From Blind to Oblivious Identity-Based Encryption: A Generic Compiler and Instantiations

Olivier Blazy[1][0000−0001−6205−8249], Estelle Blin[1][0009−0005−2577−4613], and Sayantan Mukherjee[2][?]

[1] LIX, CNRS, Inria, École Polytechnique, Institut Polytechnique de Paris, Palaiseau, France
firstname.name@polytechnique.edu
[2] Indian Institute of Technology Jammu
csayantan.mukherjee@gmail.com

Abstract. Identity-Based Encryption (IBE) schemes were introduced to simplify public-key infrastructure by using any arbitrary strings as public keys. However, a longstanding criticism of IBE is the trade-off inherent in the "key escrow" problem: the design of IBE ensures that the authority possesses a master secret key that allows it to generate secret keys for any identity and, consequently, decrypt any ciphertext. While concepts such as Blind IBE and Accountable Authority IBE attempt to mitigate this trust assumption, they fall short of fully preventing a malicious authority from passively decrypting user traffic. A major improvement was proposed by Mitrokotsa *et al.* where they formalized a stronger notion called Oblivious IBE, where the authority cannot decrypt a ciphertext without brute-forcing the identity space. However, their construction inherently needs a composite group approach and offers no generic methodology. In this work, we present the first generic compiler that transforms any blind IBE into an oblivious IBE. Our transformation establishes a fundamental connection between blindness during key extraction and obliviousness during encryption. We prove that by combining a blind IBE with a hash function which takes input over the identity space, we can force the authority to search exhaustively for the recipient's identity to decrypt. To demonstrate the versatility and practical impact of our compiler, we propose two primary instantiations in the random oracle model: the first oblivious IBE in a prime order group and a post-quantum Oblivious IBE based on lattice assumptions (inspired by a variant of GPV). In addition, we make slight modifications to get our initial instantiation to function in the standard model.
Keywords: Identity-Based Encryption, Key escrow problem, Pairing-based, Post-quantum

1 Introduction

Identity-Based Encryption (IBE), first proposed by Shamir [27] and made practical through Boneh and Franklin [10], introduced a new paradigm in cryptography. Such a scheme allows a user's identity to be an arbitrary string (an email

address, a phone number,...), and as such removes the need to have a secure way to manage public-key certificates. However, this convenience comes at a heavy price: the authority holds a master secret key (msk) that allows it to compute secret keys for any identity; in other words, this creates a **Key Escrow** problem: the authority has built-in access to all the messages hidden in encrypted communications, meaning that users have to fully trust the authority not to abuse this power.

Various works have sought to reduce or limit this trust, via notions such as *blind IBE*, *anonymous IBE*, *accountable authority IBE*, or *IBE secure against the authority*.

Recently, Mitrokotsa *et al.*. [24] introduced *Oblivious IBE*, an IBE scheme secure against an adversarial authority, giving a formal definition of obliviousness, and constructing a scheme in the standard model but in a composite-order group that achieves this security against classical adversaries. Their work significantly advances the state of the art in reducing authority trust by ensuring that even if the authority tries to decrypt a given ciphertext, it cannot do so without generating a correct secret key for every identity and try them on the ciphertext. This provides a meaningful defense against mass surveillance by the authority. However, the state of the art is very limited, as only one construction currently exists, it heavily relies on composite order groups, and only handles classical adversaries. In addition, their construction is ad-hoc, meaning there is no proposed way to transform an IBE with relevant enough properties into an oblivious one.

In other words, this means that the work of [24] leaves open several key questions:

- **Generality of the definition and instantiations.** While they define obliviousness and give a specific instantiation, they do not provide a *generic* method for converting existing IBE schemes into oblivious IBE schemes.
- **Stronger security models.** While mentioned in the original paper, the initial proposal leaves to future works the notion of "strong OIBE". We show how our design allows to achieve it.
- **Prime-order group instantiations.** No construction is known in the prime-order group that achieves oblivious IBE.
- **Post-quantum security.** Ensuring obliviousness under quantum attacks remains unsettled.

1.1 Prior work and state of the art

The inherent key escrow problem in Identity-Based Encryption (IBE), where the authority holds absolute power to decrypt any ciphertext, has driven significant research into reducing trust in the authority. We categorize some of the relevant prior work below.

Blind IBE. Green and Hohenberger introduced blind IBE [19], where a user can obtain a secret key for an identity without the authority learning the queried

identity. For that the user key generation is superseded by an interactive protocol between the user and the authority, where the user manages to retrieve its secret key in an oblivious way. This already provides privacy of identity in the key extraction phase. However, this does not ensure that the authority does not possess a trapdoor to decrypt ciphertexts. Most of the time, this is trivially untrue, as the authority retains a master secret key.

Anonymous IBE. Anonymous IBEs were considered in [1], and further expanded in works such as Boyen and Waters [11] or Camenisch and Kohlweiss [12] by ensuring anonymity of ciphertexts and hiding the identity even in the presence of authority queries. It is crucial to prevent traffic analysis. However, like for Blind IBE, this does not prevent the authority from generating either a master decryption key able to decrypt any ciphertext without considering the identity, or targeting a given user whose key it generated before.

Accountable authority IBE / IBE secure against the authority. To mitigate the risk of key leakage, Emura, *et al.* [16] defined IBE secure against the authority by adding another authority that would certify users, giving instantiations under lattice assumptions. These limit what the authority can achieve in terms of key generation or decryption. However, it may not work if the two authorities manage to collude.

Oblivious IBE. Mitrokotsa *et al.*. [24] formalized the notion of obliviousness and strong-obliviousness. They gave a concrete construction in the composite group approach, secure in the weaker paradigm against classical adversaries. This ensures the authority cannot decrypt a ciphertext unless it has explicitly engaged in a key generation for the given identity, thereby forcing a malicious entity to brute-force over the identity space to find the right decryption key.

In summary, blind IBE protects user identity during the extraction phase, anonymous IBE ensures ciphertext anonymity, and accountable authority IBE limits the authority's power adding deterrence but not prevention. Oblivious IBE bridges all three techniques, but as of now it does not achieve all of: (i) a generic transformation to oblivious IBE, (ii) strong security in the standard model, and (iii) post-quantum security.

1.2 Our contributions

In this work, we make multi-fold improvements over the state of the art, particularly over [24]:

1. **Generic Compiler from Blind IBE.** Our first contribution is a generic compiler in the random oracle model (ROM) that transforms *any blind IBE scheme* into an *oblivious IBE scheme*. This shows a set of properties that suffices to obtain obliviousness in the sense of [24].

2. **Stronger security model.** We extend the approach of [24] to achieve the notion they call *strong oblivious* IBE. This notion allows the adversary to query several encryption challenges, and solve the one of its choice. We simply show that designing, through our framework, an oblivious IBE scheme whose security relies on a self random reducible hypothesis allows to achieve this notion at no extra cost.
3. **Prime order group instantiation.** Using our compiler, we provide an instantiation of oblivious IBE in the ROM. We also provide an explicit instantiation in the standard-model under well-studied hardness assumptions, eliminating the reliance on composite order groups.
4. **Post-quantum construction.** We further present a post-quantum construction of oblivious IBE in the (quantum) random-oracle model, based on quantum-resistant assumptions such as lattices. We do that simply by starting from a post-quantum IBE and leveraging a transformation using an oblivious transfer to achieve a blind IBE fitting our framework.

1.3 Technical intuition

The core intuition behind our compiler is simple yet powerful. We observe that in most classical IBE schemes, the master secret key allows the authority to compute a *trapdoor* for any identity (which is necessary to perform user key generation). This means, that to prevent the authority from using this trapdoor on a specific ciphertext, we must ensure that the decryption process requires knowledge that the authority lacks (*i.e.* the identity of the receiver).

To achieve this, we decide to blind the ciphertext using a hash function taking in parameter the recipient's identity. Assuming a ciphertext c being the encryption of a message m for an identity $\mathtt{id}$, the payload now roughly becomes $(c \oplus \mathsf{H}(\mathtt{id}, r), r)$, where r is some fresh random value.

1. The *User* knows their identity, so can directly compute $\mathsf{H}(\mathtt{id}, r)$, and so recover m.
2. The *Authority*, on the other hand, is stuck. Because we assume a blind scheme, it does not have a-priori knowledge of the user that could be targeted. Therefore, the authority is forced to iterate through all possible identities in the identity space, each time generating keys, and checking the hash function output. In this way, this severely hinders mass-surveillance.

Conceptually, this approach bootstraps the blindness property, which protects the user during key extraction via selective-failure blindness, into a comprehensive confidentiality guarantee that shields the encrypted message from the authority.

1.4 Organization

We begin in Section 2 by establishing the necessary mathematical notations, cryptographic protocols, and security assumptions used throughout this work. In

Section 3, we review the formal definitions of Identity-Based Encryption (IBE), detail the inherent key escrow problem, and formally define the notions of blind IBE and oblivious IBE. In Section 4, we present our primary contribution: a generic compiler from selective-failure blind IBE to oblivious IBE, complete with security proofs. Sections 5.1 and 5.2 highlight the versatility of our compiler by proposing an instantiation leading to a pairing-based scheme secure in the standard model, and a post-quantum lattice-based scheme in the random oracle model. Finally, we conclude with open problems in Section 6.

2 Preliminaries

Notations and conventions We denote our security parameter as λ. For all $n \in \mathbb{N}$, we denote by 1^n the n-bit string of ones. For any bit string x, we define x_i as the i^{th} most significant bit of x. For any finite set S, we denote by $x \xleftarrow{\$} S$ sampling x according to the uniform distribution on S. For any other distribution $\mathfrak{D}$ on S, we use $x \leftarrow^{\mathfrak{D}} S$ to indicate x is sampled from S according to the distribution $\mathfrak{D}$. For any algorithm A, we define $x \leftarrow A(a_1, \ldots, a_n)$ as the execution of A with inputs $a_1, \ldots, a_n$ and then assigning the output to x. We denote algorithms with access to an oracle O by A^O. For conciseness, we will write PPT for Probabilistic Polynomial Time. A function $f : \mathbb{N} \to \mathbb{R}$ is said to be negligible, if $\forall c \in \mathbb{N}, \exists k_0 \in \mathbb{N}, \forall k \geq k_0 : |f(k)| < k^{-c}$. A problem is said to be hard, if there exists no polynomial time algorithm solving it with non-negligible probability. The security properties of our cryptographic protocols will be described by experiments. For any PPT adversary $\mathcal{A}$ trying to break the security property, we define its advantage of doing so by the success' chance to win the experiment more than half the time. We use sequences of games [28] for our security proofs.

Zero-knowledge proof We start by recalling the formal definition of a Zero-Knowledge Proof of knowledge (ZKP) scheme [18].

Definition 1 (Zero-Knowledge Proof of Knowledge [18]). *A* ZKP *scheme consists of an interactive protocol between two PPT algorithms, a prover* $\mathcal{P}$*, and a verifier* $\mathcal{V}$*. At the end of the protocol,* $\mathcal{V}$ *either accepts or rejects.* $\mathcal{P}$ *will try to convince* $\mathcal{V}$ *that they know a secret* x *such that the public statement* $f(x)$ *is true. We will use the notation* $\mathsf{ZKP}(x : f(x))$ *to describe a run of this protocol. A* ZKP *scheme should respect the three following properties.*

- *Completeness: If* x *is such that* $f(x)$ *is true and if* $\mathcal{P}$ *and* $\mathcal{V}$ *follow the protocol honestly, then* $\mathcal{V}$ *rejects with negligible probability.*
- *Soundness:* $\mathcal{V}$ *accepts with negligible probability if* $\mathcal{P}$ *is malicious and* x *does not satisfy the relation* $f(x)$*.*
- *Honest-Verifier Zero-Knowledge: There exists a PPT simulator* $\mathcal{S}$*, without the knowledge of* x*, whose output transcript is indistinguishable from the transcript of a* ZKP *protocol between an honest* $\mathcal{P}$ *and an honest* $\mathcal{V}$*.*

Intuitively, the idea of the zero-knowledge property is that the verifier $\mathcal{V}$ gains no information about the secret x beyond the validity of the statement.

Security assumptions Given the three cyclic multiplicative groups of prime order p: $\mathbb{G}, \hat{\mathbb{G}}$ and $\mathbb{G}_T$. Let g and $\hat{g}$ be generators of $\mathbb{G}, \hat{\mathbb{G}}$. The map $e : \mathbb{G} \times \hat{\mathbb{G}} \to \mathbb{G}_T$ is a non-degenerate bilinear pairing if:

- $\forall h \in \mathbb{G}, \hat{h} \in \hat{\mathbb{G}}$, and $\forall \alpha, \beta \in \mathbb{Z}_p : e(h^\alpha, \hat{h}^\beta) = e(h, \hat{h})^{\alpha\beta}$. (bilinear)
- $e(h, \hat{h}) = 1$ if and only if $h = 1$ or $\hat{h} = 1$. (non-degenerate)

Definition 2 (Decisional Diffie Hellman (DDH [9])). *The Decisional Diffie-Hellman hypothesis in a multiplicative group $(p, \mathbb{G}, g)$, states that given $(g^\alpha, g^\beta, g^\psi)$ for unknown $\alpha, \beta \xleftarrow{\$} \mathbb{Z}_p$, it is hard to decide whether $\psi = \alpha\beta$.*

Definition 3 (Symmetric External Diffie Hellman (SXDH [5])). *This variant of the DDH assumption states that DDH is hard in both $\mathbb{G}$ and $\hat{\mathbb{G}}$.*

3 Identity-Based Encryption scheme and the key escrow problem

In this section, we recall the standard security definitions of an Identity-Based Encryption ($\mathcal{IBE}$) scheme, and use that to highlight the inherent key escrow problem associated with $\mathcal{IBE}$; we need to overly trust the authority not to be malicious. We will show that the security requirements of the standard definition are insufficient against this issue, and explain why oblivious IBE were introduced.

3.1 Identity-Based Encryption schemes

An Identity-Based Encryption scheme is a cryptographic protocol between a central authority and multiple users. It is an alternative for the public key infrastructure, since to encrypt a message, one can use the identity of the receiver, so there is no need for further authentication which can be costly.

The authority starts by generating a pair of master public and secret keys $(\mathsf{mpk}, \mathsf{msk})$. Each user has a proper and unique identity $\mathtt{id}$ which lies in some identity space $\mathcal{ID}$ of size 2^N. This identity $\mathtt{id}$ will be used to encrypt a message m from the plaintext space $\mathcal{M}$ into a ciphertext c from some ciphertext space $\mathcal{C}$. The only way to decrypt c would be by using a secret key $\mathsf{sk}_{\mathtt{id}}$ that initially only the authority can compute using msk and $\mathtt{id}$, and then sent to the rightful user.

We start by recalling the formal structure of Identity-Based Encryption scheme [27].

Definition 4 (Identity-Based Encryption scheme). *Given $\mathcal{ID}, \mathcal{M}, \mathcal{C}$, the sets denoting the identity space, the plaintext space and the ciphertext space. An $\mathcal{IBE}$ scheme consists of four PPT algorithms* $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}$ *with the following properties:*

- $\mathsf{Setup}(1^\lambda)$*: The probabilistic key generation algorithm returns the master public/secret keys* $(\mathsf{mpk}, \mathsf{msk})$*.*
- $\mathsf{KeyGen}(\mathsf{msk}, \mathtt{id})$*: The probabilistic user's secret keys generation algorithm returns a secret key* $\mathsf{sk}_{\mathtt{id}}$ *with respect to identity* $\mathtt{id} \in \mathcal{ID}$*.*
- $\mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}, m)$*: The probabilistic encryption algorithm encrypts a message* $m \in \mathcal{M}$ *into a ciphertext* $c \in \mathcal{C}$ *with respect to* $\mathtt{id} \in \mathcal{ID}$*.*
- $\mathsf{Decrypt}(\mathsf{sk}_{\mathtt{id}}, \mathtt{id}, c)$*: The probabilistic decryption algorithm returns either a decryption* $m' \in \mathcal{M}$ *or the reject symbol* $\perp$*.*

The first essential property of an $\mathcal{IBE}$ is *correctness*: it should ensure that if a ciphertext c of m encrypted with $\mathtt{id}$ is decrypted with a well-generated secret key $\mathsf{sk}_{\mathtt{id}}$ for $\mathtt{id}$, then the receiver recovers m. More formally, it is given by the following definition.

Definition 5 (Correctness). *For all security parameter* λ*, all pairs* $(\mathsf{mpk}, \mathsf{msk})$ *generated by* $\mathsf{Setup}(1^\lambda)$*, all* $\mathtt{id} \in \mathcal{ID}$*, all* $m \in \mathcal{M}$*:*

$$\Pr[\mathsf{Decrypt}(\mathsf{sk}_{\mathtt{id}}, \mathtt{id}, c) \neq m] = \mathsf{negl}(\lambda)$$

where $\mathsf{sk}_{\mathtt{id}}$ *is generated by* $\mathsf{KeyGen}(\mathsf{msk}, \mathtt{id})$*, and* $c \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}, m)$*.*

Security against users. The second property should ensure that if a malicious user does not have the secret key $\mathsf{sk}_{\mathtt{id}}$ then he cannot recover a message encrypted for the identity $\mathtt{id}$. We define more formally *indistinguishability against chosen plaintext attacks* (IND-ID-CPA) through Definition 6 which involves the security game $\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{IND\text{-}ID\text{-}CPA}}$. Roughly, the idea of this security game is that the adversary cannot distinguish between the encryptions of the challenged message m_0^* and the challenged message m_1^*, for an identity challenge $\mathtt{id}^*$.

Definition 6 (Indistinguishability against chosen plaintext attacks). *An* $\mathcal{IBE}$ *is* IND-ID-CPA*-secure if for all PPT* $\mathcal{A}$*, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{IND\text{-}ID\text{-}CPA}}_{\mathcal{IBE}}(\mathcal{A}) := \left| \Pr[\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{IND\text{-}ID\text{-}CPA}}(\mathcal{A}, \lambda) = 1] - \frac{1}{2} \right|$$

where $\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{IND\text{-}ID\text{-}CPA}}(\mathcal{A}, \lambda)$ *is defined as:*

$\underline{\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{IND\text{-}ID\text{-}CPA}}(\mathcal{A}, \lambda)}$*:*	$\underline{\mathsf{O}_{\mathsf{KG}}(\mathtt{id})}$*:*
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	$\mathsf{IDList} \leftarrow \mathsf{IDList} \cup \{\mathtt{id}\}$
$\mathsf{IDList} \leftarrow \emptyset$	$\mathsf{sk}_{\mathtt{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathtt{id})$
$(\mathtt{id}^*, m_0^*, m_1^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{KG}}(\cdot), \mathsf{O}_{\mathsf{ENC}}(\cdot)}(\mathsf{mpk})$	
$b \xleftarrow{\$} \{0, 1\}$	
$c^* \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}^*, m_b^*)$	
$b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{KG}}(\cdot), \mathsf{O}_{\mathsf{ENC}}(\cdot)}(\mathsf{mpk}, c^*)$	
Return $b' = b \wedge \mathtt{id}^* \notin \mathsf{IDList}$	

The third property should ensure that a malicious user should not be able to guess which identity was used during the encryption of a plaintext. We define more formally *anonymity against chosen plaintext attacks* (ANON-ID-CPA) through Definition 7 which involves the security game $\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{ANON\text{-}ID\text{-}CPA}}$. Roughly, the idea of this security game is that the adversary cannot distinguish which challenged identities $\mathtt{id}^*_0, \mathtt{id}^*_1$ was used to encrypt the challenged plaintext m^*.

Definition 7 (Anonymity against chosen plaintext attacks). *An $\mathcal{IBE}$ is* ANON-ID-CPA*-secure if for all PPT $\mathcal{A}$, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{ANON\text{-}ID\text{-}CPA}}_{\mathcal{IBE}}(\mathcal{A}) := |\Pr[\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{ANON\text{-}ID\text{-}CPA}}(\mathcal{A},\lambda)=1] - \frac{1}{2}|$$

where $\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{ANON\text{-}ID\text{-}CPA}}(\mathcal{A},\lambda)$ is defined as:

$\underline{\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{ANON\text{-}ID\text{-}CPA}}(\mathcal{A},\lambda)}$:	$\underline{\mathsf{O}_{\mathsf{KG}}(\mathtt{id})}$:
$(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	$\mathsf{IDList} \leftarrow \mathsf{IDList} \cup \{\mathtt{id}\}$
$\mathsf{IDList} \leftarrow \emptyset$	$\mathsf{sk}_{\mathtt{id}} \leftarrow \mathsf{KeyGen}(\mathsf{msk},\mathtt{id})$
$(\mathtt{id}^*_0,\mathtt{id}^*_1,m^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{KG}}(.),\mathsf{O}_{\mathsf{ENC}}(.)}(\mathsf{mpk})$	
$b \xleftarrow{\$} \{0,1\}$	
$c^* \leftarrow \mathsf{Encrypt}(\mathsf{mpk},\mathtt{id}^*_b,m^*)$	
$b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{KG}}(.),\mathsf{O}_{\mathsf{ENC}}(.)}(\mathsf{mpk},c^*)$	
Return $b' = b \wedge \mathtt{id}^*_0, \mathtt{id}^*_1 \notin \mathsf{IDList}$	

Security issue of $\mathcal{IBE}$ against authority. Definitions 6 and 7 guarantee us that a potentially malicious user cannot obtain information about another users' encrypted data, thus ensuring confidentiality. However, it is important to note that since the authority has in its possession the master secret key msk, it can build the secret key for all identities and then decrypt each ciphertext. Standard $\mathcal{IBE}$ and their security requirements do not prevent the authority from being malicious, this is named *the key escrow problem*. To prevent this issue, [19] introduced some security definitions. Roughly, the idea would be to do the KeyGen algorithm while blinding the identity of the user towards the authority.

3.2 Blind Identity-Based Encryption schemes

We now introduce a new security definition, where the main goal will be to adapt the standard $\mathcal{IBE}$ definition to the case where an authority could be potentially malicious. We will have an interactive protocol BlindKeyGen between the authority and the user instead of the KeyGen function. During the protocol, the user part will take as input an identity id, while the authority part will take as input the master secret key msk.

Definition 8 (Blind $\mathcal{IBE}$). *A* blind *$\mathcal{IBE}$ scheme consists of three PPT algorithms* (Setup, Encrypt, Decrypt) *defined the same was as in the original $\mathcal{IBE}$ definition and an interactive protocol* BlindKeyGen *between two PPT algorithms* Aut *and* User*.*

– $\mathsf{Aut}(\mathsf{msk})$*: This algorithm helps to generate a possible secret key* $\mathsf{sk}_{\mathtt{id}}$ *for the* User *algorithm, it returns nothing.*

– $\mathsf{User}(\mathtt{id})$*: This probabilistic algorithm returns a valid secret key* $\mathsf{sk}_{\mathtt{id}}$ *or the reject symbol* $\perp$*.*

The notation $\mathsf{BlindKeyGen}(\langle \mathsf{Aut}(\mathsf{msk}), \mathsf{User}(\mathtt{id})\rangle)$ *will stand for a run of the protocol between two honest parties using the algorithms* $\mathsf{Aut}(.)$ *and* $\mathsf{User}(.)$*, whose output is the one of* $\mathsf{User}(\mathtt{id})$*.*

Remark 1. To represent a run of the protocol between two PPT $\mathcal{A}$ and $\mathcal{B}$, we will use the notation $\mathsf{BlindKeyGen}(\langle \mathcal{A}(\mathsf{msk}), \mathcal{B}(\mathtt{id})\rangle)$. $\mathcal{A}, \mathcal{B}$ are not necessarily honest and may not follow the algorithms $\mathsf{Aut}(.), \mathsf{User}(.)$ to the letter.

The correctness of a blind $\mathcal{IBE}$ is defined similarly as in Definition 5, the only difference is that we replace the secret key $\mathsf{sk}_{\mathtt{id}}$ generated by KeyGen with an $\mathsf{sk}_{\mathtt{id}}$ generated by the $\mathsf{BlindKeyGen}$ protocol with an honest authority. The first formalized security of blind $\mathcal{IBE}$ can be found in [19], it is based on two properties the leak-freeness and selective-failure blindness that we will recall in the following parts.

Leak-free key generation protocol
We need the $\mathsf{BlindKeyGen}$ protocol to respect the property of leak-freeness (LF) to ensure a malicious user can't take advantage of the $\mathsf{BlindKeyGen}$ protocol to recover more information about msk or any other secret key than the one he queried $\mathsf{sk}_{\mathtt{id}}$ with the identity $\mathtt{id}$. We define more formally the *leak-freeness* (LF) property through Definition 9 which involves the security game $\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{LF}}(\mathcal{A}, \lambda)$. Roughly, the idea is that the adversary will have to distinguish between a random ciphertext from the ciphertext space and the encryption of the challenge m^* with the identity challenge $\mathtt{id}^*$ for which it did not query a secret key.

Definition 9 (Leak-freeness). *An* $\mathcal{IBE}$ *is* LF*-secure if for all PPT* $\mathcal{A}$*, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{LF}}_{\mathcal{IBE}}(\mathcal{A}) := \left|\Pr[\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{LF}}(\mathcal{A}, \lambda) = 1] - \frac{1}{2}\right|$$

where $\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{LF}}(\mathcal{A}, \lambda)$ *is defined as:*

$\underline{\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{LF}}(\mathcal{A}, \lambda)}$:	$\underline{\mathsf{O}_{\mathsf{BKG}}(\mathtt{id})}$:
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	$\mathsf{IDList}[i] \leftarrow \mathtt{id}, i \leftarrow i + 1$
$\mathsf{IDList} \leftarrow \emptyset, i \leftarrow 0$	*Execute* $\mathsf{BlindKeyGen}(\langle \mathsf{Aut}(\mathsf{msk}), \mathcal{A}(\mathtt{id})\rangle)$
$(\mathtt{id}^*, m^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{BKG}}(.)}(\mathsf{mpk})$	
$b \xleftarrow{\$} \{0, 1\}$	
If $b = 0, c^* \xleftarrow{\$} \mathcal{C}$	
if $b = 1, c^* \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}^*, m^*)$	
$b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{BKG}}(.)}(\mathsf{mpk}, c^*, m^*)$	
Return $b' = b \wedge \mathtt{id}^* \notin \mathsf{IDList}$	

Remark 2. We can observe that the leak-freeness property implies the IND-ID-CPA and the ANON-ID-CPA property of the $\mathcal{IBE}$ scheme since we can simulate the O_{KG} oracle with an $\mathsf{O}_{\mathsf{BKG}}$ oracle.

Due to lack of space, we won't recall the original definition of [19]. We will explain in more detail how the former definition together with the properties of IND-ID-CPA and ANON-ID-CPA are linked to our definition in the extended version of the paper.

Selective-failure blind key generation protocol
Moreover, we need the BlindKeyGen protocol to respect the property of selective-failure blindness (SFB). A potentially malicious authority cannot learn anything about the user's identity during the BlindKeyGen protocol. Also, the authority cannot cause the BlindKeyGen protocol to fail in a manner dependent on the user's identity. The first formalized security of such scheme can be found in [19]. Roughly, the idea is that the adversary will receive two BlindKeyGen queries for two challenge identities $\mathtt{id}_0^*, \mathtt{id}_1^*$ and will have to answer the protocol from the authority point of view (not necessarily in an honest way). It will then receive a proof that the secret keys $\mathsf{sk}_{\mathtt{id}_0^*}, \mathsf{sk}_{\mathtt{id}_1^*}$ were correctly generated (or not). Afterwards the adversary should not be able to distinguish in which order the identities were queried. We define more formally the security game in Figure 1.

Definition 10 (Selective-failure blindness). *For a given* $\mathtt{id}$*, let* $\mathsf{SK}_{\mathtt{id}}$ *be the sets of all correct* $\mathsf{sk}_{\mathtt{id}}$*. An* $\mathcal{IBE}$ *is* SFB*-secure if for all PPT* $\mathcal{A}$*, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{SFB}}_{\mathcal{IBE}}(\mathcal{A}) := \left|\Pr[\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{SFB}}(\mathcal{A},\lambda)=1] - \frac{1}{2}\right|$$

where $\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{SFB}}(\mathcal{A},\lambda)$ *is defined as in Figure 1.*

We slightly weakened the security game of selective-failure blindness defined in the original paper of [19]. We replaced the disclosure of the two generated secret key in the challenge, by a zero-knowledge proof of knowledge (defined above in Definition 1) that in the experiment the computation of both secret key went correctly thanks to the BlindKeyGen protocol.

We stress that a scheme achieving classical selective-failure blindness also fulfills our variant as *less* information is provided to the adversary, but our variant better captures what classically appears in a real scenario as user do not explicitly leak their secret key after BlindKeyGen, but just the fact that they can decrypt (and as such the knowledge of their key).

But is it enough? As shown in [24], a lot of schemes which are proven blind still suffer from a severe drawback. We recall the attack on every ciphertext that Mitrokotsa, *et al.* made in [24] about the blind version, proposed by [19], of the scheme of Waters [29], which was proven selective-failure blind, as an example. The scheme of [29] is as follows.

Let $\mathbb{G}, \mathbb{G}_T$ be groups of prime order p, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ a bilinear non-degenerate pairing. For simplicity, let $U(\mathtt{id}) = \prod_{i=0}^{n-1} u_i^{\mathtt{id}_i}$.

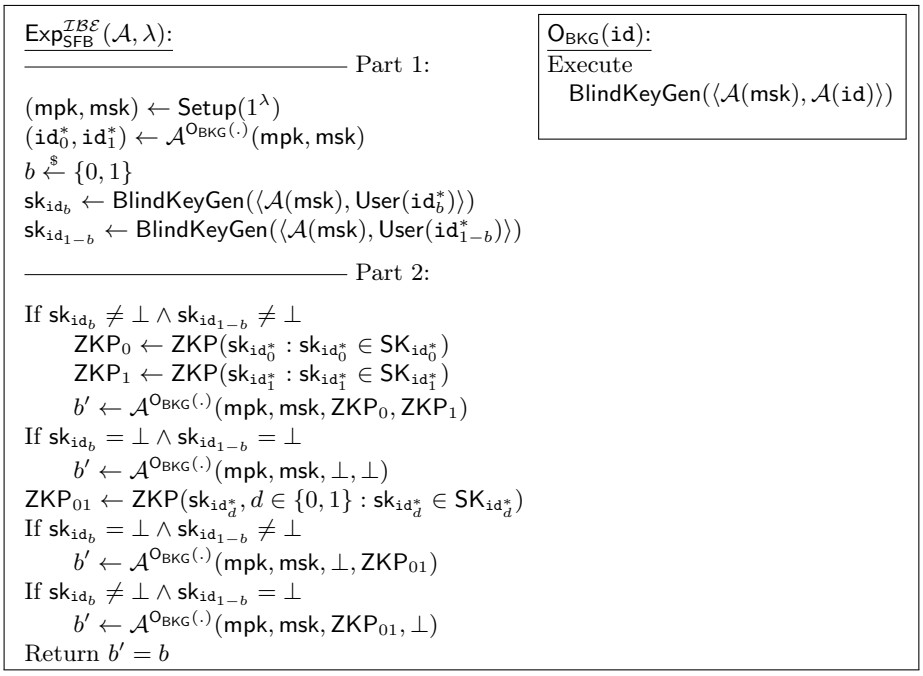


Fig. 1: SFB experiment.

- Let the master public key be $\mathsf{mpk} = (g, g_1 = g^\alpha, g_2, u', (u_i)_{0 \leq i \leq n-1})$, where each element lies in $\mathbb{G}$, and α in $\mathbb{Z}_p$.
- Let the master secret key be $\mathsf{msk} = g_2^\alpha$.
- For a plaintext $m \in \mathbb{G}_T$, its encryption for an identity $\mathtt{id}$ is $c = (c_0, c_1, c_2)$ with $c_0 = e(g_1, g_2)^t \cdot m$, $c_1 = g^t$, $c_2 = (u' \cdot h(\mathtt{id}))^t$, with $t \in \mathbb{Z}_p$.

$$e(c_1, \mathsf{msk}) = e(g^t, g_2^\alpha) = e(g^\alpha, g_2)^t = e(g_1, g_2)^t \tag{1}$$

Given Equation 1, it is evident that the authority can decrypt any given ciphertext without having to know the target identity. We can also see that during the encryption, the master secret key g_2^α is algebraically reconstructed since $e(g_1, g_2)^t = e(g, g_2^\alpha)^t$. A lot of schemes based on pairings have a similar drawback.

Despite this drawback, Definition 10 is still necessary since the authority will not learn which user already queried their secret key. But even if the scheme respects this new security requirement, the authority still has a trapdoor which enables them to decrypt *every* ciphertext. This is exactly the property exploited by [24] against Water's scheme.

3.3 Oblivious Identity-Based Encryption

To fix this issue, Mitrokotsa, *et al.* [24] proposed a new property, which they called strong-obliviousness, we will prefer to call it *obliviousness*. In an oblivi-

ous $\mathcal{IBE}$ scheme, we require that the authority should not be able to decrypt a ciphertext more efficiently than brute-forcing over the identity space $\mathcal{ID}$ (*i.e.* generating a possible secret key $\mathsf{sk}_{\mathtt{id}}$ for each $\mathtt{id} \in \mathcal{ID}$). In other words, we should have an exponentially big identity space together with the following requirements.

(i) The $\mathcal{IBE}$ scheme should be selective-failure blind, so that the authority does not learn which identity $\mathtt{id}$ already queried a secret key $\mathsf{sk}_{\mathtt{id}}$. Therefore, it will not be able to brute-force on a smaller set of identity.

(ii) The $\mathcal{IBE}$ scheme should generate anonymous ciphertext for the authority, if it was not the case, then given a ciphertext, the authority would recover the identity $\mathtt{id}$, and be able to directly generate the good user secret key $\mathsf{sk}_{\mathtt{id}}$.

The main goal is to restrict the authority's ability to distinguish between a ciphertext encrypted with a random identity and a word picked randomly in the ciphertext space. We define more formally the obliviousness of an $\mathcal{IBE}$ via the game given in Definition 11. Roughly, the idea of this security game is that the adversary cannot distinguish which identity $\mathtt{id}^*$ was used to encrypt the challenged plaintext m^*, even if this identity was used in a $\mathsf{BlindKeyGen}$ protocol to query sk_{id^*}. We simulate this condition by drawing random identities $\mathtt{id}$, storing them in a list IDList and executing the $\mathsf{BlindKeyGen}$ protocol with them; the challenge identity $\mathtt{id}^*$ is then randomly drawn from IDList.

Definition 11 (Oblivious $\mathcal{IBE}$). *An $\mathcal{IBE}$ is oblivious if for all PPT $\mathcal{A}$, the following advantage is negligible:*

$$\mathsf{Adv}^{\mathsf{OBLI}}_{\mathcal{IBE}}(\mathcal{A}) := \left|\Pr[\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{OBLI}}(\mathcal{A},\lambda)=1] - \frac{1}{2}\right|$$

where $\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{OBLI}}(\mathcal{A},\lambda)$ is defined as:

$\underline{\mathsf{Exp}^{\mathcal{IBE}}_{\mathsf{OBLI}}(\mathcal{A},\lambda)\text{:}}$	$\underline{\mathsf{O}_{\mathsf{BKG}}()\text{:}}$
$(\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	$\mathtt{id} \xleftarrow{\$} \mathcal{ID}, \mathsf{IDList}[i] \leftarrow \mathtt{id}, i \leftarrow i+1$
$\mathsf{IDList} \leftarrow \emptyset$	*Execute* $\mathsf{BlindKeyGen}(\langle \mathcal{A}(\mathsf{msk}), \mathsf{User}(\mathtt{id})\rangle)$
$(m^*, j^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{BKG}}(.),\mathsf{O}_{\mathsf{ENC}}(.)}(\mathsf{mpk},\mathsf{msk})$	$\underline{\mathsf{O}_{\mathsf{ENC}}(j \in [i], m)\text{:}}$
$b \xleftarrow{\$} \{0,1\}$	$\mathtt{id} \leftarrow \mathsf{IDList}[j]$
$\mathtt{id}^* \leftarrow \mathsf{IDList}[j^*]$	$c \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, id, m)$
If $b=0, c^* \xleftarrow{\$} \mathcal{C}$	*Return* c
if $b=1, c^* \leftarrow \mathsf{Encrypt}(\mathsf{mpk},\mathtt{id}^*,m^*)$	
$b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{BKG}}(.),\mathsf{O}_{\mathsf{ENC}}(.)}(\mathsf{mpk},\mathsf{msk},c^*,m^*)$	
Return $b'=b$	

After proposing this new and more complete definition, Mitrokotsa, *et al.* [24] also proposed the first (and only to our knowledge) pairing-based oblivious $\mathcal{IBE}$ scheme in the standard model. It is based on Wee's IBE's construction [30] and Dodis-Yampolski's verifiable random function [14]. This raises the following question:

Is there a generic transformation from a selective-failure blind IBE scheme (as per Definition 10) to an oblivious IBE scheme (as per Definition 11)?

4 Generic compiler from blind IBE to oblivious IBE

In this section, we will present a generic compiler from selective-failure blind to oblivious IBE by using a hash function, the obliviousness of the resulting scheme will reduce to the security of the selective-failure blindness of the formal $\mathcal{IBE}$ and of the hash function in the Random Oracle Model ROM. We first present our generic transformation and afterwards we will prove its security against malicious users and its security against a malicious authority.

How will we use the hash function in our generic transformation? We can use a hash function taking in parameter the identity $\mathtt{id}$ of the recipient and a random element to mask the ciphertext. This will ensure that if a potential malicious authority does not know the identity of the recipient of the ciphertext, it will have to brute force over the identity space to be able to find the hash of the identity and decrypt the ciphertext.

4.1 A generic compiler

To summarize, we want to use a hash function to forbid the algebraic reconstitution of msk in the ciphertext as described above in Equation 1. Let us construct a new $\mathcal{IBE}$ scheme from a selective-failure blind $\mathcal{IBE}$ scheme Π. We will keep the same function Setup and protocol $\mathsf{BlindKeyGen}$ from Π. For the $\mathsf{Encrypt}$ function, we will use a Hash function $\mathsf{H} : \mathcal{ID} \times \{0,1\}^\lambda \to \mathcal{C}$. The goal will be to randomize the original ciphertext with $H(\mathtt{id}, r)$, $\mathtt{id}$ being the recipient's identity and r a random element in $\{0,1\}^\lambda$. The receiver of a given ciphertext knows its own identity, hence they are able to compute $H(\mathtt{id}, r)$, for all other parties, $H(\mathtt{id}, r)$ should look random since they do not know $\mathtt{id}$. We give more formally our generic compiler in the following theorem.

Theorem 1. *Let Π be a SFB-$\mathcal{IBE}$ scheme, and $\mathsf{H} : \mathcal{ID} \times \{0,1\}^\lambda \to \mathcal{C}$ be a secure hash function. The $\mathcal{IBE}$ obtained by keeping the same $\Pi.\mathsf{Setup}()$ function and $\Pi.\mathsf{BlindKeyGen}()$ protocol, and after applying the transformation described in Figure 2 on $\Pi.\mathsf{Encrypt}(), \Pi.\mathsf{Decrypt}()$ is* oblivious.

$\underline{\mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}, m)}$:	$\underline{\mathsf{Decrypt}(\mathsf{sk}_{\mathtt{id}}, c')}$:
$c \leftarrow \Pi.\mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}, m)$	Parse c' into (c, r)
$r \xleftarrow{\$} \{0,1\}^\lambda$	$m \leftarrow \Pi.\mathsf{Decrypt}(\mathsf{sk}_{\mathtt{id}}, c \oplus \mathsf{H}(\mathtt{id}, r))$
Return $(c \oplus \mathsf{H}(\mathtt{id}, r), r)$	Return m

Fig. 2: The description of our generic transformation on an $\mathcal{IBE}$ Π.

For any $\mathcal{IBE}$ Π, we will now denote $\mathsf{OBLI}(\Pi)$ the $\mathcal{IBE}$ resulting after applying the transformation described in Figure 2 on Π, and after adding $F_{\mathcal{ID}}$ in the master public key.

Remark 3. Correctness. If Π is a correct $\mathcal{IBE}$, then $\mathsf{OBLI}(\Pi)$ is correct, since H is a deterministic function.

4.2 Security proofs

In this section, we will first prove the security against malicious users of our transformation, with the leak-free (LF) property defined in Definition 9. Afterwards, we will prove the security against a potential malicious authority with the obliviousness property defined in Definition 11.

Security against users. We can observe that for the leak-free (LF) property, the security of our transformation can be reduced to the security of the original $\mathcal{IBE}$ Π. We recall that the leak-free property as we defined it implies the (IND-ID-CPA) property and the (ANON-ID-CPA) property.

Intuitively, the idea is that a PPT adversary $\mathcal{A}_\Pi$ against the underlying $\mathcal{IBE}$ scheme Π can distinguish between a random ciphertext from the encryption of the challenge message $m*$ for an identity challenge $\mathtt{id}^*$ by using an adversary $\mathcal{A}_{\mathsf{OBLI}(\Pi)}$ against $\mathsf{OBLI}(\Pi)$. Considering that $\mathcal{A}_\Pi$ can simulate the experiment $\mathsf{Exp}^{\mathsf{OBLI}(\Pi)}_{\mathsf{IND\text{-}ID\text{-}CPA}}(\mathcal{A}_{\mathsf{OBLI}(\Pi)}, \lambda)$ and send the modified ciphertext in our transformation, $(c^* \oplus \mathsf{H}(\mathtt{id}^*, r^*), r^*)$ as the challenge to $\mathcal{A}_{\mathsf{OBLI}(\Pi)}$ and finally output the answer of $\mathcal{A}_{\mathsf{OBLI}(\Pi)}$.

Theorem 2. *Let $\mathcal{A}_{\mathsf{OBLI}(\Pi)}$ be any PPT adversary against the $\mathcal{IBE}$ $\mathsf{OBLI}(\Pi)$. There exists a PPT adversary $\mathcal{A}_\Pi$ against the original $\mathcal{IBE}$ scheme Π such that:*

$$\mathsf{Adv}^{\mathsf{LF}}_{\mathsf{OBLI}(\Pi)}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{LF}}_{\Pi}(\mathcal{A}_\Pi) \ .$$

Proof. Let's describe the behavior of an adversary $\mathcal{A}_\Pi$ during the experiment $\mathsf{Exp}^{\Pi}_{\mathsf{LF}}(\mathcal{A}_\Pi, \lambda)$. $\mathcal{A}_\Pi$ runs $\mathsf{Exp}^{\mathsf{OBLI}(\Pi)}_{\mathsf{LF}}(\mathcal{A}_{\mathsf{OBLI}(\Pi)}, \lambda)$, by giving the same mpk it received in input to $\mathcal{A}_{\mathsf{OBLI}(\Pi)}$. It simulates the oracle queries to $\mathsf{O}_{\mathsf{BKG}}(.)$, by just forwarding the queries to its own oracle. Afterwards, $\mathcal{A}_\Pi$ outputs the same challenge query $(\mathtt{id}^*, m^*)$ it receives from $\mathcal{A}_{\mathsf{OBLI}(\Pi)}$. Further on, $\mathcal{A}_\Pi$ sends $(c^* \oplus \mathsf{H}(\mathtt{id}^*, r^*), r^*)$ to $\mathcal{A}_{\mathsf{OBLI}(\Pi)}$, with c^* being the challenge ciphertext it received, and r^* being randomly drawn from $\{0,1\}^\lambda$. $\mathcal{A}_\Pi$ outputs whatever $\mathcal{A}_{\mathsf{OBLI}(\Pi)}$ outputs. We have that $\mathsf{Exp}^{\Pi}_{\mathsf{LF}}(\mathcal{A}_\Pi, \lambda) = \mathsf{Exp}^{\mathsf{OBLI}(\Pi)}_{\mathsf{LF}}(\mathcal{A}_{\mathsf{OBLI}(\Pi)}, \lambda)$. □

Security against authority. We will show that our generic transformation is oblivious, if the $\mathcal{IBE}$ scheme we used is selective-failure blind as defined in Definition 10, the number of oracle queries to $\mathsf{O}_{\mathsf{BKG}}$ is polynomially bounded, the number of oracle queries to $\mathsf{O}_{\mathsf{ENC}}$ is polynomially bounded and if the used hash function is secure.

Theorem 3. *Let H be a hash function modeled as a ROM and $\mathcal{L}_{\mathsf{H}}$ be the hash list containing the tuples $((\mathtt{id}, r), \mathsf{H}(\mathtt{id}, r))$. Let $\mathcal{A}$ be a PPT adversary against the $\mathcal{IBE}$ $\mathsf{OBLI}(\Pi)$, $\mathcal{A}_\Pi$ be a PPT adversary against the original $\mathcal{IBE}$ scheme Π and $\mathcal{A}_{\mathsf{PRF}}$ be a PPT adversary against the PRF. Let q_{enc} (resp. q_{bkg}) be the number of queries to the oracle $\mathsf{O}_{\mathsf{ENC}}$ (resp $\mathsf{O}_{\mathsf{BKG}}$) that $\mathcal{A}$ is doing.*

$$\mathsf{Adv}^{\mathsf{OBLI}}_{\mathsf{OBLI}(\Pi)}(\mathcal{A}) \leq 2 \times q_{\mathsf{bkg}} \times \mathsf{Adv}^{\mathsf{SFB}}_{\Pi}(\mathcal{A}_\Pi) + \frac{q_{\mathsf{enc}}}{2^\lambda} + \mathsf{Adv}^{\mathsf{PRF}}_{F_{\mathcal{ID}}}(\mathcal{A}_{\mathsf{PRF}}) \ .$$

The main idea of the proof is that the oracle $\mathsf{O}_{\mathsf{BKG}}$ is not giving any information to the adversary about which $\mathtt{id}$ was queried during the $\mathsf{BlindKeyGen}$ protocol. Considering the adversary does not know which identity was queried and would be used in the challenge, it cannot distinguish between the hash of $\mathtt{id}^*$ and a random value r^* from a random element h^*. We will use an intermediate lemma to prove the reduction between G_0 and G_1.

Proof. For clarity, we give the sequence of games $G_0 - G_3$ in Figure 3. If a line of the game of security is ending with G_{i-j}, this means that the line only appears in Games i up to j.

$\underline{\mathsf{Exp}^{\mathsf{OBLI}}(\mathcal{A},\lambda)}$:		$\underline{\mathsf{O}_{\mathsf{BKG}}()}$:	
$r^* \xleftarrow{\$} \{0,1\}^\lambda$	G_{2-3}	$\mathtt{id} \xleftarrow{\$} \mathcal{ID}, \mathsf{IDList}[i] \leftarrow \mathtt{id}, i \leftarrow i+1$	
$\mathtt{id}^* \xleftarrow{\$} \mathcal{ID}$	G_{1-3}	Execute $\mathsf{BlindKeyGen}(\langle(\mathcal{A}, \mathsf{msk})(\mathsf{User}, \mathtt{id})\rangle)$	
$\mathsf{H}^* \xleftarrow{\$} \mathcal{C}$	G_3	$\underline{\mathsf{O}_{\mathsf{ENC}}(j \in [i], m)}$:	
$(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$		$\mathtt{id} \leftarrow \mathsf{IDList}[j]$	
$\mathsf{IDList} \leftarrow \emptyset$		$r \xleftarrow{\$} \{0,1\}^\lambda$	G_{0-1}
$(m^*, j^*) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{BKG}}(.),\mathsf{O}_{\mathsf{ENC}}(.)}(\mathsf{mpk}, \mathsf{msk})$		$r \xleftarrow{\$} \{0,1\}^\lambda \backslash \{r^*\}$	G_{2-3}
$\mathtt{id}^* \leftarrow \mathsf{IDList}[j^*]$	G_0	$c' \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, id, m)$	
$c \leftarrow \mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}^*, m^*)$		$c \leftarrow (c' \oplus \mathsf{H}(\mathtt{id}, r), r)$	
$r^* \xleftarrow{\$} \{0,1\}^\lambda$	G_{0-1}	Return c	
$c^* \leftarrow (c \oplus \mathsf{H}(\mathtt{id}^*, r^*), r^*)$		$\underline{\mathsf{H}(\mathtt{id}, r)}$:	
$c^* \leftarrow (c \oplus h^*, r^*)$	G_3	If $\exists h$ such that $((\mathtt{id}, r), h) \in \mathcal{L}_{\mathsf{H}}$; return h	
$b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{BKG}}(.),\mathsf{O}_{\mathsf{ENC}}(.)}(\mathsf{mpk}, \mathsf{msk}, c^*, m^*)$		$\mathsf{H} \xleftarrow{\$} \mathcal{C}, \mathcal{L}_{\mathsf{H}} \leftarrow \mathcal{L}_{\mathsf{H}} \cup \{((\mathtt{id}, r), h)\}$	
Return $b' = 1$		Return h	

Fig. 3: Games $G_0 - G_3$ for the proof of Theorem 3.

Game 0. In Game G_0, we keep the original OBLI game defined in 11, therefore

$$\mathsf{Adv}^{\mathsf{OBLI}}_{\mathsf{OBLI}(\Pi)}(\mathcal{A}) \;=\; \mathsf{Adv}^{G_0}(\mathcal{A}) \;.$$

Game 1. In Game G_1, instead of drawing randomly the challenged $\mathtt{id}^*$ from IDList, we take a random $\mathtt{id}^* \in \mathcal{ID}$. Now $\mathtt{id}^*$ is not correlated anymore to IDList, which was built during the $\mathsf{O}_{\mathsf{BKG}}(.)$ queries.

Lemma 1. *Let q_{bkg} be the number of queries to the $\mathsf{O}_{\mathsf{BKG}}(.)$ oracle. There exists an adversary $\mathcal{A}_\Pi$ against the selective-failure blind property of Π such that*

$$|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})| \leq \mathsf{Adv}^{\mathsf{SFB}}_{\Pi}(\mathcal{A}_\Pi) \times 2 \times q_{\mathsf{bkg}} \;.$$

Game 2. In Game G_2, we are drawing r^* at the very beginning of the game before giving the adversary access to $\mathsf{O}_{\mathsf{ENC}}(.)$. During the encryption queries, we are picking randomly r in $\{0,1\}^\lambda \backslash \{r^*\}$ to ensure that each randomness we use in $\mathsf{O}_{\mathsf{ENC}}(.)$ will not be the same as the one used for the final challenge. Then

$$|\mathsf{Adv}^{G_1}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})| \leq \frac{q_{\mathsf{enc}}}{2^\lambda} \;.$$

GAME 3. In Game G_3, we replace $\mathsf{H}(\mathtt{id}^*, r^*)$ by a uniform h^* we drew at the beginning of the game to mask the ciphertext c. Let q_{H} be the number of query to H, then

$$|\mathsf{Adv}^{G_2}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A})| \leq q_{\mathsf{H}}/2^{\lambda} \times |\mathcal{ID}| \ .$$

Since h^* is uniformly sampled, c^* is completely random, the adversary outputs the correct bit half the time and so $\mathsf{Adv}^{G_3}(\mathcal{A})$ is statistically negligible. This concludes the proof. □

To prove the intermediate Lemma 1, we will employ a "guess-the-index" technique. The adversary $\mathcal{A}_{\Pi}$ doing the reduction will pick a random index $j \in [q_{\mathsf{bkg}}]$. It will embed the selective-failure blindness challenge into the j-th $\mathsf{O}_{\mathsf{BKG}}$ query of the adversary $\mathcal{A}$ against G_1. If the adversary $\mathcal{A}$ chooses a challenge index $j^* \neq j$, $\mathcal{A}_{\Pi}$ aborts. This yields to $\mathcal{A}_{\Pi}$ a non-negligible advantage proportional to $1/q_{\mathsf{bkg}}$.

Proof (Lemma 1). Let q_{bkg} be the number of queries to the $\mathsf{O}_{\mathsf{BKG}}(.)$ oracle. Let $\mathcal{A}_{\Pi}$ be the adversary against the selective-failure blind property of Π. $\mathcal{A}_{\Pi}$ runs G_1 in the following way. $\mathcal{A}_{\Pi}$ receives $(\mathsf{mpk}, \mathsf{msk})$ and chooses a random j, it sends $(\mathsf{mpk}, \mathsf{msk})$ to $\mathcal{A}$, and keep tracks of IDList and

But $\mathcal{A}_{\Pi}$ simulates $\mathsf{O}_{\mathsf{BKG}}$ in the following way:

- If $\mathcal{A}$ queries the ith $\mathsf{O}_{\mathsf{BKG}}$ query with $i \neq j$, it simulates the $\mathsf{O}_{\mathsf{BKG}}$ oracle "normally", by drawing a random $\mathtt{id}$ put it in IDList and initiates the $\mathsf{BlindKeyGen}$ protocol from the user side with $\mathtt{id}$.
- When $\mathcal{A}$ queries the jth $\mathsf{O}_{\mathsf{BKG}}$ query, $\mathcal{A}_{\Pi}$ draws randomly $\mathtt{id}_0$ from $\mathcal{ID}$ and put it in IDList. $\mathcal{A}_{\Pi}$ draws a random $\mathtt{id}_1$ from $\mathcal{ID}$ and returns the challenge $(\mathtt{id}_0, \mathtt{id}_1)$ for its experiment. $\mathcal{A}_{\Pi}$ receives the query $\mathsf{BlindKeyGen}(\langle \mathcal{A}_{\Pi}(\mathsf{msk}), \mathsf{User}(\mathtt{id}_b)\rangle)$ followed by the query $\mathsf{BlindKeyGen}(\langle \mathcal{A}_{\Pi}(\mathsf{msk}), \mathsf{User}(\mathtt{id}_{1-b})\rangle)$ from its experiment. $\mathcal{A}_{\Pi}$ draws a random b' and forwards the query $\mathsf{BlindKeyGen}(\langle \mathcal{A}(\mathsf{msk}), \mathsf{User}(\mathtt{id}'_b)\rangle)$ to $\mathcal{A}$. $\mathcal{A}_{\Pi}$ returns the answers of $\mathcal{A}$ in its proper experiment for the query$\mathsf{BlindKeyGen}(\langle \mathcal{A}(\mathsf{msk}), \mathsf{User}(\mathtt{id}'_b)\rangle)$. It fails the other query and ignore the challenge received from its experiment.

$\mathcal{A}_{\Pi}$ simulates the oracle $\mathsf{O}_{\mathsf{ENC}}$ by taking $\mathtt{id} \leftarrow \mathsf{IDList}[j]$ and returning the correct ciphertext $c \leftarrow (\mathsf{Encrypt}(\mathsf{mpk}, id, m) \oplus \mathsf{H}(\mathtt{id}, r), r)$.

$\mathcal{A}_{\Pi}$ receives (m^*, j^*) from $\mathcal{A}$, if $j \neq j^*$, $\mathcal{A}_{\Pi}$ aborts. Otherwise, $\mathcal{A}_{\Pi}$ sends $(c^* \oplus H(id_1, r^*), r^*)$ with $c^* = Enc(m^*, id_1)$ to $\mathcal{A}$ and outputs the same bit $\mathcal{A}$ outputs. The probability of not aborting is $1/q_{\mathsf{bkg}}$ and the probability that $b = b'$ is $1/2$, hence the lemma. □

5 From standard definition to advanced security request

In this section, we will recall two IND-ID-CPA and ANON-ID-CPA schemes, introduce a BlindKeyGen protocol for them and apply our generic compiler. The first scheme is based on asymmetric elliptic curves in the standard model, it is the asymmetric variant of the scheme introduced in 2005 by Waters [29]. The second one is based on lattices; it is the scheme introduced by Agrawal and Boyen in [3].

5.1 Oblivious IBE in a pairing-friendly environment

The generalized version of Waters' IBE [29], proposed independently by Naccache [25] and Chatterjee and Sarkar [13] consists of representing each identity $\mathtt{id}$ lying in the identity space $\mathcal{ID}$ (the set of bit strings of length $N = n \times \ell$) by n blocks of ℓ bits each.

We will only consider type-3 pairing, where we do not know efficient isomorphism between $\mathbb{G}$ and $\hat{\mathbb{G}}$. We will call a bilinear pairing asymmetric if it is a type-3 pairing and SXDH holds for it. Let GroupGen be a PPT algorithm that on input 1^λ returns $\mathcal{G} = (p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e, g, \hat{g})$ where e is an asymmetric, bilinear, non-degenerate and efficiently computable pairing

Since Waters' scheme is not ANON-ID-CPA, we are taking the transformation of Ducas [15] from symmetric to asymmetric bilinear to solve this issue. Let describe this scheme in Figure 4. Let $\mathbb{G}, \hat{\mathbb{G}}$, and $\mathbb{G}_T$ be groups of order p. Let g be a generator of $\mathbb{G}$ and $\hat{g}, \hat{f}, \hat{h}$ be generators of $\hat{\mathbb{G}}$. For simplicity, let $\mathcal{U}(\mathtt{id}) = u_0 \prod_{1 \le i \le n} u_i^{\mathtt{id}_i}$ and $\hat{\mathcal{U}}(\mathtt{id}) = \hat{u_0} \prod_{1 \le i \le n} \hat{u_i}^{\mathtt{id}_i}$.

$\underline{\mathsf{Setup}(1^\lambda)}$:
$\mathcal{G} = (p, \mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e, g, \hat{g}) \leftarrow \mathsf{GroupGen}(1^\lambda)$
$\hat{f} \xleftarrow{\$} \hat{\mathbb{G}}, \hat{h} \xleftarrow{\$} \hat{\mathbb{G}}$; $\alpha \xleftarrow{\$} \mathbb{Z}_p, (\mu_i \xleftarrow{\$} \mathbb{Z}_p)_{0 \le i \le n}$
$\mathsf{mpk} \leftarrow (g, \hat{g}, \hat{f}, \hat{h}, (u_i = g^{\mu_i})_{0 \le i \le n}, g^\alpha, \hat{g}^\alpha)$
$\mathsf{msk} \leftarrow (\hat{h}^\alpha, (\mu_i)_{0 \le i \le n}, (\hat{u}_i = \hat{g}^{\mu_i})_{0 \le i \le n}, (\hat{f}_i = \hat{f}^{\mu_i})_{0 \le i \le n})$
Return $(\mathsf{mpk}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathtt{id})}$:
Return $\mathsf{sk}_{\mathtt{id}} = (\hat{g}^\beta, \hat{h}^\alpha \cdot \hat{\mathcal{U}}(\mathtt{id})^\beta)$ with $\beta \xleftarrow{\$} \mathbb{Z}_p$

$\underline{\mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}, m \in \mathbb{G}_T)}$:
$\delta \xleftarrow{\$} \mathbb{Z}_p, c_0 \leftarrow g^\delta, c_1 \leftarrow \mathcal{U}(\mathtt{id})^\delta$
$c_2 \leftarrow e(g^\alpha, \hat{h})^\delta \cdot m$
Return (c_0, c_1, c_2)

$\underline{\mathsf{Decrypt}(\mathsf{sk}_{\mathtt{id}}, c)}$:
Parse $\mathsf{sk}_{\mathtt{id}} = (s_0, s_1)$
Parse $c = (c_0, c_1, c_2)$
Return $m' = c_2 \frac{e(c_1, s_0)}{e(c_0, s_1)}$

Fig. 4: Asymmetric Waters $\mathcal{IBE}$ scheme.

Theorem 4. *The $\mathcal{IBE}$ described in Figure 4 is* IND-ID-CPA *and* ANON-ID-CPA *under the SXDH assumption.*

A generalization of the scheme was proven IND-ID-CPA and ANON-ID-CPA in [8, Figure 7] (under the $k-$MDDH assumption, by taking the correct parameter ($k = 1$) this is equivalent to SXDH). Let us modify this $\mathcal{IBE}$ into a selective-failure blind one.

Blind IBE with asymmetric pairings Since the original BlindKeyGen protocol from [19] was for the original scheme of Waters [29] which was not with asymmetric pairings, we propose some modifications for it to work with the $\mathcal{IBE}$ scheme detailed in Figure 4. We are forced to send $n + 1$ values to hide each $\hat{g}^{\mathtt{id}_i}$, so the authority is not able to retrieve the $\mathtt{id}$ queried; but can still compute something which will lead to the reconstruction of $(\hat{g}^\beta, \hat{h}^\alpha \cdot \hat{\mathcal{U}}(\mathtt{id})^\beta)$.

$\underline{\mathsf{Aut}(\mathsf{mpk}, \mathsf{msk})}$:	$\underline{\mathsf{User}(\mathsf{mpk}, \mathtt{id})}$:
	$(\tau_i \xleftarrow{\$} \mathbb{Z}_p)_{0\leq i\leq n}, T_0 \leftarrow \hat{f}^{\tau_0} \cdot \hat{g}$
	Send $T_0, (T_i \leftarrow \hat{f}^{\tau_i} \cdot \hat{g}^{\mathtt{id}_i})_{1\leq i\leq n}$
	Execute $\mathsf{ZKP}(((\tau_i)_{0\leq i\leq n}, \mathtt{id}) : T_i)$
If the proof fails, abort.	
$\beta \xleftarrow{\$} \mathbb{Z}_p, S_0 \leftarrow \hat{g}^{\beta}$	
$S_1 \leftarrow \hat{h}^{\alpha} \cdot \prod_{i=0}^{n} T_i^{\mu_i \beta}$	
$(B_i \leftarrow \hat{f}_i^{\beta})_{0\leq i\leq n}$	
Return $(S_0, S_1, (B_i)_{0\leq i\leq n})$	
	If $e(g^{\alpha}, \hat{h}) \cdot e(\mathcal{U}(\mathtt{id}), S_0) \cdot e(g, \prod_{i=0}^{n} B_i^{\tau_i})$ $\neq e(g, S_1)$, abort.
	Return $\mathsf{sk}_{\mathtt{id}} = (S_0, S_1 / \prod_{i=0}^{n} B_i^{\tau_i})$

Fig. 5: BlindKeyGen protocol for Asymmetric Waters $\mathcal{IBE}$

For clarity, through this subsection, let us call Π, the $\mathcal{IBE}$ described in Figure 4 together with BlindKeyGen protocol described in Figure 5.

Theorem 5. *The $\mathcal{IBE}$ described in Figure 4 together with* BlindKeyGen *protocol described in Figure 5 is selective-failure blind,* i.e.

$$\mathsf{Adv}^{\mathsf{SFB}}_{\Pi}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{ZK}}_{\mathsf{ZKP}}(\mathcal{A}) + \mathsf{Adv}^{\hat{G}}_{\mathsf{SXDH}}(\mathcal{A}) \ .$$

For better readability of the following proof of selective-failure blindness, we split the experiment of SFB (defined in Definition 10) in two parts. We will only detail the first part (stopping at the line ending with Part 2 in the definition), considering the only modification we will bring to the second part will be to replace the zero-knowledge proofs by their simulators.

Sketch of the proof. As explained earlier, we will only describe the first part of the SFB experiment considering that we will change all the ZKP in the second part by their respective simulators in Game 1. For better readability, we use a variable d which will take the value b and $1-b$ to execute both BlindKeyGen. In Game 2, if a $\mathsf{sk}_{\mathtt{id}}$ was correctly generated (*i.e.* $\neq \perp$) we will replace them by random values. In Game 3, we will replace each $\hat{f}^{\tau_{d,i}}$ by random values, considering that we do not need to compute correct $\mathsf{sk}_{\mathtt{id}}$ anymore.

Proof. For clarity, we give the sequence of games $G_0 - G_3$ in Figure 6. If a line of the game of security is ending with G_{i-j}, this means that the line only appears in Games i up to j.

Game 0. In Game G_0, we keep the original SFB game defined in 10, therefore

$$\mathsf{Adv}^{\mathsf{SFB}}_{\Pi}(\mathcal{A}) \ = \ \mathsf{Adv}^{G_0}(\mathcal{A}) \ .$$

Game 1. In Game G_1, we are replacing each ZKP by their zero-knowledge simulators (for readability purpose, we will not describe this process in the figure

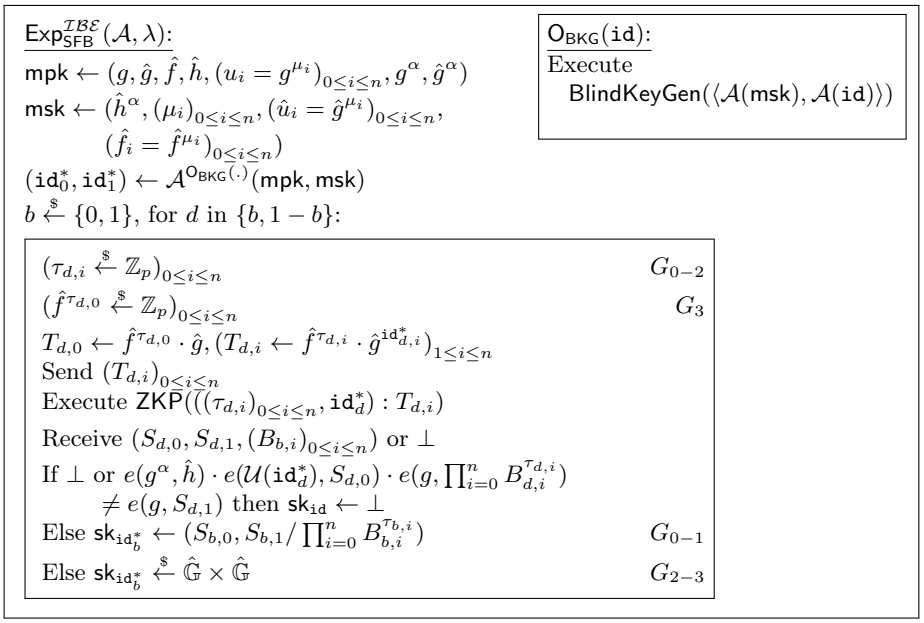


Fig. 6: Games $G_0 - G_3$ for the proof of Theorem 5.

above). We are replacing the ones being executed during the $\mathsf{BlindKeyGen}$ protocols and the ones that are in the second part of the experiment (that we did not describe). Therefore,

$$|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})| \le \mathsf{Adv}^{\mathsf{ZK}}_{\mathsf{ZKP}}(\mathcal{A}) \ .$$

Game 2. In Game G_2, we are replacing both $\mathsf{sk}_{\mathtt{id}}$ by random values in the else condition. Since the ZKP in the second part are already replaced by their simulators, nothing change for the adversary:

$$\mathsf{Adv}^{G_1}(\mathcal{A}) = \mathsf{Adv}^{G_2}(\mathcal{A}) \ .$$

Game 3. In Game G_3, we are replacing each $\hat{f}^{\tau_{d,i}}$ by random values. Since both ZKP were already replaced by their simulators, and we do not need to compute $\mathsf{sk}_{\mathtt{id}_d}$, it is not impacting the rest of the game. This is reducible to the discrete logarithm in $\hat{\mathbb{G}}$ which is supposed to be intractable thanks to the SXDH assumption, then

$$|\mathsf{Adv}^{G_2}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A})| \le \mathsf{Adv}^{\hat{G}}_{\mathsf{SXDH}}(\mathcal{A}) \ .$$

Considering that each $\hat{f}^{\tau_{d,i}}$ is now uniformly sampled, each $T_{d,i}$ are random, then $\mathcal{A}$ is only receiving random values through the game. Therefore, the adversary outputs the correct bit half the time and so $\mathsf{Adv}^{G_3}(\mathcal{A})$ is statistically negligible, which concludes the proof. □

Theorem 6. *The $\mathcal{IBE}$ described in Figure 4 together with* BlindKeyGen *protocol described in Figure 5 is leak-free if the zero-knowledge proof used in the protocol is extractable,* i.e.

$$\mathsf{Adv}^{\mathsf{LF}}_{\Pi}(\mathcal{A}) \leq \mathsf{Adv}^{\hat{G}}_{\mathsf{SXDH}}(\mathcal{A}) \ .$$

Basically, if we use an extractable ZKP, for example the one of [20] in our BlindKeyGen protocol. We will use the extractable property to recover $((\tau_i)_{0\leq i\leq n}, \mathtt{id})$ and then build $(S_0, S_1, (B_i)_{0\leq i\leq n})$ from $\mathsf{KeyGen}(\mathsf{msk}, \mathtt{id})$. From this we can replace the $\mathsf{O}_{\mathsf{BKG}}$ oracle by an O_{KG} oracle, since the $\mathsf{O}_{\mathsf{BKG}}$ oracle can be simulated with the help of an O_{KG} oracle. Afterwards we can reduce the leak-free property of our scheme to the IND-ID-CPA and ANON-ID-CPA property of the original scheme.

Proof. For clarity, we provide how we implement our oracle $\mathsf{O}_{\mathsf{BKG}}$ from Game G_0 to G_1 in the following figure.

$\underline{\mathsf{O}_{\mathsf{BKG}} \text{ in } G_0}$:	$\underline{\mathsf{O}_{\mathsf{BKG}} \text{ in } G_1}$:
Receive $(T_i)_{0\leq i\leq n}$	$(B_i \xleftarrow{\$} \hat{\mathbb{G}})_{0\leq i\leq n}$
Verify $\mathsf{ZKP}((\tau_i)_{0\leq i\leq n}, \mathtt{id}) : T_i)$	Receive $(T_i)_{0\leq i\leq n}$
If the proof fails, return $\perp$.	Verify $\mathsf{ZKP}((\tau_i)_{0\leq i\leq n}, \mathtt{id}) : T_i)$
$\beta \xleftarrow{\$} \mathbb{Z}_p, S_0 \leftarrow \hat{g}^{\beta}$	If the proof fails, return $\perp$.
$S_1 \leftarrow \hat{h}^{\alpha} \cdot \prod_{i=0}^{n} T_i^{\mu_i \beta}$	Extract $((\tau_i)_{0\leq i\leq n}, \mathtt{id})$
$(B_i \leftarrow \hat{f}_i^{\beta})_{0\leq i\leq n}$	$(\mathsf{sk}_0, \mathsf{sk}_1) \leftarrow \mathsf{O}_{\mathsf{KG}}(\mathtt{id})$
Return $(S_0, S_1, (B_i)_{0\leq i\leq n})$	$S_0 \leftarrow \mathsf{sk}_0, S_1 \leftarrow \mathsf{sk}_1 \cdot \prod_{i=0}^{n} B_i^{\tau_i}$
	Return $(S_0, S_1, (B_i)_{0\leq i\leq n})$

Game 0. In Game G_0, we keep the original $\mathsf{O}_{\mathsf{BKG}}$ oracle as defined in the LF experiment (defined in 10), therefore

$$\mathsf{Adv}^{\mathsf{LF}}_{\Pi}(\mathcal{A}) \ = \ \mathsf{Adv}^{G_0}(\mathcal{A}) \ .$$

Game 1. In Game G_1, we will use the extractability of the ZKP used in the protocol to recover the secrets $(\tau_i)_{0\leq i\leq n}$, and the secret $\mathtt{id}$. We start the game by drawing random B_i instead of computing them as we did in G_0. This is reducible to the discrete logarithm in $\hat{\mathbb{G}}$ which is supposed to be intractable thanks to the SXDH assumption, then Therefore,

$$|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})| \leq \mathsf{Adv}^{\hat{G}}_{\mathsf{SXDH}}(\mathcal{A}) \ .$$

Thanks to the extractability property we can compute S_0, S_1 from the random B_i and the secrets $(\tau_i)_{0\leq i\leq n}$, $\mathtt{id}$ so G_1 does not reveal more information than an oracle O_{KG} could reveal. Hence, we can reduce the LF-security to the IND-ID-CPA or ANON-ID-CPA properties of the original scheme, here SXDH. □

Oblivious IBE Let's now apply the transformation described in Figure 2 to our blind $\mathcal{IBE}$. We keep the same Setup function and BlindKeyGen protocol that we described just above.

$\underline{\mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}, m \in \mathbb{G}_T)}$:	$\underline{\mathsf{Decrypt}(\mathsf{sk}_{\mathtt{id}}, c)}$:
$\delta, r \xleftarrow{\$} \mathbb{Z}_p, c_0 \leftarrow g^\delta, c_1 \leftarrow \mathcal{U}(\mathtt{id})^\delta$	Parse $\mathsf{sk}_{\mathtt{id}} = (s_0, s_1)$
$c_2 \leftarrow e(g^\alpha, \hat{h})^\delta \cdot m \cdot H(\mathtt{id}, r)$	Parse $c = (c_0, c_1, c_2, r)$
Return (c_0, c_1, c_2, r)	Return $m' = c_2 \frac{e(c_1, s_0)}{e(c_0, s_1)} / H(\mathtt{id}, r)$

Since the point of weakness presented in Section 1 was only coming from c_2, we can simply hide c_2 with the hash function.

Moving back to the standard model The use of a random oracle while generic forces us to move away from the standard model. In this part, we will explain how to supersede $H(\mathtt{id}, r)$ by a standard model function. A first natural thought would be to rely on a standard model PRF. However, this leads to a problem in the reduction if the PRF cannot be evaluated by the simulator without relying on $\mathtt{id}$.

So, instead, we are going to rely on $c_1 = \mathcal{U}(\mathtt{id})^\delta$, to create a mask $M = c_1^r$. We need to be careful though, if we give r directly, then the mask is publicly computable without $\mathtt{id}$; so instead, we are going to publish a randomization of the u_i, as $n+1$ elements in $\mathbb{G}$ defined as $v_i = u_i^{r\delta}$, defining $\mathcal{V}(\mathtt{id}) = v_0 \prod v_i^{\mathtt{id}_i} = \mathcal{U}(\mathtt{id})^{r\delta} = c_1^r$.

$\underline{\mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}, m \in \mathbb{G}_T)}$:	$\underline{\mathsf{Decrypt}(\mathsf{sk}_{\mathtt{id}}, c)}$:
$\delta, r \xleftarrow{\$} \mathbb{Z}_p, c_0 \leftarrow g^\delta, c_1 \leftarrow \mathcal{U}(\mathtt{id})^\delta$	Parse $\mathsf{sk}_{\mathtt{id}} = (s_0, s_1)$
$\boldsymbol{v} = \{v_i = u_i^{r\delta} \vert i \in [0, n]\} \in \mathbb{G}^{n+1}$	Parse $c = (c_0, c_1, c_2, \boldsymbol{v})$
$c_2 \leftarrow e(g^\alpha, \hat{h})^\delta \cdot m \cdot e(\mathcal{V}(r), \hat{g})$	Return $m' = c_2 \frac{e(c_1, s_0)}{e(c_0, s_1)} / e(\mathcal{V}(\mathtt{id}), \hat{g})$
Return $(c_0, c_1, c_2, \boldsymbol{v})$	

To prove the security of our scheme, we are going to adapt the Theorem 7, to no longer rely on the random oracle. Theorems 2 can be directly used as they did not require the random oracle.

Theorem 7. *Under the fact that Blind Waters is selective-failure blind, and that both the knapsack Diffie Hellman problem and SXDH are hard, then our new protocol is oblivious.*

The proof is going to follow the same logic as the compiler one, however we need to be more careful in the order we do things: First, we show how to simulate M without knowing $\mathtt{id}^*$, so that we can rely on the SFB game without having to hide behind the ROM. Then, we apply SFB, to be able to take a truly random $\mathtt{id}^*$. Finally, we show that given an SXDH challenge, we can adapt $\boldsymbol{v}$ and M to either solve said challenge with an adversary against our scheme, or transform our output as pure random.

Proof. Game 0. In Game G_0, we keep the original OBLI game defined in 11, therefore

$$\mathsf{Adv}^{\mathsf{OBLI}}_{\mathsf{OBLI}(\Pi)}(\mathcal{A}) \;=\; \mathsf{Adv}^{G_0}(\mathcal{A}) \;.$$

GAME 1. In Game G_1, when picking δ, r for the challenge, checks whether the resulting $\boldsymbol{v}^*$ is fresh, otherwise abort.

$$|\mathsf{Adv}^{G_0}(\mathcal{A}) - \mathsf{Adv}^{G_1}(\mathcal{A})| \leq \frac{q_{\mathsf{enc}}}{2^\lambda} \ .$$

GAME 2. In Game G_2, we are now changing the way we compute M, to be c_1^r. This is the exact same value, as $\mathcal{V}(\mathtt{id})$, so the distribution in the adversary eyes is the same.

$$\mathsf{Adv}^{G_1}(\mathcal{A}) = \mathsf{Adv}^{G_2}(\mathcal{A}) \ .$$

GAME 3. In Game G_3, instead of drawing randomly the challenged $\mathtt{id}^*$ from IDList, we take a random $\mathtt{id}^* \in \mathcal{ID}$. Now $\mathtt{id}^*$ is not correlated anymore to IDList, which was built during the $\mathsf{O}_{\mathsf{BKG}}(.)$ queries. Using Lemma 1, leads to:

$$|\mathsf{Adv}^{G_3}(\mathcal{A}) - \mathsf{Adv}^{G_2}(\mathcal{A})| \leq \mathsf{Adv}^{\mathsf{SFB}}_{\Pi}(\mathcal{A}_{\Pi}) \times 2 \times q_{\mathsf{bkg}} \ .$$

GAME 4. In Game G_4, we replace $c_2 = \mathcal{U}(\mathtt{id})^\delta$ by a random element from $\mathbb{G}$. An adversary, distinguishing this from a properly computed c_2 directly solves the Knapsack Diffie-Hellman problem[3]. Hence

$$|\mathsf{Adv}^{G_4}(\mathcal{A}) - \mathsf{Adv}^{G_3}(\mathcal{A})| \leq \mathsf{Adv}^{\mathsf{kdh}}_{\Pi}(\mathcal{A}_{\Pi}) \ .$$

GAME 5. In Game G_5, we now replace c_3 by a random value. For that, we consider an SXDH challenge of the G, U, R, M. We define, $\forall i \in [0, n], v_i = R^{\mu_i}$. And set $c_0 = G, c_1 = U, c_2 = e(g^\alpha, \hat{h})^\delta \cdot m \cdot e(M, \hat{g})$. If the tuple is well-formed, there exist a, r such that $U = G^a, R = G^r, M = G^{ar} = U^r$, and in this case we are in the situation of Game 3. If the tuple is not well-formed, then M is random, and so is c_3. Hence:

$$|\mathsf{Adv}^{G_5}(\mathcal{A}) - \mathsf{Adv}^{G_4}(\mathcal{A})| \leq \mathsf{Adv}^{\mathsf{SXDH}}_{\Pi}(\mathcal{A}_{\Pi}) \ .$$

Everything in the ciphertext is now random, which leads to the conclusion. □

5.2 Post-quantum instantiation

We will now sketch a high level construction of a post-quantum $\mathcal{IBE}$ fulfilling the prerequisite of our oblivious framework. We do not directly use the GPV [17] $\mathcal{IBE}$ due to the possible attack [26] on the blinded version leading to a recovery of the master key. Instead, we will use the $\mathcal{IBE}$ in the standard model from [3] (which was expanded in [2]) that is both $\mathsf{IND\text{-}ID\text{-}CPA}$ and $\mathsf{ANON\text{-}ID\text{-}CPA}$, show how it can be made blind using an oblivious transfer. The attack does not further apply, because prevents the user from choosing the elements he can request a

[3] The Knapsack Diffie-Hellman problem was formalized, and proven in [21], and states precisely that $U'_{n+1} = (u_0, ..., u_n, U) \in \mathbb{G}^{n+1}$ is computationally indistinguishable from $K'_{n+1,1} := (u_0, ..., u_n, U) \in \mathbb{G}^n \times \{h$, such that $h = u_0 \prod u_i^{a_i}$ with $a_i \in \{0, 1\}\}$ as long as n is big enough (greater than twice the bit-length of the group order p.)

pre-image of. It should be noted that the transformation is far from a practical complexity; thus we will omit explicit parameter instantiations.

We can use any Oblivious Transfer like [22]. We also require Zero-Knowledge proofs, either compatible with Lattice operations like [23] or with circuit-based technique [4].

Reminders around Lattices. An m-dimensional lattice associated to the matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is defined as $\Lambda^{\perp}(\mathbf{A}) = \{\boldsymbol{x} \in \mathbb{Z}^m : \mathbf{A}x = 0 \in \mathbb{Z}_q^n\}$. Let denote the distribution $D_{\mathbb{Z}^m,\sigma}$ with domain $\{e \in \mathbb{Z}^m : \|e\| \leq \sigma\sqrt{m}\}$ by taking the Gaussian smoothing parameter $\sigma \geq L \cdot \omega(\sqrt{\log m})$. We recall the GPV preimage sampling functions without entering too much into details :

- $\mathsf{TrapGen}(1^\lambda)$: Output $(\mathbf{A}, \mathbf{T_A}, q, n, m)$ where $\mathbf{A}$ is a matrix in $Z_q^{n \times m}$ and $\mathbf{T_A} \in \mathbb{Z}_q^{m \times m}$ is a short "trapdoor" basis of $\Lambda^{\perp}(\mathbf{A})$.
- $\mathsf{SampleDom}(\mathbf{H}, \sigma)$: Output an $\boldsymbol{x}$ which was sampled from distribution $D_{\mathbb{Z}^m,\sigma}$ and $\mathbf{A}\boldsymbol{x}$ is uniform over $\mathbb{Z}_q^n$.
- $\mathsf{SamplePre}(\mathbf{T_A}, \sigma, \boldsymbol{y})$: Output an $\boldsymbol{x}$ such that $\mathbf{A}\boldsymbol{x} = \boldsymbol{y} \bmod q$ which was sampled from distribution $D_{\mathbb{Z}^m,\sigma}$.

The Agrawal-Boyen $\mathcal{IBE}$ [3] Let our identity space be $\mathcal{ID} = \{0,1\}^N$. For simplicity, we will write $\mathbf{H}_{\mathtt{id}}$ for $[\mathbf{H}_{1,\mathtt{id}_1} | \dots | \mathbf{H}_{N,\mathtt{id}_N}]$. The "noise" distributions χ, χ^m, and χ^{nN} are parameterized as in the Regev public-key cryptosystem. Let us now describe the Agrawal-Boyen scheme in Figure 7.

$\underline{\mathsf{Setup}(1^\lambda)}$:
$(\mathbf{A}, \mathbf{T_A}, q, n, m) \leftarrow \mathsf{TrapGen}(1^\lambda)$
$(\mathbf{H}_{i,b} \xleftarrow{\$} \mathbb{Z}_q^{n \times m})_{((i,b) \in [N] \times \{0,1\})}$
$\boldsymbol{u}_0 \xleftarrow{\$} \mathbb{Z}_q^n, \bar{\mathbf{H}} = \{\mathbf{H}_{i,b}\}$
$\mathsf{mpk} := (\mathbf{A}, \boldsymbol{u}_0, \bar{\mathbf{H}}), \mathsf{msk} := \mathbf{T_A}$
Return $(\mathsf{mpk}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}(\mathsf{msk}, \mathtt{id})}$:
$(\boldsymbol{r}_i \leftarrow \mathsf{SampleDom}(\mathbf{H}_{i,\mathtt{id}_i}, \sigma))_{(i \in [N])}$
$\boldsymbol{r}$ is the concatenation of the vectors $\boldsymbol{r}_i$
$\boldsymbol{u} \leftarrow \boldsymbol{u}_0 + \mathbf{H}_{\mathtt{id}} \cdot \boldsymbol{r}$
$\boldsymbol{e} \leftarrow \mathsf{SamplePre}(\mathbf{T_A}, \sigma, \boldsymbol{u})$
Return $\mathsf{sk}_{\mathtt{id}} = (\boldsymbol{e}, \boldsymbol{r})$

$\underline{\mathsf{Encrypt}(\mathsf{mpk}, \mathtt{id}, b \in \{0,1\})}$:
$\boldsymbol{s} \xleftarrow{\$} \mathbb{Z}_q^n, x \leftarrow^{\chi} \mathbb{Z}_q$
$\boldsymbol{y} \leftarrow^{\chi^m} \mathbb{Z}_q^m, \boldsymbol{z} \leftarrow^{\chi^{nN}} \mathbb{Z}_q^{nN}$
$c_0 = \boldsymbol{u}_0^\top \boldsymbol{s} + x + b\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q$
$\boldsymbol{c_1} = \mathbf{A}^\top \boldsymbol{s} + \boldsymbol{y} \in \mathbb{Z}_q^m$
$\boldsymbol{c_2} = \mathbf{H}_{id}^\top \boldsymbol{s} + \boldsymbol{z} \in \mathbb{Z}_q^{nN}$
Return $c = (c_0, \boldsymbol{c_1}, \boldsymbol{c_2})$

$\underline{\mathsf{Decrypt}(\mathsf{sk}_{\mathtt{id}}, c)}$:
Parse $c = (c_0, \boldsymbol{c_1}, \boldsymbol{c_2}); \mathsf{sk}_{\mathtt{id}} = (\boldsymbol{e}, \boldsymbol{r})$
$v \leftarrow c_0 - \boldsymbol{e}^\top \boldsymbol{c_1} + \boldsymbol{r}^\top \boldsymbol{c_2} \in \mathbb{Z}_q$
If $|v - \lfloor \frac{q}{2} \rfloor| \leq \frac{q}{4}$, output 1.
Otherwise, output 0.

Fig. 7: Agrawal-Boyen $\mathcal{IBE}$

It was shown in the original paper, that under LWE this scheme is both IND-sID-CPA and ANON-sID-CPA, with argument that with bigger parameters it can be made IND-ID-CPA, and ANON-ID-CPA.

Getting a Blind Identity-Based Encryption. A simple way to get a Blind Identity-Based Encryption from an $\mathcal{IBE}$ is to use an Oblivious Transfer to retrieve $\mathsf{sk}_{\mathtt{id}}$. The link between those primitives was first hinted in [19], and a construction of blind $\mathcal{IBE}$ from Oblivious Transfer was given in more details for pairings in [6], and [7]. A primary challenge in this transformation is collapsing the exponential identity space into a polynomial number of queries.

Here, as in the pairing papers, one part of the secret is generated by a bit by bit decomposition of the identity. Meaning that instead of having the whole identity space at once, a user can retrieve all his $r_{i,\mathtt{id}_i}$ in only N 1-out-of-2 oblivious transfers. From this, he can sample a $\boldsymbol{t} \in \mathbb{Z}_q^n$ by running $\mathsf{SampleDom}(\mathbf{A}, \sigma)$ and set $\boldsymbol{u} = \boldsymbol{u}_0 + \mathbf{H}_{\mathtt{id}} r + \mathbf{A}\boldsymbol{t}$ and send that to the authority; together with a ZKP that $\boldsymbol{u}$ is honestly formed, with $\boldsymbol{r}$ coming from the oblivious transfer, and $\boldsymbol{t}$ correctly sampled and initially committed.

The server can then run $\mathsf{SamplePre}(\mathbf{T}_{\mathbf{A}}, \sigma, \boldsymbol{u})$ to compute $\boldsymbol{e'}$, which the user will transform into $\boldsymbol{e} = \boldsymbol{e'} - \boldsymbol{t}$ so that $\boldsymbol{u}_0 + \mathbf{H}_{\mathtt{id}}\boldsymbol{r} = \mathbf{A}\boldsymbol{e}$ as in the original scheme. In order to avoid ROS-like attack, the user will commit to his $\boldsymbol{t}$ in the first flow, so that he cannot make adaptive queries to the $\mathsf{SamplePre}$ Oracle.

$\underline{\mathsf{Aut}(\mathsf{mpk}, \mathsf{msk})}$:	$\underline{\mathsf{User}(\mathsf{mpk}, \mathtt{id})}$:
	$\boldsymbol{t} \leftarrow \mathsf{SampleDom}(\mathbf{A}, \sigma)$
	Compute C_t, a commit to $\boldsymbol{t}$
	First flow: $\mathrm{OT}(\langle User, \mathtt{id} \rangle)$, C_t
$\forall i, b \in [k] \times \{0,1\}$:	
$\boldsymbol{r}_{i,b} \leftarrow \mathsf{SampleDom}(\mathbf{H}_{i,\mathtt{id}_i}, \sigma)$	
Second flow: $\mathrm{OT}(\langle Aut, (\boldsymbol{r_{i,b}}) \rangle)$	
	$\forall i \in [k]$ recovers $\boldsymbol{r}_{i,\mathtt{id}_i}$, and $\boldsymbol{r}$
	Sends $\boldsymbol{u} = \boldsymbol{u}_0 + \mathbf{H}_{\mathtt{id}}\boldsymbol{r} + \mathbf{A}\boldsymbol{t}$ and π.
	Execute $\mathsf{ZKP}((\boldsymbol{t}, \mathtt{id}) : (C_t, \boldsymbol{u}))$
If the proof fails, abort.	
Sends $\boldsymbol{e'} = \mathsf{SamplePre}(\mathbf{T}_{\mathbf{A}}, \sigma, \boldsymbol{u})$	
	Sets $\boldsymbol{e} = \boldsymbol{e'} - \boldsymbol{t}$
	$\mathsf{sk}_{\mathtt{id}} = (\boldsymbol{r}, \boldsymbol{e})$

Fig. 8: $\mathsf{BlindKeyGen}$ protocol for Agrawal-Boyen $\mathcal{IBE}$

This scheme is blind under the obliviousness of the Oblivious Transfer, and the Soundness, Zero-Knowledge property of the ZKP. Intuitively, the obliviousness of the Oblivious Transfer and the Zero-Knowledge property hide the identity requested, and ensure that the user does not jeopardize it, and the soundness ensures that the user does not exploit his queries to recover msk.

Oblivious post-quantum IBE. We now have all the tools needed for our compiler. Combining them, would therefore, under the $\mathsf{IND\text{-}ID\text{-}CPA}$ and $\mathsf{ANON\text{-}ID\text{-}CPA}$ of the original $\mathcal{IBE}$ scheme, the obliviousness of the Oblivious Transfer, the security of the Zero-Knowledge proof, and the ROM, lead to a lattice-based Oblivious $\mathcal{IBE}$.

6 Conclusions and Future Work

In this work, we focused on limiting the required trust in the authority of an Identity-Based Encryption. We established a fundamental connection, showing that blind IBE, which protects user identity during key extraction, and ciphertext generation, can be generically transformed into oblivious IBE, thereby preventing the authority from decrypting ciphertexts without an explicit key generation query. We can even reach the originally defined strong oblivious IBE, as long as we rely on the random oracle model.

Our contributions improve the applicability of oblivious IBE, since we provided the first generic compiler for this primitive. Furthermore, we resolved the open problem of constructing olivious IBE in the standard model in prime order groups, with an instantiation solely based on pairings. Finally, we addressed long-term security concerns by proposing a path to the first post-quantum Oblivious IBE from lattices in the quantum random oracle model.

Several open problems remain:

- An efficient post-quantum standard-model scheme: Our lattice-based construction is indeed post-quantum, however it relies on a random-oracle and sequentially composes several post-quantum primitives, resulting in a theoretically sound but concretely inefficient scheme.
- Efficiency Improvements: Our generic compiler relies on Zero-Knowledge Proofs (ZKPs), meaning that the concrete overhead can be significant (especially in the post-quantum world). Future works could focus on optimizing these components or designing direct constructions that bypass these needs for better performance.

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, Berlin, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218_13
2. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Berlin, Heidelberg (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_28
3. Agrawal, S., Boyen, X.: Identity-based encryption from lattices in the standard model (08 2009)
4. Ames, S., Hazay, C., Ishai, Y., Venkitasubramaniam, M.: Ligero: Lightweight sublinear arguments without a trusted setup. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017. pp. 2087–2104. ACM Press (Oct / Nov 2017). https://doi.org/10.1145/3133956.3134104
5. Ateniese, G., Camenisch, J., Hohenberger, S., de Medeiros, B.: Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385 (2005), https://eprint.iacr.org/2005/385

6. Blazy, O., Chevalier, C., Germouty, P.: Adaptive oblivious transfer and generalization. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 217–247. Springer, Berlin, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53890-6_8
7. Blazy, O., Chevalier, C., Germouty, P.: Almost optimal oblivious transfer from QA-NIZK. In: Gollmann, D., Miyaji, A., Kikuchi, H. (eds.) ACNS 2017. LNCS, vol. 10355, pp. 579–598. Springer, Cham (Jul 2017). https://doi.org/10.1007/978-3-319-61204-1_29
8. Blazy, O., Germouty, P., Phan, D.H.: Downgradable identity-based encryption and applications. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 44–61. Springer, Cham (Mar 2019). https://doi.org/10.1007/978-3-030-12612-4_3
9. Boneh, D.: The decision Diffie-Hellman problem. In: Third Algorithmic Number Theory Symposium (ANTS). LNCS, vol. 1423. Springer (1998), invited paper
10. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Berlin, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8_13
11. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Berlin, Heidelberg (Aug 2006). https://doi.org/10.1007/11818175_17
12. Camenisch, J., Kohlweiss, M., Rial, A., Sheedy, C.: Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 196–214. Springer, Berlin, Heidelberg (Mar 2009). https://doi.org/10.1007/978-3-642-00468-1_12
13. Chatterjee, S., Sarkar, P.: Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In: Won, D., Kim, S. (eds.) ICISC 05. LNCS, vol. 3935, pp. 424–440. Springer, Berlin, Heidelberg (Dec 2006). https://doi.org/10.1007/11734727_33
14. Dodis, Y., Yampolskiy, A.: A verifiable random function with short proofs and keys. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 416–431. Springer, Berlin, Heidelberg (Jan 2005). https://doi.org/10.1007/978-3-540-30580-4_28
15. Ducas, L.: Anonymity from asymmetry: New constructions for anonymous HIBE. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 148–164. Springer, Berlin, Heidelberg (Mar 2010). https://doi.org/10.1007/978-3-642-11925-5_11
16. Emura, K., Katsumata, S., Watanabe, Y.: Identity-based encryption with security against the KGC: A formal model and its instantiation from lattices. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) ESORICS 2019, Part II. LNCS, vol. 11736, pp. 113–133. Springer, Cham (Sep 2019). https://doi.org/10.1007/978-3-030-29962-0_6
17. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008). https://doi.org/10.1145/1374376.1374407
18. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems (extended abstract). In: 17th ACM STOC. pp. 291–304. ACM Press (May 1985). https://doi.org/10.1145/22145.22178
19. Green, M., Hohenberger, S.: Blind identity-based encryption and simulatable oblivious transfer. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 265–282. Springer, Berlin, Heidelberg (Dec 2007). https://doi.org/10.1007/978-3-540-76900-2_16

20. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Berlin, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_24
21. Guillevic, A., Vergnaud, D.: Algorithms for outsourcing pairing computation. In: Joye, M., Moradi, A. (eds.) Smart Card Research and Advanced Applications. pp. 193–211. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-16763-3_12, https://doi.org/10.1007/978-3-319-16763-3_12
22. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Adaptive oblivious transfer with access control from lattice assumptions. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 533–563. Springer, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70694-8_19
23. Lyubashevsky, V., Nguyen, N.K., Plançon, M.: Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 71–101. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_3
24. Mitrokotsa, A., Mukherjee, S., Tomy, J.: Oblivious identity-based encryption - (IBE secure against an adversarial KGC). In: Eichlseder, M., Gambs, S. (eds.) SAC 2024, Part I. LNCS, vol. 15516, pp. 282–309. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-82852-2_13
25. Naccache, D.: Secure and *practical* identity-based encryption. Cryptology ePrint Archive, Report 2005/369 (2005), https://eprint.iacr.org/2005/369
26. Rückert, M.: Lattice-based blind signatures. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 413–430. Springer, Berlin, Heidelberg (Dec 2010). https://doi.org/10.1007/978-3-642-17373-8_24
27. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 47–53. Springer, Berlin, Heidelberg (Aug 1984). https://doi.org/10.1007/3-540-39568-7_5
28. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332 (2004), https://eprint.iacr.org/2004/332
29. Waters, B.R.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Berlin, Heidelberg (May 2005). https://doi.org/10.1007/11426639_7
30. Wee, H.: Déjà Q: Encore! Un petit IBE. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part II. LNCS, vol. 9563, pp. 237–258. Springer, Berlin, Heidelberg (Jan 2016). https://doi.org/10.1007/978-3-662-49099-0_9