle|\, \mathsf{coin}' = 1\right] + \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \,\middle|\, \mathsf{coin}' = 0\right]\right)\\ &= \frac{1}{2}\left(1 - \Pr\left[\widehat{\mathsf{coin}} = 0 \,\middle|\, \mathsf{coin}' = 1\right] + \Pr\left[\widehat{\mathsf{coin}} = 0 \,\middle|\, \mathsf{coin}' = 0\right]\right)\\ &= \frac{1}{2}\left(1 - \epsilon_1 + \epsilon_0\right),\end{aligned}$$

which in turn implies

$$|\epsilon_0 - \epsilon_1| = 2\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda).$$

Game_2**.** In this game, we replace $\mathsf{ct}^* \leftarrow\!\!\$\ \mathcal{CT}$ with $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_1^*}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$. By doing this, Game_2 is the original security game conditioned on $\mathsf{coin} = 1$.

To show the difference between Game_1 and Game_2 is negligible, we construct $\mathcal{B}_2$ that breaks Priv-MisMatch security of IB-ME using $\mathcal{A}$. The boxed descriptions are only used to simulate the Priv$^+$-CCA game.

1. Upon receiving the master public key mpk, $\mathcal{B}_2$ executes $\mathcal{A}$ on input mpk.
2. $\mathcal{B}_2$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ sends σ to its $\mathcal{O}_S$ oracle and receives ek_σ. Then $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.

   (c) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$, $\mathcal{B}_2$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_2$ computes $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$ and returns it to $\mathcal{A}$.

3. When $\mathcal{A}$ sends $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_2$ picks $\sigma_0^* \leftarrow\!\!\$\ \Sigma_0^*$, sends $(\Sigma_1^*, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$ to its challenger, and receives the target receiver key and challenge ciphertext $(\mathsf{dk}_{\mathsf{rcv}_1^*}, \mathsf{ct}^*)$. Then $\mathcal{B}_2$ returns ct^* to $\mathcal{A}$. Moreover, $\mathcal{B}_2$ sends σ_0^* to its challenger and receives $\mathsf{ek}_{\sigma_0^*}$.
4. $\mathcal{B}_2$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ sends σ to its $\mathcal{O}_S$ oracle and receives ek_σ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ returns $\mathsf{dk}_{\mathsf{rcv}_1^*}$ to $\mathcal{A}$ if $\rho = \mathsf{rcv}_1^*$. Otherwise, $\mathcal{B}_2$ sends ρ to its $\mathcal{O}_R$ oracle, receives dk_ρ, and returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$ oracle, if $i = 1$ holds, $\mathcal{B}_2$ sends $(\mathsf{rcv}, \mathsf{m})$ to its $\mathcal{O}_{E^*}$ oracle and receives ct. Otherwise, $\mathcal{B}_2$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_0^*}, \mathsf{rcv}, \mathsf{m})$. Then $\mathcal{B}_2$ returns ct to $\mathcal{A}$.

   (d) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$, $\mathcal{B}_2$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_2$ computes $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$ and returns it to $\mathcal{A}$.

5. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}_2$ sends it to the challenger as its guess.

Let $\mathsf{coin}' \in \{0,1\}$ be the challenge bit for $\mathcal{B}_2$. It is clear that if $\mathsf{coin}' = 0$ (that is $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\mathsf{ek}_{\sigma_1^*}}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$), then $\mathcal{B}_2$ perfectly simulates Game_2 for $\mathcal{A}$. On the other hand, if $\mathsf{coin}' = 1$ (that is $\mathsf{ct}^* \leftarrow\!\$\, \mathcal{CT}$), then $\mathcal{B}_2$ perfectly simulates Game_1 for $\mathcal{A}$. Therefore, we have $\epsilon_1 = \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 1\right]$ and $\epsilon_2 = \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 0\right]$. Thus, we have

$$\begin{aligned}\Pr\left[\mathsf{Priv\text{-}MisMatch}^{\mathcal{B}_2}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] &= \frac{1}{2}\left(\Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 0\right] + \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 1\right]\right)\\ &= \frac{1}{2}\left(1 - \Pr\left[\widehat{\mathsf{coin}} = 0 \;\middle|\; \mathsf{coin}' = 0\right] + \Pr\left[\widehat{\mathsf{coin}} = 0 \;\middle|\; \mathsf{coin}' = 1\right]\right)\\ &= \frac{1}{2}\left(1 - \epsilon_2 + \epsilon_1\right),\end{aligned}$$

Thus, we have

$$|\epsilon_1 - \epsilon_2| = 2\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda).$$

From the above arguments, we have

$$\begin{aligned}&\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 2\right] - \frac{1}{2}\right|\\ &\le \frac{1}{2}\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 2 \wedge \mathsf{coin} = 0\right] - \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 2 \wedge \mathsf{coin} = 1\right]\right|\\ &= \frac{1}{2}|\epsilon_0 - \epsilon_2| \le \frac{1}{2}\left(|\epsilon_0 - \epsilon_1| + |\epsilon_1 - \epsilon_2|\right) = 2\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda).\end{aligned}$$

$$\left(\text{resp., } \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 2\right] - \frac{1}{2}\right| \le 2\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda)\right).$$

□

Next, we prove **Case 3 (that is $\rho \neq \mathsf{rcv}_0^* \wedge H_\infty(\Sigma_1^*) \geq \kappa$).**

Lemma 3. *If there exists an adversary $\mathcal{A}_3$ that breaks the* $\mathsf{Priv}^+\text{-}\mathsf{CPA}$ *(resp.,* $\mathsf{Priv}^+\text{-}\mathsf{CCA}$*) security for case 3, there exists an adversary $\mathcal{B}_1$ that breaks* $\mathsf{Priv\text{-}CPA}$ *(resp.,* $\mathsf{Priv\text{-}CCA}$*) security and $\mathcal{B}_2$ that breaks* $\mathsf{Priv\text{-}MisMatch}$ *security such that*

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 3}\right] - \frac{1}{2}\right| \le \mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda) + \mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)$$

$$\left(\textit{resp., } \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \textit{Case 3}\right] - \frac{1}{2}\right| \le \mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda) + \mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)\right).$$

Proof. To prove the lemma, we consider the following sequence of games Game_i for $i \in \{0,1,2\}$. We define the advantage of $\mathcal{A}$ in Game_i as

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

Game_0**.** This is the original security game conditioned on $\mathsf{coin} = 0$.

Game_1**.** In this game, we replace $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_0^*}, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$ with $\mathsf{ct}^* \leftarrow\!\$\, \mathcal{CT}$.

To show the difference between Game_0 and Game_1 is negligible, we construct $\mathcal{B}_1$ that breaks $\mathsf{Priv\text{-}CPA}$ security of $\mathsf{IB\text{-}ME}$ using $\mathcal{A}$. The boxed descriptions are only used for the reduction of $\mathsf{Priv\text{-}CCA}$.

1. Upon receiving the master public key mpk, $\mathcal{B}_1$ executes $\mathcal{A}$ on input mpk.
2. $\mathcal{B}_1$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_1$ returns it to $\mathcal{A}$.

(b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.

> (c) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_1$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to its $\mathcal{O}_D$ oracle and receives m. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.

3. When $\mathcal{A}$ sends $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_1$ picks $\sigma_0^* \leftarrow\$ \Sigma_0^*$ and $\sigma_1^* \leftarrow\$ \Sigma_1^*$, and sends $(\sigma_0^*, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$ to its challenger and receives the challenge ciphertext ct^*. Then $\mathcal{B}_1$ returns it to $\mathcal{A}$. Moreover, $\mathcal{B}_1$ sends σ_0^* and σ_1^* to its challenger and receives $\mathsf{ek}_{\sigma_0^*}$ and $\mathsf{ek}_{\sigma_1^*}$.
4. $\mathcal{B}_1$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$, $\mathcal{B}_1$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_i^*}, \mathsf{rcv}, \mathsf{m})$ and returns ct to $\mathcal{A}$.

> (d) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_1$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to its $\mathcal{O}_D$ oracle and receives m. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.

5. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}$ sends it to the challenger as its guess.

Here, since we assume the case 3, $\mathcal{A}$ might make a decryption key query ρ such that $\rho \neq \mathsf{rcv}_0^*$. Note that, if $\mathcal{A}$ makes a decryption key query $\rho = \mathsf{rcv}_1^*$, $\mathcal{B}_1$ can make a decryption key query on input ρ since rcv_1^* is not the challenge receiver identity for $\mathcal{B}_1$ in the $\mathsf{Priv\text{-}CPA}$ game. Let $\mathsf{coin}' \in \{0,1\}$ be the challenge bit for $\mathcal{B}_1$. Then, it is clear that if $\mathsf{coin}' = 0$ (that is $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\mathsf{ek}_{\sigma_0^*}}, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$), then $\mathcal{B}_1$ perfectly simulates Game_0 for $\mathcal{A}$. On the other hand, if $\mathsf{coin}' = 1$ (that is $\mathsf{ct}^* \leftarrow\$ \mathcal{CT}$), then $\mathcal{B}_1$ perfectly simulates Game_1 for $\mathcal{A}$. Therefore, we have $\epsilon_0 = \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 0\right]$ and $\epsilon_1 = \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 1\right]$. Thus, we have

$$\begin{aligned}
\Pr\left[\mathsf{Priv\text{-}CPA}^{\mathcal{B}_1}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] &= \frac{1}{2}\left(\Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 1\right] + \Pr\left[\mathsf{coin}' = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin}' = 0\right]\right) \\
&= \frac{1}{2}\left(1 - \Pr\left[\widehat{\mathsf{coin}} = 0 \;\middle|\; \mathsf{coin}' = 1\right] + \Pr\left[\widehat{\mathsf{coin}} = 0 \;\middle|\; \mathsf{coin}' = 0\right]\right) \\
&= \frac{1}{2}\left(1 - \epsilon_1 + \epsilon_0\right),
\end{aligned}$$

which in turn implies

$$|\epsilon_0 - \epsilon_1| = 2\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_1, \mathsf{IB\text{-}ME}}(\lambda).$$

Game_2**.** In this game, we replace $\mathsf{ct}^* \leftarrow\$ \mathcal{CT}$ with $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_1^*}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$. By doing this, Game_2 is the original security game conditioned on $\mathsf{coin} = 1$.

To show difference between Game_1 and Game_2 are negligible, we construct $\mathcal{B}_2$ that breaks $\mathsf{Priv\text{-}MisMatch}$ security of $\mathsf{IB\text{-}ME}$ using $\mathcal{A}$. The boxed descriptions are only used to simulate the $\mathsf{Priv}^+\text{-}\mathsf{CCA}$ game.

1. Upon receiving the master public key mpk, $\mathcal{B}_2$ executes $\mathcal{A}$ on input mpk.
2. $\mathcal{B}_2$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.

> (c) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$, $\mathcal{B}_2$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_2$ computes $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$ and returns it to $\mathcal{A}$.

3. When $\mathcal{A}$ sends $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_2$ picks $\sigma_0^* \leftarrow\!\!\$\ \Sigma_0^*$, and sends $(\Sigma_1^*, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$ to its challenger and receives the challenge ciphertext $\mathsf{ct}^*, \mathsf{dk}_{\rho_1^*}$. Then $\mathcal{B}_2$ returns ct^* to $\mathcal{A}$. Moreover, $\mathcal{B}_2$ sends σ_0^* to its challenger and receives $\mathsf{ek}_{\sigma_0^*}$.
4. $\mathcal{B}_2$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ returns $\mathsf{dk}_{\rho_1^*}$ to $\mathcal{A}$ if $\rho = \rho_1^*$. Otherwise, $\mathcal{B}_2$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$, $\mathcal{B}_2$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to its challenger and receives ct if $i = 1$. Otherwise, $\mathcal{B}_2$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_0^*}, \mathsf{rcv}, \mathsf{m})$. Then $\mathcal{B}_2$ returns ct to $\mathcal{A}$.

   (d) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$, $\mathcal{B}_2$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_2$ computes $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$ and returns it to $\mathcal{A}$.
5. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}_2$ sends it to the challenger as its guess.

It is clear that if $\mathsf{coin} = 0$ (that is $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\mathsf{ek}_{\sigma_1^*}}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$), then $\mathcal{B}_2$ perfectly simulates Game_2 for $\mathcal{A}$. On the other hand, if $\mathsf{coin} = 1$ (that is $\mathsf{ct}^* \leftarrow\!\!\$\ \mathcal{CT}$), then $\mathcal{B}_2$ perfectly simulates Game_1 for $\mathcal{A}$. Therefore, we have $\epsilon_1 = \Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}} \,\middle|\, \mathsf{coin} = 1\right]$ and $\epsilon_2 = \Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}} \,\middle|\, \mathsf{coin} = 0\right]$. Thus, we have

$$\begin{aligned}\Pr\left[\mathsf{Priv\text{-}MisMatch}^{\mathcal{B}_2}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] &= \frac{1}{2}\left(\Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}} \,\middle|\, \mathsf{coin} = 0\right] + \Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}} \,\middle|\, \mathsf{coin} = 1\right]\right) \\ &= \frac{1}{2}\left(1 - \Pr\left[\widehat{\mathsf{coin}} = 0 \,\middle|\, \mathsf{coin} = 0\right] + \Pr\left[\widehat{\mathsf{coin}} = 0 \,\middle|\, \mathsf{coin} = 1\right]\right) \\ &= \frac{1}{2}\left(1 - \epsilon_2 + \epsilon_1\right).\end{aligned}$$

Thus, we have

$$|\epsilon_1 - \epsilon_2| = 2\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda).$$

From the above arguments, we have

$$\begin{aligned}&\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \,\middle|\, \text{Case } 3\right] - \frac{1}{2}\right| \\ &\le \frac{1}{2}\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \,\middle|\, \text{Case } 3 \wedge \mathsf{coin} = 0\right] - \Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \,\middle|\, \text{Case } 3 \wedge \mathsf{coin} = 1\right]\right| \\ &= \frac{1}{2}|\epsilon_0 - \epsilon_2| \le \frac{1}{2}(|\epsilon_0 - \epsilon_1| + |\epsilon_1 - \epsilon_2|) = \mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda) + \mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)\end{aligned}$$

$$\left(\text{resp., } \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \,\middle|\, \text{Case } 3\right] - \frac{1}{2}\right| \le \mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda) + \mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)\right).$$

□

Finally, we prove **Case 4 (that is** $\rho \neq \mathsf{rcv}_1^* \wedge H_\infty(\Sigma_0^*) \ge \kappa$**).**

Lemma 4. *If there exists an adversary $\mathcal{A}_4$ that breaks the* Priv^+-CPA *(resp.,* Priv^+-CCA*) security for case 4, there exists an adversary $\mathcal{B}_1$ that breaks* $\mathsf{Priv\text{-}MisMatch}$ *security and $\mathcal{B}_2$ that breaks* Priv-CPA *(resp.,* Priv-CCA*) security such that*

$$\left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \,\middle|\, \textit{Case } 4\right] - \frac{1}{2}\right| \le \mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda) + \mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)$$

$$\left(\textit{resp., } \left|\Pr\left[\mathsf{Priv}^+\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \,\middle|\, \textit{Case } 4\right] - \frac{1}{2}\right| \le \mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda) + \mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)\right).$$

Proof. To prove the lemma, we consider the following sequence of games Game_i for $i \in \{0,1,2\}$. We define the advantage of $\mathcal{A}$ in Game_i as

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

Game_0**.** This is the original security game for $\mathsf{coin} = 0$.

Game_1**.** In this game, we replace $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_0^*}, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$ with $\mathsf{ct}^* \leftarrow\!\!\$\ \mathcal{CT}$.

To show difference between Game_0 and Game_1 are negligible, we construct $\mathcal{B}_1$ that breaks Priv-MisMatch security of IB-ME using $\mathcal{A}$. The boxed descriptions are only used to simulate the Priv^+-CCA game.

1. Upon receiving the master public key mpk, $\mathcal{B}_1$ executes $\mathcal{A}$ on input mpk.
2. $\mathcal{B}_1$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.

   (c) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$, $\mathcal{B}_1$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_1$ computes $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$ and returns it to $\mathcal{A}$.

3. When $\mathcal{A}$ sends $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_2$ picks $\sigma_1^* \leftarrow\!\!\$\ \Sigma_1^*$, and sends $(\Sigma_0^*, \mathsf{rcv}_0^*, \mathsf{m}_0^*)$ to its challenger and receives the challenge ciphertext $\mathsf{ct}^*, \mathsf{dk}_{\rho_0^*}$. Then $\mathcal{B}_1$ returns ct^* to $\mathcal{A}$. Moreover, $\mathcal{B}_1$ sends σ_1^* to its challenger and receives $\mathsf{ek}_{\sigma_1^*}$.
4. $\mathcal{B}_1$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_1$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_1$ returns $\mathsf{dk}_{\rho_0^*}$ to $\mathcal{A}$ if $\rho = \rho_0^*$. Otherwise, $\mathcal{B}_1$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_1$ returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$, $\mathcal{B}_1$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to its challenger and receives ct if $i = 0$. Otherwise, $\mathcal{B}_1$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_1^*}, \mathsf{rcv}, \mathsf{m})$. Then $\mathcal{B}_1$ returns ct to $\mathcal{A}$.

   (d) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$, $\mathcal{B}_1$ sends ρ to its $\mathcal{O}_R$ oracle and receives dk_ρ. Then, $\mathcal{B}_1$ computes $\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$ and returns it to $\mathcal{A}$.

5. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}_1$ sends it to the challenger as its guess.

It is clear that if $\mathsf{coin} = 0$ (that is $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\mathsf{ek}_{\sigma_1^*}}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$), then $\mathcal{B}_1$ perfectly simulates Game_0 for $\mathcal{A}$. On the other hand, if $\mathsf{coin} = 1$ (that is $\mathsf{ct}^* \leftarrow\!\!\$\ \mathcal{CT}$), then $\mathcal{B}_1$ perfectly simulates Game_1 for $\mathcal{A}$. Therefore, we have $\epsilon_0 = \Pr\Big[\mathsf{coin} = \widehat{\mathsf{coin}} \;\Big|\; \mathsf{coin} = 0\Big]$ and $\epsilon_1 = \Pr\Big[\mathsf{coin} = \widehat{\mathsf{coin}} \;\Big|\; \mathsf{coin} = 1\Big]$. Thus, we have

$$\begin{aligned}
\Pr\Big[\mathsf{Priv\text{-}MisMatch}_{\mathsf{IB\text{-}ME}}^{\mathcal{B}_1}(\lambda) \Rightarrow 1\Big] &= \frac{1}{2}\left(\Pr\Big[\mathsf{coin} = \widehat{\mathsf{coin}} \;\Big|\; \mathsf{coin} = 1\Big] + \Pr\Big[\mathsf{coin} = \widehat{\mathsf{coin}} \;\Big|\; \mathsf{coin} = 0\Big]\right) \\
&= \frac{1}{2}\left(1 - \Pr\Big[\widehat{\mathsf{coin}} = 0 \;\Big|\; \mathsf{coin} = 1\Big] + \Pr\Big[\widehat{\mathsf{coin}} = 0 \;\Big|\; \mathsf{coin} = 0\Big]\right) \\
&= \frac{1}{2}\left(1 - \epsilon_1 + \epsilon_0\right).
\end{aligned}$$

Thus, we have

$$|\epsilon_0 - \epsilon_1| = 2\mathsf{Adv}_{\mathcal{B}_1, \mathsf{IB\text{-}ME}}^{\mathsf{priv\text{-}mismatch}}(\lambda).$$

Game_2**.** In this game, we replace $\mathsf{ct}^* \leftarrow\!\!\$\ \mathcal{CT}$ with $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_1^*}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$. By doing this, Game_2 is the original security game for $\mathsf{coin} = 1$.

To show difference between Game_1 and Game_2 are negligible, we construct $\mathcal{B}_2$ that breaks Priv-CPA security of IB-ME using $\mathcal{A}$. The boxed descriptions are only used for the reduction of Priv-CCA.

1. Upon receiving the master public key mpk, $\mathcal{B}_2$ executes $\mathcal{A}$ on input mpk.
2. $\mathcal{B}_2$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.

   (c) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_2$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to its $\mathcal{O}_D$ oracle and receives m. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.

3. When $\mathcal{A}$ sends $(\Sigma_0^*, \Sigma_1^*, \mathsf{rcv}_0^*, \mathsf{rcv}_1^*, \mathsf{m}_0^*, \mathsf{m}_1^*)$ to request a challenge ciphertext, $\mathcal{B}_2$ picks $\sigma_0^* \leftarrow\!\!\$\ \Sigma_0^*$ and $\sigma_1^* \leftarrow\!\!\$\ \Sigma_1^*$, and sends $(\sigma_1^*, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$ to its challenger and receives the challenge ciphertext ct^*. Then $\mathcal{B}_2$ returns it to $\mathcal{A}$. Moreover, $\mathcal{B}_2$ sends σ_0^* and σ_1^* to its challenger and receives $\mathsf{ek}_{\sigma_0^*}$ and $\mathsf{ek}_{\sigma_1^*}$.
4. $\mathcal{B}_2$ answers queries from $\mathcal{A}$ as follows.
   (a) When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}_2$ sends σ to its challenger and receives ek_σ. Then $\mathcal{B}_2 2$ returns it to $\mathcal{A}$.
   (b) When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}_2$ sends ρ to its challenger and receives dk_ρ. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.
   (c) When $\mathcal{A}$ sends $(i, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_{E^*}$, $\mathcal{B}_2$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma_i^*}, \mathsf{rcv}, \mathsf{m})$ and returns ct to $\mathcal{A}$.

   (d) When $\mathcal{A}$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to $\mathcal{O}_D$ oracle, $\mathcal{B}_2$ sends $\mathsf{snd}, \rho, \mathsf{ct}$ to its $\mathcal{O}_D$ oracle and receives m. Then, $\mathcal{B}_2$ returns it to $\mathcal{A}$.

5. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}_2$ sends it to the challenger as its guess.

First, $\mathcal{A}$ can make a decryption key query on input ρ_0^* since ρ_0^* is not the challenge receiver identity of the $\mathsf{Priv\text{-}CPA}$ game. Then, it is clear that if $\mathsf{coin} = 0$ (that is $\mathsf{ct}^* = \mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\mathsf{ek}_{\sigma_1^*}}, \mathsf{rcv}_1^*, \mathsf{m}_1^*)$), then $\mathcal{B}_2$ perfectly simulates Game_2 for $\mathcal{A}$. On the other hand, if $\mathsf{coin} = 1$ (that is $\mathsf{ct}^* \leftarrow\!\!\$\ \mathcal{CT}$), then $\mathcal{B}_2$ perfectly simulates Game_1 for $\mathcal{A}$. Therefore, we have $\epsilon_1 = \Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin} = 1\right]$ and $\epsilon_2 = \Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin} = 0\right]$. Thus, we have

$$\begin{aligned}
\Pr\left[\mathsf{Priv\text{-}CPA}^{\mathcal{B}_2}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] &= \frac{1}{2}\left(\Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin} = 0\right] + \Pr\left[\mathsf{coin} = \widehat{\mathsf{coin}} \;\middle|\; \mathsf{coin} = 1\right]\right) \\
&= \frac{1}{2}\left(1 - \Pr\left[\widehat{\mathsf{coin}} = 0 \;\middle|\; \mathsf{coin} = 0\right] + \Pr\left[\widehat{\mathsf{coin}} = 0 \;\middle|\; \mathsf{coin} = 1\right]\right) \\
&= \frac{1}{2}\left(1 - \epsilon_2 + \epsilon_1\right).
\end{aligned}$$

$$\left(\text{resp., } \Pr\left[\mathsf{Priv\text{-}CPA}^{\mathcal{B}_2}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1\right] = \frac{1}{2}(1 - \epsilon_2 + \epsilon_1)\right)$$

Thus, we have

$$|\epsilon_1 - \epsilon_2| = 2\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda) \quad \left(\text{resp., } |\epsilon_1 - \epsilon_2| = 2\mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)\right).$$

From the above arguments, we have

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 4\right] - \frac{1}{2}\right| \\
&\le \frac{1}{2}\left|\Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 4 \wedge \mathsf{coin} = 0\right] - \Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CPA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; \text{Case } 4 \wedge \mathsf{coin} = 1\right]\right| \\
&= \frac{1}{2}|\epsilon_0 - \epsilon_2| = \frac{1}{2}(|\epsilon_0 - \epsilon_1| + |\epsilon_1 - \epsilon_2|) = \mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda) + \mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda).
\end{aligned}$$

$$\left(\text{resp., } \left|\Pr\left[\mathsf{Priv}^{+}\text{-}\mathsf{CCA}^{\mathcal{A}}_{\mathsf{IB\text{-}ME}}(\lambda) \Rightarrow 1 \;\middle|\; -\right] \frac{1}{2}\right| \leq \mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}}(\lambda) + \mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{B}_2,\mathsf{IB\text{-}ME}}(\lambda)\right)$$

□

Putting Lemmata 1 to 4, we have

$$\mathsf{Adv}^{\mathsf{priv}^{+}\text{-}\mathsf{cpa}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) \leq 4\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda) + 4\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{B},\mathsf{IB\text{-}ME}}(\lambda).$$

□

4 Improved IB-ME Scheme from BDH in the ROM

This section shows an improved IB-ME scheme from the BDH assumption in the ROM. Our idea is to combine the Boneh-Franklin IBE scheme [5] and the Sakai-Ohgishi-Kasahara IB-NIKE scheme [30]. We also introduce several optimizations to reduce secret key and ciphertext sizes. To achieve stronger security, we employ the FO transformation [19]. Interestingly, the FO transformation allows us to achieve not only Priv-CCA security at minimum costs but also Auth-oCMA security for free. We also provide a formal proof of its Priv-MisMatch security. As a result, we obtain a highly efficient and strongly secure IB-ME scheme compared to the scheme of Ateniese et al. [2].

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 outputs $\mathsf{ek}_\sigma \coloneqq \mathsf{u}_\sigma^x$.

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

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma, \mathsf{rcv}, \mathsf{m})$**:** 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

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

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

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

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

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

That is, it holds that

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

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 $\mathsf{Priv\text{-}CCA}$, $\mathsf{Priv\text{-}MisMatch}$ and $\mathsf{Auth\text{-}oCMA}$ secure in the ROM.

Theorem 2. *Suppose the hash function* G *is a random oracle. If there exists an adversary* $\mathcal{A}$ *that breaks the* $\mathsf{Priv\text{-}CCA}$ *security of* $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$*, there exists an adversary* $\mathcal{B}$ *that breaks the* BDH *assumption for* $\mathcal{G}$ *such that*

$$\mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda) \leq 3\hat{e}(1+q_R)q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda) + \frac{q_{\mathsf{Dec}}}{p} + \frac{3q_{\mathsf{G}}}{2^{\lambda}}.$$

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* $\mathcal{A}$ *makes to are the maximum number of queries* $\mathcal{A}$ *sends to* $\mathcal{O}_R$*,* $\mathcal{O}_D$*,* $\hat{\mathsf{H}}$ *and* G *oracles, respectively.*

To prove the $\mathsf{Priv\text{-}CCA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$, we use the intermediate scheme $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$, which is a simplified version of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$. We prove that $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ is $\mathsf{Priv\text{-}CPA}$ secure under the BDH assumption, and then prove the $\mathsf{Priv\text{-}CCA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ assuming the $\mathsf{Priv\text{-}CPA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$.

Basic IB-ME scheme. The IB-ME scheme $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ is as follows. The differences between $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ and $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ are that $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}.\mathsf{Enc}$ samples uniform randomness r instead of generating it with a hash function G, and $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}.\mathsf{Dec}$ does not perform the ciphertext validity check (i.e., do 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 $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}.\mathsf{Setup}$ except that G is not chosen.
$\mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma)$**:** It is identical to $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}.\mathsf{SKGen}$.
$\mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho)$**:** It is identical to $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}.\mathsf{RKGen}$.
$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma, \mathsf{rcv}, \mathsf{m})$**:** It chooses $r \leftarrow\!\!\$\, \mathbb{Z}_p$ and computes $\mathsf{u}_{\mathsf{rcv}} := \mathsf{H}_2(\mathsf{rcv})$ and

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

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

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

Finally, it outputs m.

We can easily verify that $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ is correct. We now show that $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$ is $\mathsf{Priv\text{-}CPA}$ secure.

Theorem 3. *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* $\mathsf{Priv\text{-}CPA}$ *security of* $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$*, there exists an adversary* $\mathcal{B}$ *that breaks the BDH assumption for* $\mathcal{G}$ *such that*

$$\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda) \leq \hat{e}(1+q_R)q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda),$$

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 of Theorem 3. To prove the theorem, we consider the following sequence of games Game_i for $i \in \{0,1,2\}$. 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

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

Game_1**.** In this game, we add abort conditions. We guess the challenge identity ρ^* 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 $\mathcal{A}$.
– When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, it searches an entry $(\rho, \mathsf{u}_\rho, b, d) \in \mathcal{L}_{\mathsf{H}_2}$[14]. If $d = 0$, the game aborts. Otherwise (i.e., $d = 1$), it computes $\mathsf{dk}_\rho := (g_2^x)^b$ and returns it to $\mathcal{A}$.
– When $\mathcal{A}$ outputs $(\sigma^*, \mathsf{rcv}^*, \mathsf{m}^*)$ to request a challenge ciphertext, it searches $(\mathsf{rcv}^*, \mathsf{u}_{\mathsf{rcv}^*}, b, d)$ 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 Game_1 is equal to the advantage of $\mathcal{A}$ in Game_0 conditioning on the game does not abort. Therefore, we have

$$\epsilon_1 = \epsilon_0 \cdot \Pr[\neg\mathsf{abort}].$$

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 the game does not abort when $\mathcal{A}$ request 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 $\Pr[\neg\mathsf{abort}] \leq \frac{1}{\hat{e}(1+q_R)}$ for large q_R. Therefore, we have

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

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

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

Let BadQ be the event that $\mathcal{A}$ queries $(\cdot, \mathsf{rcv}^*, R^*, U^*, \cdot)$ to the oracle $\hat{\mathsf{H}}$ where $U^* := e(R^*, \mathsf{dk}_{\mathsf{rcv}^*})$. 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

$$|\epsilon_2 - \epsilon_1| \leq \Pr[\mathsf{BadQ}].$$

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, $\mathcal{B}$ flip 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 $Z \leftarrow\$ \{0,1\}^{\mathsf{msgLen}}$ 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 $\mathsf{ek}_\sigma := (g_1^\alpha)^b$. Then, $\mathcal{B}$ returns ek_σ 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 $\mathsf{dk}_\rho := (g_2^\alpha)^b$. Then, $\mathcal{B}$ returns dk_ρ to $\mathcal{A}$.
   (f) When $\mathcal{A}$ outputs $(\sigma^*, \mathsf{rcv}^*, \mathsf{m}^*)$ to request a challenge ciphertext, $\mathcal{B}$ searches $(\mathsf{rcv}^*, \mathsf{u}_{\mathsf{rcv}^*}, b^*, d^*) \in \mathcal{L}_{\mathsf{H}_2}$. 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 $\mathsf{ct}_1 \leftarrow\$ \mathcal{CT}$, and returns $\mathsf{ct}_{\mathsf{coin}}$ to $\mathcal{A}$.

[14] 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.)

3. Finally, $\mathcal{A}$ outputs a guess $\widehat{\mathsf{coin}}$. Then, $\mathcal{B}$ picks an entry $(\cdot, \mathsf{rcv}^*, R^*, U^*, \cdot) \in \mathcal{L}_{\hat{\mathsf{H}}}$ 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 $\mathsf{dk}_{\mathsf{rcv}^*} = (\mathsf{u}_{\mathsf{rcv}^*})^{\alpha} = (g_2^{\alpha\beta})^{b^*}$ and $R^* = g_1^{\gamma}$, and thus

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

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

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

In Game_2, both ct_0 and ct_1 are chosen at random from the ciphertext space. Since coin is information-theoretically hidden from $\mathcal{A}$, we have $\epsilon_2 = 0$.

Putting everything together, we obtain

$$\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda) \leq \hat{e}(1 + q_R) q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda).$$

□

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 [19].

Proof of Theorem 2. To prove the theorem, we consider the following sequence of games Game_i for $i \in \{0, \cdots, 5\}$. 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

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

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

$$\mathsf{ek}_{\sigma} := \mathsf{H}_1(\sigma)^x, \qquad \mathsf{ct} \leftarrow \mathsf{IB\text{-}ME}^{\mathsf{Basic}}.\mathsf{Enc}(\mathsf{mpk}^{15}, \mathsf{ek}_{\sigma}, \rho, \mathsf{m}||\mathsf{k}; r).$$

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

[15] 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 $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$.

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_0. To generate ct_0, the challenger chooses $r^* \leftarrow\!\!\$\ \mathbb{Z}_p$ and computes

$$\mathsf{ek}_{\sigma^*} := \mathsf{H}_1(\sigma^*)^x, \mathsf{ct}_0 \leftarrow \mathsf{IB\text{-}ME}^{\mathsf{Basic}}.\mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_{\sigma^*}, \mathsf{rcv}^*, \mathsf{m}^*||\mathsf{k}^*; r^*).$$

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 requests 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 $\mathsf{Priv\text{-}CPA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{Basic}}$.

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 $\mathsf{Priv\text{-}CCA}$ 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 $(\sigma^*, \mathsf{rcv}^*, \mathsf{m}^*)$, $\mathcal{B}_1$ sends $(\sigma^*, \mathsf{rcv}^*, \mathsf{m}^*||\mathsf{k}^*)$ to its challenger, receiving the challenge ciphertext ct^*. $\mathcal{B}_1$ 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., ct^* is sampled from $\mathcal{CT}$). Since k^* is uniformly distributed and independent from $\mathcal{B}_1$'s view when ct^* is sampled from $\mathcal{CT}$, $\Pr[\mathsf{Fail}] \leq q_\mathsf{G}/2^\lambda$. 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

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

Therefore, we have

$$|\epsilon_5 - \epsilon_4| \leq \Pr[\mathsf{BadQ}] \leq 2\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B}_1,\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda) + \frac{2q_{\mathsf{G}}}{2^\lambda}.$$

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

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

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

$$\begin{aligned}\mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda) &\leq 3\mathsf{Adv}^{\mathsf{priv\text{-}cpa}}_{\mathcal{B},\mathsf{IB\text{-}ME}^{\mathsf{Basic}}}(\lambda) + \frac{q_D}{p} + \frac{3q_{\mathsf{G}}}{2^\lambda} \\ &= 3\hat{e}(1+q_R)q_{\hat{\mathsf{H}}} \cdot \mathsf{Adv}^{\mathsf{bdh}}_{\mathcal{B},\mathcal{G}}(\lambda) + \frac{q_{\mathsf{Dec}}}{p} + \frac{3q_{\mathsf{G}}}{2^\lambda}.\end{aligned}$$

□

Theorem 4. $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ *is* Priv-MisMatch *secure in the ROM. Formally, a κ-admissible adversary $\mathcal{A}$ attacking the* Priv-MisMatch *security of* $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ *has advantage*

$$\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda) \leq \frac{q_{\hat{\mathsf{H}}} + q_{\mathsf{G}}}{2^{\kappa-1}}.$$

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 for $i \in \{0,1,2\}$. 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

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

Game_1**.** In this game, the challenger aborts the game if σ_0^* or σ_1^* are sent to $\hat{\mathsf{H}}$ or G oracle before $\mathcal{A}$ requests the challenge ciphertext. Since both are chosen independently at random and from κ-distribution, we have

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

Game_2**.** In this game, the challenge ciphertext ct_0 is computed as $\mathsf{ct}_0 \leftarrow (g_1^{r_0}, (\mathsf{m}_0||\mathsf{k}_0) \oplus Z_0)$ for random $r_0 \leftarrow\!\!\$\ \mathbb{Z}_p$ and $Z_0 \leftarrow\!\!\$\ \{0,1\}^{\mathsf{msgLen}+\lambda}$. $\mathcal{A}$ may notice this change when it sends σ_0^* or σ_1^* to $\hat{\mathsf{H}}$ or G oracle. Since σ^* is chosen independently at random from κ-distribution, we have

$$|\epsilon_2 - \epsilon_1| \leq \frac{q_{\hat{\mathsf{H}}} + q_{\mathsf{G}}}{2^\kappa}.$$

In Game_2, both ct_0 and ct_1 are distributed uniformly at random. This means that coin is information-theoretically hidden from $\mathcal{A}$, so we have

$$\epsilon_2 = 0.$$

Putting everything together, we obtain

$$\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{BDH}}}(\lambda) \leq \frac{q_{\hat{\mathsf{H}}} + q_{\mathsf{G}}}{2^{\kappa-1}}.$$

□

Theorem 5. *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. 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 $\mathsf{ek}_\sigma := \mathsf{H}_1(\sigma)^x$. 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 $(\mathsf{snd}, \mathsf{rcv}, g_1^r, e(X^r, \mathsf{u}_{\mathsf{rcv}}), e(\mathsf{ek}_\sigma, \mathsf{u}_{\mathsf{rcv}}), *) \in \mathcal{L}_{\hat{\mathsf{H}}}$, it aborts the game. (In this case, it cannot program the random oracle.)
  - Else, it updates

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

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

  Finally, it returns ek_σ to $\mathcal{A}$.
- When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, it computes $\mathsf{dk}_\rho := \mathsf{H}_2(\rho)^x$. 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 $(\mathsf{snd}, \mathsf{rcv}, g_1^r, e(g_1^r, \mathsf{dk}_\rho), e(\mathsf{H}_1(\mathsf{snd}), \mathsf{dk}_\rho), *) \in \mathcal{L}_{\hat{\mathsf{H}}}$, it aborts the game.
  - Else, for each entry, it updates

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

  Finally, it returns dk_ρ 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 ek_σ[16] and computes $\mathsf{u}_{\mathsf{rcv}} := \mathsf{H}_2(\mathsf{rcv})$ and

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

  2. If $\sigma \notin \mathcal{L}_S$ and $\mathsf{rcv} \in \mathcal{L}_R$, it retrieves $\mathsf{dk}_{\mathsf{rcv}}$[17] and computes $\mathsf{u}_{\mathsf{snd}} := \mathsf{H}_1(\mathsf{snd})$ and

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

  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 $\mathcal{L}_E \leftarrow \mathcal{L}_E \cup \{(\sigma, \mathsf{rcv}, \mathsf{m}||\mathsf{k}, \mathsf{ctxt})\}$.

[16] Since $\sigma \in \mathcal{L}_S$, the challenger already has computed ek_σ.
[17] Since $\mathsf{rcv} \in \mathcal{L}_R$, the challenger already has computed $\mathsf{dk}_{\mathsf{rcv}}$.

Let Fail be the event that Game_1 aborts if $(\mathsf{snd}, \mathsf{rcv}, g_1^r, e(X^r, \mathsf{u}_{\mathsf{rcv}}), e(\mathsf{ek}_\sigma, \mathsf{u}_{\mathsf{rcv}}), *) \in \mathcal{L}_{\hat{\mathsf{H}}}$ 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 $X := g_1^\alpha$ (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 $\mathsf{mpk} := (G, \mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}, \mathsf{G}, X)$.
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 $\mathsf{u}_\sigma := g_1^b$ 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 $\mathsf{u}_\rho := (g_2^\beta)^{\hat{b}}$, 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 $D := V^{\frac{1}{b\hat{b}}}$ 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 $\mathsf{ek}_\sigma = (g_1^\alpha)^b$ 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 $\mathsf{dk}_\rho = (g_2^\alpha)^{\hat{b}}$ 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 $Z^* := \hat{\mathsf{H}}(\mathsf{snd}^*, \rho^*, R^*, e(R^*, \mathsf{dk}_{\rho^*}), e(\mathsf{H}_1(\mathsf{snd}^*), \mathsf{dk}_{\rho^*}))$.

Let BadQ be the event that $\mathcal{A}$ makes a query $(\sigma^*, \rho^*, \cdot, \cdot, V^*)$ to the oracle $\hat{\mathsf{H}}$ where $V^* := e(\mathsf{u}_{\sigma^*}, \mathsf{u}_{\rho^*})^x$. 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 $X := g_1^\alpha$ (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 $\mathsf{mpk} := (G, \mathsf{H}_1, \mathsf{H}_2, \hat{\mathsf{H}}, \mathsf{G}, X)$.
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 $\mathsf{u}_\sigma := g_1^b$ 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 $\mathsf{u}_\rho := (g_2^\beta)^{\hat{b}}$ 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 $\mathsf{ek}_\sigma = (g_1^\alpha)^b$ and returns it 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 $\mathsf{dk}_\rho = (g_2^\alpha)^{\hat{b}}$ and return it to $\mathcal{A}$.
3. $\mathcal{A}$ outputs $(\mathsf{snd}^*, \rho^*, \mathsf{ct}^* := (R^*, \mathsf{ctxt}^*))$. $\mathcal{B}_2$ sets $\sigma^* := \mathsf{snd}^*$. 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 $D := V^{\frac{1}{b^* \hat{b}^*}}$ 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 $\mathsf{G}(\mathsf{m}^*, \mathsf{k}^*, \mathsf{snd}^*, \rho^*)$ 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

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

□

5 Improved IB-ME Scheme from IBE and IBS in the QROM

This section shows a new generic construction of IB-ME based on IBE, IBS, which we call $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$. To achieve $\mathsf{Priv\text{-}MisMatch}$ security, we hide messages with random oracle.

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 $\mathcal{M}_{\mathsf{IBE}} = \{0,1\}^{\mathsf{msgLen}+\mathsf{sigLen}+\mathsf{seedLen}}$.
- An IBS scheme $\mathsf{IBS} = (\mathsf{IBS.Setup}, \mathsf{IBS.KGen}, \mathsf{IBS.Sign}, \mathsf{IBS.Ver})$ with $\mathcal{ID}_{\mathsf{IBS}} = \{0,1\}^*$ and sigLen bits signatures.
- A hash function $\mathsf{H} : \mathcal{ID} \to \mathcal{CT}$.

The proposed IB-ME scheme $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ is as follows.

$\mathsf{Setup}(1^\lambda)$**:** It computes $(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$ and $(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}) \leftarrow \mathsf{IBS.Setup}(1^\lambda)$, and outputs $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{IBS}})$ and $\mathsf{msk} := (\mathsf{msk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBS}})$.

$\mathsf{SKGen}(\mathsf{mpk}, \mathsf{msk}, \sigma)$**:** It outputs $\mathsf{ek}_\sigma \leftarrow \mathsf{IBS.KGen}(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}, \sigma)$.

$\mathsf{RKGen}(\mathsf{mpk}, \mathsf{msk}, \rho)$**:** It outputs $\mathsf{dk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{msk}_{\mathsf{IBE}}, \rho)$.

$\mathsf{Enc}(\mathsf{mpk}, \mathsf{ek}_\sigma, \mathsf{rcv}, \mathsf{m})$**:** It computes $\mathsf{sig} \leftarrow \mathsf{IBS.Sign}(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{ek}_\sigma, \mathsf{m}||\mathsf{rcv})$, $\mathsf{ct} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{rcv}, \mathsf{m}||\mathsf{sig})$ and $\hat{\mathsf{ct}} := \mathsf{ct} \oplus \mathsf{H}(\sigma)$. It outputs $\hat{\mathsf{ct}}$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{dk}_\rho, \mathsf{snd}, \mathsf{ct})$**:** It computes $\mathsf{ct}' := \hat{\mathsf{ct}} \oplus \mathsf{H}(\mathsf{snd})$. Then, it computes $\mathsf{m}'||\mathsf{sig}' \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{dk}_\rho, \mathsf{ct}')$. If the output is equal to $\perp$, it outputs $\perp$. Else, it computes $b \leftarrow \mathsf{IBS.Ver}(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{snd}, \mathsf{m}'||\rho, \mathsf{sig}')$. If $b = 1$, it outputs m'; otherwise, it outputs $\perp$.

Correctness. We can verify that $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ is correct with negligible correctness errors. The condition $\sigma = \mathsf{snd}$ ensures $\mathsf{ct}' = \hat{\mathsf{ct}} \oplus \mathsf{H}(\mathsf{snd}) = \hat{\mathsf{ct}} \oplus \mathsf{H}(\sigma) = \mathsf{ct}$. Furthermore, the condition $\mathsf{rcv} = \rho$ and the correctness of the IBE scheme, for any messages, we have $\mathsf{m}'||\mathsf{sig}' = \mathsf{m}||\mathsf{sig}$ with all but negligible probability. Finally, the correctness of the IBS scheme ensures $\mathsf{IBS.Ver}(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{snd}, \mathsf{m}'||\rho, \mathsf{sig}') = 1$. Therefore, the decryption algorithm finally outputs the encrypted message m with a probability of all but negligible.

5.2 Security Proof

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

Theorem 6. *If there exists an adversary $\mathcal{A}$ that breaks the* $\mathsf{Priv\text{-}CCA}$ *security of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$*, there exists an adversary $\mathcal{B}$ that breaks the* $\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}$ *security of* IBE *such that*

$$\mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) = \mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cca}}_{\mathcal{B},\mathsf{IBE}}(\lambda).$$

Proof. Let $\mathcal{A}$ be an adversary that breaks the $\mathsf{Priv\text{-}CCA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$. We show an adversary $\mathcal{B}$ that breaks the $\mathsf{ANO\text{-}IND\text{-}ID\text{-}CCA}$ security of IBE by using $\mathcal{A}$. The description of $\mathcal{B}$ is as follows.

1. Upon receiving the master public key $\mathsf{mpk}_{\mathsf{IBE}}$, $\mathcal{B}$ generates $(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}) \leftarrow \mathsf{IBS.Setup}(1^\lambda)$ and executes $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk}_{\mathsf{IBE}}, \mathsf{mpk}_{\mathsf{IBS}})$.
2. $\mathcal{B}$ answers queries from $\mathcal{A}$ as follows.
   - When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}$ computes $\mathsf{ek}_\sigma \leftarrow \mathsf{IBS.KGen}(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{msk}_{\mathsf{IBS}}, \sigma)$ and returns it to $\mathcal{A}$.
   - When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}$ sends ρ to $\mathcal{O}_{SK}$ oracle and receives dk_ρ. Then $\mathcal{B}$ returns it to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(\mathsf{snd}, \rho, \hat{\mathsf{ct}})$ to $\mathcal{O}_D$ oracle, $\mathcal{B}$ simulates the decryption process by using its decryption oracle. Note that when $\hat{\mathsf{ct}} = \hat{\mathsf{ct}}^*$, it must be $(\mathsf{snd}, \rho) \neq (\sigma^*, \mathsf{rcv}^*)$. When $\mathsf{snd} = \sigma^*$, the challenge IBE ciphertext ct^* is decrypted with the receiver $\rho \neq \mathsf{rcv}^*$; when $\rho = \mathsf{rcv}^*$, $\mathsf{snd} \neq \sigma^*$ and it implies $\hat{\mathsf{ct}}^* \oplus \mathsf{H}(\mathsf{snd}) \neq \mathsf{ct}^*$. Therefore, in any case, the decryption process can be simulated.

3. When $\mathcal{A}$ sends $(\sigma^*, \mathsf{rcv}^*, \mathsf{m}^*)$ to request a challenge ciphertext, $\mathcal{B}$ first computes $\mathsf{sig}^* \leftarrow \mathsf{IBS.Sign}(\mathsf{mpk}_{\mathsf{IBS}}, \mathsf{ek}_{\sigma^*}, \mathsf{m}^*)$ and sends $(\mathsf{rcv}^*, \mathsf{m}^*||\mathsf{sig}^*)$ to its challenger and receives the challenge ciphertext ct^*. Then, it computes $\hat{\mathsf{ct}} := \mathsf{ct}^* \oplus \mathsf{H}(\sigma^*)$ which is sent to $\mathcal{A}$.
4. Finally, when $\mathcal{A}$ outputs $\widehat{\mathsf{coin}}$, $\mathcal{B}$ sends it to the challenger as its guess.

We can verify that $\mathcal{B}$ perfectly simulates the Priv-CCA game against $\mathcal{A}$. Moreover, $\mathsf{rcv}^* \notin \mathcal{L}_R$ implies $\mathsf{rcv}^* \notin \mathcal{L}_{SK}$. Therefore, if $\mathcal{A}$ breaks the Priv-CCA security, $\mathcal{B}$ also breaks the ANO-IND-ID-CCA security, that is,

$$\mathsf{Adv}^{\mathsf{priv\text{-}cca}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) = \mathsf{Adv}^{\mathsf{ano\text{-}ind\text{-}id\text{-}cca}}_{\mathcal{B},\mathsf{IBE}}(\lambda).$$

□

Theorem 7. $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ *is* Priv-MisMatch *secure in the QROM. Formally, a κ-admissible adversary $\mathcal{A}$ attacking the* Priv-MisMatch *security of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ *has advantage*

$$\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}}(\lambda) \leq \frac{q_{\mathsf{H}}^2}{2^{\kappa-1}}.$$

where q_{H} is the maximum number of queries $\mathcal{A}$ makes to the H *oracle.*

Proof. To prove the theorem, we consider the following sequence of games Game_i for $i \in \{0, 1, 2\}$. 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

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

Game_1. In this game, the challenger aborts the game if σ_0^* or σ_1^* are sent to H oracle before $\mathcal{A}$ requests the challenge ciphertext. Since both are chosen independently at random and from κ-distribution, we have

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

Game_2. In this game, the challenge ciphertext $\hat{\mathsf{ct}}^*$ is replaced with a random ciphertext. $\mathcal{A}$ may notice this change when it sends σ_0^* or σ_1^* to H oracle. Since σ^* is chosen independently at random from κ-distribution, we have

$$|\epsilon_2 - \epsilon_1| \leq \frac{q_{\mathsf{H}}^2}{2^{\kappa}}.$$

In Game_2, the challenge ciphertext is distributed uniformly at random regardless of the value of the challenge bit. This means that coin is information-theoretically hidden from $\mathcal{A}$, so we have

$$\epsilon_2 = 0.$$

Putting everything together, we obtain

$$\mathsf{Adv}^{\mathsf{priv\text{-}mismatch}}_{\mathcal{A},\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}}(\lambda) \leq \frac{q_{\mathsf{H}}^2}{2^{\kappa-1}}.$$

□

Theorem 8. *If there exists an adversary $\mathcal{A}$ that breaks the* Auth-iCMA *security of* $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$*, there exists an adversary $\mathcal{B}$ that breaks the* EUF-ID-CMA *security of* IBS *such that*

$$\mathsf{Adv}^{\mathsf{auth\text{-}icma}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) = \mathsf{Adv}^{\mathsf{euf\text{-}id\text{-}cma}}_{\mathcal{B},\mathsf{IBS}}(\lambda).$$

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	Mismatch	Enc. key	Dec. key	Ciphertext
Ateniese et al. [2]	CPA	oNMA		$\|\mathbb{G}_1\|$	$3\|\mathbb{G}_2\|$	$2\|\mathbb{G}_1\| + \lambda$
Boyen and Li [6] (IBE [5]+IBS [10])	CPA	iCMA	$\surd$	$\|\mathbb{G}_1\|$	$\|\mathbb{G}_2\|$	$3\|\mathbb{G}_1\| + 3\lambda$
$\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$(§ 4)	CCA	oCMA	$\surd$	$\|\mathbb{G}_1\|$	$\|\mathbb{G}_2\|$	$\|\mathbb{G}_1\| + \lambda$
$\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$(§ 5) (IBE [5]+IBS [10])	CCA	iCMA	$\surd$	$\|\mathbb{G}_1\|$	$\|\mathbb{G}_2\|$	$3\|\mathbb{G}_1\| + \lambda$

Proof. Let $\mathcal{A}$ be an adversary that breaks the $\mathsf{Auth\text{-}iCMA}$ security of $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$. We show an adversary $\mathcal{B}$ that breaks the $\mathsf{EUF\text{-}ID\text{-}CMA}$ security of IBS by using $\mathcal{A}$. The description of $\mathcal{B}$ is as follows.

1. Upon receiving the master public key $\mathsf{mpk_{IBS}}$, $\mathcal{B}$ generates $(\mathsf{mpk_{IBE}}, \mathsf{msk_{IBE}}) \leftarrow \mathsf{IBE.Setup}(\lambda)$ and executes $\mathcal{A}$ on input $\mathsf{mpk} := (\mathsf{mpk_{IBE}}, \mathsf{mpk_{IBS}})$.
2. $\mathcal{B}$ answers queries from $\mathcal{A}$ as follows.
   - When $\mathcal{A}$ sends σ to $\mathcal{O}_S$ oracle, $\mathcal{B}$ sends σ to its key generation oracle $\mathcal{O}_{SK}$ oracle and receives ek_σ. Then $\mathcal{B}$ returns it to $\mathcal{A}$.
   - When $\mathcal{A}$ sends ρ to $\mathcal{O}_R$ oracle, $\mathcal{B}$ computes $\mathsf{dk}_\rho \leftarrow \mathsf{IBE.KGen}(\mathsf{mpk_{IBE}}, \mathsf{msk_{IBE}}, \rho)$ and returns it to $\mathcal{A}$.
   - When $\mathcal{A}$ sends $(\sigma, \mathsf{rcv}, \mathsf{m})$ to $\mathcal{O}_E$ oracle, $\mathcal{B}$ first sends $(\sigma, \mathsf{m}||\mathsf{rcv})$ to its signing oracle and receives sig. Then, it computes $\mathsf{ct} \leftarrow \mathsf{IBE.Enc}(\mathsf{mpk_{IBE}}, \mathsf{rcv}, \mathsf{m}||\mathsf{sig})$ and $\hat{\mathsf{ct}} := \mathsf{ct} \oplus \mathsf{H}(\sigma)$. It returns $\hat{\mathsf{ct}}$ to $\mathcal{A}$.
3. When $\mathcal{A}$ outputs $(\mathsf{snd}^*, \rho^*, \hat{\mathsf{ct}}^*)$ as a forgery, $\mathcal{B}$ computes $\mathsf{ct}^* := \hat{\mathsf{ct}}^* \oplus \mathsf{H}(\mathsf{snd}^*)$, $\mathsf{m}^*||\mathsf{sig}^* \leftarrow \mathsf{IBE.Dec}(\mathsf{mpk_{IBE}}, \mathsf{dk}_\rho, \mathsf{ct}^*)$. If the output is not $\perp$, it computes $b^* \leftarrow \mathsf{IBS.Ver}(\mathsf{mpk_{IBS}}, \mathsf{snd}^*, \mathsf{m}^*||\rho^*, \mathsf{sig}^*)$. If $b^* = 1$, it outputs $(\mathsf{m}^*||\rho^*, \mathsf{sig}^*)$ as its forgery.

We can verify that $\mathcal{B}$ perfectly simulates the $\mathsf{Auth\text{-}iCMA}$ game. If $\mathcal{A}$ creates a valid forgery, we have $\mathsf{snd}^* \notin \mathcal{L}_S$, $(\mathsf{snd}^*, \rho^*, \mathsf{m}^*) \notin \mathcal{L}_E$, and $\mathsf{IBS.Ver}(\mathsf{mpk_{IBS}}, \mathsf{snd}^*, \mathsf{m}^*||\rho, \mathsf{sig}^*) = 1$. $\mathsf{snd}^* \notin \mathcal{L}_S$ implies $\mathsf{snd}^* \notin \mathcal{L}_{SK}$, and $(\mathsf{snd}^*, \rho^*, \mathsf{m}^*) \notin \mathcal{L}_E$ implies $(\mathsf{snd}^*, \mathsf{m}^*||\rho^*) \notin \mathcal{L}_{SIG}$. Therefore, if $\mathcal{A}$ breaks the $\mathsf{Auth\text{-}iCMA}$ security, $\mathcal{B}$ also breaks the $\mathsf{EUF\text{-}ID\text{-}CMA}$ security. Thus, we have

$$\mathsf{Adv}^{\mathsf{auth\text{-}icma}}_{\mathcal{A},\mathsf{IB\text{-}ME}}(\lambda) = \mathsf{Adv}^{\mathsf{euf\text{-}id\text{-}cma}}_{\mathcal{B},\mathsf{IBS}}(\lambda).$$

□

6 Comparison

We compare our IB-ME schemes, $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ and $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$, with the existing schemes by Ateniese et al. [2], Wang et al. [32] and Boyen and Li [6], which are based on standard assumptions. Their security and secret key and ciphertext sizes are summarized in Tables 2 and 3.

IB-ME from the BDH assumption in the ROM. We compare $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ and $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ with the Ateniese et al. scheme and the Boyen and Li scheme. We instantiate $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ and the Boyen and Li scheme with the Boneh-Franklin IBE scheme [5], the Cha-Cheon IBS scheme [10] and a RO-based reusable extractor. Table 2 summarizes their properties. Among them, $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ is the best in terms of key and ciphertext sizes, as they are only one group element. In addition, it achieves stronger $\mathsf{Priv\text{-}CCA}$ and $\mathsf{Auth\text{-}oCMA}$ security. We can see that $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ is a pure improvement of the Ateniese et al. scheme. $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ has about twice the ciphertext of $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$, but achieves $\mathsf{Auth\text{-}iCMA}$ security (that is, secure even if the receiver's key is compromised), which is stronger than $\mathsf{Auth\text{-}oCMA}$ security. Thus, $\mathsf{IB\text{-}ME}^{\mathsf{BDH}}$ and $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ offer a trade-off between efficiency and security level. Compared with the Boyen and Li scheme, $\mathsf{IB\text{-}ME}^{\mathsf{IBE+IBS}}$ is better because it achieves $\mathsf{Priv\text{-}CCA}$ security and offers more compact ciphertexts.

Table 3: Comparison of IB-ME schemes from lattices in the ROM. The data sizes are provided in bytes. The column "Ciphertext" indicates the difference between the length of ciphertext and that of plaintext. All achieve 80-bit security.

Schemes	Security			Space complexity		
	Priv	Auth	Mismatch	Enc. key	Dec. key	Ciphertext
Wang et al. [32] (LATTE-3 [35]+Falcon-IBS†)	CPA	iCMA		1595	41472	16085
Boyen and Li [6] (DLP-0 [14, 15]+Falcon-IBS†)	CPA	iCMA	$\surd$	1595	1152	4117
IB-ME$^{\mathsf{IBE+IBS}}$(§ 5) (DLP-0 [14, 15]+Falcon-IBS†)	CCA	iCMA	$\surd$	1595	1152	4021

†: IBE scheme derived from Falcon-512 [29] via the signature-to-IBS conversion [24]. We assume that the secret key of Falcon is a seed of 32 bytes.

IB-ME from lattices in the QROM. We compare post-quantum lattice-based IB-ME schemes in the QROM derived from our IB-ME$^{\mathsf{IBE+IBS}}$, the Wang et al. scheme, and the Boyen and Li scheme. Our scheme and Boyen and Li scheme are instantiated with a lattice-based anonymous IBE scheme by Ducas, Lyubashevsky and Prest (DLP) [14, 15] while the Wang et al. scheme is based on a lattice-based anonymous HIBE scheme LATTE [35][18]. All use a lattice-based IBS scheme derived from Falcon [29] through signature-to-IBS conversion [24]. Table 3 summarizes their security and space complexity. Our scheme offers small secret keys and ciphertexts of less than 5 kilobytes. Compared to the Wang et al. scheme, our decryption key and ciphertext are only 2.8% and 25.2% of theirs, respectively. This is due to the fact that our scheme is simply based on IBE, not HIBE. Compared to the Boyen and Li scheme, our scheme offers similar space complexity, but the ciphertext is 64 bytes (=2λ bits) shorter than their scheme. Therefore, our construction is considered to be more sophisticated than that of Boyen and Li. It should be noted that our scheme achieves Priv-CCA security differently from existing schemes.

Acknowledgements The authors thank the anonymous reviewers for their constructive comments and suggestions. This research was in part conducted under a contract of "Research and development on new generation cryptography for secure wireless communication services" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)," which was supported by the Ministry of Internal Affairs and Communications, 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 JP22H03590, Japan.

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Gilbert [21], pages 553–572.
2. 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.
3. 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.
4. 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.

[18] LATTE is based on the anonymous HIBE scheme by Cash et al. [9].

5. 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.
6. 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.
7. 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.
8. 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.
9. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Gilbert [21], pages 523–552.
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. Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter identity-based encryption via asymmetric pairings. *DCC*, 73(3):911–947, 2014.
13. Sohto Chiku, Keisuke Hara, and Junji Shikata. Hierarchical identity-based matchmaking encryption. In *IEICE Technical Report*, volume 123, pages 60–67. The Institute of Electronics, Information and Communication Engineers, July 2023.
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. European Telecommunications Standards Institute. Quantum-safe identity-based encryption. Technical report, The European Telecommunications Standards Institute, 2019.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. Henri Gilbert, editor. *EUROCRYPT 2010*, volume 6110 of *LNCS*. Springer, Berlin, Heidelberg, May / June 2010.
22. Dennis Hofheinz, Dingding Jia, and Jiaxin Pan. Identity-based encryption tightly secure under chosen-ciphertext attacks. In Peyrin and Galbraith [28], pages 190–220.
23. Shuichi Katsumata, Shota Yamada, and Takashi Yamakawa. Tighter security proofs for GPV-IBE in the quantum random oracle model. In Peyrin and Galbraith [28], pages 253–282.
24. Eike Kiltz and Gregory Neven. Identity-based signatures. In Marc Joye and Gregory Neven, editors, *Identity-Based Cryptography*, volume 2 of *Cryptology and Information Security Series*, pages 31–44. IOS Press, 2009.
25. 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.
26. John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002.
27. 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.

28. Thomas Peyrin and Steven Galbraith, editors. *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*. Springer, Cham, December 2018.
29. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.
30. Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *The 2000 Symposium on Cryptography and Information Security*, January 2000.
31. 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.
32. 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.
33. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Berlin, Heidelberg, August 2009.
34. 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.
35. Raymond K. Zhao, Sarah McCarthy, Ron Steinfeld, Amin Sakzad, and Máire O'Neill. Quantum-safe HIBE: does it cost a latte? Cryptology ePrint Archive, Report 2021/222, 2021.
36. 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.