New Anonymity Notions for Identity-Based Encryption

Malika Izabachène, David Pointcheval

The extended abstract of this full version appeared in
Proceedings of the 6th Conference on Security and Cryptography for Networks (SCN '08)
September 10–12, 2008, Amalfi, Italy – R. Ostrovsky Eds. Springer-Verlag, LNCS 5229, pages 375–391.

New Anonymity Notions for Identity-Based Encryption

Malika Izabachène and David Pointcheval

Ecole Normale Supérieure – LIENS/CNRS/INRIA, France
{Malika.Izabachene,David.Pointcheval}@ens.fr

Abstract. Identity-based encryption is a very convenient tool to avoid key management. Recipient-privacy is also a major concern nowadays. To combine both, anonymous identity-based encryption has been proposed. This paper extends this notion to stronger adversaries (the authority itself). We discuss this new notion, together with a new kind of non-malleability with respect to the identity, for several existing schemes. Interestingly enough, such a new anonymity property has an independent application to password-authenticated key exchange. We thus come up with a new generic framework for password-authenticated key exchange, and a concrete construction based on pairings.

1 Introduction

Motivation. The idea of using identities instead of public keys in order to avoid the (costly) use of certificates comes from Shamir [19]. He indeed suggested *Identity-based Encryption* ($\mathcal{IBE}$), that would allow a user to encrypt a message using any string, that would specify the recipient, as encryption parameter, such that this recipient only can decrypt the ciphertext. Identity-based cryptography thus provides this interesting feature that one does not need authenticated public keys. Key managament is made simpler. Note however that a drawback is an *authority* that is required to generate the private keys for the users, according to their identities. This authority thus has the ability to decrypt any ciphertext. Privacy cannot be achieved with respect to this authority. Nevertheless, privacy of the plaintext is not the unique goal in cryptography, with encryption schemes. Privacy of the recipient may also be a requirement. Such a *key-privacy* notion has already been defined in the public-key setting in [3]. It has more recently been extended to the identity-based setting in [1], under the notion of *anonymity*. However, the security model in this $\mathcal{IBE}$ setting still trusts the authority. Whereas trusting the authority is intrinsic for privacy of the plaintext, it is not for the privacy of the recipient: a stronger anonymity notion is possible, with respect to the authority, but is it achievable for practical $\mathcal{IBE}$?

For efficiency reasons, the use of *Key Encapsulation Mechanisms* $\mathcal{KEM}$ have been shown as a preferable approach [21]. It consists in generating an ephemeral key and an encrypted version of the latter. The ephemeral key is thereafter used with a *Data Encryption Method* $\mathcal{DEM}$ to encrypt the message. In such a context, we are interested in the *semantic security* of the ephemeral key, and the *anonymity* of the recipient. In the identity-based context, Bentahar *et al.* [7] defined *Identity-based Key Encapsulation Mechanisms* $\mathcal{IB}$-$\mathcal{KEM}$. An anonymity notion with respect to the authority would then be an interesting feature. Interestingly enough, this notion of anonymity with respect to the authority might have side applications. One of them is PAKE [6], for *password-authenticated key exchange*. Such a protocol allows two players to establish a private channel, using a short secret as a sole authentication means. The latter is thus subject to exhaustive search, but such a short secret is very convenient for human beings.

Related Work. The idea of *identity-based encryption* is due to Shamir [19], in 1984. The first goal was to simplify public key management. However, the first practical solutions appeared in 2001 only [10, 15]. Thereafter, many schemes have been proposed, based on pairing, factoring and lattices. Since such schemes were dealing with encryption, the main security notion was the semantic security [17].

Even if recipient-anonymity had already been addressed for public-key encryption [3] in 2001, anonymity for $\mathcal{IBE}$ has been proposed recently by Abdalla *et al.* [1], but as a simple extension of the previous public-key setting definition. In 2006, Gentry [16] and Boyen and Waters [12] presented the first anonymous $\mathcal{IBE}$ schemes without random oracles.

Our contributions. As already noticed in [1], *anonymity* might have some side applications to searchable encryption. In this paper, we deal with anonymity for $\mathcal{IB}$-$\mathcal{KEM}$, even with respect to the authority, the so-called *Key Anonymity with respect to the Authority* and denoted KwrtA-Anonymity: we first provide a formal security model, and then we discuss this security notion with existing schemes. We also consider a new non-malleability notion for the identity, that we call *identity-based non-malleability*: if one encrypts a message (or a key) for user U, one has no idea about the value obtained by another user U', whatever the relation between U and U' (or the identities) is.

Thereafter, we show that these security notions can also have side applications to password-authenticated key exchange. Such a KwrtA-*anonymous* and *identity-based non-malleability* $\mathcal{IB}$-$\mathcal{KEM}$ scheme can indeed be plugged into a password-authenticated two-party key exchange protocol, in the same vein as the IPAKE construction [14] did with trapdoor hard-to-invert group isomorphisms. Our security result holds in a stronger security model than usual (with an adaptive selection of passive and active attacks, as in [18]), but the construction still assumes the random-oracle model [5], as in [14].

Eventually, we provide an $\mathcal{IB}$-$\mathcal{KEM}$, that is both KwrtA-*anonymous* and *identity-based non-malleable*, in addition to the full-identity semantic security, against chosen-plaintext adversaries. This thus leads to a new password-authenticated two-party key exchange protocol.

2 Anonymous Identity-Based Encryption

Anonymity for public-key encryption schemes has first been introduced by Bellare *et al.* [3], under the *key privacy* security notion, and has been extended to identity-based encryption by Abdalla *et al.* [1].

In these papers, anonymity meant that even if the adversary chooses a message and two identities (or two public keys), and the challenger encrypts the message with one of the identities (or keys), the adversary cannot guess which one has actually been involved in the computation. This notion is quite strong for public-key encryption, but not that strong in the identity-based setting since it does not capture anonymity with respect to the authority that knows the master secret key, and even chooses the public parameters PK.

Unfortunately, the previous definitions cannot be trivially extended: the adversary can easily break anonymity if he knows the expected plaintext, and just hesitates between two identities, since he can decrypt any ciphertext. Anonymity can only be expected against the server if the plaintexts follow a non-trivial distribution. Since we will deal with key-encapsulation mechanisms, this non-trivial distribution is already implicit for the ephemeral keys.

This enhanced security notion will be called *Key Anonymity with respect to the Authority* and denoted KwrtA-Anonymity. This section defines precisely this notion for identity-based key encapsulation mechanisms.

2.1 Identity-Based Encryption and Key Encapsulation Mechanisms

We first review the definitions of identity-based encryption, and more specifically of identity-based key encapsulation mechanisms [7]. In the following, we assume that identities are bit strings in a dictionary Dic.

Definition 1 (Identity-Based Encryption). An $\mathcal{IBE}$ scheme is specified by four algorithms:

$\mathsf{Setup}_{\mathsf{IBE}}(1^\lambda)$**.** Takes as input a security parameter λ. It outputs the public parameters PK, as well as a master secret key MK.

$\mathsf{Extract}_{\mathsf{IBE}}(\mathsf{MK}, \mathsf{ID})$**.** Takes as input the master secret key MK, and the identity ID of the user. It outputs the user's decryption key usk.

$\mathsf{Encrypt}_{\mathsf{IBE}}(\mathsf{PK}, \mathsf{ID}, M)$**.** Takes as input the public parameter PK, the identity of the recipient, and a message M to be encrypted. It outputs a ciphertext.

$\mathsf{Decrypt}_{\mathsf{IBE}}(\mathsf{usk}, c)$**.** Takes as input the user's decryption key and a ciphertext c. It outputs the decryption or $\perp$, if the ciphertext is not valid.

In [20] Shoup proposed a more efficient framework for public-key encryption, the so-called KEM/DEM, for *key encapsulation mechanism/data encapsulation method*. More recently, Bentahar *et al.* [7] extended this concept to the identity-based setting, and therefore proposed some constructions of $\mathcal{IB}$-$\mathcal{KEM}$ semantically secure. We will use the following formalism:

Definition 2 (Identity-Based Key Encapsulation Mechanism).
An $\mathcal{IB}$-$\mathcal{KEM}$ scheme is specified by the following four algorithms:

$\mathsf{Setup}_{\mathsf{IBK}}(1^\lambda)$**.** Takes as input a security parameter λ. It outputs the public parameters PK, as well as a master secret key MK.

$\mathsf{Extract}_{\mathsf{IBK}}(\mathsf{MK}, \mathsf{ID})$**.** Takes as input the master secret key MK and an identity ID of the user. It outputs the user's decryption key usk.

$\mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID})$**.** Takes as input the public parameters PK and the identity of the recipient. It outputs a pair (K, c), where K is the ephemeral session key and c is the encapsulation of that key.

$\mathsf{Decaps}_{\mathsf{IBK}}(\mathsf{usk}, c)$**.** Takes as input the user's decryption key usk and a ciphertext c. It outputs the key K encapsulated in c or $\perp$, if the ciphertext is not valid.
We also formally define the function $\mathsf{Decaps}_{\mathsf{IBK}}(\mathsf{ID}, c)$, which takes as input a user identity ID and a ciphertext c. It first extracts the decryption key usk associated to ID, and then decapsulates c under usk.

We first review the notion of semantic security for $\mathcal{IB}$-$\mathcal{KEM}$, then we deal with anonymity, and an additional security notion, that we call *identity-based non-malleability*.

2.2 Security Notions

We directly describe the security notions for identity-based key encapsulation mechanisms, but one can easily derive them for identity-based encryption.

Semantic Security. The semantic security formalizes the privacy of the key. The security game, in the strongest security model (*i.e.* chosen-ciphertext and full-identity attacks) is the following one:

Setup : The challenger runs the $\mathsf{Setup}_{\mathsf{IBK}}$ algorithm on input 1^λ to obtain the public parameters PK, and the master secret key MK. It publishes PK.

Find stage: The adversary $\mathcal{A}$ adaptively issues the following queries:

- $\mathsf{Extract}$ query on input an ID: The challenger runs the $\mathsf{Extract}$ algorithm on input $(\mathsf{MK}, \mathsf{ID})$, and provides the associated decryption key usk.
- Decaps query on input an ID and a ciphertext c: The challenger first extracts the decryption key for ID, and then decrypts the ciphertext c with this key. It outputs the resulting ephemeral key, or $\perp$.

$\mathcal{A}$ outputs a target identity ID^*, on which no $\mathsf{Extract}$-query has been asked.

Challenge: The challenger randomly gets $(K_0, c^*) \leftarrow \mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID}^*)$ and $(K_1, c') \leftarrow \mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID}^*)$. It flips a bit b and outputs (K_b, c^*).

Guess stage: The adversary can issue the same queries as in the Find stage, with the restriction that no $\mathsf{Extract}$-query on input ID^* and no Decaps-query on input (ID^*, c^*) can be asked. The adversary finally outputs its guess $b' \in \{0, 1\}$ for b.

We then define the advantage of $\mathcal{A}$ in breaking the *Semantic Security* of an $\mathcal{IB}$-$\mathcal{KEM}$ scheme with its ability in deciding whether it actually received the real ephemeral key associated to c^* or a random one. We denote this security notion by IND, which can thereafter be combined with various oracle accesses, in order to define selective/full-identity and chosen plaintext/ciphertext attacks. More formally, we want the advantage below, to be negligible:

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{IBK}}(\mathcal{A}) = 2 \times \Pr_b \left[\begin{array}{c} (\mathsf{PK}, \mathsf{MK}) \leftarrow \mathsf{Setup}_{\mathsf{IBK}}(1^\lambda); (\mathsf{ID}^*, s) \leftarrow \mathcal{A}_1(\mathsf{PK}) \\ (K_0, c^*) \leftarrow \mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID}^*); \\ (K_1, c') \leftarrow \mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID}^*) \\ b' \leftarrow \mathcal{A}_2(K_b, c^*, s) : b = b' \end{array} \right] - 1.$$

In the following, we will need a very weak notion, that we call *weak semantic security*, during which attack that adversary has to choose in advance the target identity ID^* (selective-ID), and has no oracle access at all: no Decaps queries, and no $\mathsf{Extract}$ queries.

Anonymity. Anonymity against $\mathcal{IBE}$ means that for a chosen plaintext, and given a ciphertext c encrypted under ID_0 or ID_1 of adversary's choice, the adversary should not be able to decide which identity has been involved. With an appropriate $\mathcal{DEM}$ encryption scheme, the key encapsulation anonymity version can be defined as follows:

Setup: The challenger runs $\mathsf{Setup}_{\mathsf{IBK}}$ on input 1^λ to obtain the public parameters PK, and the master secret key MK. It publishes PK.

Find stage: The adversary $\mathcal{A}$ adaptively issues $\mathsf{Extract}$ and Decaps queries. $\mathcal{A}$ outputs two identities ID_0, ID_1, on which no $\mathsf{Extract}$-query has been asked before.

Challenge: The challenger randomly selects $b \in \{0, 1\}$ and gets an encapsulated pair (K^*, c^*) under ID_b. It returns (K^*, c^*).

Guess stage: The adversary can issue the same queries as in the Find stage, subject to the restriction that no $\mathsf{Extract}$-query is allowed to be asked on ID_0 or ID_1, and no Decaps-query can be asked on input (ID_0, c^*), or (ID_1, c^*). It finally outputs its guess $b' \in \{0, 1\}$ for b.

We say that an $\mathcal{IB}$-$\mathcal{KEM}$ scheme provides key-anonymity if the advantage of $\mathcal{A}$ in deciding which identity is actually involved in the above experiment is negligible:

$$\mathsf{Adv}^{\mathsf{anon}}_{\mathsf{IBK}}(\mathcal{A}) = 2 \times \Pr_b \left[\begin{array}{c} (\mathsf{PK}, \mathsf{MK}) \leftarrow \mathsf{Setup}_{\mathsf{IBK}}(1^\lambda); \\ (\mathsf{ID}_0, \mathsf{ID}_1, s) \leftarrow \mathcal{A}_1(\mathsf{PK}) \\ (K^*, c^*) \leftarrow \mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID}_b); \\ b' \leftarrow \mathcal{A}_2(K^*, c^*, s) : b = b' \end{array} \right] - 1.$$

As already noticed, this anonymity notion does not provide any security with respect to the authority, since the above security notion assumes that the adversary has no idea about MK.

KwrtA-Anonymity. We therefore enhance the previous security model, in order to consider the authority as a possible adversary. However, it is clear that given (K^*, c^*), the authority can check the involved ID. We thus truncate the input to c^* only:

Find stage: The adversary generates (valid, see below) public parameters PK. $\mathcal{A}$ outputs PK and two identities ID_0, ID_1.

Challenge : The challenger randomly selects $b \in \{0,1\}$, and generates a ciphertext for ID_b, $(K^*, c^*) \leftarrow \mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID}_b)$. It outputs c^*.

Guess stage: The adversary finally outputs its guess $b' \in \{0,1\}$.

We say that an $\mathcal{IB}$-$\mathcal{KEM}$ scheme provides *Key Anonymity with respect to the Authority* (denoted $\mathsf{KwrtA\text{-}Anonymity}$) if the advantage of $\mathcal{A}$ in deciding which identity is involved in the experiment above is negligible:

$$\mathsf{Adv}^{\mathsf{kwrta-anon}}_{\mathsf{IBK}}(\mathcal{A}) = 2 \times \Pr_b \left[\begin{array}{c} (\mathsf{PK}, \mathsf{ID}_0, \mathsf{ID}_1, s) \leftarrow \mathcal{A}_1(1^\lambda) \text{ s.t. } \mathsf{Valid}_{\mathsf{IBK}}(\mathsf{PK}) \\ (K^*, c^*) \leftarrow \mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID}_b); \\ b' \leftarrow \mathcal{A}_2(c^*, s) : b = b' \end{array} \right] - 1.$$

We emphasis that in the above experiment, the adversary has to generate valid public parameters PK. Note that against $\mathsf{KwrtA\text{-}Anonymity}$ (*vs.* anonymity), on the one hand, the new adversary may know the master key MK, but on the other hand, it must make its decision from c^* only. Therefore, these two security notions are not really comparable. Furthermore, since the adversary generates PK, one has to be able to check the honest generation. In some cases, PK is a truly random value, without redundancy; in some other cases, appropriate redundancy should be proven. We thus define an additional algorithm:

$\mathsf{Valid}_{\mathsf{IBK}}(\mathsf{PK})$**.** Takes as input the public parameters PK, and checks whether they satisfy the required properties.

Identity-based Non-Malleability. In the application we will study later, a new security notion for identity-based encryption will appear. It basically states that when one sends a ciphertext to a user ID, one has no idea how user ID' will decrypt it, even for identities chosen by the adversary. This means that when one computes an encapsulation, it provides an ephemeral session key with a unique recipient, and not several secret keys with several partners. We define the *identity-based non-malleability* game as follows:

Setup: The challenger runs $\mathsf{Setup}_{\mathsf{IBK}}$ on input 1^λ to obtain the public parameters PK, and the master secret key MK. It publishes PK.

Attack: The adversary $\mathcal{A}$ adaptively issues $\mathsf{Extract}$ and Decaps queries, and outputs a ciphertext c, and two pairs (K_0, ID_0), and (K_1, ID_1).

The adversary wins this game if the two formal equalities hold:

$$K_0 = \mathsf{Decaps}_{\mathsf{IBK}}(\mathsf{ID}_0, c) \text{ and } K_1 = \mathsf{Decaps}_{\mathsf{IBK}}(\mathsf{ID}_1, c).$$

We thus define the success of $\mathcal{A}$ in breaking the Identity-based Non-Malleability of an $\mathcal{IB}$-$\mathcal{KEM}$ scheme by:

$$\mathsf{Succ}^{\mathsf{id\text{-}nm}}_{\mathsf{IBK}}(\mathcal{A}) = \Pr \left[\begin{array}{c} (\mathsf{PK}, \mathsf{MK}) \leftarrow \mathsf{Setup}_{\mathsf{IBK}}(1^\lambda); \\ (c, (K_0, \mathsf{ID}_0), (K_1, \mathsf{ID}_1)) \leftarrow \mathcal{A}(\mathsf{PK}) : \\ K_0 = \mathsf{Decaps}_{\mathsf{IBK}}(\mathsf{ID}_0, c) \wedge K_1 = \mathsf{Decaps}_{\mathsf{IBK}}(\mathsf{ID}_1, c) \end{array} \right].$$

Note that this security notion is for a normal user, and not for the authority itself. Indeed, it would clearly be incompatible with $\mathsf{KwrtA\text{-}Anonymity}$.

3 Anonymous and Non-Malleable $\mathcal{IB}$-$\mathcal{KEM}$

Since the first practical $\mathcal{IBE}$ schemes, new features, and new efficient/security criteria have been defined. An efficient anonymous $\mathcal{IBE}$ with a tight security proof in the standard model is one of the open problems. In this section, we first review some candidates, and then propose a new scheme that satisfies all the above requirements: semantic security, various anonymity notions and identity-based non-malleability.

3.1 Backgrounds on Pairings

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of large prime order p. We suppose that these two groups are equipped with a pairing, *i.e.* a non-degenerated and efficiently computable bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. In the following, we use multiplicative notation for $\mathbb{G}_1$ and $\mathbb{G}_2$: $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$, for all $a, b \in \mathbb{Z}_p$, and any $g_1 \in \mathbb{G}_1$ and $g \in \mathbb{G}_2$. For the sake of generality, we consider the asymmetric case, where $\mathbb{G}_1 \neq \mathbb{G}_2$, but most of the schemes below also apply in the symmetric setting, where $\mathbb{G}_1 = \mathbb{G}_2$.

3.2 Diffie-Hellman Assumptions

The co-CDH-Problem. Let g_1 and g_2 two generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. We define the co-Diffie-Hellman value $\mathsf{co\text{-}CDH}_{g_1,g_2}(u)$, for $u = g_1^x \in \mathbb{G}_1$, the element $v = g_2^x \in \mathbb{G}_2$.

The $\mathsf{co\text{-}CDH}_{\mathbb{G}_1,\mathbb{G}_2}$ problem can be formalized as follows: given $g_1, u \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, output $v = \mathsf{co\text{-}CDH}_{g_1,g_2}(u)$. We define the success probability of $\mathcal{A}$ in breaking the $\mathsf{co\text{-}CDH}_{\mathbb{G}_1,\mathbb{G}_2}$-problem as:

$$\mathsf{Succ}^{\mathsf{co-cdh}}_{\mathbb{G}_1,\mathbb{G}_2}(\mathcal{A}) = \Pr\left[g_1 \stackrel{R}{\leftarrow} \mathbb{G}_1; g_2 \stackrel{R}{\leftarrow} \mathbb{G}_2, x \stackrel{R}{\leftarrow} \mathbb{Z}_p; v \leftarrow \mathcal{A}(g_1, g_2, g_1^x) : v = g_2^x\right].$$

Note that when $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$, the $\mathsf{co\text{-}CDH}_{\mathbb{G},\mathbb{G}}$-problem is exactly the usual *Computational Diffie-Hellman Problem* in $\mathbb{G}$, which can still be difficult. However, the decisional version is easy, granted the pairing.

We can indeed define the $\mathsf{co\text{-}DH}_{\mathbb{G}_1,\mathbb{G}_2}$-language of the quadruples $(a, b, c, d) \in \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_2$, such that $d = \mathsf{co\text{-}CDH}_{a,b}(c)$.

The Common co-CDH-Problem. Given two elements, it is simple to complete a co-CDH-quadruple (g_1, g_2, u, v). However, finding two such quadruples with constraints may not be simple. We thus define a new problem, called the *Common* co-CDH-*Problem*, as follows: Given $g, h \in \mathbb{G}$, and $V \in \mathbb{G}_T$, output $k_0 \neq k_1 \in \mathbb{Z}_p$, $K_0, K_1 \in \mathbb{G}_T$ and a common $c \in \mathbb{G}$, such that:

$$(gh^{k_0}, V, c, K_0), (gh^{k_1}, V, c, K_1) \in \mathsf{co\text{-}DH}_{\mathbb{G},\mathbb{G}_T}.$$

We define the success of $\mathcal{A}$ in breaking the $\mathsf{Common\text{-}co\text{-}CDH}_{\mathbb{G},\hat{e}}$-*Problem* as:

$$\mathsf{Succ}^{\mathsf{common\text{-}co\text{-}cdh}}_{\mathbb{G},\hat{e}}(\mathcal{A}) = \Pr\left[\begin{array}{l} g, h \in \mathbb{G}; V \in \mathbb{G}_T; (c, k_0, k_1, K_0, K_1) \leftarrow \mathcal{A}(g, h, V): \\ \quad k_0 \neq k_1 \wedge (gh^{k_0}, V, c, K_0) \in \mathsf{co\text{-}DH}_{\mathbb{G},\mathbb{G}_T} \\ \quad \wedge (gh^{k_1}, V, c, K_1) \in \mathsf{co\text{-}DH}_{\mathbb{G},\mathbb{G}_T} \end{array}\right]$$

The CBDH-Problem. Diffie-Hellman variants have been proposed in groups equipped with pairings, and namely in the symmetric case: let g be a generator of $\mathbb{G}$. We define the Bilinear Diffie-Hellman value of g^x, g^y, g^z, for $x, y, z \in \mathbb{Z}_p$, in base g, the element $V = \hat{e}(g, g)^{xyz} \in \mathbb{G}_T$.

The $\mathsf{CBDH}_{\mathbb{G},\hat{e}}$ problem can be formalized as follows: given $g, X = g^x, Y = g^y, Z = g^z \in \mathbb{G}$, output $V = \hat{e}(g, g)^{xyz}$. We define the success probability of $\mathcal{A}$ in breaking the $\mathsf{CBDH}_{\mathbb{G},\hat{e}}$-problem as:

$$\mathsf{Succ}^{\mathsf{cbdh}}_{\mathbb{G},\hat{e}}(\mathcal{A}) = \Pr\left[g \stackrel{R}{\leftarrow} \mathbb{G}; x, y, z \stackrel{R}{\leftarrow} \mathbb{Z}_p; V \leftarrow \mathcal{A}(g, g^x, g^y, g^z) : v = \hat{e}(g, g)^{xyz}\right].$$

THE DBDH-PROBLEM. The decisional version can then be intractable too: given $g, X = g^x, Y = g^y, Z = g^z \in \mathbb{G}$, and $V \in \mathbb{G}_T$, decide whether $V = \hat{e}(g,g)^{xyz}$, or not. We define the advantage of $\mathcal{A}$ in breaking the $\mathsf{DBDH}_{\mathbb{G},\hat{e}}$-problem as:

$$\mathsf{Adv}^{\mathsf{dbdh}}_{\mathbb{G},\hat{e}}(\mathcal{A}) = \Pr\left[g \stackrel{R}{\leftarrow} \mathbb{G}; x,y,z \stackrel{R}{\leftarrow} \mathbb{Z}_p; V = \hat{e}(g,g)^{xyz} : 1 \leftarrow \mathcal{A}(g, g^x, g^y, g^z, V)\right] - \Pr\left[g \stackrel{R}{\leftarrow} \mathbb{G}; x,y,z \stackrel{R}{\leftarrow} \mathbb{Z}_p; V \stackrel{R}{\leftarrow} \mathbb{G}_T : 1 \leftarrow \mathcal{A}(g, g^x, g^y, g^z, V)\right].$$

THE SUCCESSIVE-POWER VERSION. For our scheme to be semantically secure, we will need a stronger variant of the above DBDH problem, given access to a sequence of powers, similarly to the Strong Diffie-Hellman problem [9]: More precisely, given g, g^x, g^y, g^z, and $g^{z/x}, g^{z/x^2}, \ldots, g^{z/x^q}$, as well as V, from some $V \in \mathbb{G}_T$, where q is a parameter, decide whether $V = \hat{e}(g,g)^{xyz}$, or a random element. We define the advantage of $\mathcal{A}$ in breaking the q-SP-$\mathsf{DBDH}_{\mathbb{G},\hat{e}}$-assumption as:

$$\mathsf{Adv}^{q\text{-}\mathsf{spdbdh}}_{\mathbb{G},\hat{e}}(\mathcal{A}) = \Pr\left[\begin{array}{l} g \stackrel{R}{\leftarrow} \mathbb{G}; x,y,z \stackrel{R}{\leftarrow} \mathbb{Z}_p; V = \hat{e}(g,g)^{xyz} : \\ \quad 1 \leftarrow \mathcal{A}(g, g^x, g^y, g^z, g^{z/x}, \cdots, g^{z/x^q}, V) \end{array}\right] - \Pr\left[\begin{array}{l} g \stackrel{R}{\leftarrow} \mathbb{G}; x,y,z \stackrel{R}{\leftarrow} \mathbb{Z}_p; V \stackrel{R}{\leftarrow} \mathbb{G}_T : \\ \quad 1 \leftarrow \mathcal{A}(g, g^x, g^y, g^z, g^{z/x}, \cdots, g^{z/x^q}, V) \end{array}\right].$$

It is clear that such a sequence of powers should not provide much information to the adversary. And thus, for any polynomial-time adversary $\mathcal{A}$, the above advantage is negligible. We provide the proofs that our two new problems are intractable for generic adversaries in the Appendix B.

3.3 Previous $\mathcal{IBE}$ Schemes

Let us review several $\mathcal{IBE}$, and see which properties they satisfy. For the sake of simplicity, for all of them, we review the key encapsulation mechanisms. In several schemes, we will need a deterministic map F from identities onto the group $\mathbb{G}$, possibly with parameter PK.

THE BONEH-FRANKLIN SCHEME [10]. In this scheme, $\mathsf{MK} = s \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and $\mathsf{PK} = g^s$. The map $F(\mathsf{ID})$ is independent of PK. This is a function onto $\mathbb{G}$, modeled as a random oracle in the security analysis. The ciphertext $c = g^r \in \mathbb{G}$ corresponds to the key $K = \hat{e}(F(\mathsf{ID}), \mathsf{PK})^r = \mathsf{BDH}_g(\mathsf{PK}, c, F(\mathsf{ID})) = \hat{e}(\mathsf{usk}_{\mathsf{ID}}, c)$, where $\mathsf{usk}_{\mathsf{ID}} = F(\mathsf{ID})^s = \text{co-}\mathsf{CDH}_{g,F(\mathsf{ID})}(\mathsf{PK}) \in \mathbb{G}$.

It is quite similar to the ElGamal encryption, and thus the *semantic security* relies on the $\mathsf{DBDH}_{\mathbb{G},\hat{e}}$, but against chosen-plaintext attacks only, in the random oracle model, even with access to the Extract-query, which is similar to the Boneh-Lynn-Shacham signature [11] (secure against chosen-message attacks under the $\mathsf{CDH}_{\mathbb{G}}$ problem).

Since the ciphertext is totally independent of the identity, this scheme is *KwrtA-anonymous*, in the information-theoretical sense. Nevertheless, the basic anonymity is similar to the semantic security, and relies on the $\mathsf{DBDH}_{\mathbb{G},\hat{e}}$. However, since the ciphertext does not involve the identity, it is easy to break the *identity-based non-malleability*: knowing r and $c = g^r$, one easily computes $K = \mathsf{BDH}_g(\mathsf{PK}, c, F(\mathsf{ID})) = \hat{e}(F(\mathsf{ID}), \mathsf{PK})^r$, for any ID of ones choice.

THE BONEH-BOYEN SCHEME [8]. In this scheme, $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$, $g, g_2, h \stackrel{R}{\leftarrow} \mathbb{G}$, and $\mathsf{PK} = (g, g_1 = g^\alpha, g_2, h)$, while $\mathsf{MK} = g_2^\alpha$. The map F_{PK} is defined by $F_{\mathsf{PK}}(\mathsf{ID}) = g_1^{\mathsf{ID}} \cdot h$. The ciphertext $c = (g^s, F_{\mathsf{PK}}(\mathsf{ID})^s)$ corresponds to the key

$$K = \hat{e}(g_1, g_2)^s = \hat{e}(c_1, \mathsf{usk}_2)/\hat{e}(\mathsf{usk}_1, c_2),$$

if one gets $\mathsf{usk}_{\mathsf{ID}} = (g^r, \mathsf{MK} \cdot F_{\mathsf{PK}}(\mathsf{ID})^r)$, for any $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$.

As above, the *semantic security* relies on the $\mathsf{DBDH}_{\mathbb{G},\hat{e}}$ assumption, in the standard model, but against selective-ID chosen-plaintext attacks, even with access to the Extract-query (the underlying signature scheme is selective-forgery secure against chosen-message attacks under the CBDH assumption).

However, because of the redundancy in the ciphertext, which matches with one identity only, this scheme is not *anonymous*: one just has to check, for a candidate ID, and a ciphertext $c = (c_1, c_2)$, whether $(g, F_{\mathsf{PK}}(\mathsf{ID}), c_1, c_2)$ is a Diffie-Hellman tuple, by $\hat{e}(c_1, F_{\mathsf{PK}}(\mathsf{ID})) \stackrel{?}{=} \hat{e}(c_2, g)$. Since this attack did not need a candidate key K, a fortiori, this scheme is not *KwrtA-anonymous*.

On the other hand, since the ciphertext focuses to a specific recipient, one has no idea how another ID' would decrypt it, because of its randomness r' in the decryption key: for wrong user, with $\mathsf{usk}' = (g^{r'}, g_2^{\alpha} F_{\mathsf{PK}}(\mathsf{ID}')^{r'})$, and $c = (g^s, F_{\mathsf{PK}}(\mathsf{ID}')^{s'})$ ($s' \neq s$ since ID' is not the intended recipient), $K' = K \times H^{r'}$, for $H \neq 1$, and r' totally random. Therefore, it is *identity-based non-malleable* in the information-theoretical sense.

The Gentry Scheme [16]. In 2006, two schemes have been proposed, with provable anonymity. Gentry's scheme is one of them: $g, h \stackrel{R}{\leftarrow} \mathbb{G}$ and $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$. The public parameters are $\mathsf{PK} = (g, g_1 = g^{\alpha}, h)$ and $\mathsf{MK} = \alpha$. The map F_{PK} is defined by $F_{\mathsf{PK}}(\mathsf{ID}) = g_1 \cdot g^{-\mathsf{ID}} = g^{\alpha - \mathsf{ID}}$. The ciphertext $c = (F_{\mathsf{PK}}(\mathsf{ID})^s, \hat{e}(g,g)^s)$ is the encapsulation of $K = \hat{e}(g,h)^s$, and thus, setting $(\mathsf{usk}_1, \mathsf{usk}_2) = (r, (hg^{-r})^{1/(\alpha - \mathsf{ID})})$, for any $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$, $K = \hat{e}(c_1, \mathsf{usk}_2) \cdot {c_2}^{\mathsf{usk}_1}$.

The scheme is *semantically secure* and *anonymous* against chosen plaintext attacks, even with access to the Extract-query, under the truncated decisional augmented bilinear Diffie-Hellman exponent assumption (see [16] for details).

However, the scheme is not *KwrtA-anonymous*, since using bilinear maps combined with the redundancy inside the ciphertext provides a test for any target identity ID', since knowing α, $\mathcal{A}$ can test whether

$$ {c_2}^{\alpha - \mathsf{ID}'} = e(g,g)^{s(\alpha - \mathsf{ID}')} \stackrel{?}{=} e(c_1, g) = e(g^{s(\alpha - \mathsf{ID}')}, g). $$

Since the ciphertext is specific to the recipient, $\mathcal{A}$ has no idea how an other ID' decrypts $c = (c_1 = F_{\mathsf{PK}}(\mathsf{ID}')^{s'}, c_2 = e(g,g)^s)$, since

$$ K' = \hat{e}(c_1, \mathsf{usk}_2') \cdot {c_2}^{\mathsf{usk}_1'} = K \cdot (\hat{e}(g,g)^{\mathsf{usk}_1'} / \hat{e}(g,h))^{s - s'}, $$

is a random element in $\mathbb{G}_T$. Thus, the scheme is *identity-based non-malleable* in the information-theoretical sense.

The Boyen-Waters scheme [13]. Boyen and Waters proposed another provably anonymous scheme: ω, t_1, t_2, t_3 and $t_4 \stackrel{R}{\leftarrow} \mathbb{Z}_p$ are set to be the master secret key and $\Omega = \hat{e}(g,g)^{t_1 \cdot t_2 \cdot \omega}, g, g_0, g_1, v_1 = g^{t_1}, v_2 = g^{t_2}, v_3 = g^{t_3}$ are the public parameters PK, with g a random generator of $\mathbb{G}$ and $g_0, g_1 \stackrel{R}{\leftarrow} \mathbb{G}$. The map F_{PK} is defined by $F_{\mathsf{PK}}(\mathsf{ID}) = g_0 \cdot \mathsf{ID}$. To encrypt a key, one chooses a random $s \in \mathbb{Z}_p$ and sets $K = \Omega^s$, its encapsulation has the following form: $c = (c_0, c_1, c_2, c_3, c_4)$, with $c_0 = F_{\mathsf{PK}}(\mathsf{ID})^s$, $c_1 = v_1^{s - s_1}$, $c_2 = v_2^{s_1}$, $c_3 = v_3^{s - s_2}$, and $c_4 = v_4^{s_2}$. To decapsulate the key, one has to compute

$$ \begin{aligned} K^{-1} = \Omega^{-s} &= \hat{e}(g,g)^{-\omega t_1 t_2 s} \\ &= \hat{e}(c_0, \mathsf{usk}_0) \times \hat{e}(c_1, \mathsf{usk}_1) \times \hat{e}(c_2, \mathsf{usk}_2) \times \hat{e}(c_3, \mathsf{usk}_3) \times \hat{e}(c_4, \mathsf{usk}_4) \end{aligned} $$

with $\mathsf{usk}_{\mathsf{ID}} = (\mathsf{usk}_0, \mathsf{usk}_1, \mathsf{usk}_2, \mathsf{usk}_3, \mathsf{usk}_4)$, where:

$$ \begin{aligned} \mathsf{usk}_0 &= g^{r_1 t_1 t_2 + r_2 t_3 t_4} \\ \mathsf{usk}_1 &= g^{-\omega t_2} F_{\mathsf{PK}}(\mathsf{ID})^{-r_1 t_2} \quad & \mathsf{usk}_2 &= g^{-\omega t_1} F_{\mathsf{PK}}(\mathsf{ID})^{-r_1 t_1} \\ \mathsf{usk}_3 &= F_{\mathsf{PK}}(\mathsf{ID})^{-r_2 t_4} & \mathsf{usk}_4 &= F_{\mathsf{PK}}(\mathsf{ID})^{-r_2 t_3} \end{aligned} $$

for any $r_1, r_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p$. This scheme is *semantically secure* under $\mathsf{DBDH}_{\mathbb{G},\hat{e}}$, and *anonymous* under the decision linear assumption (we do not give more details since this scheme is totally different from ours below. The reader is refereed to [13]). However, it is not *KwrtA-anonymous*: since knowing the master key and given a ciphertext $c = (c_0, c_1, c_2, c_3, c_4)$, one can decide for a target identity whether c_0, c_1, c_2 or/and c_0, c_3, c_4 is a linear tuple in basis v_0, v_1, v_2 and v_0, v_3, v_4 respectively.

Since the key is completely independent of the identity and c_0 is determined by the identity (among other elements), the same argument than for the two previous schemes holds: it is *identity-based non-malleable* in an information-theoretically sense.

Note that for all the above schemes, the public parameters consist of independent elements in appropriate groups. The validity check $\mathsf{Valid}_{\mathsf{IBK}}(\mathsf{PK})$ is thus trivial.

3.4 Our Scheme

None of the previous schemes satisfies both *KwrtA-anonymity* and *identity-based non-malleability*. In this section, we describe our scheme, and show that it achieves all the security properties: *semantic security, anonymity, KwrtA-anonymity* and *identity-based non-malleability*. For the sake of simplicity, we use a symmetric pairing:.

$\mathsf{Setup}_{\mathsf{IBK}}$**.** The setup algorithm chooses two random generators $g, h \in \mathbb{G}$, and a random exponent $\omega \in \mathbb{Z}_p$. It keeps this exponent as the master key $\mathsf{MK} = \omega$. The corresponding system parameters are: $\mathsf{PK} = (g, g_1 = g^\omega, h)$. It defines the identity-function: $F(\mathsf{ID}) = g_1 \cdot g^{\mathsf{ID}} = g^{\omega+\mathsf{ID}}$.
Note that, as above, the public parameters consist of independent elements in appropriate groups. The validity check $\mathsf{Valid}_{\mathsf{IBK}}(\mathsf{PK})$ is thus trivial.

$\mathsf{Extract}_{\mathsf{IBK}}(\mathsf{MK}, \mathsf{ID})$**.** To issue a private key for identity ID, the key extraction authority computes the private key, $\mathsf{usk}_{\mathsf{ID}} = h^{1/(\omega+\mathsf{ID})}$.

$\mathsf{Encaps}_{\mathsf{IBK}}(\mathsf{PK}, \mathsf{ID})$**.** In order to generate an ephemeral key with an identity ID, the algorithm chooses a random exponent $r \in \mathbb{Z}_p$, and creates the ciphertext as: $c = F(\mathsf{ID})^r$, that corresponds to the key $K = \hat{e}(g, h)^r$.

$\mathsf{Decaps}_{\mathsf{IBK}}(\mathsf{usk}_{\mathsf{ID}}, c)$**.** The decryption algorithm extracts the ephemeral key K from a ciphertext c by computing: $K = \hat{e}(\mathsf{usk}_{\mathsf{ID}}, c)$.

Correctness. Let us check the decryption process:

$$K = \hat{e}(\mathsf{usk}_{\mathsf{ID}}, c) = \hat{e}(h^{1/(\omega+\mathsf{ID})}, g^{r(\omega+\mathsf{ID})}) = \hat{e}(h, g)^r.$$

Semantic Security. It is worth to precise that we do not require to be able to simulate any oracle for making use of $\mathcal{IB}$-$\mathcal{KEM}$ schemes in the next section. The *weak semantic security* will be enough:

Theorem 3. *The weak semantic security of our scheme (under selective-ID, chosen-plaintext and no-identity attacks) relies on the* $\mathsf{DBDH}_{\mathbb{G},\hat{e}}$*-problem, in the standard model.*

Proof. Given u, $A = u^a$, $B = u^b$, $C = u^c$, and $V \in \mathbb{G}_T$ the input to the $\mathsf{DBDH}_{\mathbb{G},\hat{e}}$*-Problem*, and the target identity ID^*, we set $g = A = u^a$, $h = C = u^c = g^{c/a}$, $g_1 = u^t \cdot A^{-\mathsf{ID}^*} = u^{t-a\mathsf{ID}^*}$, and $c = B$. This implicitly defines $\mathsf{MK} = t/a - \mathsf{ID}^*$, for a randomly chosen $t \stackrel{R}{\leftarrow} \mathbb{Z}_p$. Therefore, $F_{\mathsf{PK}}(\mathsf{ID}^*) = g_1 g^{\mathsf{ID}^*} = u^t \cdot A^{-\mathsf{ID}^*} \cdot A^{\mathsf{ID}^*} = u^t$, and the randomness r of the challenge ciphertext $c = F_{\mathsf{PK}}(\mathsf{ID}^*)^r = u^{tr} = u^b = B$ is $r = b/t$. The corresponding encapsulated key should thus be

$$K = \hat{e}(h, g)^r = \hat{e}(u^c, u^a)^{b/t} = \hat{e}(u, u)^{abc/t}.$$

By letting $(V^{1/t}, c)$ be the output of the challenger, an adversary able to break the semantic security (without $\mathsf{Extract}$-queries) helps us to decide whether V is the Bilinear Diffie-Hellman value or not. □

In order to show the usual semantic security (under full-ID, but chosen-plaintext attacks), we have to be able to simulate the Extract-oracle, which thus requires additional inputs. But first, we modify a little bit the scheme, by using $H(\mathsf{ID})$, instead of ID in the above description, where H is a random oracle [5] onto $\mathbb{Z}_p$.

Theorem 4. *The semantic security of our scheme (by using $H(\mathsf{ID})$, instead of ID) under full-ID and chosen-plaintext (no* Decaps *queries) relies on the* successive-power *version, in the random oracle model.*

Proof. Given u, $A = u^a$, $B = u^b$, $C = u^c$, $C_i = C^{1/a^i}$, for $i = 1, \ldots, q$, and $V \in \mathbb{G}_T$ the input to the q-SP-DBDH$_{\mathbb{G},\hat{e}}$-problem, we first compute $\{V_i = \hat{e}(u,u)^{bc/a^i}\}_{i=0\ldots q}$, since $V_0 = \hat{e}(B,C)$, and $V_i = \hat{e}(B,C_i)$, for $i = 1, \ldots, q$. Then, we set $g = A = u^a$ and $g_1 = u^t \cdot A^{-x^*}$, for randomly chosen $t, x^* \stackrel{R}{\leftarrow} \mathbb{Z}_p$. This implicitly defines $\mathsf{MK} = t/a - x^*$. We also choose random elements $x_1, \ldots, x_q \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, and set $P(X) = \prod(tX + x_i)$, a polynomial of degree q, where the number of random oracle queries is $q+1$. We then set $h = C^{P(1/a)} = u^{cP(1/a)}$, which can be easily computed granted C, C_1, ..., C_q.

First, all the random oracle queries will be answered by an $x^* + x_i$, or x^* (for a unique randomly chosen query): we hope to assign x^* to $H(\mathsf{ID}^*)$, the target identity, which happens with probability $1/q$. Let us assume this situation:

- By definition, as above, $F_{\mathsf{PK}}(\mathsf{ID}^*) = g_1 g^{H(\mathsf{ID}^*)} = u^t \cdot A^{-x^*} \cdot A^{x^*} = u^t$;
- For all the other identities, $H(\mathsf{ID}_j) = x_j$, and then usk_j can be computed as

$$h^{1/(\mathsf{MK}+x^*+x_j)} = C^{P(1/a)/(\mathsf{MK}+x^*+x_j)} = C^{P(1/a)/(t/a+x_j)} = C^{P_j(1/a)},$$

  where P_j is a polynomial of degree $q-1$. Then usk_j can be easily computed granted C, C_1, ..., C_{q-1}. Hence the simulation of the Extract-oracle.

As above, the challenge ciphertext is set $c = B = u^b = F_{\mathsf{PK}}(\mathsf{ID}^*)^r$ for $r = b/t$. The corresponding encapsulated key should thus be

$$K = \hat{e}(g,h)^r = \hat{e}(u^a, u^{cP(1/a)})^{b/t} = (\hat{e}(u,u)^{abc})^{P(1/a)/t}.$$

Let us expand $P(X) = \sum_{i=0}^{i=q} p_i X^i$, and then

$$K = \hat{e}(u,u)^{abc \cdot p_0/t} \times \prod_{i=1}^{i=q} \hat{e}(u,u)^{bc/a^{i-1} \cdot p_i/t} = \left(\hat{e}(u,u)^{abc}\right)^{p_0/t} \times \prod_{i=1}^{i=q} V_{i-1}^{p_i/t}.$$

If $V = \hat{e}(u,u)^{abc}$, the correct key is $V^{p_0/t} \times \prod_{i=1}^{i=q} V_{i-1}^{p_i/t}$. In the random case, the same computation leads to a totally random key (note that $p_0 = \prod x_i \neq 0 \bmod p$). Then, by letting $(V^{p_0/t} \times \prod_{i=1}^{i=q} V_{i-1}^{p_i/t}, c)$ be the output of the challenger, an adversary able to break the semantic security helps us to decide whether V is the Bilinear Diffie-Hellman value or not. We thus break the q-SP-DBDH$_{\mathbb{G},\hat{e}}$-problem. □

Anonymity. The usual anonymity notion relies on the same assumption as the semantic security. Since the ciphertext consists of $c = F(\mathsf{ID})^r$, a random element in $\mathbb{G}$, whatever the identity ID. It is thus clearly *KwrtA-anonymous*, in the information-theoretical sense.

Theorem 5. *Our scheme is unconditionally KwrtA-anonymous.*

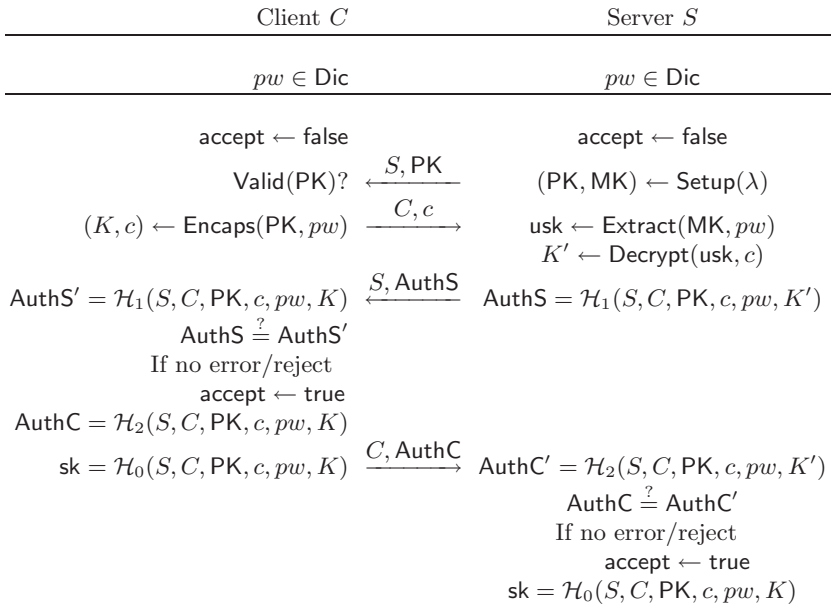

Fig. 1. IBK-PAKE: a Password-Authenticated Key-Exchange Protocol

IDENDITY-BASED NON-MALLEABILITY. Let us consider the ciphertext c, and its decryption with respect to ID_i for $i \in \{0, 1\}$. In the following, r_i is formally defined by $c = F(\mathsf{ID}_i)^{r_i}$, and $K_i = \hat{e}(g, h)^{r_i}$. Thus, the identity-based non-malleability relies on the intractability of finding $c, \{\mathsf{ID}_i, K_i\}$, with $\mathsf{ID}_0 \neq \mathsf{ID}_1$ such that $r_i = \log_{\hat{e}(g,h)}(K_i) = \log_{F(\mathsf{ID}_i)}(c)$. This thus leads to a solution of the *Common* co-CDH-*Problem*.

Theorem 6. *The identity-based non-malleability of our scheme relies on the* Common co-CDH-Problem *in groups* $\mathbb{G}$ *and* $\mathbb{G}_T$.

4 $\mathcal{IBK} - \mathcal{PAKE}$: Our PAKE Protocol

The previous sections focused on identity-based key encapsulation mechanisms, and new anonymity properties. We now show how a weakly semantically secure $\mathcal{IB}$-$\mathcal{KEM}$, that is both KwrtA-anonymous and identity-based non-malleable, can be used to build a password-authenticated key exchange.

4.1 Description of our Scheme

Our new scheme is generic. It basically consists in generating the session key using this $\mathcal{IB}$-$\mathcal{KEM}$, under the common password as the identity, see Figure 1. The other party can easily recover the session key. Security notions for semantic security and perfect forward secrecy follow from the (weak) semantic security and anonymity properties of the $\mathcal{IB}$-$\mathcal{KEM}$ scheme.

4.2 Security Analysis

Communication Model. We assume to have a fixed set of protocol *participants*, and each of them can be either a client or a server. They are all allowed to participate to several different, possibly concurrent,

executions of the key exchange protocol. We model this by allowing each participant an unlimited number of *instances* able to initiate or participate to executions of the protocol.

In the password-based scenario, the two parties share a low-entropy secret pw which is drawn from a small dictionary Dic. In the following, we assume that the distribution is uniform. More complex distributions could be considered.

We use the security model introduced by Bellare *et al.* [4], improved by Abdalla *et al.* [2] to consider the Real-or-Random security notion instead of the Find-then-Guess. In this model, the adversary $\mathcal{A}$ has the entire control of the network, which is formalized by allowing $\mathcal{A}$ to ask the following query, $\mathsf{Send}(U, m)$, that models $\mathcal{A}$ sending the message m to instance U. The adversary $\mathcal{A}$ gets back the response U generates in processing the message m according to the protocol. A query $\mathsf{Send}(U, \mathsf{INIT})$ initializes the key exchange algorithm, by activating the first player in the protocol.

From the original security model, we suppress the $\mathsf{Execute}$-queries. Even if they were important to model passive attacks *vs.* active attacks, we consider a stronger security model where the adversary always uses Send-queries, either for simply forwarding a flow generated by a honest user, or for modifying/manufacturing a flow. Thereafter, if the whole transcript of an execution of the protocol turns out to consist of forwarded flows only, this execution is then considered as a *passive* attack: it is similar to an $\mathsf{Execute}$-query in previous models [4]. If one flow has been modified or manufactured, the session corresponds to an *active* attack.

As a consequence, in addition to the usual security model with $\mathsf{Execute}$-queries, the adversary can adaptively decide, during an execution of the protocol, whether the session will correspond to a passive attack, or to an active one, and not from the beginning of the session only (as in [18]). An attack game will consist of a mix of passive and active attacks, in a concurrent manner.

However, as usual, we will be essentially interested in active attacks: q_{activeC} and q_{activeS} will, respectively, denote the number of active attacks in which the adversary played against the client and the server, respectively. We want to show that $q_{\mathsf{activeC}} + q_{\mathsf{activeS}}$ is an upper-bound on the number of passwords the adversary may have tried.

Security Notions. Two main security notions have been defined for key exchange protocols. The first is the semantic security of the key, which means that the exchanged key is unknown to anybody other than the players. The second one is unilateral or mutual authentication, which means that either one, or both, of the participants actually know the key. In the following, we focus on the semantic security, also known as *AKE Security*.

The semantic security of the session key is modeled by an additional query $\mathsf{Test}(U)$. Since we are working in the Real-or-Random scenario, this Test-query can be asked as many times as the adversary $\mathcal{A}$ wants, but to *fresh* instances only. The freshness notion captures the intuitive fact that a session key is not "obviously" known to the adversary. More formally an instance is said to be fresh if it has successfully completed execution and

1. Neither it nor its partner was corrupted before the session started
2. or, the attack, on this session, was passive.

Two instances are partners if they run a key exchange protocol together. This is formally modeled by the notion of session ID: the session ID is a string defined from the transcript (usually, it consists of the first flows, sent and received), and two instances are partners if they share the same session IDs.

The Test-query is answered as follows: a (private) coin b has been flipped once for all at the beginning of the attack game, if $b = 1$ (Real), then the actual session key sk is sent back, if $b = 0$ (Random), or a

random value is returned. Note that for consistency reasons, in the random case, the same random value is sent to partners.

We denote the AKE advantage as the probability that $\mathcal{A}$ correctly guesses the value of b with its output b': $\mathsf{Adv}^{\mathsf{ake}}(\mathcal{A}) = 2\Pr[b = b'] - 1$.

The adversary will also have access to the Corrupt-query that leaks the password: it is useful to model the perfect forward secrecy. The latter notion means that a session key remains secret even after the leakage of the long-term secret.

Security Result. For our protocol, we can state the following security result, which proof can be found in the Appendix A.

Theorem 7 (AKE Security). *Let us consider an Identity-Based Key Encapsulation Mechanism* $\mathsf{IBK} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encaps}, \mathsf{Decaps})$ *that is weakly semantically secure (selective-ID, chosen-plaintext attacks and no* Extract*-queries), anonymous,* KwrtA*-anonymous, and identity-based non-malleable, then our protocol* IBK-PAKE*, provides semantic security and perfect forward secrecy:*

$$Adv^{\mathsf{ake}}_{\mathsf{ibk-pake}}(\mathcal{A}) \leq 4 \times \frac{q_{\mathsf{active}}}{N} + \mathsf{negl}(),$$

where $q_{\mathsf{active}} = q_{\mathsf{activeC}} + q_{\mathsf{activeS}}$ *is the number of active attacks and* N *is the size of the dictionary.*

5 Conclusion

In this paper, we have first introduced two new security notions for identity-based key encapsulation mechanisms: the first one is an enhancement of the usual anonymity, the second one formalizes a kind on non-malleability, with respect to the recipient identity.

Then, we proposed the first scheme that is full-ID semantically secure against chosen-message attacks, and that achieves our new security notions.

We furthermore showed that these new security notions could be useful for identity-based schemes as a tool: we provided a new framework for password-authenticated key exchange, with an identity-based key encapsulation mechanism as a core sub-routine.

Acknowledgment

We would like to thank the anonymous referees for their fruitful comments. This work has been partially supported by European Commission through the IST Program under Contract IST-2002-507932 ECRYPT, and by the French ANR-07-SESU-008-01 PAMPA Project.

References

1. Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *CRYPTO 2005*, *LNCS* 3621, pages 205–222. Springer, 2005.
2. Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-based authenticated key exchange in the three-party setting. In *PKC 2005*, *LNCS* 3386, pages 65–84. Springer, 2005.
3. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT 2001*, *LNCS* 2248, pages 566–582. Springer, 2001.
4. Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT 2000*, *LNCS* 1807, pages 139–155. Springer, 2000.

5. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93*, pages 62–73. ACM Press, 1993.
6. Steven M. Bellovin and Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *1992 IEEE Symposium on Security and Privacy*, pages 72–84. IEEE Computer Society Press, 1992.
7. Kamel Bentaha, Pooya Farshim, John Malone-Lee, and Nigel P. Smart. Generic constructions of identity-based and certificateless KEMs. *Journal of Cryptology*, 21(2):178–199, April 2008.
8. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *EUROCRYPT 2004*, *LNCS* 3027, pages 223–238. Springer, 2004.
9. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *EUROCRYPT 2004*, *LNCS* 3027, pages 56–73. Springer, 2004.
10. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO 2001*, *LNCS* 2139, pages 213–229. Springer, 2001.
11. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In *ASIACRYPT 2001*, *LNCS* 2248, pages 514–532. Springer, 2001.
12. Dan Boneh and Brent R. Waters. Conjunctive, subset, and range queries on encrypted data. Cryptology ePrint Archive, 2006.
13. Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *CRYPTO 2006*, *LNCS* 4117, pages 290–307. Springer, 2006.
14. Dario Catalano, David Pointcheval, and Thomas Pornin. IPAKE: Isomorphisms for password-based authenticated key exchange. *Journal of Cryptology*, 20(1):115–149, January 2007.
15. Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Cryptography and Coding, 8th IMA International Conference*, *LNCS* 2260, pages 360–363. Springer, 2001.
16. Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT 2006*, *LNCS* 4004, pages 445–464. Springer, 2006.
17. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
18. David Pointcheval and Sébastien Zimmer. Multi-factor authenticated key exchange. In *ACNS 2008*, *LNCS* 5037, pages 277–295. Springer, 2008.
19. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO '84*, *LNCS* 196, pages 47–53. Springer, 1985.
20. Victor Shoup. Using hash functions as a hedge against chosen ciphertext attack. In *EUROCRYPT 2000*, *LNCS* 1807, pages 275–288. Springer, 2000.
21. Victor Shoup. ISO 18033-2: An emerging standard for public-key encryption. December 2004. Final Committee Draft.

A Proof of Theorem 7

This proof goes with a sequence of games, starting from $\mathbf{G}_0$ which represents the real attack game (with a mix of passive and active attacks). In all the games, we focus on the two following events:

- EvS (for semantic security). This event occurs if the adversary correctly guesses the bit b involved in the Test-queries;
- EvA (for authentication). This event occurs if an instance accepts a session with no partner (with the adversary).

Our goal is to estimate the probability of event EvS in the real attack game. To this aim, we modify step by step the simulation, to end up with a trivial game the adversary cannot win. We will denote by Δ_n the difference of the probabilities of any event Ev in the game $\mathbf{G}_n$ and the previous one. Note that such an event must be observable when the distance is computationally bounded, but can be any event when the distance is statistically bounded.

Game $\mathbf{G}_0$: This is the real attack game. By definition, we have:

$$\mathsf{Adv}^{\mathsf{ake}}_{\mathsf{ibk-pake}}(\mathcal{A}) = 2\Pr[\mathsf{EvS}_0] - 1.$$

Game $\mathbf{G}_1$: In this game, we simulate Send-queries as shown in Figure 2. The Test-queries are answered according to the bit b: if $b = 1$ (Real), then the value sk computed during the simulation is returned; if $b = 0$ (Random), $\mathcal{H}'_0(S, C, \mathsf{PK}, c)$ is returned, where $\mathcal{H}'_0$ is a private random oracle. This is a truly random value. But since the input is the session ID, thus satisfies the constraint that Test-queries on two partners give the same random value, in the random case. The goal of the following games will be to make the simulations of sk in the real and random cases identical.

Game $\mathbf{G}_2$: In a first step, we cancel games in which some collision appears. The former are related to the session IDs, which collisions would be problematic, since it would lead to identical session keys to different sessions, the latter are just for making the security analysis simpler:

- collisions on the partial trancript (S, C, PK, c).
- collisions on the output of $\mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_0$.

$$\Delta_2 \leq \frac{q_{\mathcal{H}}^2}{2^\ell} + \gamma q_{\mathsf{session}}^2,$$

where $q_{\mathcal{H}}$ is the number of hash queries, and q_{session} the number of sessions. We furthermore denote by γ an upper-bound on the guessing probabilities of PK and c, and by ℓ the length of the hash function outputs.

Game $\mathbf{G}_3$: In this game, we deal with sessions in which the adversary remained passive during the first two flows. We denote by q_{passive} the number of such sessions. We will denote by OG the passive flows, for *Oracle-Generated*. More precisely, in the following, $\mathsf{OG}(C, n)$, resp. $\mathsf{OG}(S, n)$, means that the n-th flow, received by the client, resp. by the server, has been generated by the simulation: we thus know the random coins, but the adversary does not.

The adversary may actively participate to the protocol during the third or fourth flows, and trying to build a valid authenticator in the name of either the server (with AuthS) or the client (with AuthC). Intuitively, since the adversary has no idea about the secret information, MK and K, the probability to build valid authenticators is negligible, even if it would know the password.

We thus modify, in these sessions only (which can be determined before applying the following modifications), the way the authenticators and session keys are computed. More precisely, we replace the public oracles $\mathcal{H}_i$ (for $i = 0, 1, 2$) by private ones $\mathcal{H}'_i$ (for $i = 0, 1, 2$), on truncated inputs: $(S, C, \mathsf{PK}, c, pw, K)$ is replaced by the session ID that is public, (S, C, PK, c). We claim that such a modification cannot be noticed by the adversary, under the *semantic security* of the $\mathcal{IB}$-$\mathcal{KEM}$ scheme.

Let us first formally modify the simulation. We thereafter prove the above claim.

▶**Rule S2**$^{(3)}$

- if $\neg\mathsf{Corrupt} \wedge \mathsf{OG}(C, 1) \wedge \mathsf{OG}(S, 2)$, compute $\mathsf{AuthS} = \mathcal{H}'_1(S, C, \mathsf{PK}, c)$.
- else,
  get $\mathsf{usk} \leftarrow \mathsf{Extract}(\mathsf{MK}, pw)$;
  decrypt the ciphertext, $K' \leftarrow \mathsf{Decaps}(\mathsf{usk}, c)$;
  compute $\mathsf{AuthS} = \mathcal{H}_1(S, C, \mathsf{PK}, c, pw, K')$.

▶**Rule C2**$^{(3)}$

- if $\neg\mathsf{Corrupt} \wedge \mathsf{OG}(C, 1) \wedge \mathsf{OG}(S, 2)$, compute $\mathsf{AuthS}' = \mathcal{H}'_1(S, C, \mathsf{PK}, c)$,
  $\mathsf{sk} = \mathcal{H}'_0(S, C, \mathsf{PK}, c)$, and $\mathsf{AuthC} = \mathcal{H}'_2(S, C, \mathsf{PK}, c)$
- else, compute $\mathsf{AuthS}' = \mathcal{H}_1(S, C, \mathsf{PK}, c, pw, K)$,
  $\mathsf{sk} = \mathcal{H}_0(S, C, \mathsf{PK}, c, pw, K)$, $\mathsf{AuthC} = \mathcal{H}_2(S, C, \mathsf{PK}, c, pw, K)$

Send-queries to C

We answer to the Send-queries to a C-instance as follows:

- A $\mathsf{SendC}(C; S, \mathsf{PK})$ query is processed according to the following rule:
  - ▶**Rule C1**$^{(1)}$
    - Get $(K, c) \leftarrow \mathsf{Encaps}(\mathsf{PK}, pw)$.

  Answer with c.
- A $\mathsf{SendC}(C; S, \mathsf{AuthS})$-query is processed according to the following rule:
  - ▶**Rule C2**$^{(1)}$
    - Compute $\mathsf{AuthS}' = \mathcal{H}_1(C, S, \mathsf{PK}, c, pw, K)$, $\mathsf{sk} = \mathcal{H}_0(C, S, \mathsf{PK}, c, pw, K)$, and $\mathsf{AuthC} = \mathcal{H}_2(C, S, \mathsf{PK}, c, pw, K)$

  Check wether AuthS' is equal to AuthS.
  If the equality holds,
  accept with session key sk, and answer with AuthC;
  else reject.
- All the other cases are ignored.

Send-queries to S

We answer to the Send-queries to an S-instance as follows:

- A $\mathsf{SendS}(S; \mathsf{INIT})$-query is processed according to the following rules:
  - ▶**Rule S1**$^{(1)}$
    - Get $(\mathsf{PK}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$.

  Answer with PK.
- A $\mathsf{SendS}(S; C, c)$-query is processed according to the following rules:
  - ▶**Rule S2**$^{(1)}$
    - Get $\mathsf{usk} \leftarrow \mathsf{Extract}(\mathsf{MK}, pw)$;
    - Decapsulate the key, $K' \leftarrow \mathsf{Decaps}(\mathsf{usk}, c)$;
    - Compute $\mathsf{AuthS} = \mathcal{H}_1(C, S, \mathsf{PK}, c, pw, K')$.

  Answer with AuthS.
- A $\mathsf{SendS}(S; C, \mathsf{AuthC})$-query is processed according to the following rules:
  - ▶**Rule S3**$^{(1)}$
    - Compute $\mathsf{AuthC}' = \mathcal{H}_2(C, S, \mathsf{PK}, c, pw, K')$, and $\mathsf{sk} = \mathcal{H}_0(C, S, \mathsf{PK}, c, pw, K')$

  Check wether AuthC' is equal to AuthC.
  If the equality holds,
  accept with session key sk, and terminate;
  Else reject.
- All the other cases are ignored.

$\mathcal{H}$

For a hash-query $\mathcal{H}_n(q)$ (resp. $\mathcal{H}'_n(q)$), such that a record (n, q, r) appears in $\Lambda_{\mathcal{H}_1}$ (resp. $\Lambda_{\mathcal{H}'}$), the answer is r. Otherwise one chooses a random element r in the appropriate range, answers with it, and adds the record (n, q, r) to $\Lambda_{\mathcal{H}_1}$ (resp. $\Lambda_{\mathcal{H}'}$).

Fig. 2. Simulation of the $\mathsf{IBK\text{-}PAKE}$ protocol

►**Rule S3**$^{(3)}$
- if $\neg\mathsf{Corrupt} \wedge \mathsf{OG}(C,1) \wedge \mathsf{OG}(S,2) \wedge \mathsf{OG}(C,3)$,
  compute $\mathsf{AuthC} = \mathcal{H}'_2(S, C, \mathsf{PK}, c)$ and $\mathsf{sk} = \mathcal{H}'_0(S, C, \mathsf{PK}, c)$
- else, compute $\mathsf{AuthC}' = \mathcal{H}_2(S, C, \mathsf{PK}, c, pw, K')$
  and $\mathsf{sk} = \mathcal{H}_0(S, C, \mathsf{PK}, c, pw, K')$.

Games $\mathbf{G}_3$ and $\mathbf{G}_2$ are identical except if the adversary remarks the use of private oracles. But this is possible if and only if $\mathcal{A}$ asks queries to $\mathcal{H}_i$ on some transcripts (S, C, PK, c), with the appropriate pair (pw, K). We call this event AskHP (for *Passive*, since the adversary was passive during the first flows). In order to estimate the probability of event AskHP, we define an auxiliary game that fills the gap between the two games $\mathbf{G}_2$ and $\mathbf{G}_3$ with the semantic security of the $\mathcal{IB}$-$\mathcal{KEM}$ scheme: if $\mathcal{A}$ can distinguish both executions, then we can construct a *selective-ID, chosen-plaintext adversary against the semantic security* of the underlying $\mathcal{IB}$-$\mathcal{KEM}$ scheme, without any $\mathsf{Extract}$-query (our previous weak semantic security notion).

We thus suppose by contradiction that there exists $\mathcal{A}'$ that can distinguish the games $\mathbf{G}_3$ and $\mathbf{G}_2$, with non-negligeable advantage. We construct an adversary $\mathcal{B}$ against the semantic security of the $\mathcal{IB}$-$\mathcal{KEM}$ scheme.

Game $\mathbf{G}_{2.1}$: We make the modification, from game $\mathbf{G}_2$, for one session only, the i-th one, for i chosen between 1 and q_{session}, before any corruption:

►**Rule S1**$^{(2.1)}$
- if this is the i-th session,
  let the challenger run $\mathsf{Setup}(1^\lambda)$, with the target identity pw, to get PK;
- else,
  get $(\mathsf{PK}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda)$.

►**Rule C1**$^{(2.1)}$
- if this is the i-th session and $\mathsf{OG}(C,1)$,
  ask for pw to the challenger, that answers with a pair (K, c) (if $d = 1$, $K = K_1$ really corresponds to c, but if $d = 0$, $K = K_0$ is independent of c).
- else,
  get $(K, c) \leftarrow \mathsf{Encaps}(\mathsf{PK}, pw)$.

►**Rule S2**$^{(2.1)}$
- if this is the i-th session, and $\mathsf{OG}(C,1) \wedge \mathsf{OG}(S,2)$,
  compute $\mathsf{AuthS} = \mathcal{H}_1(S, C, \mathsf{PK}, c, pw, K)$.
- else,
  get $\mathsf{usk} \leftarrow \mathsf{Extract}(\mathsf{MK}, pw)$;
  decrypt the ciphertext, $K' \leftarrow \mathsf{Decaps}(\mathsf{usk}, c)$;
  compute $\mathsf{AuthS} = \mathcal{H}_1(S, C, \mathsf{PK}, c, pw, K')$.

Note that in the real case ($d = 1$), we are exactly as in game $\mathbf{G}_2$. In the random case, it is clear that the probability to ask $\mathcal{H}_i$ on appropriate K for this session is negligible (upper-bounded by $\gamma q_{\mathcal{H}}$, where γ is an upper-bound on the guessing probability for K) since no information leaks about it.

Thus,

$$\Delta_3 \leq \Pr[\mathsf{AskHP}_2] \leq q_{\mathsf{session}} \times \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{IBK}}(t) + \gamma q_{\mathcal{H}} \leq \mathsf{negl}().$$

Game $\mathbf{G}_4$: In this game, we just formally modify the simulation, but without any modification to the view of the adversary, since we take a random ciphertext c according to the appropriate distribution, that may depend on the password. Our goal will then be to make it independent of the password, when there is no corruption (the flag $\mathsf{Corrupt}$ specifies whether the adversary got the password or not with a corruption query). We define two distributions $\mathcal{C}_{\mathsf{PK}}$ and $\mathcal{C}_{\mathsf{PK},pw}$ as follows:

$$\mathcal{C}_{\mathsf{PK}} = \{\mathsf{Encrypt}(\mathsf{PK}, pw, m) \mid m \in \mathcal{M}\}, \text{and}$$

$$\mathcal{C}_{\mathsf{PK},pw} = \{\mathsf{Encrypt}(\mathsf{PK}, pw, m) \mid m \in \mathcal{M}, pw \in \mathsf{Dic}\}.$$

▶**Rule C1**$^{(4)}$
- if $\neg\mathsf{Corrupt} \wedge \mathsf{OG}(C, 1)$, get $c \leftarrow \mathcal{C}_{\mathsf{PK},pw}$;
- else, get $(K, c) \leftarrow \mathsf{Encaps}(\mathsf{PK}, pw)$.

Since K is not needed in the later flows, in the case that $\mathsf{OG}(C, 1)$ and $\mathsf{OG}(S, 2)$, this simulation does not change anything, $\Delta_4 = 0$.

Game $\mathbf{G}_5$: We now make the simulation of the client, that receives an oracle-generated public key PK, independent of the password. Our new game will be indistinguishable from the previous one, under the *KwrtA-anonymity* of the underlying $\mathcal{IB}$-$\mathcal{KEM}$ scheme.

▶**Rule C1**$^{(5)}$
- if $\neg\mathsf{Corrupt} \wedge \mathsf{OG}(C, 1)$, get $c \leftarrow \mathcal{C}_{\mathsf{PK}}$;
- else, get $(K, c) \leftarrow \mathsf{Encaps}(\mathsf{PK}, pw)$.

The distance between $\mathbf{G}_5$ and $\mathbf{G}_4$ is negligible except if $\mathcal{A}$ is able to distinguish the distribution $\mathcal{C}_{\mathsf{PK}}$ from the distribution of $\mathcal{C}_{\mathsf{PK},pw}$, where pw is possibly known to the adversary, but PK has been generated by a trusted party. Using a classical hybrid sequence of games, we can easily show that

$$\Delta_5 \leq q_{\mathsf{passive}} \times \mathsf{Adv}^{\mathsf{anon}}\mathsf{IBK}(t) \leq \mathsf{negl}().$$

Note that in the previous hybrid sequence of games (in game $\mathbf{G}_{2.1}$), we had to start the simulation specific to the i-th hybrid game before knowing whether it would be a passive session or not, hence the factor q_{session}. In this hybrid sequence, the modification starts only when we already know that it is a passive session, hence the factor q_{passive} only.

Game $\mathbf{G}_6$: In this game, we consider attacks in which the adversary starts to be active in the second flow: it chooses the ciphertext c. We will show that there is a small chance to send the correct authenticator AuthC, except if the correct password has been guessed, and used in the ciphertext c: at most one password can be tested.

More precisely, we replace the hash oracles $\mathcal{H}_i$ by private ones as soon as PK has been oracle-generated.

▶**Rule S2**$^{(6)}$
- if $\neg\mathsf{Corrupt} \wedge \mathsf{OG}(C, 1)$, compute $\mathsf{AuthS} = \mathcal{H}'_1(S, C, \mathsf{PK}, c)$.
- else,
  get $\mathsf{usk} \leftarrow \mathsf{Extract}(\mathsf{MK}, pw)$;
  decrypt the ciphertext, $K' \leftarrow \mathsf{Decaps}(\mathsf{usk}, c)$;
  compute $\mathsf{AuthS} = \mathcal{H}_1(S, C, \mathsf{PK}, c, pw, K')$.

►**Rule C2**$^{(6)}$

– if $\neg\mathsf{Corrupt} \wedge \mathsf{OG}(C,1)$, compute $\mathsf{AuthS}' = \mathcal{H}'_1(S,C,\mathsf{PK},c)$,
$\mathsf{sk} = \mathcal{H}'_0(S,C,\mathsf{PK},c)$, $\mathsf{AuthC} = \mathcal{H}'_2(S,C,\mathsf{PK},c)$
– else, compute $\mathsf{AuthS}' = \mathcal{H}_1(S,C,\mathsf{PK},c,pw,K)$,
$\mathsf{sk} = \mathcal{H}_0(S,C,\mathsf{PK},c,pw,K)$, $\mathsf{AuthC} = \mathcal{H}_2(S,C,\mathsf{PK},c,pw,K)$

Games $\mathbf{G}_6$ and $\mathbf{G}_5$ are identical except if the adversary remarks the use of the private oracles $\mathcal{H}'_i$ in sessions in which he chooses the ciphertext c, but not PK. Such an event happens only if $\mathcal{A}$ asks the $\mathcal{H}$ queries on some transcript (S,C,PK,c) for the appropriate pair (pw,K), *i.e.* such that $K = \mathsf{Decaps}(pw,c)$, for the password. This means that there exists at least one password pw such that $(S,C,\mathsf{PK},c,pw,K) \in \Lambda_{\mathcal{H}_1}$, and $K = \mathsf{Decaps}(pw,c)$. We call this event AskHS_6. To make our analysis clearer, we distinguish the two following cases:

– for all the sessions (S,C,PK,c), where c is chosen by the adversary, there is at most one password π such that $(S,C,\mathsf{PK},c,\pi,K = \mathsf{Decaps}(\pi,c)) \in \Lambda_{\mathcal{H}_1}$. We call this event AskHSU_6. If it occurs, then at most one password is tested by active attacks against the server. Our goal is to show that:

$$\Pr[\mathsf{AskHSU}_6] \leq \frac{q_{\mathsf{activeS}}}{N}.$$

– for some session (S,C,PK,c), there are at least two passwords π_1,π_2 such that $(S,C,\mathsf{PK},c,\pi_i,K_i = \mathsf{Decaps}(\pi_i,c)) \in \Lambda_{\mathcal{H}_1}$. We call this event AskHSM. If this event occurs, then $\mathcal{A}$ can construct a tuple $\{c,(\pi_1,K_1),(\pi_2,K_2)\}$ such that $K_1 = \mathsf{Decaps}(\pi_1,c)$ and $K_2 = \mathsf{Decaps}(\pi_2,c)$, which contradicts the identity-based non-malleability of the underlying $\mathcal{IB}$-$\mathcal{KEM}$ scheme:

$$\Pr[\mathsf{AskHSM}_6] \leq \mathsf{Adv}^{\mathsf{id\text{-}nm}}_{\mathsf{IBK}}(t).$$

Hence,

$$\Delta_6 \leq \Pr[\mathsf{AskHS}_6] \leq \mathsf{Adv}^{\mathsf{id\text{-}nm}}_{\mathsf{IBK}}(t) + \Pr[\mathsf{AskHSU}_6].$$

Game $\mathbf{G}_7$: In this game, we now consider the active attacks against the client, from the beginning of the session: the adversary sends the first flow, it thus controls PK, and possibly knows MK. As in game $\mathbf{G}_5$, we want to make the generation of the ciphertext c independent of the password, even in this case. However, we have to make the following computations independent of K first: we replace the hash oracles by private oracles in all the sessions where nobody is corrupted:

►**Rule S2**$^{(7)}$

– if $\neg\mathsf{Corrupt}$, compute $\mathsf{AuthS} = \mathcal{H}'_1(S,C,\mathsf{PK},c)$;
– else,
get $\mathsf{usk} \leftarrow \mathsf{Extract}(\mathsf{MK},pw)$;
decrypt the ciphertext, $K' \leftarrow \mathsf{Decaps}(\mathsf{usk},c)$;
compute $\mathsf{AuthS} = \mathcal{H}_1(S,C,\mathsf{PK},c,pw,K')$.

►**Rule C2**$^{(7)}$

– if $\neg\mathsf{Corrupt}$, compute $\mathsf{AuthS}' = \mathcal{H}'_1(S,C,\mathsf{PK},c)$,
$\mathsf{sk} = \mathcal{H}'_0(S,C,\mathsf{PK},c)$, $\mathsf{AuthC} = \mathcal{H}'_2(S,C,\mathsf{PK},c)$;
– else, compute $\mathsf{AuthS}' = \mathcal{H}_1(S,C,\mathsf{PK},c,pw,K)$,
$\mathsf{sk} = \mathcal{H}_0(S,C,\mathsf{PK},c,pw,K)$, $\mathsf{AuthC} = \mathcal{H}_2(S,C,\mathsf{PK},c,pw,K)$

►**Rule S3**$^{(7)}$
- if $\neg$Corrupt, compute $\mathsf{AuthC} = \mathcal{H}'_2(S, C, \mathsf{PK}, c)$ and $\mathsf{sk} = \mathcal{H}'_0(S, C, \mathsf{PK}, c)$;
- else, compute $\mathsf{AuthC}' = \mathcal{H}_2(S, C, \mathsf{PK}, c, pw, K')$ and $\mathsf{sk} = \mathcal{H}_0(S, C, \mathsf{PK}, c, pw, K')$.

Games $\mathbf{G}_7$ and $\mathbf{G}_6$ are identical except when PK is not oracle-generated: all public oracles are replaced by private ones. $\mathcal{A}$ can see the difference only if it asks $\mathcal{H}_1$ on some transcript (S, C, PK, c) for an appropriate pair (pw, K), such that $K = \mathsf{Decaps}(pw, c)$ in $\Lambda_{\mathcal{H}_1}$. This query is required for a valid authenticator AuthS.

As we have canceled executions with collisions from game $\mathbf{G}_2$, for each authenticator AuthS sent by the adversary, there is at most one tuple $(S, C, \mathsf{PK}, c, pw, K)$ such that $\mathsf{AuthS} = \mathcal{H}_1(S, C, \mathsf{PK}, c, pw, K)$. Such an event is denoted AskHC.

$$\Delta_7 \leq \Pr[\mathsf{AskHC}_7].$$

Game $\mathbf{G}_8$: Since we still use the password for defining the distribution of c, we cannot say anything about the probability of events AskHC and AskHSU. So, as in game $\mathbf{G}_5$, we now make the generation of the ciphertext c independent of the password, even in the case that PK is generated by the adversary. Since the adversary chooses the public key PK, the distance now involves the *KwrtA-anonymity* of the $\mathcal{IB}$-$\mathcal{KEM}$ scheme:

►**Rule C1**$^{(8)}$
- if $\neg$Corrupt, get $c \leftarrow \mathcal{C}_{\mathsf{PK}}$;
- else, get $(K, c) \leftarrow \mathsf{Encaps}(\mathsf{PK}, pw)$.

The distance between $\mathbf{G}_8$ and $\mathbf{G}_7$ is negligible except if $\mathcal{A}$ is able to distinguish the distribution $\mathcal{C}_{\mathsf{PK}}$ from the distribution of $\mathcal{C}_{\mathsf{PK},pw}$, where pw is possibly known to the adversary, and PK has been chosen by the adversary. Using a similar hybrid sequence of games as done in Game $\mathbf{G}_5$, but when the adversary knows MK, we can easily show that

$$\Delta_8 \leq q_{\mathsf{activeC}} \times \mathsf{Adv}^{\mathsf{kwrta-anon}}_{\mathsf{IBK}}(t) \leq \mathsf{negl}().$$

Game $\mathbf{G}_9$: First note that unless a corruption happened, the authenticators and the session keys are computed using the private oracles, without the password in the input. Note that the password is no longer used in any other stage of the simulation unless there has been a corruption:

- the server generates public parameters,
- the client generates a ciphertext in $\mathcal{C}_{\mathsf{PK}}$.

We can thus choose the password at the very end only, or when a corruption happens: $\Delta_9 = 0$. And then, we can evaluate whether events AskHC or AskHSU were raised or not:

- Event AskHSU means that for each session (S, C, PK, c), where c is chosen by the adversary (active attack against the server, there is at most one appropriate pair (pw, K). And thus, only one password can raise this event for each session against the server:
$$\Pr[\mathsf{AskHSU}_9] \leq \frac{q_{\mathsf{activeS}}}{N}$$
- Event AskHC means that $\mathcal{A}$ sent an authenticator AuthS (active attack against the client) that includes the good password. But only one value for the password can raise this event for each session against the client, since collisions on $\mathcal{H}_1$ were canceled:
$$\Pr[\mathsf{AskHC}_9] \leq \frac{q_{\mathsf{activeC}}}{N}.$$

Game $\mathbf{G}_{10}$: In order to conclude, we need to replace hash oracles by the private ones (at least for the session keys) for all the sessions that are fresh. We already dealt with all the cases, except when the password has been corrupted. In the security model, we also included passive sessions as fresh sessions, even when players are corrupted.

▶**Rule S2**$^{(10)}$
- if $\neg\mathsf{Corrupt} \vee (\mathsf{OG}(C,1) \wedge \mathsf{OG}(S,2))$,
  compute $\mathsf{AuthS} = \mathcal{H}'_1(S, C, \mathsf{PK}, c)$;
- else,
  get $\mathsf{usk} \leftarrow \mathsf{Extract}(\mathsf{MK}, pw)$;
  decrypt the ciphertext, $K' \leftarrow \mathsf{Decaps}(\mathsf{usk}, c)$;
  compute $\mathsf{AuthS} = \mathcal{H}_1(S, C, \mathsf{PK}, c, pw, K')$.

▶**Rule C2**$^{(10)}$
- if $\neg\mathsf{Corrupt} \vee (\mathsf{OG}(C,1) \wedge \mathsf{OG}(S,2))$,
  compute $\mathsf{AuthS}' = \mathcal{H}'_1(S, C, \mathsf{PK}, c)$,
  $\mathsf{sk} = \mathcal{H}'_0(S, C, \mathsf{PK}, c)$, $\mathsf{AuthC} = \mathcal{H}'_2(S, C, \mathsf{PK}, c)$;
- else, compute $\mathsf{AuthS}' = \mathcal{H}_1(S, C, \mathsf{PK}, c, pw, K)$,
  $\mathsf{sk} = \mathcal{H}_0(S, C, \mathsf{PK}, c, pw, K)$, $\mathsf{AuthC} = \mathcal{H}_2(S, C, \mathsf{PK}, c, pw, K)$

▶**Rule S3**$^{(10)}$
- if $\neg\mathsf{Corrupt} \vee (\mathsf{OG}(C,1) \wedge \mathsf{OG}(S,2))$,
  compute $\mathsf{AuthC} = \mathcal{H}'_2(S, C, \mathsf{PK}, c)$
  and $\mathsf{sk} = \mathcal{H}'_0(S, C, \mathsf{PK}, c)$;
- else, compute $\mathsf{AuthC}' = \mathcal{H}_2(S, C, \mathsf{PK}, c, pw, K')$
  and $\mathsf{sk} = \mathcal{H}_0(S, C, \mathsf{PK}, c, pw, K')$.

Games $\mathbf{G}_{10}$ and $\mathbf{G}_9$ are identical except when the two first flows are oracle-generated and the password is compromised: we replace the hash oracles by private oracles. But exactly as in game $\mathbf{G}_3$, the distinction relies on the weak semantic security of the $\mathcal{IB}$-$\mathcal{KEM}$ scheme:

$$\Delta_{10} \leq q_{\mathsf{session}} \times \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{IBK}}(t) + \gamma q_{\mathcal{H}} \leq \mathsf{negl}().$$

Since in the latter game, all the session keys are computed with a private oracle, there is no difference between the real and the random cases for the Test-queries:

$$\Pr[S_{10}] = \frac{1}{2}.$$

B Analysis in the Generic Model

B.1 The Common co-CDH-Problem

Let us first recall the Common co-CDH-Problem: given $g, h \in \mathbb{G}$, and $V \in \mathbb{G}_T$, output $c \in \mathbb{G}$, $k_0 \neq k_1 \in \mathbb{Z}_p$, and $K_0, K_1 \in \mathbb{G}_T$ such that:

$$gh^{k_i} = \mathsf{co\text{-}CDH}_{K_i,c}(V) \text{ for } i = 0, 1.$$

We define the success of $\mathcal{A}$ in breaking the $\mathsf{Common\text{-}co\text{-}CDH}_{\mathbb{G},\hat{e}}$-*Problem*, denoted by $\mathsf{Succ}^{\mathsf{common\text{-}co\text{-}cdh}}_{\mathbb{G},\hat{e}}(\mathcal{A})$ as:

$$\Pr\left[\begin{array}{l} g, h \xleftarrow{R} \mathbb{G}; V \in \mathbb{G}_T; (c, k_0, k_1, K_0, K_1) \leftarrow \mathcal{A}(g, h, V) : \\ \quad k_0 \neq k_1 \wedge gh^{k_0} = \mathsf{co\text{-}CDH}_{K_0,c}(V) \wedge gh^{k_1} = \mathsf{co\text{-}CDH}_{K_1,c}(V) \end{array}\right].$$

Theorem 8. *Let $\mathcal{A}$ be an adversary that makes at most q group operation queries (internal laws in $\mathbb{G}$ or $\mathbb{G}_T$, or pairing operations). On inputs $g, h \in \mathbb{G}$, and $V \in \mathbb{G}_T$, the probability that $\mathcal{A}$ outputs a solution (k_0, k_1, K_0, K_1, c) to the Common* co-CDH*-Problem is bounded by*

$$\frac{(3q+4)^2+3}{p} \leq \mathcal{O}\left(\frac{q^2}{p}\right).$$

Proof. Let $\mathcal{A}$ be an adversary against the Common co-CDH-Problem. We define a simulator $\mathcal{B}$ that emulates the group oracles: $\mathcal{B}$ maintains two lists L_1 and L_T of polynomials $L_1 = \{(F_{1,i}, \xi_{1,i}), i = 1, \cdots, t_1\}$ and $L_T = \{(F_{T,i}, \xi_{T,i}), i = 1, \cdots, t_T\}$ such that at step t, $t_1 + t_T \leq 3 \cdot t + 4$. The entries $\xi_{1,i}, \xi_{T,i}$ are set to be distinct random strings and are used to represent elements in $\mathbb{G}$ and $\mathbb{G}_T$ respectively. At the beginning of the game, $\mathcal{B}$ just sets two polynomials $F_{1,0} = 1$ and $F_{1,1} = x_1$, which refer to a generator g and a random element $h = g^{x_1}$ in $\mathbb{G}$, respectively. Similarly, $\mathcal{B}$ defines two polynomials $F_{T,0} = 1$ and $F_{T,1} = X_1$ associated to elements $U = e(g, g)$ and $V = e(g, g)^{X_1}$ in $\mathbb{G}_T$.

For any oracle query, $\mathcal{B}$ updates the lists L_1 and L_T:

- **Group Operation in $\mathbb{G}$:** when $\mathcal{A}$ asks for the addition of two elements in $\mathbb{G}$, it gives two representations ξ_i and ξ_j. Theses two strings are either associated to the polynomials $F_{1,i}, F_{1,j}$ ($(F_{1,i}, \xi_i)$ and $(F_{1,j}, \xi_j)$ are in L_1) or one defines a new variable $x_{1,i}$ and set $F_{1,i} = x_{1,i}$ associated to ξ_i and thus adds $(F_{1,i}, \xi_i)$ to L_1 (or for index j). We thus assume that $(F_{1,i}, \xi_i)$ and $(F_{1,j}, \xi_j)$ are in L_1.
  Then, it computes the sum of the polynomials, $F_{1,k} = F_{1,i} + F_{1,j}$. If the resulting polynomial $F_{1,k}$ already appears in the list for some index $l \leq t_1$, then it sets $\xi_{1,k} \leftarrow \xi_{1,l}$, else it chooses a new random string in $\{0,1\}^{\log_2 p}$ for $\xi_{1,k}$. Note that group operations in $\mathbb{G}$ result in multivariate polynomials of degree at most one in variables $x_1, \cdots, x_m$, for some integer $m \leq t_1$.
- **Pairing:** when $\mathcal{A}$ requests a pairing query. It gives two representations $\xi_{1,i}$ and $\xi_{1,j}$. As above, by possibly setting the undefined elements, we can assume that $(F_{1,i}, \xi_i)$ and $(F_{1,j}, \xi_j)$ are in L_1. Then, $\mathcal{B}$ computes the product of the polynomials, $F_{T,k} = F_{1,i} \cdot F_{1,j}$. If the resulting polynomial already appears in the list for some index $l \leq t_T$, then it sets $\xi_{T,k} \leftarrow \xi_{T,l}$, else it chooses a new random string $\xi_{T,k}$ in $\{0,1\}^{\log_2 p}$ for $F_{T,k}$.
  Since we know that polynomials in L_1 are of degree 1 in the variables $x_1, \ldots$, the polynomials we create with this simulation are of degree 2 in the same variables.
- **Group operation in $\mathbb{G}_T$:** when $\mathcal{A}$ asks for the addition of two elements in $\mathbb{G}_T$, it gives two representations ξ_i and ξ_j. As above, by possibly setting the undefined elements (and new variables X_i or X_j), we can assume that $(F_{T,i}, \xi_i)$ and $(F_{T,j}, \xi_j)$ are in L_T. Then, $\mathcal{B}$ computes the sum of the polynomials, $F_{T,k} = F_{T,i} + F_{T,j}$. If the resulting polynomial already appears in the list for some index $l \leq t_T$, then it sets $\xi_{T,k} \leftarrow \xi_{T,l}$, else it chooses a new random string $\xi_{T,k}$ in $\{0,1\}^{\log_2 p}$ for $F_{T,k}$.
  In the previous simulation, we created polynomials in L_T of degree 2 in the variables $x_1, \ldots$. We can thus add these polynomials: they remain polynomials of degree 2 in the variables $x_1, \ldots$. We can also add these polynomials with the initial polynomials $F_{T,1}, F_{T,2}, \ldots$ and the new variables X_i: polynomials of degree 1 in the variables $X_1, \ldots$.

As a consequence, any polynomial F in L_T can be split in two polynomials $A \in \mathbb{Z}_p[x_1, \cdots, x_m]$ (of degree 2) and $B \in \mathbb{Z}_p[X_1, \cdots, X_n]$ (of degree 1) such that $F = A + B$.

Note that for each group operation query, the oracle adds at most three new variables in the list. Thus if q is the number of queries we have $t_1 + t_T \leq 3q + 4$.

To evaluate the success of any adversary in distinguishing the above simulation from the real oracles, one has to define the event raised in case of deviation. This happens if the evaluations of two polynomials on

the initial vector $(x_1, \ldots, X_1, \ldots)$ refer to the same value: the oracles would output the same representation whereas our simulation just compares the polynomials and would thus output different representations. More precisely, the simulation can be detected if there exists a pair of polynomials (F, F') such that for a random choice of $x_1, \cdots, x_n, X_1, \cdots, X_m$ in $\mathbb{Z}_p$,

$$F(x_1, \cdots, x_n, X_1, \cdots, X_m) = F'(x_1, \cdots, x_n, X_1, \cdots, X_m) \text{ whereas } F \neq F'.$$

Since polynomials are of degree at most 2, this can happen with probability less than $2/p$ for each pair of polynomials: after q queries, the probability that the adversary distinguishes the two executions is bounded by $2 \cdot (3q+4)^2/p$.

Unless the adversary $\mathcal{A}$ detects the simulation, it terminates by outputting a tuple $(k_0, k_1, \xi_0^T, \xi_1^T, \xi_c^1)$, where k_0, k_1 are in $\mathbb{Z}_p$. $\mathcal{B}$ retrieves, in the list, the polynomials associated to ξ_0^T (representation of K_0), ξ_1^T (representation of K_1) and ξ_c^1 (representation of c), if they exist. Otherwise, as before, it adds new variables. Let thus F_0, F_1 and P be the polynomials associated to ξ_0^T, ξ_1^T and ξ_c^1 respectively: $P \in \mathbb{Z}_p[x_1, x_2, \cdots, x_m]$ of degree one, and $F_i \in \mathbb{Z}_p[x_1, \cdots, x_m, X_1, \cdots, X_n]$ of degree two. More precisely, as noted before, we can split $F_i = A_i + B_i$, with $A_i \in \mathbb{Z}_p[x_1, x_2, \cdots, x_m]$ of degree two, and $B_i \in \mathbb{Z}_p[X_1, \cdots, X_n]$ of degree one.

If $\mathcal{A}$ is successful, this means that for some β_i, we have:

$$c^{\beta_i} = g h^{k_i} \text{ and } V^{\beta_i} = K_i$$

The equalities above implies the following ones:

$$\begin{cases} \beta_i \cdot P(x_1, x_2, \cdots, x_n) = 1 + k_i x_1 \\ X_1 \cdot \beta_i = A_i(x_1, x_2, \cdots, x_n) + B_i(X_1, \cdots, X_m) \end{cases}$$

After substitution, we obtain

$$(A_i(x_1, x_2, \cdots, x_m) + B_i(X_1, \cdots, X_n)) \cdot P(x_1, x_2, \cdots, x_m) - (1 + k_i x_1) \cdot X_1 = 0.$$

At this point, either the success probability of the adversary is negligible (the above polynomial is non-zero), or

$$A_i(x_1, x_2, \cdots, x_m) = 0, \qquad B_i(X_1, X_2, \cdots, X_n) = \beta_i \cdot X_1$$

where β_i is now known to be a constant. Since P is a common polynomial, one gets

$$(1 + k_1 x_1) \cdot \beta_0 - (1 + k_0 x_1) \cdot \beta_1 = 0.$$

Again, either the success probability of the adversary is negligible (the above polynomial is non-zero), or $\beta_0 = \beta_1$ and $k_1 \beta_0 = k_0 \beta_1$, which implies that $k_0 = k_1$. However, a successful attack does not allow that, which concludes the proof.

C Analysis of the Successive-Power Problem.

The Successive-Power problem is the following: given g, g^x, g^y, g^z, and $g^{z/x}$, g^{z/x^2}, …, g^{z/x^q}, as well as V, from some $V \in \mathbb{G}_T$, where q is a parameter, decide whether $V = \hat{e}(g,g)^{xyz}$, or V is a random element of $\mathbb{G}_T$.

Theorem 9. *Let $\mathcal{A}$ be an adversary that makes at most t group operation queries. On input $g, g^x, g^y, g^{\frac{z}{x^i}}$, for $i \in \{0, \cdots, q\}$, the advantage of $\mathcal{A}$ in distinguishing the distribution of $V = \hat{e}(g,g)^{xyz}$ from the random distribution in $\mathbb{G}_T$ is bounded by*

$$\frac{(3t+q+7)^2}{p} \leq \mathcal{O}\left(\frac{t^2+q^2}{p}\right)$$

Proof. As in previous proof, we construct an algorithm $\mathcal{B}$ that interacts with $\mathcal{A}$, using lists of pairs $L_1 = \{(F_{1,i}, \xi_{1,i})\}$ and $L_T = \{(F_{T,i}, \xi_{T,i})\}$, but this time, we use fractions of polynomials. It starts with $F_{1,1} = 1, F_{1,2} = X, F_{1,3} = Y, F_{1,i} = \frac{Z}{X^{i-4}}$ for $i = \{4, \cdots, q+4\}$, and $F_{T,1} = 1$, $F_{T,2} = T_0$, $F_{T,3} = T_1$. X, Y, Z are unknown variables. For a random bit b, T_b is also a really new unknown variable, whereas $T_{1-b} = XYZ$ (but considered as an independent variable too. The adversary has to guess b.

When $\mathcal{A}$ terminates, it outputs its guess b', and then $\mathcal{B}$ chooses a random assignment $x, y, z, t_b \in \mathbb{Z}_p$, for X, Y, Z, and T_b but sets $T_{1-b} = xyz$.

In the simulated game, the advantage of the adversary is clearly zero: all the polynomials built during the simulation are independent to XYZ.

One thus have just to evaluate the probability the adversary can detect that it is interacting with a simulator: after t queries, the number of polynomials is upper-bounded by $3t+q+7$, which concludes the proof.