Identity-Based Matchmaking Encryption without Random Oracles*

Danilo Francati[1], Alessio Guidi[2], Luigi Russo[3], and Daniele Venturi[2]

[1] Aarhus University, Aarhus, Denmark
dfrancati@cs.au.dk
[2] Sapienza University of Rome, Rome, Italy
venturi@di.uniroma1.it
[3] EURECOM, Sophia Antipolis, France
russol@eurecom.fr

Abstract. Identity-based matchmaking encryption (IB-ME) is a generalization of identity-based encryption where the sender and the receiver can both specify a target identity: If both the chosen target identities match the one of the other party, the plaintext is revealed, and otherwise the sender's identity, the target identity, and the plaintext remain hidden. Previous work showed how to construct IB-ME in the random oracle model. We give the first construction in the plain model, based on standard assumptions over bilinear groups.

Keywords: Identity-based encryption · matchmaking encryption · plain model.

* This paper appears in the Proceedings of the 22nd International Conference on Cryptology in India (INDOCRYPT 2021)

1 Introduction

Identity-based encryption (IBE) [6] extends the standard concept of public-key encryption to a setting where the receiver's public key is an arbitrary string representing its identity. This allows a sender to encrypt a message while specifying the identity $\mathsf{rcv} \in \{0,1\}^*$ of the intended receiver. A receiver with identity $\rho \in \{0,1\}^*$ obtains a decryption key dk_ρ from an authority, which allows to correctly decrypt the ciphertext so long as $\rho = \mathsf{rcv}$.

Identity-based matchmaking encryption (IB-ME) [2] is a generalization of IBE in which the sender's identity $\sigma \in \{0,1\}^*$ can also be embedded in the ciphertext. The receiver can now additionally specify a target sender's identity $\mathsf{snd} \in \{0,1\}^*$ on the fly, and obtain the message so long as there is a match in both directions (*i.e.*, $\rho = \mathsf{rcv}$ and $\sigma = \mathsf{snd}$). An IB-ME should satisfy two main security properties:

- *Privacy:* In case of mismatch (*i.e.*, either $\rho \neq \mathsf{rcv}$ or $\sigma \neq \mathsf{snd}$) both the sender's identity and the plaintext remain hidden.
- *Authenticity:* The sender obtains from the authority an encryption key ek_σ associated to its identity, with the guarantee that it should be hard to forge a valid ciphertext embedding σ without knowing such a key.

IB-ME finds applications in settings where IBE with strong anonymity guarantees is required. For instance, Ateniese *et al.* [2] show how to use IB-ME in order to construct a privacy-preserving bulletin board that can be used by newspapers and organizations to collect information from anonymous sources.

1.1 Our Contribution

The work of Ateniese *et al.* [2] shows how to construct IB-ME under the Bilinear Diffie-Hellman assumption. This leaves the following open problem:

Can we construct IB-ME in the plain model?

We answer the above question to the positive by providing the first construction of IB-ME without random oracles (see Section 4). On a high level, our result is obtained in two steps:

- First, we give a construction of an IB-ME satisfying privacy based on the Decisional Augumented Bilinear Diffie-Hellman Exponent assumption over bilinear groups. Our scheme builds upon the anonymous IBE of Gentry [4]. Very roughly, we add the functionality that the receiver can decrypt a ciphertext only if it knows (or guesses) the sender's identity. This is achieved by adding a second layer of encryption using a one-time pad derived from the sender's identity via a randomness extractor. While it seems that this idea can be applied generically to any anonymous IBE, our security analysis crucially relies on specific properties of Gentry's scheme (e.g., homomorphism).

- Second, we exhibit a generic transform taking as input any private IB-ME and outputting an IB-ME satisfying both privacy and authenticity. The main idea is to let ek_σ consist of a signature over the sender's identity σ (computed using the authority's master secret key). Hence, the sender encrypts the message using the underlying IB-ME but additionally proves in zero knowledge that it knows a valid signature of the string representing its identity. Privacy follows by the privacy property of the underlying IB-ME along with the zero knowledge property; authenticity follows by knowledge soundness.

An additional contribution of our work is to significantly strengthen the definition of privacy for IB-ME. In particular, the previous definition only guarantees privacy when the receiver's identity ρ does not match the target identity rcv specified by the sender. We give a stronger definition that allows to characterize privacy in a meaningful way also in case the target identity snd chosen by the receiver does not match the identity σ of the sender. We refer the reader to Section 3 for more details.

1.2 Related Work

Ateniese *et al.* [2] define the more general concept of ME, in which both the sender and the receiver (each with its own attributes) can specify policies the other party must satisfy in order for the message to be revealed. Differently than IB-ME, the policy chosen by the receiver cannot be chosen on the fly, but is associated to a secret key that is generated by the authority.

As pointed out in [2], the general concept of ME implies both (anonymous) ciphertext-policy and key-policy attribute-based encryption [7,5]. The implication holds in the identity-based setting too: IB-ME can be seen as a more expressive version of (anonymous) IBE [1], in which both the sender and the receiver can specify a target communicating entity (in a privacy-preserving way).

2 Preliminaries

2.1 Notation

We use the notation $[n] \overset{\text{def}}{=} \{1, \ldots, n\}$. Capital boldface letters (such as $\mathbf{X}$) are used to denote random variables, small letters (such as x) to denote concrete values, calligraphic letters (such as $\mathcal{X}$) to denote sets, and serif letters (such as A) to denote algorithms. All of our algorithms are modeled as (possibly interactive) Turing machines; if algorithm A has oracle access to some oracle O, we write $\mathcal{Q}_\mathsf{O}$ and $\mathcal{O}_\mathsf{O}$ for the set of queries asked by A to O and for the set of outputs returned by O, respectively.

For a string $x \in \{0,1\}^*$, we let $|x|$ be its length; if $\mathcal{X}$ is a set, $|\mathcal{X}|$ represents the cardinality of $\mathcal{X}$. When x is chosen randomly in $\mathcal{X}$, we write $x \leftarrow\$\, \mathcal{X}$. If A is an algorithm, we write $y \leftarrow\$\, \mathsf{A}(x)$ to denote a run of A on input x and output y; if A is randomized, y is a random variable and $\mathsf{A}(x; r)$ denotes a run of A on input x and (uniform) randomness r. An algorithm A is *probabilistic polynomial-time*

(PPT) if A is randomized and for any input $x, r \in \{0,1\}^*$ the computation of $\mathsf{A}(x; r)$ terminates in a polynomial number of steps (in the input size).

Negligible functions. We denote by $\lambda \in \mathbb{N}$ the security parameter and we implicitly assume that every algorithm takes as input the security parameter (written in unary). A function $\nu : \mathbb{N} \to [0,1]$ is called *negligible* in the security parameter λ if it vanishes faster than the inverse of any polynomial in λ, i.e. $\nu(\lambda) \in O(1/p(\lambda))$ for all positive polynomials $p(\lambda)$. We sometimes write $\mathsf{negl}(\lambda)$ (resp., $\mathsf{poly}(\lambda)$) to denote an unspecified negligible function (resp., polynomial function) in the security parameter.

Unpredictability and indistinguishability. The min-entropy of a random variable $\mathbf{X} \in \mathcal{X}$ is $\mathbb{H}_\infty(\mathbf{X}) \overset{\text{def}}{=} -\log \max_{x \in \mathcal{X}} \mathbb{P}[X = x]$, and it measures the best chance to predict $\mathbf{X}$ (by a computationally unbounded algorithm). We say that $\mathbf{X}$ and $\mathbf{Y}$ are *computationally* indistinguishable, denoted $\mathbf{X} \approx_c \mathbf{Y}$, if for all PPT distinguishers D we have $\varDelta_{\mathsf{D}}(\mathbf{X}; \mathbf{Y}) \in \mathsf{negl}(\lambda)$, where

$$\varDelta_{\mathsf{D}}(\mathbf{X}; \mathbf{Y}) \overset{\text{def}}{=} \left| \mathbb{P}\big[\mathsf{D}(1^\lambda, \mathbf{X}) = 1\big] - \mathbb{P}\big[\mathsf{D}(1^\lambda, \mathbf{Y}) = 1\big] \right|.$$

2.2 Signature Schemes

A signature scheme with message space $\mathcal{M}$ is made of the following polynomial-time algorithms.

$\mathsf{KGen}(1^\lambda)$: Upon input the security parameter 1^λ, the randomized key generation algorithm outputs a secret and a public key $(\mathsf{sk}, \mathsf{pk})$.

$\mathsf{Sign}(\mathsf{sk}, m)$: Upon input the secret key sk and the message $m \in \mathcal{M}$, the deterministic signing algorithm produces a signature s.

$\mathsf{Ver}(\mathsf{pk}, m, s)$: Upon input the public key pk, the message $m \in \mathcal{M}$, and the signature s, the deterministic verification algorithm returns a decision bit.

A signature scheme should satisfy two properties. The first property says that honestly generated signatures always verify correctly. The second property, called unforgeability, says that it should be hard to forge a signature on a fresh message, even after seeing signatures on polynomially many messages.

Definition 1 (Correctness of signatures). *A signature scheme $\varPi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$ with message space $\mathcal{M}$ is correct if $\forall \lambda \in \mathbb{N}$, $\forall (\mathsf{sk}, \mathsf{pk})$ output by $\mathsf{KGen}(1^\lambda)$, and $\forall m \in \mathcal{M}$, the following holds: $\mathbb{P}[\mathsf{Ver}(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = 1] = 1$.*

Definition 2 (Unforgeability of signatures). *A signature scheme $\varPi = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$ is existentially unforgeable under chosen-message attacks (EUF-CMA) if for all PPT adversaries A:*

$$\mathbb{P}\big[\mathbf{G}^{\mathsf{euf}}_{\varPi, \mathsf{A}}(\lambda) = 1\big] \le \mathsf{negl}(\lambda),$$

where $\mathbf{G}^{\mathsf{euf}}_{\varPi, \mathsf{A}}(\lambda)$ is the following experiment:

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda)$.
- $(m, s) \leftarrow\!\!\$\ \mathsf{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(1^\lambda, \mathsf{pk})$
- *If $m \notin \mathcal{Q}_{\mathsf{Sign}}$, and $\mathsf{Ver}(\mathsf{pk}, m, s) = 1$, output 1, else output 0.*

2.3 Non-Interactive Zero Knowledge

Let R be a relation, corresponding to an NP language L. A non-interactive zero-knowledge (NIZK) proof system for R is a tuple of polynomial-time algorithms $\Pi = (\mathsf{I}, \mathsf{P}, \mathsf{V})$ specified as follows. (i) The randomized algorithm I takes as input the security parameter and outputs a common reference string ω; (ii) The randomized algorithm $\mathsf{P}(\omega, (y, x))$, given $(y, x) \in R$ outputs a proof π; (iii) The deterministic algorithm $\mathsf{V}(\omega, (y, \pi))$, given an instance y and a proof π outputs either 0 (for "reject") or 1 (for "accept"). We say that a NIZK for relation R is *correct* if for all $\lambda \in \mathbb{N}$, every ω output by $\mathsf{I}(1^\lambda)$, and any $(y, x) \in R$, we have that $\mathsf{V}(\omega, (y, \mathsf{P}(\omega, (y, x)))) = 1$.

We define two properties of a NIZK proof system. The first property, called adaptive multi-theorem zero knowledge, says that honest proofs do not reveal anything beyond the fact that $y \in L$. The second property, called knowledge soundness, requires that every adversary creating a valid proof for some statement, must know the corresponding witness.

Definition 3 (Adaptive multi-theorem zero-knowledge). *A NIZK Π for a relation R satisfies adaptive multi-theorem zero-knowledge if there exists a PPT simulator $\mathsf{Z} := (\mathsf{Z}_0, \mathsf{Z}_1)$ such that the following holds:*

- *Algorithm Z_0 outputs ω and a simulation trapdoor ζ.*
- *For all PPT distinguishers D, we have that*

$$\Big| \mathbb{P}\Big[\mathsf{D}^{\mathsf{P}(\omega,(\cdot,\cdot))}(\omega) = 1 : \ \omega \leftarrow\!\!\$\ \mathsf{I}(1^\lambda)\Big] \\ - \mathbb{P}\Big[\mathsf{D}^{\mathsf{O}(\zeta,(\cdot,\cdot))}(\omega) = 1 : \ (\omega, \zeta) \leftarrow\!\!\$\ \mathsf{Z}_0(1^\lambda)\Big]\Big| \leq \mathsf{negl}(\lambda),$$

where the oracle $\mathsf{O}(\zeta, (\cdot, \cdot))$ takes as input a pair (y, x) and returns $\mathsf{Z}_1(\zeta, y)$ if $(y, x) \in R$ (and $\perp$ otherwise).

Definition 4 (Knowledge soundness). *A NIZK Π for a relation R satisfies knowledge soundness if there exists a PPT extractor $\mathsf{K} = (\mathsf{K}_0, \mathsf{K}_1)$ such that the following holds:*

- *Algorithm K_0 outputs ω and an extraction trapdoor ξ, such that the distribution of ω is computationally indistinguishable to that of $\mathsf{I}(1^\lambda)$.*
- *For all PPT adversaries A, we have that*

$$\mathbb{P}\left[\begin{array}{c} \mathsf{V}(\omega, (y, \pi)) = 1 \wedge \\ (y, x) \notin R \end{array} : \begin{array}{c} (\omega, \xi) \leftarrow\!\!\$\ \mathsf{K}_0(1^\lambda) \\ (y, \pi) \leftarrow\!\!\$\ \mathsf{A}(\omega) \\ x \leftarrow\!\!\$\ \mathsf{K}_1(\xi, y, \pi) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

2.4 Reusable Computational Extractors

A computational extractor is a polynomial time algorithm $\mathsf{Ext} : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ that on input a seed $s \in \mathcal{S}$ and a value $x \in \mathcal{X}$ outputs $\mathsf{Ext}_s(x) = y \in \mathcal{Y}$. The security of computational extractors guarantees that $y \in \mathcal{Y}$ is pseudorandom when the seed is sampled at random from $\mathcal{S}$ and x is sampled from an input distribution $\mathbf{X}$ (defined over the input space $\mathcal{X}$) of min-entropy $\mathbb{H}_\infty(\mathbf{X}) \geq k$, even if the seed is made public. In this work, we will rely on so-called *reusable* [3], computational extractors, that produce random looking outputs even if evaluated multiple times on the same input. The formal definition is provided below.

Definition 5 (Reusable computational extractors). *An algorithm* $\mathsf{Ext} : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ *is a* (k, q)*-reusable-extractor if for all random variables* $\mathbf{X} \in \mathcal{X}$ *such that* $\mathbb{H}_\infty(\mathbf{X}) \geq k$*, and for all PPT distinguishers* D*, it holds that*

$$\Delta_{\mathsf{D}}((s_1, \ldots, s_q, \mathsf{Ext}_{s_1}(x), \ldots, \mathsf{Ext}_{s_q}(x)); (s_1, \ldots, s_q, y_1, \ldots, y_q)) \leq \mathsf{negl}(\lambda),$$

where $x \leftarrow\!\!\$\, \mathbf{X}$*,* $s_i \leftarrow\!\!\$\, \mathcal{S}$*, and* $y_i \leftarrow\!\!\$\, \mathcal{Y}$ *(for all* $i \in [q]$*).*

2.5 Augumented Bilinear Diffie-Hellman Exponent Assumption

Our IB-ME construction is based on the hardness of the decisional truncated ABDHE assumption, which we recall below.

Definition 6 (Decisional truncated q-ABDHE assumption). *Let* $\mathbb{G}$ *and* $\mathbb{G}_T$ *be two groups of prime order* p*. Let* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ *be an admissible bilinear map, and let* g, g' *be generators of* $\mathbb{G}$*. The decisional truncated* q*-ABDHE problem is hard in* $(\mathbb{G}, \mathbb{G}_T, e)$ *if for every PPT adversary* A*:*

$$\Big| \mathbb{P}\big[\mathsf{A}(g', g'_{q+2}, g, g_1, \ldots, g_q, e(g_{q+1}, g')) = 0\big] - \mathbb{P}\big[\mathsf{A}(g', g'_{q+2}, g, g_1, \ldots, g_q, Z) = 0\big] \Big| \leq \mathsf{negl}(\lambda),$$

where $g_i = g^{(\alpha^i)}$*,* $g, g' \leftarrow\!\!\$\, \mathbb{G}$*,* $\alpha \leftarrow\!\!\$\, \mathbb{Z}_p$ *and* $Z \in \mathbb{G}_T$*.*

3 Identity-Based Matchmaking Encryption

We recall below the definition of IB-ME presented in [2]. In IB-ME (i.e., ME in the identity-based setting), attributes and policies are treated as binary strings. We denote with rcv and snd the target identities (i.e., policies) chosen by the sender and the receiver, respectively. We say that a match (resp. mismatch) occurs when $\sigma = \mathsf{snd}$ and $\rho = \mathsf{rcv}$ (resp. $\sigma \neq \mathsf{snd}$ or $\rho \neq \mathsf{rcv}$). The receiver can choose the target identity snd on the fly.

3.1 Syntax

More formally, an IB-ME scheme is composed of the following 5 polynomial-time algorithms:

$\mathsf{Setup}(1^\lambda)$**:** Upon input the security parameter 1^λ, the randomized setup algorithm outputs the master public key mpk and the master secret key msk. We implicitly assume that all other algorithms take mpk as input.

$\mathsf{SKGen}(\mathsf{msk}, \sigma)$**:** Upon input the master secret key msk, and identity σ, the randomized sender-key generator outputs an encryption key ek_σ for σ.

$\mathsf{RKGen}(\mathsf{msk}, \rho)$**:** Upon input the master secret key msk, and identity ρ, the randomized receiver-key generator outputs a decryption key dk_ρ for ρ.

$\mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{rcv}, m)$**:** Upon input the encryption key ek_σ for identity σ, a target identity rcv, and a message $m \in \mathcal{M}$, the randomized encryption algorithm produces a ciphertext c linked to both σ and rcv.

$\mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{snd}, c)$**:** Upon input the decryption key dk_ρ for identity ρ, a target identity snd, and a ciphertext c, the deterministic decryption algorithm outputs either a message m or $\perp$.

Correctness. Correctness of IB-ME simply says that in case of a match the receiver obtains the plaintext.

Definition 7 (Correctness of IB-ME). *An IB-ME $\Pi = (\mathsf{Setup}, \mathsf{SKGen}, \mathsf{RKGen}, \mathsf{Enc}, \mathsf{Dec})$ is correct if $\forall \lambda \in \mathbb{N}$, $\forall(\mathsf{mpk}, \mathsf{msk})$ output by $\mathsf{Setup}(1^\lambda)$, $\forall m \in \mathcal{M}$, $\forall \sigma, \rho, \mathsf{rcv}, \mathsf{snd} \in \{0,1\}^*$ such that $\sigma = \mathsf{snd}$ and $\rho = \mathsf{rcv}$:*

$$\mathbb{P}[\mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{snd}, \mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{rcv}, m)) = m] \geq 1 - \mathsf{negl}(\lambda),$$

where $\mathsf{ek}_\sigma \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma)$ and $\mathsf{dk}_\rho \leftarrow\$ \mathsf{RKGen}(\mathsf{msk}, \rho)$.

3.2 Security

We now define privacy and authenticity of IB-ME. Recall that privacy captures secrecy of the sender's inputs $(\sigma, \mathsf{rcv}, m)$. This is formalized by asking the adversary to distinguish between $\mathsf{Enc}(\mathsf{ek}_{\sigma_0}, \mathsf{rcv}_0, m_0)$ and $\mathsf{Enc}(\mathsf{ek}_{\sigma_1}, \mathsf{rcv}_1, m_1)$ where $(m_0, m_1, \sigma_0, \sigma_1, \mathsf{rcv}_0, \mathsf{rcv}_1)$ are chosen by the attacker.

```
$\mathbf{G}^{\text{ib-priv}}_{\Pi,\mathsf{A}}(\lambda)$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
$(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \sigma_0, \sigma_1, \alpha) \leftarrow\$ \mathsf{A}_1^{\mathsf{O}_1,\mathsf{O}_2}(1^\lambda, \mathsf{mpk})$
$b \leftarrow\$ \{0,1\}$
$\mathsf{ek}_{\sigma_b} \leftarrow\$ \mathsf{SKGen}(\mathsf{msk}, \sigma_b)$
$c \leftarrow\$ \mathsf{Enc}(\mathsf{ek}_{\sigma_b}, \mathsf{rcv}_b, m_b)$
$b' \leftarrow\$ \mathsf{A}_2^{\mathsf{O}_1,\mathsf{O}_2}(1^\lambda, c, \alpha)$
If $(b' = b)$ return 1
Else return 0
```

```
$\mathbf{G}^{\text{ib-auth}}_{\Pi,\mathsf{A}}(\lambda)$
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
$(c, \rho, \mathsf{snd}) \leftarrow\$ \mathsf{A}^{\mathsf{O}_1,\mathsf{O}_2}(1^\lambda, \mathsf{mpk})$
$\mathsf{dk}_\rho \leftarrow\$ \mathsf{RKGen}(\mathsf{msk}, \rho)$
$m = \mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{snd}, c)$
If $\forall \sigma \in \mathcal{Q}_{\mathsf{O}_1} : (\sigma \neq \mathsf{snd}) \wedge (m \neq \perp)$
    return 1
Else return 0
```

Fig. 1: Games defining CPA-privacy and CPA-authenticity security of IB-ME. Oracles O_1, O_2 are implemented by $\mathsf{SKGen}(\mathsf{msk}, \cdot)$, $\mathsf{RKGen}(\mathsf{msk}, \cdot)$.

Definition 8 (Privacy of IB-ME [2]). *We say that an IB-ME Π satisfies* privacy *if for all valid PPT adversaries* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$*:*

$$\left| \mathbb{P}\left[\mathbf{G}^{\mathsf{ib\text{-}priv}}_{\Pi,\mathsf{A}}(\lambda) = 1 \right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda),$$

where game $\mathbf{G}^{\mathsf{ib\text{-}priv}}_{\Pi,\mathsf{A}}(\lambda)$ *is defined in Fig. 1. Adversary* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ *is called valid if* $\forall \rho \in \mathcal{Q}_{\mathsf{O}_2}$ *it satisfies the following invariant:*

$$\rho \neq \mathsf{rcv}_0 \wedge \rho \neq \mathsf{rcv}_1 \tag{1}$$

Note that, when a match occurs, IB-ME reveals all the inputs of the encryption algorithm (as the sender's and receiver's identities match). Hence, the above definition only guarantees privacy when a match does not occur (mismatch case). However, as discussed in [2], since the receiver can choose a target identity snd on the fly during the decryption process, we need to restrict privacy only to the case when the adversary holds a decryption key dk_ρ for an identity ρ that does not satisfy both target identities rcv_0 and rcv_1 (see Eq. (1)). This is because otherwise an adversary can submit a challenge $(m, m, \sigma_0, \sigma_1, \mathsf{rcv}, \mathsf{rcv})$ such that $\sigma_0 \neq \sigma_1$, and then ask for the decryption key dk_ρ for the identity $\rho = \mathsf{rcv}$ (i.e., the adversary's identity satisfies the sender's policy). Then, the adversary can retrieve the challenge bit b by simply decrypting the challenge ciphertext c under the target identity snd_0.

The definition of authenticity intuitively says that an adversary cannot compute a valid ciphertext under the identity σ, if it does not hold the corresponding encryption key ek_σ produced by the authority.

Definition 9 (Authenticity of IB-ME [2]). *We say that an IB-ME Π satisfies* authenticity *if for all PPT adversaries* A*:*

$$\mathbb{P}\left[\mathbf{G}^{\mathsf{ib\text{-}auth}}_{\Pi,\mathsf{A}}(\lambda) = 1 \right] \leq \mathsf{negl}(\lambda),$$

where game $\mathbf{G}^{\mathsf{ib\text{-}auth}}_{\Pi,\mathsf{A}}(\lambda)$ *is defined in Fig. 1.*

Note that the secret encryption key ek_σ is needed only when authenticity is required. For applications where authenticity is not required, we can simply let $\mathsf{ek}_\sigma = \sigma = \mathsf{SKGen}(\mathsf{msk}, \sigma)$ and $\mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{rcv}, m) = \mathsf{Enc}(\sigma, \mathsf{rcv}, m)$. We also observe that Definition 9 is slightly stronger than the definition of authenticity given in [2]. In particular, the adversary is allowed to obtain the decryption key dk_σ for the identity $\sigma = \mathsf{snd}$ where snd is the receiver's target identity included in the forgery (c, ρ, snd).

3.3 A Stronger Flavor of Privacy

As we argue below, the above definition of privacy provides an unsatisfactory level of security and does not match the intuitive privacy guarantee of matchmaking encryption. In particular, Definition 8 guarantees privacy only when the receiver does not hold a decryption key dk_ρ for an identity ρ that allows to decrypt the challenge ciphertext. This is reminiscent of anonymous IBE (where anonymity refers to secrecy of the sender's identity). Indeed, we can use an anonymous IBE $\Pi' = (\mathsf{Setup}', \mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$ to build an IB-ME $\Pi_{\mathsf{bad}} = (\mathsf{Setup}, \mathsf{SKGen}, \mathsf{RKGen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

1. The IB-ME encryption algorithm $\mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{rcv}, m)$ produces a ciphertext $c \leftarrow\$ \mathsf{Enc}'(\mathsf{rcv}, m||\sigma)$ where $\mathsf{ek}_\sigma = \sigma$ and $(\mathsf{msk}, \mathsf{mpk}) \leftarrow\$ \mathsf{Setup}(1^\lambda) = \mathsf{Setup}'(1^\lambda)$.
2. The IB-ME decryption algorithm $\mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{snd}, c)$ computes $m||\sigma = \mathsf{Dec}(\mathsf{dk}_\rho, c)$ where $\mathsf{dk}_\rho \leftarrow\$ \mathsf{RKGen}(\mathsf{msk}, \rho) = \mathsf{KGen}'(\mathsf{msk}, \rho)$. Finally, it outputs m if $\sigma = \mathsf{snd}$. Otherwise, it returns $\perp$.

It is easy to see that the above IB-ME satisfies privacy as per Definition 8, as security of the anonymous IBE Π' implies that $\mathsf{Enc}(\mathsf{ek}_{\sigma_0}, \mathsf{rcv}_0, m_0) = \mathsf{Enc}'(\mathsf{rcv}_0, m_0||\sigma_0) \approx_c \mathsf{Enc}'(\mathsf{rcv}_1, m_1||\sigma_1) = \mathsf{Enc}(\mathsf{ek}_{\sigma_1}, \mathsf{rcv}_1, m_1)$. However, Π_{bad} does not meet the intuitive privacy guarantee of IB-ME. Suppose a receiver, holding an identity ρ, tries to decrypt a ciphertext c computed as $\mathsf{Enc}'(\mathsf{rcv}, m||\sigma)$ where $\rho = \mathsf{rcv}$. Regardless of the selected target identity snd, the receiver will learn the sender's identity σ by simply decrypting c using the decryption key dk_ρ.

This gap is due to the fact that Definition 8 does not take into account the case in which the receiver's target identity snd is not satisfied by σ. Unfortunately, this seems inherent in that when σ_0 and σ_1 are chosen by the adversary, the attacker can simply try to decrypt the challenge ciphertext by choosing on the fly a target identity $\mathsf{snd} = \sigma_0 \neq \sigma_1$. Ateniese *et al.* [2, Remark 1] noticed this gap and informally argued that their IB-ME construction hides the message and the sender's identity to an honest receiver that uses an invalid target identity snd. For readers familiar with [2], the latter follows by the fact that their construction leverages a random oracle to derive a one-time key from the sender's identity σ. Intuitively, this allows to hide σ to an honest receiver that does not evaluate the random oracle on the same input $\mathsf{snd} = \sigma$ (i.e., to a receiver that does not choose the correct target identity $\mathsf{snd} = \sigma$).

A stronger definition of privacy. We introduce a stronger flavor of privacy, which we dub *enhanced* privacy. Enhanced privacy captures privacy of IB-ME according to every possible mismatch condition for the receiver. The main challenge is to capture the scenario in which the adversary wants to leak information from a ciphertext $c \leftarrow\$ \mathsf{Enc}(\mathsf{ek}_{\sigma_b}, \mathsf{rcv}_b, m_b)$ while holding a decryption key dk_ρ such that $\rho = \mathsf{rcv}_b$ for $b \in \{0, 1\}$. As explained in [2, Section 5], an adversary that matches the target identity chosen by the sender, can always choose on the fly a target identity snd such that $\mathsf{snd} = \sigma_0 \neq \sigma_1$ and leak the bit b by decrypting the challenge ciphertext. In order to rule out the above trivial attack, our definition of enhanced privacy modifies the mismatch condition in such a way that the sender's identities σ_0, σ_1 are hidden when the adversary holds a decryption key dk_ρ for the identity $\rho = \mathsf{rcv}$. This does not allow the attacker to choose $\mathsf{snd} = \sigma_0 \neq \sigma_1$, since σ_0, σ_1 are kept secret.

More formally, the security game for enhanced privacy (see Fig. 2) is identical to that of privacy (see Fig. 1) except that the challenge sender's attributes σ_0 and

```
$\mathbf{G}^{\mathsf{ib\text{-}priv}^+}_{\Pi,\mathsf{A}}(\lambda)$
$(\mathsf{mpk}, \mathsf{msk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$
$(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1, \alpha) \xleftarrow{\$} \mathsf{A}_1^{\mathsf{O}_1,\mathsf{O}_2}(1^\lambda, \mathsf{mpk})$
$\sigma_0 \xleftarrow{\$} \mathbf{ID}_0, \sigma_1 \xleftarrow{\$} \mathbf{ID}_1$
$\mathsf{ek}_{\sigma_0} \xleftarrow{\$} \mathsf{SKGen}(\mathsf{msk}, \sigma_0), \mathsf{ek}_{\sigma_1} \xleftarrow{\$} \mathsf{SKGen}(\mathsf{msk}, \sigma_1)$
$b \xleftarrow{\$} \{0, 1\}$
$c \xleftarrow{\$} \mathsf{Enc}(\mathsf{ek}_{\sigma_b}, \mathsf{rcv}_b, m_b)$
$b' \xleftarrow{\$} \mathsf{A}_2^{\mathsf{O}_1,\mathsf{O}_2,\{\mathsf{O}_3^i\}_{i\in\{0,1\}}}(1^\lambda, c, \alpha)$
If $(b' = b)$
    return 1
Else return 0
```

Fig. 2: Games defining enhanced privacy of IB-ME. Oracles O_1, O_2 are implemented by $\mathsf{SKGen}(\mathsf{msk}, \cdot)$, $\mathsf{RKGen}(\mathsf{msk}, \cdot)$. Oracle $\mathsf{O}_3^i(m, \mathsf{rcv})$ is implemented by $\mathsf{Enc}(\mathsf{ek}_{\sigma_i}, \mathsf{rcv}, m)$ for $i \in \{0, 1\}$.

σ_1 are replaced with two adversarial distributions $\mathbf{ID}_0$ and $\mathbf{ID}_1$. The challenger privately samples $(\sigma_0, \sigma_1) \xleftarrow{\$} \mathbf{ID}_0 \times \mathbf{ID}_1$ and proceeds as usual by computing $c \xleftarrow{\$} \mathsf{Enc}(\mathsf{ek}_{\sigma_b}, \mathsf{rcv}_b, m_b)$ for $b \xleftarrow{\$} \{0, 1\}$. To capture secrecy of σ_i for $i \in \{0, 1\}$, and avoid trivial attacks when the adversary holds dk_ρ such that $\rho = \mathsf{rcv}_i$, we require the distributions $\mathbf{ID}_i$ to have a non-trivial amount of min-entropy $\mathbb{H}_\infty(\mathbf{ID}_i) \geq \omega(\log(\lambda))$. In particular, an adversary is considered valid if for every identity ρ for which it knows the corresponding decryption key dk_ρ: (i) Either $\rho \neq \mathsf{rcv}_0$ and $\rho \neq \mathsf{rcv}_1$, or (ii) the distributions $\mathbf{ID}_0$ and $\mathbf{ID}_1$ have a non-trivial amount of min-entropy $\mathbb{H}_\infty(\mathbf{ID}_i) \geq \omega(\log(\lambda))$ for $i \in \{0, 1\}$, or (iii) $\rho \neq \mathsf{rcv}_0$ and $\mathbf{ID}_1$ has a non-trivial amount of min-entropy $\mathbb{H}_\infty(\mathbf{ID}_1) \geq \omega(\log(\lambda))$, or (iv) $\rho \neq \mathsf{rcv}_1$ and $\mathbf{ID}_0$ has a non-trivial amount of min-entropy $\mathbb{H}_\infty(\mathbf{ID}_0) \geq \omega(\log(\lambda))$.

Definition 10 (Enhanced privacy of IB-ME). *We say that an IB-ME Π satisfies* enhanced privacy *if for all valid PPT adversaries* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$*:*

$$\left| \mathbb{P}\left[\mathbf{G}^{\mathsf{ib\text{-}priv}^+}_{\Pi,\mathsf{A}}(\lambda) = 1 \right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda),$$

where game $\mathbf{G}^{\mathsf{ib\text{-}priv}^+}_{\Pi,\mathsf{A}}(\lambda)$ *is depicted in Fig. 2. Adversary* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ *is called valid if* $\forall \rho \in \mathcal{Q}_{\mathsf{O}_2}$ *it satisfies the following invariant:*

$$\begin{aligned} (\rho \neq \mathsf{rcv}_0 \wedge \rho \neq \mathsf{rcv}_1) &\vee (\mathbb{H}_\infty(\mathbf{ID}_0), \mathbb{H}_\infty(\mathbf{ID}_1) \geq \omega(\log(\lambda))) \\ &\vee (\rho \neq \mathsf{rcv}_0 \wedge \mathbb{H}_\infty(\mathbf{ID}_1) \geq \omega(\log(\lambda))) \\ &\vee (\rho \neq \mathsf{rcv}_1 \wedge \mathbb{H}_\infty(\mathbf{ID}_0) \geq \omega(\log(\lambda))). \end{aligned} \tag{2}$$

Note that, in the second query phase, the adversary has oracle access to $\mathsf{Enc}(\mathsf{ek}_{\sigma_0}, \cdot, \cdot)$ and $\mathsf{Enc}(\mathsf{ek}_{\sigma_0}, \cdot, \cdot)$. This is crucial in order to give the attacker the possibility to obtain ciphertexts under arbitrary messages and target identities when the identity σ_i is unknown (i.e., $\mathbb{H}_\infty(\mathbf{ID}_i) \geq \omega(\log(\lambda))$).

Remark 1. Observe that enhanced privacy (cf. Definition 10) is stronger than privacy (cf. Definition 8). Indeed, enhanced privacy rules out all the adversaries that choose two constant distributions $\mathbf{ID}_0 = \sigma_0$ and $\mathbf{ID}_1 = \sigma_1$ and always play the security experiment with respect to the first mismatch condition ($\rho \neq \mathsf{rcv}_0 \wedge \rho \neq \mathsf{rcv}_1$) of Eq. (2). Those are all the adversaries ruled out by Definition 8.

Remark 2. The contrived IB-ME Π_{bad} described at the beginning of Section 3.3 does not satisfy enhanced privacy. To see this, consider the adversary that plays the experiment $\mathbf{G}^{\mathsf{ib\text{-}priv}^+}_{\Pi_{\mathsf{bad}},\mathsf{A}}(\lambda)$ of Fig. 2 with respect to the second mismatch condition ($\mathbb{H}_\infty(\mathbf{ID}_0) \geq \omega(\log(\lambda) \wedge \mathbb{H}_\infty(\mathbf{ID}_1) \geq \omega(\log(\lambda)))$ of Eq. (2) as follows:

- Output a challenge $(m, m, \mathsf{rcv}, \mathsf{rcv}, \mathbf{ID}_0, \mathbf{ID}_1)$ such that $\mathbf{ID}_0$, $\mathbf{ID}_1$ have an empty intersection (i.e., there does not exist an identity σ that is output by both distributions) and $\mathbb{H}_\infty(\mathbf{ID}_0) \geq \omega(\log(\lambda))$, $\mathbb{H}_\infty(\mathbf{ID}_1) \geq \omega(\log(\lambda))$.
- Ask to $\mathsf{O}_2(\cdot) = \mathsf{RKGen}(\mathsf{msk}, \cdot) = \mathsf{KGen}'(\mathsf{msk}, \cdot)$ the decryption key dk_ρ for $\rho = \mathsf{rcv}$ (observe that this is a valid query when $\mathbb{H}_\infty(\mathbf{ID}_0) \geq \omega(\log(\lambda))$ and $\mathbb{H}_\infty(\mathbf{ID}_1) \geq \omega(\log(\lambda))$).
- Decrypt the challenge ciphertext c by executing $m||\sigma = \mathsf{Dec}'(\mathsf{dk}_\rho, c)$ using the decryption algorithm of the underlying IBE, and output $b' = 0$ if $\sigma \in \mathbf{ID}_0$. Otherwise, output $b' = 1$.

Since the encryption algorithm $\mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{rcv}, m)$ of Π_{bad} encrypts a ciphertext by running $\mathsf{Enc}'(\mathsf{rcv}, m||\sigma)$ (see Item 1 in the description of Π_{bad}) where Enc' is the encryption algorithm of the underlying IBE, the above adversary outputs $b' = b$ with overwhelming probability.

4 Construction without Random Oracles

In this section, we describe our constructions of IB-ME and prove their security. We start by giving a direct construction of an IB-ME satisfying enhanced privacy in the plain model. Hence, we show how to add authenticity generically via a generic transform (while preserving enhanced privacy).

4.1 Achieving Privacy

Our construction is based on the anonymous IBE of Gentry [4]. At a high level, in this scheme one encrypts a message m under the target identity rcv by computing $m \cdot g^s$ where s is sampled at random. During decryption, a receiver holding the correct decryption key dk_ρ for $\rho = \mathsf{rcv}$ is able to compute the inverse g^{-s} of g^s (by leveraging auxiliary information included in the ciphertext) and therefore obtain the message. Our IB-ME leverages the homomorphic properties of the IBE scheme to encrypt the message as $m \cdot g^s \cdot g_\sigma$, where g_σ is output by a reusable extractor $\mathsf{Ext}_x(\sigma)$. This way, a receiver also needs to choose the correct target identity $\mathsf{snd} = \sigma$ to recompute g_σ and recover m. Since our construction will not meet authenticity directly, we will assume that $\sigma = \mathsf{ek}_\sigma = \mathsf{SKGen}(\mathsf{msk}, \sigma)$ and $\mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{rcv}, m) = \mathsf{Enc}(\sigma, \mathsf{rcv}, m)$.

Construction 1 *Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p, and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a symmetric pairing, and let $\mathsf{Ext} : \mathcal{S} \times \mathbb{Z}_p \to \mathbb{G}_T$.*

$\mathsf{Setup}(1^\lambda)$**:** *Sample random generators $g \in \mathbb{G}$ and $\alpha, y \leftarrow\$ \mathbb{Z}_p$. Compute $g_\alpha = g^\alpha \in \mathbb{G}$, and $h = g^y$. Output $\mathsf{mpk} = (g, g_\alpha, h)$ and $\mathsf{msk} = (\alpha, y)$.*

$\mathsf{SKGen}(\mathsf{msk}, \sigma)$**:** *Upon input $\mathsf{msk} = (\alpha, y)$ and $\sigma \in \{0,1\}^*$, return $\mathsf{ek}_\sigma = \sigma$.*

$\mathsf{RKGen}(\mathsf{msk}, \rho)$**:** *Upon input $\mathsf{msk} = (\alpha, y)$ and $\rho \in \mathbb{Z}_p$, sample $r_\rho \in \mathbb{Z}_p$ and output $\mathsf{dk}_\rho = (h_\rho, r_\rho)$, where $h_\rho = g^{\frac{y - r_\rho}{\alpha - \rho}}$. If an indentity $\rho \in \mathbb{Z}_p$ is queried multiple times, we require RKGen to use the same value r_ρ (this can be accomplished by leveraging a PRF).*

$\mathsf{Enc}(\mathsf{ek}_\sigma, \mathsf{rcv}, m)$**:** *Upon input $\mathsf{ek}_\sigma = \sigma \in \{0,1\}^*$, $\mathsf{rcv} \in \mathbb{Z}_p$, and $m \in \mathbb{G}_T$, sample $s \leftarrow\$ \mathbb{Z}_p$, $x \leftarrow\$ \mathcal{S}$, compute $g_\sigma = \mathsf{Ext}_x(\sigma)$, and return $c = (c_1, c_2, c_3, c_4)$ where*

$$c_1 = (g_\alpha \cdot g^{-\mathsf{rcv}})^s, \quad c_2 = e(g,g)^s, \quad c_3 = x, \quad c_4 = m \cdot e(g,h)^{-s} \cdot g_\sigma.$$

$\mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{snd}, c)$**:** *Upon input $\mathsf{dk}_\rho = (h_\rho, r_\rho)$, $\mathsf{snd} \in \{0,1\}^*$, and $c = (c_1, c_2, c_3, c_4)$, return $m = c_4 \cdot e(c_1, h_\rho) \cdot c_2^{r_\rho} \cdot g_{\mathsf{snd}}^{-1}$ where $g_{\mathsf{snd}} = \mathsf{Ext}_{c_3}(\mathsf{snd})$.*

Correctness (cf. Definition 7) follows because $\forall \sigma, \mathsf{rcv}, \rho, \mathsf{snd} \in \mathbb{Z}_p$, $(h_\rho, r_\rho) = \mathsf{dk}_\rho \leftarrow\$ \mathsf{RKGen}(\mathsf{msk}, \rho)$ such that $\mathsf{snd} = \sigma$ and $\mathsf{rcv} = \rho$, we have:

$$g_{\mathsf{snd}} = \mathsf{Ext}_{c_3}(\mathsf{snd}) = \mathsf{Ext}_x(\sigma) = g_\sigma, \text{ and}$$
$$e(c_1, h_\rho) \cdot c_2^{r_\rho} = e(g^{s(\alpha-\rho)}, g^{\frac{y-r_\rho}{\alpha-\rho}}) \cdot e(g,g)^{s \cdot r_\rho} = e(g,h)^s.$$

The theorem below says that the above scheme satisfies enhanced privacy. The proof of security leverages both the homomorphic properties and the ciphertext structure of Gentry's scheme. For this reason, our technique does not extend directly to any anonymous IBE scheme.

Theorem 1. *Assuming that Ext is an $(\omega(\log(\lambda)), q_{\mathsf{ext}})$-reusable-extractor, and that the truncated decisional q_{abdhe}-ABDHE problem is hard, then the IB-ME Π from Construction 1 satisfies enhanced privacy, so long as $q_{\mathsf{abdhe}} = q_{\mathsf{O}_2} + 1$ and $q_{\mathsf{ext}} = \max\{q_{\mathsf{O}_3^0}, q_{\mathsf{O}_3^1}\} + 1$ (where q_{O} is the number of queries submitted to oracle O in the game of Fig. 2).*

Proof. For brevity, let $\mathbf{G}^{\text{ib-priv}^+}_{\Pi,\mathsf{A}}(\lambda) = \mathbf{G}(\lambda)$ be the experiment of Fig. 2. Recall that, in order to be valid, the adversary A must satisfy at least one of the four mismatch conditions given in Eq. (2); we define the events corresponding to each condition below:

$$\mathbf{Mismatch}_1 : \forall \rho \in \mathcal{Q}_{\mathsf{O}_2}, \rho \neq \mathsf{rcv}_0 \wedge \rho \neq \mathsf{rcv}_1 \tag{3}$$
$$\mathbf{Mismatch}_2 : \mathbb{H}_\infty(\mathbf{ID}_0), \mathbb{H}_\infty(\mathbf{ID}_1) \geq \omega(\log(\lambda)) \tag{4}$$
$$\mathbf{Mismatch}_3 : \forall \rho \in \mathcal{Q}_{\mathsf{O}_2}, \rho \neq \mathsf{rcv}_0 \wedge \mathbb{H}_\infty(\mathbf{ID}_1) \geq \omega(\log(\lambda)) \tag{5}$$
$$\mathbf{Mismatch}_4 : \forall \rho \in \mathcal{Q}_{\mathsf{O}_2}, \rho \neq \mathsf{rcv}_1 \wedge \mathbb{H}_\infty(\mathbf{ID}_0) \geq \omega(\log(\lambda)). \tag{6}$$

Lemma 1. $\left|\mathbb{P}\left[\mathbf{G}^{\text{ib-priv}^+}_{\Pi,\mathsf{A}}(\lambda) = 1 \middle| \mathbf{Mismatch}_1\right] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda).$

Proof. We consider a sequence of hybrid experiments. For the rest of this proof, we think of the experiments as conditioned on the event $\mathbf{Mismatch}_1$ of Eq. (3).

$\mathbf{H}_1(\lambda)$: This is identical to $\mathbf{G}(\lambda)$. Without loss of generality we assume the adversary A does not make any query to oracles $\{\mathsf{O}_3^i\}$ for $i \in \{0,1\}$. This is because, according to Eq. (3), A can choose two constant distributions $\sigma_0 = \mathbf{ID}_0, \sigma_1 = \mathbf{ID}_1$ and simulate the oracle $\mathsf{O}_3^i(m, \mathsf{rcv})$ as $\mathsf{Enc}(\mathsf{ek}_{\sigma_i}, \mathsf{rcv}, m)$ where $\mathsf{ek}_{\sigma_i} \leftarrow\!\!\$\ \mathsf{O}_1(1^\lambda, \sigma_i)$, for $i \in \{0,1\}$.

$\mathbf{H}_2(\lambda)$: Same as $\mathbf{H}_1(\lambda)$, except that, after receiving the challenge $(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \sigma_0 = \mathbf{ID}_0, \sigma_1 = \mathbf{ID}_1)$ from the adversary (recall we assume that $\mathbf{ID}_0, \mathbf{ID}_1$ are constant distributions), the challenger produces the challenge ciphertext $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where c_4^* is computed as

$$c_4^* = (m_b \cdot g_{\sigma_b}^*)/(e(c_1^*, h_{\mathsf{rcv}_b}) \cdot {c_2^*}^{r^*_{\mathsf{rcv}_b}}) \tag{7}$$

for $(h^*_{\mathsf{rcv}_b}, r^*_{\mathsf{rcv}_b}) \leftarrow\!\!\$\ \mathsf{RKGen}(\mathsf{msk}, \mathsf{rcv}_b)$ and $g^*_{\sigma_b} = \mathsf{Ext}_{c_3^*}(\sigma_b)$. Observe that the value $1/(e(c_1^*, h_{\mathsf{rcv}_b}) \cdot {c_2^*}^{r^*_{\mathsf{rcv}_b}})$ in Eq. (7) can be computed by running $e(g,h)^{-s}$ as in the decryption algorithm.

$\mathbf{H}_3(\lambda)$: Same as $\mathbf{H}_2(\lambda)$, except for the following differences.

Setup: The challenger samples a random polynomial $f(x) \leftarrow\!\!\$\ \mathbb{Z}_p[x]$ of degree $q = q_{\mathsf{abdhe}}$, $\alpha \leftarrow\!\!\$\ \mathbb{Z}_p$, and sets $g_\alpha = g^\alpha$ and $h = g^{f(\alpha)}$. Then, it returns $\mathsf{mpk} = (g, g_\alpha, h)$ and keeps $\mathsf{msk} = (\alpha, y)$ where $y = f(\alpha)$.

$\mathsf{RKGen} = \mathsf{O}_2$**:** On input $\rho \in \mathbb{Z}_p$ for $\mathsf{RKGen} = \mathsf{O}_2$, the challenger defines the polynomial $F_\rho(x) = (f(x) - f(\rho))/(x - \rho)$ of degree $q-1$ and computes $h_\rho = g^{F_\rho(\alpha)}$ and $r_\rho = f(\rho)$. Finally, it returns $\mathsf{dk}_\rho = (h_\rho, r_\rho)$.

Challenge: The challenger receives the challenge $(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \sigma_0 = \mathbf{ID}_0, \sigma_1 = \mathbf{ID}_1)$. It samples $b \leftarrow\!\!\$\ \{0,1\}$ and it defines the degree $q+1$ polynomial

$$F^*(x) = \frac{x^{q+2} - \mathsf{rcv}_b^{q+2}}{x - \mathsf{rcv}_b} = \sum_{i=0}^{q+1} F_i^* \cdot x^i,$$

where F_i^* is the i-th coefficient of F^*. It computes the challenge ciphertext $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ as $c_1^* = g'^{\alpha^{q+2}} \cdot g'^{-\mathsf{rcv}_b^{q+2}}$ and $c_2^* = e(g', g)^{\alpha^{q+2}} \cdot e(g', \prod_{i=0}^{q} (g^{\alpha^i})^{F_i^*})$, where $g' \leftarrow\!\!\$\ \mathbb{G}$, and c_3^*, c_4^* are computed as described in experiment $\mathbf{H}_2(\lambda)$.

$\mathbf{H}_4(\lambda)$: Same as $\mathbf{H}_3(\lambda)$, except that the challenger generates c_1^* and c_2^* in the challenge ciphertext using different randomness. More in details, the challenger computes $c_1^* = (g_\alpha \cdot g^{\mathsf{rcv}_b})^{s_1}$ and $c_2^* = e(g,g)^{s_2}$ for $s_1 \leftarrow\!\!\$\ \mathbb{Z}_p$ and $s_2 \leftarrow\!\!\$\ \mathbb{Z}_p \setminus \{s_1\}$.

Claim. $\{\mathbf{H}_1(\lambda)\}_{\lambda \in \mathbb{N}} \equiv \{\mathbf{H}_2(\lambda)\}_{\lambda \in \mathbb{N}}$.

Proof. The difference between $\mathbf{H}_1(\lambda)$ and $\mathbf{H}_2(\lambda)$ is purely conceptional. Hence, the claim follows.

Claim. $\{\mathbf{H}_2(\lambda)\}_{\lambda \in \mathbb{N}} \equiv \{\mathbf{H}_3(\lambda)\}_{\lambda \in \mathbb{N}}$.

Proof. We show that $\mathbf{H}_2(\lambda)$ and $\mathbf{H}_3(\lambda)$ are identically distributed. The distribution of mpk and msk in $\mathbf{H}_3(\lambda)$ is perfectly simulated since f is a random polynomial. The challenger evaluates the polynomial $f(x)$ on points $\mathcal{I} = \{\alpha, \mathsf{rcv}_b\} \cup \mathcal{Q}_{\mathsf{O}_2}$. Let $q = q_{\mathsf{abdhe}}$. Since $|\mathcal{I}| \leq q+1$ and f are random polynomials of degree q, we have that $\{f(i)\}_{i \in \mathcal{I}}$ are uniform and independent as in $\mathbf{H}_2(\lambda)$.

As for the challenge ciphertext, note that c_3^*, c_4^* are computed in the same way in both experiments. Hence, we focus on c_1^*, c_2^*. In $\mathbf{H}_3(\lambda)$ we can write c_1^* and c_2^* as follows

$$c_1^* = ({g'}^{\alpha^{q+2}} \cdot {g'}^{-\mathsf{rcv}_b^{q+2}}) = g^{t(\alpha - \mathsf{rcv}_b)F^*(\alpha)}$$
$$c_2^* = e(g', g)^{\alpha^{q+2}} \cdot e(g', \prod_{i=0}^{q} (g^{\alpha^i})^{F_i^*}) = e(g^t, g^{F^*(\alpha)})$$

where $g' = g^t$. By setting the randomness $s = t \cdot F^*(\alpha)$ (note that s is random since g' is random) we obtain that c_1^*, c_2^* of $\mathbf{H}_3(\lambda)$ are identically distributed to the ones of $\mathbf{H}_2(\lambda)$. This concludes the proof.

Claim. $\{\mathbf{H}_3(\lambda)\}_{\lambda \in \mathbb{N}} \approx_c \{\mathbf{H}_4(\lambda)\}_{\lambda \in \mathbb{N}}$.

Proof. For the sake of clarity, let $q = q_{\mathsf{abdhe}}$. Assume there exists a distinguisher D that is able to distinguish between $\mathbf{H}_3(\lambda)$ and $\mathbf{H}_4(\lambda)$ with non-negligible advantage. We build an adversary A that solves the q-ABDHE problem. A receives as input $(g', {g'}^{\alpha^{q+2}}, g, g^{\alpha}, \ldots, g^{\alpha^q}, Z)$ and proceeds as in $\mathbf{H}_3(\lambda)$ except for the following differences.

- At setup, it samples a random polynomial $f(x) \leftarrow\!\!\$\; \mathbb{Z}_p[x]$ of degree q and sets $h = g^{f(\alpha)}$. Note that h can be computed without knowing α using the values $g, g^{\alpha}, \ldots, g^{\alpha^q}$. Send $\mathsf{mpk} = (g, g_{\alpha} = g^{\alpha}, h)$ to D. Note that the distribution of mpk is perfectly simulated and this implicitly defines the secret key $\mathsf{msk} = (\alpha, y)$ where $y = f(\alpha)$.
- On input $\rho \in \mathbb{Z}_p$ for $\mathsf{RKGen} = \mathsf{O}_2$, it answers as in $\mathbf{H}_3(\lambda)$ except that $h_\rho = g^{F_\rho(\alpha)}$ is computed without knowing α using $g, g^{\alpha}, \ldots, g^{\alpha^q}$. Note that dk_ρ is a correctly simulated decryption key.
- During the challenge phase, it receives $(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \sigma_0 = \mathbf{ID}_0, \sigma_1 = \mathbf{ID}_1)$. Hence, A samples $b \leftarrow\!\!\$\; \{0,1\}$ and defines the degree $q+1$ polynomial

$$F^*(x) = \frac{x^{q+2} - \mathsf{rcv}_b^{q+2}}{x - \mathsf{rcv}_b} = \sum_{i=0}^{q+1} F_i^* \cdot x^i$$

as in $\mathbf{H}_3(\lambda)$. Finally, A computes the challenge ciphertext $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ as in $\mathbf{H}_3(\lambda)$ except that it sets $c_1^* = {g'}^{\alpha^q+2} \cdot {g'}^{-\mathsf{rcv}_b^{q+2}}$ and $c_2^* = Z \cdot e(g', \prod_{i=0}^{q} (g^{\alpha^i})^{F_i^*})$. Note that c_1^*, c_2^* can be computed using the input for the q-ABDHE problem.

As in the proof of the previous claim, if c_1^*, c_2^* are correctly distributed, so are c_3^*, c_4^*. We write c_1^* as $c_1^* = {g'}^{\alpha^{q+2}} \cdot {g'}^{-\mathsf{rcv}_b^{q+2}} = g^{t(\alpha - \mathsf{rcv}_b)F^*(\alpha)} = g^{(\alpha - \mathsf{rcv}_b)s_1}$, for

$s_1 = t \cdot F^*(\alpha)$. Note that s_1 is random since g' is a random generator of $\mathbb{G}$. If $Z = e(g', g)^{\alpha^{q+1}}$, the ciphertext c^* is distributed as in $\mathbf{H}_3(\lambda)$ since c_1^* and c_2^* are computed using the same randomness. Indeed, we have

$$c_2^* = Z \cdot e(g', \prod_{i=0}^{q} (g^{\alpha^i})^{F_i^*}) = e(g^t, g^{F^*(\alpha)}) = e(g, g)^{s_1}.$$

On the other hand, if $Z \leftarrow\!\!\$\ \mathbb{G}_T$ so is c_2^* as in $\mathbf{H}_4(\lambda)$. This concludes the proof.

In the last experiment, c_1^*, c_2^*, and c_3^* look like three random elements in $\mathbb{G}$, $\mathbb{G}_T$, and $\mathcal{S}$, respectively. Since c_1^* and c_2^* are random, the inequalities $c_2^* \neq e(c_1^*, g)^{\frac{1}{\alpha - \mathsf{rcv}_0}}$ and $c_2^* \neq e(c_1^*, g)^{\frac{1}{\alpha - \mathsf{rcv}_1}}$ hold with overwhelming probability. When the above inequalities hold, the value $e(c_1^*, h^*_{\mathsf{rcv}_b}) \cdot (c_2^*)^{r^*_{\mathsf{rcv}_b}}$ (used to compute c_4^*) is uniformly distributed in $\mathbb{G}_T$ since $r^*_{\mathsf{rcv}_b}$ is random and independent from the A's view (since A can not ask for decryption key $\mathsf{dk}_{\mathsf{rcv}_0}$ and $\mathsf{dk}_{\mathsf{rcv}_1}$). As a consequence, the tuple $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ does not leak any information about b (except with negligible probability). Hence, Lemma 1 follows by combining the above claims.

Lemma 2. $\left| \mathbb{P}\left[\mathbf{G}^{\mathsf{ib\text{-}priv}^+}_{\Pi,\mathsf{A}}(\lambda) = 1 \middle| \mathbf{Mismatch}_2 \right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)$.

Proof. Without loss of generality, assume $q_{\mathsf{O}_3^b} \geq q_{\mathsf{O}_3^{1-b}}$. Hence, we have $q_{\mathsf{ext}} = q_{\mathsf{O}_3^b} + 1$. We consider a sequence of hybrid experiments. For the rest of this proof, we think of the experiments as conditioned on the event $\mathbf{Mismatch}_2$ of Eq. (4).

$\mathbf{H}_1(\lambda)$: This is identical to $\mathbf{G}(\lambda)$.

$\mathbf{H}_2(\lambda)$: Same as $\mathbf{H}_1(\lambda)$, except that the challenger changes how it produces the challenge and the answers of oracles O_3^0 and O_3^1 for $i \in \{0, 1\}$. Let $\mathcal{L}_0$ and $\mathcal{L}_1$ be two empty sets:
- When computing the challenge $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ for bit b, the challenger adds c_3^* to $\mathcal{L}_b$.
- On input (m, rcv) for O_3^i, the challenger computes $c = (c_1, c_2, c_3, c_4)$ as in $\mathbf{H}_1(\lambda)$. Then, if $c_3 \in \mathcal{L}_i$, the challenger aborts. Otherwise, it adds c_3 to $\mathcal{L}_i$ and proceeds as in $\mathbf{H}_1(\lambda)$.

$\mathbf{H}_3(\lambda)$: Same as $\mathbf{H}_2(\lambda)$, except that the challenger changes how it produces the challenge and the answers of oracles O_3^b where b is the challenge bit.
- When computing the challenge $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ for bit b, the challenger samples $g^*_{\sigma_b}$ at random from $\mathbb{G}_T$.
- On input (m, rcv) for O_3^b, the challenger samples $s \leftarrow\!\!\$\ \mathbb{Z}_p$ and computes (c_1, c_2, c_3) under the randomness s (note that c_1, c_2, c_3 are computed as usual). Then, it samples $g_{\sigma_b} \leftarrow\!\!\$\ \mathbb{G}_T$ and it computes $c_4 = m \cdot e(g, h)^{-s} \cdot g_{\sigma_b}$.

$\mathbf{H}_4(\lambda)$: Same as $\mathbf{H}_3(\lambda)$, except that the challenger changes the answers of oracles O_3^{1-b} where b is the challenge bit.
- On input (m, rcv) for O_3^{1-b}, the challenger samples $s \leftarrow\!\!\$\ \mathbb{Z}_p$ and computes (c_1, c_2, c_3) under the randomness s (note that c_1, c_2, c_3 are computed as usual). Then, it samples $g_{\sigma_{1-b}} \leftarrow\!\!\$\ \mathbb{Z}_p$ and it computes $c_4 = m \cdot e(g, h)^{-s} \cdot g_{\sigma_{1-b}}$.

$\mathbf{H}_5(\lambda)$**:** Same as $\mathbf{H}_4(\lambda)$, except that, after receiving the challenge $(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1)$ from the adversary, the challenger produces the challenge ciphertext $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ where c_4^* is computed as

$$c_4^* = (m_b \cdot g_{\sigma_b}^*)/(e(c_1^*, h_{\mathsf{rcv}_b}) \cdot c_2^{*\, r_{\mathsf{rcv}_b}^*}), \tag{8}$$

where $(h_{\mathsf{rcv}_b}^*, r_{\mathsf{rcv}_b}^*) \xleftarrow{\$} \mathsf{RKGen}(\mathsf{msk}, \mathsf{rcv}_b)$ and $g_{\sigma_b}^* \xleftarrow{\$} \mathbb{G}_T$. Note that the value $1/(e(c_1^*, h_{\mathsf{rcv}_b}^*) \cdot c_2^{*\, r_{\mathsf{rcv}_b}^*})$ in Eq. (8) can be computed by running $e(g,h)^{-s}$ in the decryption algorithm. The same approach is used to answer the queries submitted to O_3^0 and O_3^1. On input (m, rcv) for O_3^i for $i \in \{0,1\}$, the challenger computes $c = (c_1, c_2, c_3, c_4)$ except that c_4 is computed as $c_4 = (m \cdot g_{\sigma_i})/(e(c_1, h_{\mathsf{rcv}}) \cdot {c_2}^{r_{\mathsf{rcv}}})$, where $(h_{\mathsf{rcv}}, r_{\mathsf{rcv}}) \xleftarrow{\$} \mathsf{RKGen}(\mathsf{msk}, \mathsf{rcv})$ and $g_{\sigma_i} \xleftarrow{\$} \mathbb{G}_T$

$\mathbf{H}_6(\lambda)$**:** Same as $\mathbf{H}_5(\lambda)$, except for the following differences.

Setup: The challenger samples a random polynomial $f(x) \xleftarrow{\$} \mathbb{Z}_p[x]$ of degree $q = q_{\mathsf{abdhe}}$, $\alpha \xleftarrow{\$} \mathbb{Z}_p$, and sets $g_\alpha = g^\alpha$ and $h = g^{f(\alpha)}$. It returns $\mathsf{mpk} = (g, g_\alpha, h)$ and keeps $\mathsf{msk} = (\alpha, y)$ where $y = f(\alpha)$.

$\mathsf{RKGen}(1^\lambda, \cdot) = \mathsf{O}_2(\cdot)$**:** On input $\rho \in \mathbb{Z}_p$ for $\mathsf{RKGen} = \mathsf{O}_2$, the challenger defines the polynomial $F_\rho(x) = (f(x) - f(\rho))/(x - \rho)$ of degree $q-1$ and computes $h_\rho = g^{F_\rho(\alpha)}$ and $r_\rho = f(\rho)$ Finally, it returns $\mathsf{dk}_\rho = (h_\rho, r_\rho)$.

Challenge: The challenger receives the challenge $(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1)$. It samples $b \xleftarrow{\$} \{0,1\}$ and it defines the degree $q+1$ polynomial

$$F^*(x) = \frac{x^{q+2} - \mathsf{rcv}_b^{q+2}}{x - \mathsf{rcv}_b} = \sum_{i=0}^{q+1} F_i^* \cdot x^i,$$

where F_i^* is the i-th coefficient of F^*. It computes the challenge ciphertext $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ as $c_1^* = g'^{\alpha^{q+2}} \cdot g'^{-\mathsf{rcv}_b^{q+2}}$ and $c_2^* = e(g', g)^{\alpha^{q+2}} \cdot e(g', \prod_{i=0}^{q} (g^{\alpha^i})^{F_i^*})$, where $g' \xleftarrow{\$} \mathbb{G}$, and c_3^*, c_4^* are computed as described in experiment $\mathbf{H}_5(\lambda)$.

$\mathsf{Enc}(\mathsf{ek}_{\sigma_i}, \cdot, \cdot) = \mathsf{O}_3^i(\cdot, \cdot)$**:** On input (m, rcv) for $\mathsf{Enc} = \mathsf{O}_3^i$, the challenger generates the decryption key $\mathsf{dk}_{\mathsf{rcv}} = (h_{\mathsf{rcv}}, r_{\mathsf{rcv}}) \xleftarrow{\$} \mathsf{O}_2(1^\lambda, \mathsf{rcv})$ and computes $c = (c_1, c_2, c_3, c_4)$ as in $\mathbf{H}_5(\lambda)$,i.e.

$$\begin{aligned} c_1 &= (g_\alpha \cdot g^{-\mathsf{rcv}})^s, \quad c_2 = e(g,g)^s, \quad c_3 = x \\ c_4 &= (m \cdot g_{\sigma_i})/(e(c_1, h_{\mathsf{rcv}}) \cdot {c_2}^{r_{\mathsf{rcv}}}), \end{aligned}$$

where $s \xleftarrow{\$} \mathbb{Z}_p$, $g_{\sigma_i} \xleftarrow{\$} \mathbb{G}_T$.

$\mathbf{H}_7(\lambda)$**:** Same as $\mathbf{H}_6(\lambda)$, except that the challenger generates c_1^* and c_2^* in the challenge ciphertext using different randomness. More in details, the challenger compute $c_1^* = (g_\alpha \cdot g^{\mathsf{rcv}_b})^{s_1}$ and $c_2^* = e(g,g)^{s_2}$ for $s_1 \xleftarrow{\$} \mathbb{Z}_p$ and $s_2 \xleftarrow{\$} \mathbb{Z}_p \setminus \{s_1\}$.

Claim. $\{\mathbf{H}_1(\lambda)\}_{\lambda \in \mathbb{N}} \approx_c \{\mathbf{H}_2(\lambda)\}_{\lambda \in \mathbb{N}}$.

Proof. The claim follows by simply observing that each time c_3 is sampled at random. Hence, since the adversary submits at most a polynomial number of queries to oracles O_3^0 and O_3^1, the probability that $c_3 \in \mathcal{L}_0$ or $c_3 \in \mathcal{L}_1$ is negligible.

Claim. $\{\mathbf{H}_2(\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{H}_3(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. Assume there exists D telling apart the two experiments with non-negligible advantage. We build an adversary A that breaks the security of the reusable extractor.

1. A proceeds as in experiment $\mathbf{H}_2(\lambda)$ until the challenge phase.
2. During the challenge phase, it receives $(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1)$. Hence, A samples $b \leftarrow\!\!\$\ \{0,1\}$ and sends $\mathbf{ID}_b$ to the challenger. It receives $(x_1, \ldots, x_{q_{\mathsf{ext}}}, g_1, \ldots, g_{q_{\mathsf{ext}}})$, where A has to determine if $g_i = \mathsf{Ext}_{x_i}(\sigma_b)$ for $\sigma_b \leftarrow\!\!\$\ \mathbf{ID}_b$. Hence:
   - It samples $\sigma_{1-b} \leftarrow\!\!\$\ \mathbf{ID}_{1-b}$ and it creates an empty set $\mathcal{L}_b$.
   - It computes $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ as in $\mathbf{H}_2(\lambda)$ except that $c_3^* = x_1$ and $c_4^* = m_b \cdot e(g,h)^{s^*} \cdot g_1$, where s^* is the randomness used to compute c_1^* and c_2^*.
3. During the second query phase, the adversary answers to the queries submitted as usual except for O_3^b:
   - On input the i-th query (m, rcv) for O_3^b, the adversary computes $c = (c_1, c_2, c_3, c_4)$ as in $\mathbf{H}_2(\lambda)$ except that $c_3 = x_i$ and $c_4 = m \cdot e(g,h)^{-s} \cdot g_i$, where s is the randomness used to compute c_1 and c_2.

Note that $q_{\mathsf{ext}} = q_{\mathsf{O}_3^b}$ and $\mathbb{H}_\infty(\mathbf{ID}_b) \geq \omega(\log(\lambda))$. It is easy to see that if $(g_1, \ldots, g_{q_{\mathsf{ext}}}) = (\mathsf{Ext}_{x_1}(\sigma_b), \ldots, \mathsf{Ext}_{x_{q_{\mathsf{ext}}}}(\sigma_b))$ then A perfectly simulates experiment $\mathbf{H}_2(\lambda)$. On the other hand, if $(g_1, \ldots, g_{q_{\mathsf{ext}}})$ are random elements, then$\mathsf{A}$ perfectly simulates $\mathbf{H}_3(\lambda)$. Hence, A breaks the security of the reausable extractors with the same advantage of D. This concludes the proof.

Claim. $\{\mathbf{H}_3(\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{H}_4(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. Identical to the analogous step in the proof of Lemma 1, and therefore omitted.

Claim. $\{\mathbf{H}_4(\lambda)\}_{\lambda\in\mathbb{N}} \equiv \{\mathbf{H}_5(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. The difference between the two hybrids is purely conceptional. Hence, the claim follows.

Claim. $\{\mathbf{H}_5(\lambda)\}_{\lambda\in\mathbb{N}} \equiv \{\mathbf{H}_6(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. Similarly to the proof of a previous claim, we have that the setup phase, the challenge phase, and the queries to oracle O_2 are perfectly simulated. It follows that the answers returned by O_3^i in $\mathbf{H}_5(\lambda)$ are identical to the ones in $\mathbf{H}_6(\lambda)$, for $i \in \{0,1\}$. This concludes the proof.

Claim. $\{\mathbf{H}_6(\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{H}_7(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. Similar to the proof of the corresponding step in Lemma 1. The only differences are that oracles O_3^i must be simulated as defined in $\mathbf{H}_5(\lambda)$ and that the challenge ciphertext $c = (c_1^*, c_2^*, c_3^*, c_4^*)$ can be simulated by sampling $g_{\sigma_b}^*$ uniformly at random from $\mathbb{G}_T$ as in $\mathbf{H}_5(\lambda)$.

In the last experiment, c_1^*, c_2^*, c_3^* are random elements in $\mathbb{G}$, $\mathbb{G}_T$, and $\mathcal{S}$, respectively. Moreover, since $g_{\sigma_b}^*$ (used to compute c_4^*) is sampled at random, we conclude that the tuple $c^* = (c_1^*, c_2^*, c_3^*, c_4^*)$ does not leak any information about b (except with negligible probability). Hence, Lemma 2 follows by combining the above claims.

Lemma 3. $\left|\mathbb{P}\left[\mathbf{G}_{\Pi,\mathsf{A}}^{\mathsf{ib\text{-}priv}^+}(\lambda) = 1 \middle| \mathbf{Mismatch}_3\right] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$.

Proof. As in the proof of Lemma 2, we assume $q_{\mathsf{O}_3^b} \geq q_{\mathsf{O}_3^{1-b}}$. Hence, we have $q_{\mathsf{ext}} = q_{\mathsf{O}_3^b} + 1$. We consider a sequence of hybrid experiments, where $\mathbf{H}^b(\lambda)$ is the experiment with challenge bit is $b \in \{0,1\}$. For the rest of this proof, we think of the experiments as conditioned on the event $\mathbf{Mismatch}_3$ of Eq. (5).

$\mathbf{H}_1^0(\lambda)$: This is identical to $\mathbf{G}(\lambda)$ with challenge bit $b = 0$. Without loss of generality we assume the adversary A does not make any query to oracles O_3^0. Similarly to $\mathbf{H}_1(\lambda)$ in the proof of Lemma 1, according to Eq. (5), the adversary A can choose a constant distribution $\sigma_0 = \mathbf{ID}_0$ and simulate the oracle $\mathsf{O}_3^0(m, \mathsf{rcv})$ as $\mathsf{Enc}(\mathsf{ek}_{\sigma_0}, \mathsf{rcv}, m)$ where $\mathsf{ek}_{\sigma_0} \leftarrow\!\!\$\, \mathsf{O}_1(1^\lambda, \sigma_0)$. Observe that $\mathbf{H}_1^0(\lambda)$ is identical to $\mathbf{H}_1(\lambda)$ in the proof of Lemma 1, except that we fix the challenge bit $b = 0$ and we assume only $\sigma_0 = \mathbf{ID}_0$ as constant distribution.

$\mathbf{H}_i^0(\lambda)$, for $i \in \{2,3,4\}$: Each hybrid $\mathbf{H}_i^0(\lambda)$ is defined as $\mathbf{H}_i(\lambda)$ in the proof of Lemma 1 for $i \in \{2,3,4\}$ except that we fix the bit $b = 0$ and, similarly to $\mathbf{H}_1^0(\lambda)$ we assume only $\sigma_0 = \mathbf{ID}_0$ is constant (and thus there are no queries submitted to O_3^0).

$\mathbf{H}_5^0(\lambda)$: Same as $\mathbf{H}_4^0(\lambda)$ except that the challenger changes how it produces the answers of oracle O_3^1. Let $\mathcal{L}_1$ be an empty set:

- On input (m, rcv) for O_3^1, the challenger computes $c = (c_1, c_2, c_3, c_4)$ as in $\mathbf{H}_1^0(\lambda)$. Then, if $c_3 \in \mathcal{L}_1$, the challenger aborts. Otherwise, it adds c_3 to $\mathcal{L}_1$ and proceeds as in $\mathbf{H}_4^0(\lambda)$.

Note that $\mathbf{H}_5^0(\lambda)$ is defined similarly to $\mathbf{H}_2(\lambda)$ in the proof of Lemma 2.

$\mathbf{H}_6^0(\lambda)$: Same as $\mathbf{H}_5^0(\lambda)$ except that the challenger changes the answers of oracles O_3^1 as follows.

- On input (m, rcv) for O_3^1, the challenger computes (c_1, c_2, c_3, c_4) as in $\mathbf{H}_5^0(\lambda)$ except that $g_{\sigma_1} \leftarrow\!\!\$\, \mathbb{G}_T$ (note that g_{σ_1} is used to compute c_4).

Note that $\mathbf{H}_6^0(\lambda)$ is defined similarly to $\mathbf{H}_4(\lambda)$ in the proof of Lemma 2.

$\mathbf{H}_i^1(\lambda)$, for $i \in \{7,8,9,10,11,12,13\}$ Each hybrid $\mathbf{H}_i^1(\lambda)$ is defined as $\mathbf{H}_{i-4}(\lambda)$ in the proof of Lemma 2 except that we fix the bit $b = 0$ and, similarly to $\mathbf{H}_1^0(\lambda)$ we assume only $\sigma_0 = \mathbf{ID}_0$ is constant (and thus there are no queries submitted to O_3^0). Note that $\mathbf{H}_5^1(\lambda)$ is identical to $\mathbf{G}(\lambda)$ with challenge bit $b = 1$.

Claim. $\{\mathbf{H}_1^0(\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{H}_4^0(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. Identical to the proof of a previous claim, except that we set the challenge bit $b = 0$ and we simulate O_3^1 as defined in Construction 1. In particular, one can show:

$$\mathbf{H}_1^0(\lambda) \equiv \mathbf{H}_2^0 \equiv \mathbf{H}_3^0(\lambda) \approx_c \mathbf{H}_4^0(1^\lambda).$$

Claim. $\{\mathbf{H}_7^1(\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{H}_{13}^1(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. Identical to the proof of a previous claim, except that we set the challenge bit $b = 1$. In particular, one can show:

$$\mathbf{H}_7^1(1^\lambda) \approx_c \mathbf{H}_8^1 \approx_c \mathbf{H}_9^1(1^\lambda) \approx_c \mathbf{H}_{10}^1(1^\lambda) \equiv \mathbf{H}_{11}^1(1^\lambda) \equiv \mathbf{H}_{12}^1(1^\lambda) \approx_c \mathbf{H}_{13}^1(1^\lambda).$$

Claim. $\{\mathbf{H}_4^0(\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{H}_5^0(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. Similar to the proof of a previous claim and therefore omitted.

Claim. $\{\mathbf{H}_5^0(\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{H}_6^0(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. Similar to the proof of a previous claim and therefore omitted.

Claim. $\{\mathbf{H}_6^0(\lambda)\}_{\lambda\in\mathbb{N}} \equiv \{\mathbf{H}_{13}^1(\lambda)\}_{\lambda\in\mathbb{N}}$.

Proof. By leveraging the same argument used at the end of the proof of Lemma 1 and Lemma 2, we conclude that in both experiments $\mathbf{H}_6^0(\lambda)$ and $\mathbf{H}_{13}^1(\lambda)$ the challenge ciphertext $c^* = (c_1^*, c_3^*, c_3^*, c_4^*)$ is uniform in $\mathbb{G}_1 \times \mathbb{G}_2 \times \mathcal{S} \times \mathbb{G}_T$ to the eyes of the adversary. Hence, the two hybrid experiments are identically distributed. This concludes the proof.

Lemma 4. $\left|\mathbb{P}\left[\mathbf{G}_{\Pi,\mathsf{A}}^{\mathsf{ib\text{-}priv}^+}(\lambda) = 1 \middle| \mathbf{Mismatch}_4\right] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$.

Proof. The proof is symmetrical to that of Lemma 3, and therefore omitted.

Theorem 1 now follows by combining the above lemmas.

4.2 Adding Authenticity

We show how to add authenticity to any IB-ME scheme satisfying enhanced privacy. Without loss of generality, we assume that the encryption keys ek_σ of the underlying IB-ME are defined as in Construction 1.

Construction 2 *Let* $\Pi = (\mathsf{Setup}, \mathsf{SKGen}, \mathsf{RKGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an IB-ME with encryption keys* ek_σ *of the form* $\mathsf{ek}_\sigma = \sigma$, $\Pi' = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Ver})$ *be a signature scheme and* $\Pi'' = (\mathsf{I}, \mathsf{P}, \mathsf{V})$ *be a NIZK argument for the following NP relation:*

$$R = \left\{ ((\mathsf{mpk}, \mathsf{pk}, c), (\sigma, s)) \; : \; \begin{array}{c} \exists \mathsf{rcv}, m, r, \;\; s.t. \\ c = \mathsf{Enc}(\mathsf{mpk}, \sigma, \mathsf{rcv}, m; r) \wedge \mathsf{Ver}(\mathsf{pk}, \sigma, s) = 1 \end{array} \right\}.$$

Consider the following IB-ME $\Pi^* = (\mathsf{Setup}^*, \mathsf{SKGen}^*, \mathsf{RKGen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$.

$\mathsf{Setup}^*(1^\lambda)$**:** *Output* $\mathsf{msk}^* = (\mathsf{msk}, \mathsf{sk})$ *and* $\mathsf{mpk}^* = (\mathsf{mpk}, \omega, \mathsf{pk})$ *where* $(\mathsf{msk}, \mathsf{mpk}) \leftarrow\!\!\$\, \mathsf{Setup}(1^\lambda)$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\, \mathsf{KGen}(1^\lambda)$ *and* $\omega \leftarrow\!\!\$\, \mathsf{I}(1^\lambda)$.

$\mathsf{SKGen}^*(\mathsf{msk}, \sigma)$**:** *Upon input* $\mathsf{msk}^* = (\mathsf{msk}, \mathsf{sk})$ *and* $\sigma \in \{0,1\}^*$, *return* $\mathsf{ek}_\sigma = (s, \sigma)$ *where* $s = \mathsf{Sign}(\mathsf{sk}, \sigma)$.

$\mathsf{RKGen}^*(\mathsf{msk}, \rho)$**:** *Upon input* $\mathsf{msk}^* = (\mathsf{msk}, \mathsf{sk})$ *and* $\rho \in \{0,1\}^*$*, return* $\mathsf{dk}_\rho \leftarrow\$ \mathsf{RKGen}(\mathsf{msk}, \rho)$*.*

$\mathsf{Enc}^*(\mathsf{ek}_\sigma, \mathsf{rcv}, m)$**:** *Upon input* $\mathsf{ek}_\sigma = (s, \sigma)$*,* $\mathsf{rcv} \in \{0,1\}^*$*, and* $m \in \{0,1\}^*$*, output* $c^* = (c, \pi)$ *where* $c \leftarrow\$ \mathsf{Enc}(\sigma, \mathsf{rcv}, m)$ *and* $\pi \leftarrow\$ \mathsf{P}(\omega, (\mathsf{mpk}, \mathsf{pk}, c), (\sigma, s))$*.*

$\mathsf{Dec}^*(\mathsf{dk}_\rho, \mathsf{snd}, c)$**:** *Upon input* dk_ρ*,* $\mathsf{snd} \in \{0,1\}^*$*, and* $c^* = (c, \pi)$*, output* $m = \mathsf{Dec}(\mathsf{dk}_\rho, \mathsf{snd}, c)$ *if* $\mathsf{V}(\omega, (\mathsf{mpk}, \mathsf{pk}, c), \pi) = 1$*. Otherwise, return* $\perp$*.*

Correctness is immediate. As for security, we establish the following results.

Theorem 2. *If Π satisfies enhanced privacy and Π'' satisfies adaptive multi-theorem zero knowledge, then the IB-ME scheme Π^* from Construction 2 satisfies enhanced privacy.*

Proof. Consider the following hybrid experiments.

$\mathbf{H}_0(\lambda)$**:** This is identical to the experiment $\mathbf{G}^{\mathsf{ib\text{-}priv}^+}_{\Pi^*,\mathsf{A}^*}(\lambda)$.

$\mathbf{H}_1(\lambda)$**:** Same as $\mathbf{H}_0(\lambda)$ but now the challenger uses the simulator $\mathsf{Z} = (\mathsf{Z}_0, \mathsf{Z}_1)$ to generate the CRS and to compute the proofs. Formally, the challenger runs $(\omega, \zeta) \leftarrow\$ \mathsf{Z}_0(1^\lambda)$ at the beginning of the experiment; when the adversary outputs the challenge $(m_0, m_1, \mathsf{rcv}_0, \mathsf{rcv}_1, \mathbf{ID}_0, \mathbf{ID}_1)$, the challenger generates the ciphertext $c^* = (c, \pi)$, where $c \leftarrow\$ \mathsf{Enc}^*(\sigma_b, \mathsf{rcv}_b, m_b)$, $\sigma_b \leftarrow\$ \mathbf{ID}_b$, and $\pi \leftarrow\$ \mathsf{Z}_1(\zeta, (\mathsf{mpk}, \mathsf{pk}, c))$.

Claim. $\{\mathbf{H}_0(\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathbf{H}_1(\lambda)\}_{\lambda\in\mathbb{N}}$

Proof. The claim follows from the adaptive multi-theorem zero-knowledge property of the NIZK. The reduction is standard, and therefore omitted.

Claim. $|\Pr[\mathbf{H}_1(\lambda) = 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$.

Proof. The claim follows from the enhanced privacy property of the IB-ME. The reduction is standard, and therefore omitted.

By combining the above claims, Construction 2 satisfies enhanced privacy.

Theorem 3. *If Π' is EUF-CMA and Π'' satisfies knowledge soundness, then the IB-ME scheme Π^* from Construction 2 satisfies authenticity.*

Proof. Assume that Construction 2 does not satisfy authenticity, i.e., there exists a PPT attacker A^* that has a non negligible advantage in experiment $\mathbf{G}^{\mathsf{ib\text{-}auth}}_{\Pi^*,\mathsf{A}^*}(\lambda)$. We build an attacker A' that breaks the EUF-CMA security of the signature scheme Π'. Attacker A' proceeds as follows:

1. Upon receiving pk from the challenger, generate $(\mathsf{msk}, \mathsf{mpk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$, $(\omega, \xi) \leftarrow\$ \mathsf{K}_0(1^\lambda)$ and forward $\mathsf{mpk}^* = (\mathsf{mpk}, \mathsf{pk}, \omega)$ to A^*.
2. A' answers to the incoming queries as follows:
   - On input $\sigma \in \{0,1\}^*$ for $\mathsf{O}^*_1 = \mathsf{SKGen}^*$, forward the query σ to the signing oracle in order to obtain a valid signature s. Finally, return to A^* the key $\mathsf{ek}_\sigma = (s, \sigma)$.

- On input $\rho \in \{0,1\}^*$ for $\mathsf{O}_2^* = \mathsf{RKGen}^*$, return $\mathsf{dk}_\rho \leftarrow\!\!\$\ \mathsf{RKGen}(\mathsf{msk}, \rho)$ to A^*.

3. Upon receiving the forgery $(c^* = (c, \pi), \rho^*, \mathsf{snd}^*)$ check whether $\mathsf{V}(\omega, (\mathsf{mpk}, \mathsf{pk}, c), \pi) = 0$ or $\mathsf{Dec}(\mathsf{dk}_{\rho^*}, \mathsf{snd}^*, c) = \perp$ where $\mathsf{dk}_\rho^* \leftarrow\!\!\$\ \mathsf{RKGen}(\mathsf{msk}, \rho^*)$. If true, abort. Otherwise, extract $(s^*, \sigma^*) \leftarrow\!\!\$\ \mathsf{K}_1(\xi, (\mathsf{mpk}, \mathsf{pk}, c), \pi)$ and return (σ^*, s^*) as forgery to the challenger.

Except with negligible probability, the oracle queries of A^* are perfectly simulated by A'. This is because the CRS ω is computed via K_0 in the reduction, which yields a CRS that is computationally close to an honestly generated CRS. This means that with non-negligible probability the ciphertext $c^* = (c, \pi)$ returned by A^* as a forgery for snd^* is valid. Now, by knowledge soundness of the underlying NIZK proof, except with negligible probability, we must have that s^* is a valid signature for σ^* (note that $\sigma^* = \mathsf{snd}^*$) with respect to the public key pk sampled by the challenger. Furthermore, this is a valid forgery because A^* never queried O_1 on the identity σ^* which implies that A' has never asked for a signature of σ^* to the challenger. Hence, (σ^*, s^*) is a valid forgery for the EUF-CMA game. This concludes the proof.

Acknowledgements. The first author was partially supported by the Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM). The last author was supported by Sapienza University of Rome under the grant SPECTRA.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (Aug 2005)
2. Ateniese, G., Francati, D., Nuñez, D., Venturi, D.: Match me if you can: Matchmaking encryption and its applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 701–731. Springer, Heidelberg (Aug 2019)
3. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 621–630. ACM Press (May / Jun 2009)
4. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (May / Jun 2006)
5. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 89–98. ACM Press (Oct / Nov 2006), available as Cryptology ePrint Archive Report 2006/309
6. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (Aug 1984)

7. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (Mar 2011)