On Constructing Certificateless Cryptosystems from Identity Based Encryption

Benoît Libert* and Jean-Jacques Quisquater

UCL, Microelectronics Laboratory, Crypto Group
Place du Levant, 3, B-1348, Louvain-La-Neuve, Belgium.
{benoit.libert,jean-jacques.quisquater}@uclouvain.be

Abstract. Certificateless cryptography (CL-PKC) is a concept that aims at enjoying the advantages of identity based cryptography without suffering from its inherent key escrow. Several methods were recently suggested to generically construct a certificateless encryption (CLE) scheme by combining identity based schemes with ordinary public key cryptosystems. Whilst the security of one of these generic compositions was proved in a relaxed security model, we show that all them are insecure against chosen-ciphertext attacks in the strongest model of Al-Riyami and Paterson. We show how to easily fix these problems and give a method to achieve generic CLE constructions which are provably CCA-secure in the random oracle model. We finally propose a new efficient pairing-based scheme that performs better than previous proposals without precomputation. We also prove its security in the random oracle model.

Keywords. Certificateless encryption, provable security, bilinear maps.

1 Introduction

In 2003, Al-Riyami and Paterson [2] invented a paradigm called certificateless public key cryptography (CL-PKC) which is intermediate between identity-based [27, 12] and traditional PKI-supported cryptography. The concept was introduced to suppress the inherent key-escrow property of identity-based cryptosystems (ID-PKC) without losing their most attractive advantage which is the absence of digital certificates and their important management overhead.

Independently of [2] and a bit earlier, Gentry [22] introduced a different but related concept named certificate based encryption (CBE) for which a signature analogue was studied in [24]. This approach is closer to the context of a traditional PKI model as it involves a certification authority (CA) providing an efficient implicit certification service for clients' public keys.

Although very different at first glance, the CBE and CLE concepts were first argued [2] to be closely related and both constructions of [2, 22] use the properties of pairings. A subsequent work of Yum and Lee considered the relations between identity-based (IBE), certificate based (CBE) and certificateless encryption schemes (CLE) and established a result of essential equivalence [31]

* This author thanks the DGTRE's First Europe Program in Belgium.

between the three primitives but this result does not hold for the strongest security model developed in [2] for CLE schemes. The same authors also proposed generic constructions of certificateless signatures [30] and encryption schemes [29] but only established the security of their designs in security models that are seemingly undermined w.r.t. the original model considered in [2] for the public key encryption case.

A more recent work [3] thoroughly investigated the connections between the CLE and CBE paradigms by proposing a simplified definition and a revised security model for certificate based encryption before proving that any secure certificateless encryption (CLE) scheme can be turned into a secure CBE in the amended model.

Among other related results, we mention a paper [16] describing a somewhat similar scheme to [3], another work [9] that investigates identity-based and certificateless extensions of key encapsulation mechanisms. Both works [9, 16] considered a model of security which is noticeably weaker (albeit realistic in practice) than the original one [2]. A very recent paper by Baek et al. [4] also showed how to devise a certificateless encryption scheme without pairings. The latter construction enjoys a better efficiency than pairing-based proposals [2, 3, 16] but is supported by a weaker security model and prevents users from generating their public key independently from the system's authority. Finally, Dent and Kudla [17] investigated the feasibility of provably secure CLE schemes in the standard model and ruled out the use of some particular proof techniques for achieving this purpose in accordance with intuitive arguments given in [16].

The contribution of the present paper to the area of certificateless cryptography is two-fold. It first identifies some weaknesses in generic constructions independently considered in [1] and [29]. It shows that one of these flaws is also present in the second provably secure CLE scheme of Al-Riyami and Paterson [3] where it can be very easily fixed. The paper then explains how to obtain generic constructions which are provably secure in the random oracle model. It does so by first giving a generic random oracle-using conversion to turn any CLE scheme which is only secure against chosen-plaintext attacks into an IND-CCA scheme in the full model of Al-Riyami and Paterson [2].

The second contribution of the paper is to describe a new efficient pairing-based scheme yielding some advantages over previous constructions [2, 3, 16, 9]: its encryption operation does not require to compute a pairing (only the decryption algorithm does) and is thus generally faster than in previous proposals [2, 3, 16, 9]. The security proof of the new scheme is nevertheless obtained under a stronger computational assumption than for previous schemes in the literature.

In the forthcoming sections of this paper, we first review the formal definition and adversarial model of CLE schemes in section 2. Section 3 illustrates the power of their security model by showing how several generic constructions studied so far are insecure in it. We explain in section 4 how to repair them and we prove the security of the fixed constructions in the random oracle model. Our new certificateless cryptosystem is then depicted in section 5 where security proofs in the random oracle model are detailed.

2 Preliminaries

We now recall the components of a certificateless encryption scheme before detailing the relevant formal security model [2].

2.1 Definition of certificateless encryption (CLE)

Definition 1. *A certificateless encryption scheme (CLE) is a 7-tuple of algorithms which are the following:*

Setup: *is a probabilistic algorithm run by a Key Generation Center (KGC), that, given a security parameter k, returns a randomly chosen master key* mk *and a list of public parameters* params.

Partial-Private-Key-Extract: *is a possibly probabilistic algorithm, run by the KGC, that takes as input a user's identifier* ID_A *and the master key* mk *to return his/her partial private key* d_A.

Set-Secret-Value: *is a probabilistic algorithm that, given a list of public parameters* params*, returns a randomly chosen secret value* x_A *for that user. This algorithm and the next two are performed by the user himself.*

Set-Private-Key: *is a deterministic private key generation algorithm that, given public parameters* params*, a user's partial private key* d_A *and secret value* x_A*, outputs a private key* S_A.

Set-Public-Key: *is a deterministic public key generation algorithm that, given public parameters* params *and a user's secret value* x_A*, computes his/her public key* pk_A*. The latter's well-formedness (i.e. its belonging to a specific group or set) must be publicly verifiable given* params.

Encrypt: *is a probabilistic algorithm taking as input a plaintext m, parameters* params*, a receiver's identity* ID_A *and his public key* pk_A *to produce a ciphertext* $C = \mathsf{Encrypt}(m, \mathsf{params}, \mathsf{ID}_A, \mathsf{pk}_A)$.

Decrypt: *is a deterministic algorithm that, given a ciphertext C, a list of public paramaters* params *and user* ID_A*'s private key, outputs a plaintext m or a distinguished symbol* $\perp$.

For completeness, it is obviously required that $\mathsf{Decrypt}(C, \mathsf{params}, S_A) = m$ *whenever* $C = \mathsf{Encrypt}(m, \mathsf{params}, \mathsf{ID}_A, \mathsf{pk}_A)$ *for all messages* $m \in \mathcal{M}$ *and public keys* $\mathsf{pk}_A = \mathsf{Set\text{-}Public\text{-}Key}(\mathsf{params}, x_A)$ *for which the matching private key is* $S_A = \mathsf{Set\text{-}Private\text{-}Key}(\mathsf{params}, \mathsf{Partial\text{-}Private\text{-}Key\text{-}Extract}(\mathsf{ID}_A), x_A)$ *and the secret value is* $x_A = \mathsf{Set\text{-}Secret\text{-}Value}(\mathsf{params})$.

Unlike Setup and Partial-Private-Key-Extract that are run by a Key Generation Center (KGC), algorithms Set-Secret-Value, Set-Private-Key and Set-Public-Key are executed by the user whose private key remains hidden from the KGC.

The recent pairing-free scheme of Baek et al. [4] fits a slightly different model where users have to obtain their partial private key and a partial public key before generating their full public key. This approach is closer to the "self-certified" paradigm [23] which is another approach suggested by Girault in 1991 to use public key cryptography without traditional digital certificates and without involving an escrow authority.

2.2 Security model

In [2], two kinds of adversaries are distinguished against CLE schemes. A Type I adversary ignores the KGC's master key but can replace public keys of arbitrary identities with other public keys of her choosing. Such an adversarial behavior seems natural as, in the absence of digital certificates, anyone can alter public directories by replacing public keys without being caught or detected. As attackers against IBE schemes (recalled in appendix A), Type I adversaries can also obtain partial and full private keys of arbitrary identities.

In contrast, a Type II adversary knows the KGC's master key (and does not need a partial key exposure oracle) and may still obtain full private keys for arbitrary identities but is disallowed to replace public keys during the game.

For both types of adversaries, depending on the strength of the attack, we may or may not provide them with an oracle decrypting arbitrary ciphertexts using the private key associated with arbitrary identities.

In the chosen-ciphertext scenario, the authors of [2] consider decryption oracles that should be able (thanks to suitable knowledge extractors) to output consistent answers even for identities whose public key has been replaced and for which they do not know the new private key. The latter requirement might look too strong but it may be argued that decryption queries involving identities of replaced public key are far more useful to a Type I attacker (especially when the latter does not know the private key associated with the new public key).

In the security analysis of generic constructions in section 3.1, we will illustrate the importance of considering adversaries who replace public keys instead of merely corrupting their owner and learning his/her secret value.

Definition 2. *A CLE scheme is IND-CCA secure if no probabilistic polynomial time (PPT) adversary $\mathcal{A}$ of Type I or II has a non-negligible advantage in the following game:*

1. *Given a security parameter k, the challenger runs* $\mathsf{Setup}(k)$ *and then delivers the resulting parameters* params *to $\mathcal{A}$ who also receives the master key* mk *if she is of Type II. Otherwise,* mk *is kept secret.*
2. *$\mathcal{A}$ is given access to*
   - *a public key broadcast oracle* Public-Key-Broadcast *taking as input identities and returning the matching public keys.*
   - *a partial key exposure oracle* Partial-Private-Key-Extract *(if she is of Type I as such an oracle is useless otherwise) returning partial private keys associated with users' identities.*
   - *a private key exposure oracle* Private-Key-Extract *revealing private keys of entities whose public key was not replaced.*
   - *a decryption oracle* Decrypt *which, given a ciphertext and an identity* (C, ID)*, returns the decryption of C using the private key corresponding to the current value of entity* ID*'s public key.*

   If $\mathcal{A}$ is of Type I, she has also access to a public key replacement oracle Public-Key-Replace *which, given an identifier* ID *and a valid public key* pk'*, replaces user* ID*'s public key with* pk'.

3. *$\mathcal{A}$ outputs messages m_0, m_1 together with an identity* ID^* *of uncorrupted private key. If $\mathcal{A}$ is of Type I,* ID^* *may not have been submitted to both oracles* Public-Key-Replace *and* Partial-Private-Key-Extract*. She gets a ciphertext* $C^* = \mathsf{Encrypt}(m_b, \mathsf{params}, \mathsf{ID}^*, \mathsf{pk}^*)$ *where* $b \xleftarrow{R} \{0,1\}$ *and* pk^* *is the public key currently associated with* ID^*.
4. *She then issues a new sequence of queries but is not permitted to ask for the decryption of C^* for the combination* $(\mathsf{ID}^*, \mathsf{pk}^*)$ *under which m_b was encrypted at step 3. Moreover no private key exposure query can be made on* ID^* *at any time and, in a Type I attack,* ID^* *may not be submitted to both oracles* Public-Key-Replace *and* Partial-Private-Key-Extract.
5. *$\mathcal{A}$ eventually outputs a bit b' and wins if $b' = b$. As usual, her advantage is* $Adv_{CLE}^{\mathsf{ind-cca}}(\mathcal{A}) := 2 \times Pr[b' = b] - 1$.

The above definition captures a chosen-ciphertext scenario. The weaker chosen-plaintext security (or IND-CPA security) notion is formalized by a similar game where attackers have no decryption oracles.

The security models considered in [4, 16, 29] are weaker in that they disallow Type I attackers to ever extract the partial private key of the target entity. In contrast, the above model allows them to do so as long as they do not additionally replace the associated public key. Besides, the models of [16, 29] only require challengers to correctly handle decryption queries for entities whose public key was not replaced. From here on, we will stick to the model of definition 2.

3 On the power of public key replacement oracles

This section underlines the strength of the security model captured by definition 2. We first explain simple attacks that compromise the security of some generic constructions of certificateless encryption. We then exemplify that allowing decryption queries even for entities whose public keys have been replaced also harms the security of the scheme proposed by Al-Riyami and Paterson published in [3]. We also show how to very easily fix the problem.

3.1 The case of generic constructions

In [1] and [29], generic constructions of certificateless encryption were independently proposed. Their idea is basically to combine strongly secure identity-based and traditional public key encryption schemes in a sequential or parallel fashion. More precisely, let $\Pi^{IBE} = (\mathsf{Setup}^{IBE}, \mathsf{Extract}^{IBE}, \mathcal{E}^{IBE}, \mathcal{D}^{IBE})$ be an IBE scheme (see appendix A for details on the formal syntax of such a primitive) and $\Pi^{PKE} = (\mathcal{K}^{PKE}, \mathcal{E}_{pk}^{PKE}, \mathcal{D}_{\mathsf{sk}}^{PKE})$ denote a traditional public key encryption scheme (the latter being made of a key generation algorithm $\mathcal{K}^{PKE}$, a probabilistic encryption algorithm $\mathcal{E}_{pk}^{PKE}$ and the deterministic decryption algorithm $\mathcal{D}_{\mathsf{sk}}^{PKE}$), a CLE scheme Π^{CLE} can be obtained with the present sequential composition. Its security was proved by Yum and Lee [29] in a model where adversaries are restricted not to issue a partial key exposure query on the target identity ID^* (recall that such a query is allowed in the strong model if entity ID^*'s public key is never replaced) nor to require the correct decryption of ciphertexts encrypted under identities of replaced public keys.

Setup: is an algorithm running the setup algorithm of Π^{IBE}. The message space of Π^{CLE} is the message space of Π^{PKE} while its ciphertext space is the one of Π^{IBE}. Both schemes have to be compatible in that the plaintext space of Π^{IBE} must contain the ciphertext space of Π^{PKE}.

Partial-Private-Key-Extract: is the private key generation algorithm of Π^{IBE}.

Set-Secret-Value and Set-Public-Key: run the key generation procedure of Π^{PKE} to obtain a private key sk and a public key pk. The former is the secret value and the latter becomes the public key.

Set-Private-Key: returns $S_A := (d_A, \mathsf{sk}_A)$ where d_A is obtained by running the key generation algorithm of Π^{IBE} for the identity ID_A and sk_A is entity A's secret value obtained from Π^{PKE}'s key generation algorithm.

Encrypt: to encrypt $m \in \mathcal{M}^{PKE}$ using the identifier $\mathsf{ID}_A \in \{0,1\}^*$ and the public key pk_A,

1. Check that pk_A has the right shape for Π^{PKE}.
2. Compute and output the ciphertext $C = \mathcal{E}^{IBE}_{\mathsf{ID}_A}(\mathcal{E}^{PKE}_{\mathsf{pk}_A}(m))$ where $\mathcal{E}^{IBE}_{\mathsf{ID}_A}$ and $\mathcal{E}^{PKE}_{\mathsf{pk}_A}$ respectively denote the encryption algorithms of Π^{IBE} and Π^{PKE} for the identity ID_A and the public key pk_A.

Decrypt: to decrypt C using $S_A = (d_A, \mathsf{sk}_A)$,

1. Compute $\mathcal{D}^{IBE}_{d_A}(C)$ using the decryption algorithm of Π^{IBE}. If the result is $\perp$, return $\perp$ and reject the ciphertext.
2. Otherwise, compute $\mathcal{D}^{PKE}_{\mathsf{sk}_A}(\mathcal{D}^{IBE}_{d_A}(C))$ using the decryption algorithm of Π^{PKE} and return the result.

This construction is insecure against Type I attacks in the full model of definition 2 even if its building blocks Π^{IBE} and Π^{PKE} are each IND-CCA secure in their model. We show it using simple arguments such as those given in [18, 32] against the security of naive multiple-encryptions. Let $C^* = \mathcal{E}^{IBE}_{\mathsf{ID}^*}(\mathcal{E}^{PKE}_{\mathsf{pk}^*}(m_b^*))$ be the challenge ciphertext in the game of definition 2 where m_b^* (for a random bit $b \in \{0,1\}$) denotes one of the messages produced by the adversary $\mathcal{A}_I$ in her challenge request. Assume that $\mathcal{A}_I$ never replaces the public key of ID^* but rather extracts the partial private key d_{ID^*} after the challenge phase. She then obtains $\mathcal{E}_1 = \mathcal{D}^{IBE}_{d_{\mathsf{ID}^*}}(C^*) = \mathcal{E}^{PKE}_{\mathsf{pk}^*}(m_b)$ and she may compute another encryption $C' = \mathcal{E}^{IBE}_{\mathsf{ID}^*}(\mathcal{E}_1) \neq C^*$ of the same plaintext and obtain m_b^*.

This does not contradict the result of [29] that considers a weaker model where attackers may not extract the partial private key for the target identity.

In [1], a reverse-ordered composition (that we call $\mathsf{Generic\text{-}CLE\text{-}2}$) where ciphertexts have the form $C = \mathcal{E}^{PKE}_{\mathsf{pk}_A}(\mathcal{E}^{IBE}_{\mathsf{ID}}(m))$ is suggested. This composition is vulnerable against an attacker replacing the target entity's public key before the challenge phase. Knowing the secret value sk^* in the challenge phase, the adversary obtains $\mathcal{E}^{IBE}_{\mathsf{ID}^*}(m_b)$ that is re-encrypted into $C' = \mathcal{E}^{PKE}_{\mathsf{pk}^*}(\mathcal{E}^{IBE}_{\mathsf{ID}^*}(m_b)) \neq C^*$ which may be submitted to the decryption oracle even though entity ID^*'s public key was replaced in the model of [2].

In [1], a ‘parallel’ construction (that we will call Generic-CLE-3) was also considered. It encrypts a plaintext m into

$$C = \langle \mathcal{E}^{PKE}_{\mathsf{pk}_A}(m_1), \mathcal{E}^{IBE}_{\mathsf{ID}}(m_2) \rangle$$

where m_1 and m_2 are subject to the constraint $m = m_1 \oplus m_2$. This parallel approach is vulnerable to a similar attack to those outlined by Dodis and Katz [18] or Zhang et al. [32] against multiple-encryption schemes: if $C^* = \langle \mathcal{E}_1^*, \mathcal{E}_2^* \rangle$ is the challenge ciphertext in the IND-CCA game, both kinds of adversaries $\mathcal{A}_I$ or $\mathcal{A}_{II}$ may first request the decryption of $C_1' = \langle \mathcal{E}_1^*, \mathcal{E}^{IBE}_{\mathsf{ID}}(0^{IBE}) \rangle$ and then the decryption of $C_2' = \langle \mathcal{E}^{PKE}_{\mathsf{pk}}(0^{PKE}), \mathcal{E}_2^* \rangle$, where 0^{PKE} and 0^{IBE} are plaintexts made of zeros in Π^{IBE} and Π^{PKE}. By combining the results m_1' and m_2' of both decryption requests into $m_1' \oplus m_2'$, the adversary $\mathcal{A}_I$ gets back the plaintext encrypted in C^*. This attack works even if Π^{IBE} and Π^{PKE} are both IND-CCA secure and it does not even require $\mathcal{A}_I$ to replace any public key. Unlike the previous two attacks, it also works in the weaker models of [16, 29].

In [18], Dodis and Katz gave generic techniques to counteract such attacks and build IND-CCA secure (possibly parallel) multiple-encryption schemes from public key encryption schemes which are individually IND-CCA. They showed that their methods apply to the design of certificate-based encryption schemes [22] without resorting to the random oracle model. Because of the strong constraint imposed on decryption oracles in definition 2, those techniques do not seem to directly apply in the present context (although they do so in the relaxed models considered in [16, 29]). In security proofs, the difficulty is that the simulator does not know the secret value of entities whose public key was replaced.

3.2 The second Al-Riyami-Paterson scheme

In [3], the inventors of the certificateless paradigm proposed a variant (named FullCLE*) of their original scheme that is significantly more efficient. It again uses *bilinear map groups* which are groups $(\mathbb{G}_1, \mathbb{G}_2)$ of prime order q for which there exists a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying the following properties:

1. Bilinearity: $\forall\ P, Q \in \mathbb{G}_1,\ \forall\ a, b \in \mathbb{Z}_p^*$, we have $\hat{e}(P^a, Q^b) = \hat{e}(P, Q)^{ab}$
2. Non-degeneracy: if P generates $\mathbb{G}_1$, then $\hat{e}(P, P)$ generates $\mathbb{G}_2$
3. Computability: $\forall\ P, Q \in \mathbb{G}_1,\ \hat{e}(P, Q)$ can be efficiently computed

In FullCLE*, public keys are made of a single group element $Y_A = x_A P \in \mathbb{G}_1$, for a secret value $x_A \in \mathbb{Z}_q^*$, and checking their validity only requires an elliptic curve scalar multiplication. The plaintext is actually scrambled twice using two distinct superposed one-time masks. In some sense, this scheme may be regarded as an optimized composition of the Boneh-Franklin IBE [12] with an ElGamal-like cryptosystem [21]. In order to achieve the security in the sense of definition 2, the authors of [3] again applied the Fujisaki-Okamoto conversion [20].

In more details, the KGC has a master key $s \in \mathbb{Z}_q^*$ and a master public key $P_{pub} = sP$. It computes partial private keys as $d_A = sh_1(\mathsf{ID}_A)$, where h_1 :

$\{0,1\}^* \rightarrow \mathbb{G}_1^*$ maps public identifiers onto the group $\mathbb{G}_1$, while end-users' private keys consist of a secret value x_A and a partial private key d_A. In accordance with the Fujisaki-Okamoto construction, messages m are encrypted into

$$C = \langle U, V, W \rangle = \langle rP, \sigma \oplus h_2(\hat{e}(P_{pub}, h_1(\mathsf{ID}_A))^r) \oplus h_2'(rY_A), m \oplus h_4(\sigma) \rangle$$

where $r = h_3(\sigma, m)$ for a random string $\sigma \xleftarrow{R} \{0,1\}^{k_1}$ (for some $k_1 \in \mathbb{N}$) and hash functions $h_2 : \mathbb{G}_2 \rightarrow \{0,1\}^{k_1}$, $h_2' : \mathbb{G}_1 \rightarrow \{0,1\}^{k_1}$, $h_3 : \{0,1\}^{n+k_1} \rightarrow \mathbb{Z}_q^*$, $h_4 : \{0,1\}^{k_1} \rightarrow \{0,1\}^n$.

It turns out that the original Fujisaki-Okamoto padding [20] does not suffice to achieve the security level modelled in definition 2. We find that a Type I adversary $\mathcal{A}_I$ can break the non-malleability of $\mathsf{FullCLE}^*$ in the scenario of definition 2 by replacing twice the target identity's public key. If the challenge ciphertext is $C^* = \langle U^*, V^*, W^* \rangle$ and x^* denotes the secret value of the target identity ID^* (which is known to a Type I adversary $\mathcal{A}$ replacing entity ID^*'s public key before the challenge phase), the attacker can replace entity ID^*'s public key with $x'P$ after the challenge phase and then ask for the decryption of $C' = \langle U^*, V^* \oplus h_2'(x^*U^*) \oplus h_2'(x'U^*), W^* \rangle$ (which is an encryption of the same plaintext as C^* for the combination $(\mathsf{ID}^*, x'P)$). Since decryption queries remain allowed even for entities of a replaced public key, $\mathcal{A}_I$ can issue a decryption query on $C' \neq C$ for the identity ID' and recover the plaintext.

Fortunately, such an attack is easily defeated by hashing the recipient's public key along with his identity and the pair (σ, m) when computing r in the encryption algorithm. A variant of $\mathsf{FullCLE}^*$ independently proposed by Cheng and Comley [16] is immune to the latter attack because it scrambles σ with a hash value of both rY_A and $\hat{e}(P_{pub}, Q_{\mathsf{ID}_A})^r$ instead of using separate masks.

These observations shed new lights on the power of attackers replacing entities' public keys instead of merely obtaining their secret value. Indeed, the $\mathsf{FullCLE}^*$ scheme remains secure in a model where attackers cannot replace public keys but are rather provided with an oracle returning secret values of arbitrary identities. The latter model is thus strictly weaker than the one of [2].

4 Secure combinations in the random oracle model

We now explain how to obtain generic constructions that withstand the attacks outlined in section 3.1 and that are provably secure in the random oracle model.

We first show a generic random oracle-based transformation that turns any IND-CPA certificateless encryption scheme into a secure CLE system in the chosen-ciphertext scenario of definition 2. We then show that all the generic compositions recalled in section 3.1 are IND-CPA if they start from chosen-plaintext secure IBE and public key encryption schemes.

4.1 From chosen-plaintext to chosen-ciphertext security

This transformation is a modification of the first Fujisaki-Okamoto conversion [19] which provides IND-CCA secure public key encryption schemes from IND-CPA ones. Our modification is to include the recipient's identity and public key among the inputs of the hash function deriving random coins from the message and a random string in the encryption algorithm.

To handle decryption queries of the chosen-ciphertext attacker, the strategy of the plaintext extractor is essentially the following: for every new random oracle query on a string $(m||\sigma||\mathsf{pk}||\mathsf{ID})$, it returns a random value r and runs the encryption algorithm of the weakly secure CLE scheme with the identity ID and the public key pk (that may have been replaced or not) to encrypt $(m||\sigma)$ using the randomness r. The resulting ciphertext C is stored in a list. By doing so, the simulator anticipates subsequent decryption queries, knowing that any valid ciphertext submitted in a decryption query was previously computed and stored in the list with all but negligible probability. The latter strategy allows us to handle decryption queries even when the relevant public key was replaced. It is a *generic knowledge extractor* (in the random oracle model) while previous works [2–4] that considered the treatment of this kind of decryption requests only used knowledge extractors that were specific to their schemes.

Theorem 1. *Let Π^{CLE} be an IND-CPA certificateless encryption scheme and suppose that*

$$\mathcal{E}_{\mathit{ID,pk}}^{\mathsf{params}}(M,R) \quad \textit{and} \quad \mathcal{D}_{S_{\mathsf{ID}}}^{\mathsf{params}}$$

are its encryption and decryption algorithms where ID and pk respectively denote the recipient's identity and his public key, M is a message of $n+k_0$ bits, R is a random string of ℓ bits while S_{ID} is the recipient's private decryption key. Then, an IND-CCA certificateless scheme $\overline{\Pi}^{CLE}$ can be obtained using modified encryption and decryption algorithms

$$\overline{\mathcal{E}}_{\mathit{ID,pk}}^{\mathsf{params}}(m,\sigma) = \mathcal{E}_{\mathit{ID,pk}}^{\mathsf{params}}(m||\sigma, H(m||\sigma||\mathsf{pk}||\mathsf{ID}))$$

where $H : \{0,1\}^* \to \{0,1\}^\ell$ is a random oracle, $m \in \{0,1\}^n$ is the plaintext and $\sigma \in \{0,1\}^{k_0}$ is a random string. The modified decryption algorithm is

$$\overline{\mathcal{D}}_{S_{\mathsf{ID}}}^{\mathsf{params}}(C) = m \quad \textit{if} \quad C = \mathcal{E}_{\mathit{ID,pk}}^{\mathsf{params}}(m||\sigma, H(m||\sigma||\mathsf{pk}||\mathsf{ID}))$$
$$\textit{and } \perp \textit{ otherwise}$$

where $(m||\sigma) = \mathcal{D}_{S_{\mathsf{ID}}}^{\mathsf{params}}(C)$.

More precisely, assume that a Type I (resp. Type II) IND-CCA attacker $\mathcal{A}$ has advantage ϵ over $\overline{\Pi}^{CLE}$ when running in time τ, making q_D decryption queries and q_H random oracle queries. It implies a Type I (resp. Type II) IND-CPA attacker $\mathcal{B}$ with advantage

$$\epsilon' > (\epsilon - q_H/2^{k_0-1})(1-2^{-\ell_0})^{q_D}$$

over Π^{CLE} when running in time $\tau' < \tau + O(q_H \tau_{\mathcal{E}})$, where $\tau_{\mathcal{E}}$ is the the cost the original encryption algorithm and

$$\ell_0 = \log_2 \big(\min_{\substack{m \in \{0,1\}^{n+k_0} \\ \mathsf{ID},\mathsf{pk}}} [\#\{\mathcal{E}_{\mathit{ID,pk}}^{\mathsf{params}}(m,r) | r \in \{0,1\}^\ell\}] \big)$$

is the logarithm of the cardinality of the smallest set of encrypted values that can be obtained for fixed plaintext, identity and public key.

Proof. The proof is quite similar to the one of theorem 3 in [19] but we have to show that the adapted conversion generically works in our context. We outline how $\mathcal{B}$ uses $\mathcal{A}$ to succeed in a chosen-plaintext attack against her challenger $\mathcal{CH}$. $\mathcal{B}$ starts by forwarding to $\mathcal{A}$ the public parameters (together with the KGC's master key in the scenario of a Type II attack) she obtains from $\mathcal{CH}$. Recall that Π^{CLE} can be itself a random oracle-using scheme. All random oracles pertaining to Π^{CLE} are thus controlled by $\mathcal{CH}$. The chosen-ciphertext attacker $\mathcal{A}$ also has access to a decryption oracle and an additional random oracle H that are simulated by $\mathcal{B}$ as follows:

- random oracle queries related to Π^{CLE} as well as public key broadcast, public key replacement (in the case of Type I attacks) and partial/full private key exposure queries are passed to $\mathcal{CH}$ whose answers are relayed to $\mathcal{A}$.
- Whenever $\mathcal{A}$ submits a string $(m||\sigma||\mathsf{pk}||\mathsf{ID})$ to the H oracle, $\mathcal{B}$ first checks if H was previously queried on the same input and returns the previously answered value if it was. Otherwise, $\mathcal{B}$ returns a randomly chosen $r \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$. She then runs the encryption algorithm of Π^{CLE} to compute

  $$C = \mathcal{E}_{\mathsf{ID},\mathsf{pk}}^{\mathsf{params}}(m||\sigma, r)$$

  which is a $\overline{\Pi}^{CLE}$ encryption of m under the public key pk and the identity ID using the randomness $\sigma \in \{0,1\}^{k_0}$ (as well as a Π^{CLE} encryption of $(m||\sigma)$ for the randomness r). In order to anticipate subsequent decryption queries, a record containing the input $(m||\sigma||\mathsf{pk}||\mathsf{ID})$, the output r and the ciphertext C is stored in a list L_H. Note that $\mathcal{B}$ might need $\mathcal{CH}$ to answer queries for random oracles related to Π^{CLE} to be able to compute C.
- Decryption queries for a ciphertext C and an identity ID: $\mathcal{B}$ first recovers the public key pk currently associated with ID (by issuing a public key query to $\mathcal{CH}$). She then searches in list L_H for a tuple of the form $((m||x||\mathsf{pk}||\mathsf{ID}), r, C)$ in order to return the corresponding m if such a tuple exists and $\perp$ otherwise.

When $\mathcal{A}$ decides that phase 1 is over, she outputs messages (m_0, m_1) and an identity ID^* (whose private key was not exposed and that was not submitted to both the Public-Key-Replace and Partial-Private-Key-Extract oracles). At that point, $\mathcal{B}$ obtains the current value pk^* of entity ID^*'s public key (by issuing a Public-Key-Broadcast query to $\mathcal{CH}$) before randomly choosing two strings $\sigma_0, \sigma_1 \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$ and in turn sending her challenge request $(M_0 = (m_0||\sigma_0), M_1 = (m_1||\sigma_1), \mathsf{ID}^*)$ to $\mathcal{CH}$. The latter then returns a Π^{CLE} encryption C^* of $M_b = (m_b||\sigma_b)$ for the identity ID^* and the current public key pk^* using some randomness $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$.

As in the proof of theorem 2 in [19], if $\mathcal{A}$ ever queries H on the input $(m_d||\sigma_d||\mathsf{pk}^*||\mathsf{ID}^*)$ for $d \in \{0,1\}$, $\mathcal{B}$ halts and outputs the corresponding bit d as a result which is very likely to be correct in this case: since $\mathcal{A}$ has absolutely no information on $\sigma_{\overline{b}}$ ($\overline{b}$ being the complement bit of b), one can show as in [19] that $\mathcal{A}$ only asks for the hash value $H(m_{\overline{b}}||\sigma_{\overline{b}}||\mathsf{pk}^*||\mathsf{ID}^*)$ with probability $q_H/2^{k_0}$ throughout the game). On the other hand, if such an H-query never occurs, $\mathcal{B}$ outputs exactly the same result b' as $\mathcal{A}$ and obviously succeeds against $\mathcal{CH}$ if $\mathcal{A}$ yields a correct guess $b' = b$.

The probability for $\mathcal{B}$ to wrongly reject a ciphertext during the game is smaller than $1-(1-2^{-\ell_0})^{q_D}$. Indeed, for a given decryption query on a ciphertext C and an identity ID, assume that $(m||\sigma) = \mathcal{D}^{\mathsf{params}}_{S_{\mathsf{ID}}}(C)$ and does not figure (together with ID and pk) in list L_H. The probability that $H(m||\sigma||\mathsf{pk}||\mathsf{ID})$ takes a value encrypting $(m||\sigma)$ into C is at most $2^{-\ell_0}$ (as at most $2^{\ell-\ell_0}$ distinct random values $r \in R$ may encrypt a given ciphertext into the same ciphertext by the definition of ℓ_0) .

It comes that $\mathcal{B}$'s advantage against $\mathcal{CH}$ is at least

$$\epsilon' > (\epsilon - q_H/2^{k_0-1})(1-2^{-\ell_0})^{q_D}$$

and that her running time is bounded by $\tau' < \tau + O(q_H \tau_{\mathcal{E}})$ where $\tau_{\mathcal{E}}$ is the time complexity of the encryption algorithm of the basic scheme Π^{CLE}. She also has to issue $q_D + 1$ public key broadcast oracle queries to $\mathcal{CH}$ and q_H queries to random oracles pertaining to Π^{CLE}. □

4.2 Generic IND-CPA secure compositions

From now, we only have to consider constructions that are only secure against chosen-plaintext attacks. By applying to them the random oracle-using conversion, we end up with provably secure constructions in the random oracle model.

Let $\Pi^{IBE} = (\mathsf{Setup}^{IBE}, \mathsf{Extract}^{IBE}, \mathcal{E}^{IBE}, \mathcal{D}^{IBE})$ be an IBE scheme and $\Pi^{PKE} = (\mathcal{K}^{PKE}, \mathcal{E}^{PKE}_{pk}, \mathcal{D}^{PKE}_{\mathsf{sk}})$ be a traditional public key encryption scheme.

Theorem 2. *If Π^{IBE} is IND-ID-CPA and Π^{PKE} is IND-CPA, then the* Generic-CLE-1 *is IND-CPA.*

The proof of the above theorem (detailed in the full paper) separately consider Type I and Type II adversaries.

Lemma 1. *A Type I IND-CPA adversary $\mathcal{A}_I$ having an advantage ϵ over* Generic-CLE-1 *implies either an IND-ID-CPA adversary with advantage $\epsilon/(2q_{\mathsf{ID}})$ over Π^{IBE} or an IND-CPA adversary with advantage $\epsilon/(2q_{\mathsf{ID}})$ over Π^{PKE}, where q_{ID} is the total number of distinct identities involved in $\mathcal{A}_I$'s requests.*

Lemma 2. *A Type II IND-CPA adversary $\mathcal{A}_{II}$ with advantage ϵ over* Generic-CLE-1 *implies an IND-CPA adversary $\mathcal{B}$ with advantage ϵ/q_{ID} over Π^{PKE}, where q_{ID} is the total number of distinct identities involved in $\mathcal{A}_{II}$'s requests.*

The proofs of chosen-plaintext security of Generic-CLE-2 and Generic-CLE-3 are very similar. In lemmas 1 and 2, q_{ID} can be the number of random oracle queries for hash functions mapping identifiers onto cyclic subgroups or finite fields if we assume that any query involving a given identity comes after a hash query on it.

This shows how to obtain a secure generic construction in the random oracle model. In the case of Generic-CLE-1, if the encryption schemes of Π^{PKE} and Π^{IBE} use distinct sets of randomness R_1 and R_2, the enhanced CLE scheme may use a random oracle $H : \{0,1\}^* \to R_1 \times R_2$ so that an encryption of a plaintext m using the random string σ is given by

$$\overline{\mathcal{E}}^{CLE}_{\mathsf{ID},\mathsf{pk}}(m||\sigma) = \mathcal{E}^{IBE}_{\mathsf{ID}}(\mathcal{E}^{PKE}_{\mathsf{pk}}(m||\sigma, r_1), r_2)$$

where $(r_1||r_2) = H(m||\sigma||\mathsf{pk}||\mathsf{ID})$. In the case of Generic-CLE-3, we have

$$\overline{\mathcal{E}}^{CLE}_{\mathsf{ID},\mathsf{pk}}(m||\sigma) = \langle \mathcal{E}^{PKE}_{\mathsf{pk}}(m_1, r_1), \mathcal{E}^{IBE}_{\mathsf{ID}}(m_2, r_2)\rangle$$

with $m_1 \oplus m_2 = m||\sigma$.

5 A new efficient construction

We present here our new efficient certificateless encryption scheme that we call NewFullCLE. Its security relies on the intractability of the following problem that was introduced in [10] by Boneh and Boyen.

Definition 3 ([10]). *The* **p-*Bilinear Diffie-Hellman Inversion problem*** *(p-BDHI) is, given* $\langle P, \alpha P, \alpha^2 P, \ldots, \alpha^p P\rangle \in \mathbb{G}_1^{p+1}$*, to compute* $\hat{e}(P,P)^{1/\alpha} \in \mathbb{G}_2$*.*

5.1 The scheme

Similarly to FullCLE*, NewFullCLE may be viewed as an optimized combination of an IBE with a traditional ElGamal-like [21] cryptosystem.

Setup: given security parameters k,k_0 so that k_0 is polynomial in k, this algorithm chooses a k-bit prime number q, bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2)$ of order q, a generator $P \in \mathbb{G}_1$ and hash functions $h_1 : \{0,1\}^* \to \mathbb{Z}_q^*$, $h_2 : {\mathbb{G}_2}^2 \to \{0,1\}^{n+k_0}$, $h_3 : \{0,1\}^* \to \mathbb{Z}_q^*$. A master key $\mathsf{mk} := s \xleftarrow{R} \mathbb{Z}_q^*$ and a public key $P_{pub} = sP \in \mathbb{G}_1$ are also chosen. The group element $g = \hat{e}(P,P) \in \mathbb{G}_2$ is also included among the public parameters which are

$$\mathsf{params} := \{q, k, k_0, \mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, g, \hat{e}, h_1, h_2, h_3, n, \mathcal{M}, \mathcal{C}\}$$

where $\mathcal{M} := \{0,1\}^n$, $\mathcal{C} := \mathbb{G}_1 \times \{0,1\}^{n+k_0}$ respectively denote cleartext and ciphertext spaces.

Partial-Private-Key-Extract: takes as input entity A's identifier $\mathsf{ID}_A \in \{0,1\}^*$ and extracts A's partial private key $d_A = \frac{1}{s+h_1(\mathsf{ID}_A)}P \in \mathbb{G}_1$.

Set-Secret-Value: given params and A as inputs, this algorithm picks $x_A \xleftarrow{R} \mathbb{Z}_q^*$ which is returned as user A's secret value.

Set-Private-Key: given params, user A's partial private key $d_A \in \mathbb{G}_1$ and his secret value $x_A \in \mathbb{Z}_q^*$, this algorithm returns the pair $S_A = (x_A, d_A) \in \mathbb{Z}_q^* \times \mathbb{G}_1$ as a private key.

Set-Public-Key: takes as input params and entity A's secret value $x_A \in \mathbb{Z}_q^*$ and produces A's public key $\mathsf{pk}_A := y_A = g^{x_A} \in \mathbb{G}_2$.

Encrypt: to encrypt $m \in \{0,1\}^n$ using the identifier $\mathsf{ID}_A \in \{0,1\}^*$ and the public key $\mathsf{pk}_A = y_A = g^{x_A}$, the sender

1. Checks that $y_A^q = 1_{\mathbb{G}_2}$.
2. Picks $\sigma \xleftarrow{R} \{0,1\}^{k_0}$, computes $r = h_3(m||\sigma||\mathsf{pk}_A||\mathsf{ID}_A) \in \mathbb{Z}_q^*$ and the ciphertext is

$$C = \langle c_1, c_2 \rangle = \langle rh_1(\mathsf{ID}_A)P + rP_{pub}, (m||\sigma) \oplus h_2(g^r||y_A^r) \rangle$$

Decrypt: given $C = \langle c_1, c_2 \rangle$, the receiver computes $\omega = \hat{e}(c_1, d_A)$ and then $(m||\sigma) = c_2 \oplus h_2(\omega||\omega^{x_A}) \in \{0,1\}^{n+k_0}$. The message is accepted iff $c_1 = r(h_1(\mathsf{ID}_A)P + P_{pub})$ with $r = h_3(m||\sigma||\mathsf{pk}_A||\mathsf{ID}_A) \in \mathbb{Z}_q^*$.

In this construction, partial private keys are signatures computed using a signature scheme independently considered in [11] and [33]. The NewFullCLE scheme is constructed on the Sakai-Kasahara IBE [26, 14, 15] which bears itself similarities with the second IBE scheme that was proved to be selective-ID secure [13, 10] without random oracles by Boneh and Boyen [10]. As for the Cheng-Chen [14] variant of the Sakai-Kasahara IBE, its security proof holds in the random oracle model [8]. The consistency of the construction is easy to check as we have

$$\hat{e}\big(rh_1(\mathsf{ID}_A)P + rP_{pub}, \frac{1}{s + h_1(\mathsf{ID}_A)}P\big) = \hat{e}(P,P)^r.$$

Including g^r among the inputs of h_2 in step 2 of the encryption algorithm is necessary to achieve a security reduction under the p-BDHI assumption. The string $(m||\sigma)$ could be hidden by a hash value of only y_A^r but the security would have to rely on a newly defined fancy assumption.

Interestingly, hashing g^r along with y_A^r is no longer necessary if the scheme is transformed into a certificate-based encryption scheme [22]. This is due to particularities of the certificate-based security model which is not detailed here.

5.2 Efficiency issues

As for the FullCLE* scheme proposed by Al-Riyami and Paterson [3], the validity of the public key can be checked very efficiently. As in [3], assuming that the bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2)$ are chosen by a higher level authority and commonly used by several distinct KGCs, end-users may generate their public key independently of any authority in the system.

The encryption algorithm only entails two exponentiations in $\mathbb{G}_2$ and a multi-exponentiation in $\mathbb{G}_1$. It has a comparable efficiency to the pairing-free scheme of [4]. The receiver has to compute a pairing, an exponentiation in $\mathbb{G}_2$ beside a multi-exponentiation in $\mathbb{G}_1$. The decryption operation may be optimized by the receiver who can pre-compute and store $h_1(\mathsf{ID}_A)P + P_{pub}$ in such a way that a simple scalar multiplication in $\mathbb{G}_1$ suffices to verify the validity of the ciphertext. Such a pre-computation also enables a speed up the encryption operation for senders who encrypt several messages under the same public key.

From a computational point of view, NewFullCLE has the same efficiency as FullCLE* [3] if pre-computations are used in both schemes (although NewFullCLE might be more efficient on curves of embedding degree 2 as an exponentiation in $\mathbb{G}_T$ is cheaper than a scalar multiplication in $\mathbb{G}_1$ in this case) as the pairing can be computed in advance for each identity in FullCLE*. However, our construction performs better in the absence of pre-computations as its encryption procedure does not compute any pairing. The encryption algorithm is also faster than its counterpart in schemes of [16, 9] for similar parameters and without pre-computations. Moreover, NewFullCLE does not need a special (and much less efficient) hash function mapping strings onto a cyclic group (and it thus benefits from a faster partial private key generation algorithm) while all schemes have comparable decryption complexities.

Regarding key sizes, users' public keys lie in $\mathbb{G}_2$ and thus have longer representations (typically 1024 bits without optimizations) than elements in $\mathbb{G}_1$. However, pairing compression techniques due to Barreto and Scott [7] allow them to be compressed to a third (say 342 bits) of their original length on supersingular curves in characteristic 3 or even to 1/6 of their length using ordinary curves such as those of Barreto and Naehrig [6]. Those compression techniques additionally increase the speed of exponentiations in $\mathbb{G}_2$.

The version of the scheme depicted in section 5.1 uses symmetric pairings (and thus supersingular curves). However, it can be implemented with asymmetric pairings as well. In environments where bandwidth is of primary concern, the size of ciphertexts can be minimized at the expense of a longer system-wide public key (which is less likely to transit across the network). In such a setting, asymmetric pairings $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ and ordinary curves such as MNT curves or BN curves [25, 6] should be used as long as a publicly computable but non-necessarily invertible isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ is available.

Regarding the latter criterion, NewFullCLE seems to be more suitable than previous proposals [2, 3, 16, 9] for an implementation with asymmetric pairings. Indeed, Smart and Vercauteren [28] recently underlined the hardness of finding ordinary pairing-friendly groups[1] $(\mathbb{G}_1, \mathbb{G}_2)$ equipped with a publicly computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ as well as an efficient algorithm to hash onto $\mathbb{G}_2$. Our scheme avoids these problems as it does not require to hash onto $\mathbb{G}_2$ or $\mathbb{G}_1$. Concretely, users' public keys have lie in $\mathbb{G}_T$ while the system-wide public key and entities' partial private keys should respectively be $P_{pub} = sP_2$ and $d_A = 1/(h_1(\mathsf{ID}_A) + s)P_2$ for generators $P_2 \in \mathbb{G}_2$ and $P_1 = \psi(P_2) \in \mathbb{G}_1$. In that bandwidth-optimized version of the scheme, users'public keys can be about 512-bit long on MNT curves [25] or even shorter on BN curves [6]. Ciphertexts are 331 bits longer than plaintexts if $k_0 = 160$.

5.3 Security results

We give a security statement (formally proven in the full version of the paper) under the p-Bilinear Diffie-Hellman Inversion assumption.

[1] More precisely, we mean groups allowing the use of the most efficient implementation techniques for ordinary curves [5].

Theorem 3. *In the random oracle model, the* NewFullCLE *scheme is secure in the sense of definition 2 under the p-BDHI assumption.*

6 Conclusion

This paper investigated the problem of generically constructing a certificateless cryptosystem which is secure in the strongest model by combining secure IBE schemes with a traditional public key cryptosystem.

It pinpointed security problems in three simple generic constructions and fixed them using a generic random oracle-using conversion (which extends the Fujisaki-Okamoto transformation) ensuring the security in the strongest sense given any scheme only withstanding chosen-plaintext attacks. We finally described a new scheme offering computational advantages over previous pairing-based constructions.

The feasibility of a CLE scheme provably fitting the model of [2] without random oracles still remains a challenging open problem.

References

1. S. S. Al-Riyami. Cryptographic schemes based on elliptic curve pairings. PhD thesis, University of London, 2004.
2. S. S. Al-Riyami and K. Paterson. Certificateless public key cryptography. In *Asiacrypt'03*, volume 2894 of *LNCS*, pages 452–473. Springer, 2003.
3. S. S. Al-Riyami and K. Paterson. CBE from CL-PKE: A generic construction and efficient schemes. In *PKC'05*, volume 3386 of *LNCS*, pages 398–415. Springer, 2005.
4. J. Baek, R. Safavi-Naini, and W. Susilo. Certificateless public key encryption without pairing. In *ISC'05*, volume 3650 of *LNCS*, pages 134–148. Springer, 2005.
5. P. S. L. M. Barreto, B. Lynn, and M. Scott. On the selection of pairing-friendly groups. In *SAC'03*, volume 3006 of *LNCS*, pages 17–25. Springer, 2003.
6. P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *SAC'05*. To Appear.
7. P. S. L. M. Barreto and M. Scott. Compressed pairings. In *Crypto'04*, volume 3152 of *LNCS*, pages 140–156. Springer, 2004.
8. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73, ACM Press, 1993.
9. K. Bentahar, P. Farshim, J. Malone-Lee, and N. P. Smart. Generic construction of identity-based and certificateless KEMs. Cryptology ePrint Archive, Report 2005/058, 2005. `http://eprint.iacr.org/2005/058`.
10. D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *Eurocrypt'04*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
11. D. Boneh and X. Boyen. Short signatures without random oracles. In *Eurocrypt'04*, volume 3027 of *LNCS*, pages 56–73. Springer, 2004.
12. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Crypto'01*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

13. R. Canetti, S. Halevi, and J. Katz. A forward secure public key encryption scheme. In *Eurocrypt'03*, volume 2656 of *LNCS*, pages 254–271. Springer, 2003.
14. L. Chen and Z. Cheng. Security proof of Sakai-Kasahara's identity-based encryption scheme. In *IMA Int. Conf. 2005*, volume 3796 of *LNCS*, pages 442–459. Springer, 2005. Also available from `http://eprint.iacr.org/2005/226`.
15. L. Chen, Z. Cheng, J. Malone-Lee, and N. P. Smart. An efficient ID-KEM based on the Sakai–Kasahara key construction. Cryptology ePrint Archive, Report 2005/224, 2005. `http://eprint.iacr.org/2005/224`.
16. Z. Cheng and R. Comley. Efficient certificateless public key encryption. Cryptology ePrint Archive, Report 2005/012, 2005. `http://eprint.iacr.org/2005/012`.
17. A. Dent and C. Kudla. On Proofs of Security for Certificateless Cryptosystems. Cryptology ePrint Archive, Report 2005/348, 2005. `http://eprint.iacr.org/2005/348`.
18. Y. Dodis and J. Katz. Chosen-ciphertext security of multiple encryption. In *TCC'05*, volume 3378 of *LNCS*, pages 188–209. Springer, 2005.
19. E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. In *PKC'99*, volume 1560 of *LNCS*, pages 53–68. Springer, 1999.
20. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Crypto'99*, volume 1666 of *LNCS*, pages 537–554. Springer, 1999.
21. T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Crypto'84*, volume 196 of *LNCS*, pages 10–18. Springer, 1985.
22. C. Gentry. Certificate-based encryption and the certificate revocation problem. In *Eurorypt'03*, volume 2656 of *LNCS*, pages 272–293. Springer, 2003.
23. M. Girault. Self-certified public keys. In *Eurocrypt'91*, volume 547 of *LNCS*, pages 490–497. Springer, 1991.
24. G. Kang and S. H. H. J. H. Park. A certificate-based signature scheme. In *CT-RSA'04*, volume 2964 of *LNCS*, pages 99–111. Springer, 2004.
25. A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals*, E84-A(5):1234–1243, 2001.
26. R. Sakai and M. Kasahara. ID-based cryptosystems with pairing on elliptic curve. In *SCIS'03*, Hamamatsu, Japan, 2003. `http://eprint.iacr.org/2003/054`.
27. A. Shamir. Identity based cryptosystems and signature schemes. In *Crypto'84*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.
28. N. P. Smart and F. Vercauteren. On computable isomorphisms in efficient pairing based systems. Cryptology ePrint Archive, Report 2005/116, 2005. `http://eprint.iacr.org/2005/116`.
29. D. H. Yum and P. J. Lee. Generic construction of certificateless encryption. In *ICCSA'04*, volume 3043 of *LNCS*, pages 802–811. Springer, 2004.
30. D. H. Yum and P. J. Lee. Generic construction of certificateless signature. In *ACISP'04*, volume 3108 of *LNCS*, pages 200–211. Springer, 2004.
31. D. H. Yum and P. J. Lee. Identity-based cryptography in public key management. In *EuroPKI'04*, volume 3093 of *LNCS*, pages 71–84. Springer, 2004.
32. R. Zhang, G. Hanaoka, J. Shikata and H. Imai. On the Security of Multiple Encryption or CCA-security+CCA-security=CCA-security? In *PKC'04*, volume 2947 of *LNCS*, pages 360–374. Springer, 2004.
33. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In *PKC'04*, volume 2947 of *LNCS*, pages 277–290. Springer, 2004.

Appendix: formal model of identity based encryption

We recall here the formalism introduced in [12] for identity based encryption. Such a primitive is described by the following definition.

Definition 4. *An identity based encryption (IBE) scheme consists of a 4-uple of algorithms* (Setup^{IBE}, $\mathit{Extract}^{IBE}$, $\mathcal{E}^{IBE}$, $\mathcal{D}^{IBE}$) *with the following specifications.*

Setup^{IBE}**:** *is a probabilistic algorithm run by a private key generator (PKG) that takes as input a security parameter to output a set of public parameters* params *including the master public key P_{pub} of the PKG. The algorithm also outputs the PKG's master key* mk *that is kept secret.*

$\mathsf{Extract}^{IBE}$**:** *is a key generation algorithm run by the PKG on input of a master key* mk *and a user's identity* ID *to return the user's private key d_{ID}.*

$\mathcal{E}^{IBE}$**:** *this probabilistic algorithm takes as input a plaintext M, a recipient's identity* ID *and the set of public parameters* params *to output a ciphertext C.*

$\mathcal{D}^{IBE}$**:** *is a deterministic decryption algorithm taking as input a ciphertext C, the system-wide parameters* params *and the private decryption key d_{ID} to return a plaintext M or a distinguished symbol $\perp$ if C is not a valid ciphertext.*

For consistency purposes, it is required that $M = \mathcal{D}^{IBE}(C, d_{\mathsf{ID}}, \mathsf{params})$ if $C = \mathcal{E}^{IBE}(M, \mathsf{ID}, \mathsf{params})$ for all messages M whenever $d_{\mathsf{ID}} = \mathsf{Extract}^{IBE}(\mathsf{mk}, \mathsf{ID})$.

The models of chosen-plaintext and chosen-ciphertext security were extended to the IBE setting by Boneh and Franklin themselves [12]. Their model considers a "find-then-guess" game between a challenger and an adversary who may adaptively choose the identity on which she will be challenged after having seen private keys for several arbitrary identities.

Definition 5. *An IBE scheme is **IND-ID-CCA secure** if no PPT adversary has a non-negligible advantage in the following game.*

1. *The challenger runs the* Setup *algorithm on input of a security parameter k and sends the domain-wide parameters* params *to the adversary $\mathcal{A}$.*
2. *In a find stage, $\mathcal{A}$ starts probing the following oracles:*
   - *Key extraction oracle: given an identity ID, it returns the extracted private key associated with it.*
   - *Decryption oracle: given an identity $\mathit{ID} \in \{0,1\}^*$ and a ciphertext C, it generates the private key d_{ID} for ID and returns either a plaintext M or a distinguished symbol $\perp$ indicating that the ciphertext was ill-formed.*

   $\mathcal{A}$ can present her queries adaptively. At some point, she produces two plaintexts $M_0, M_1 \in \mathcal{M}$ and an identity ID^* for which she has not requested the private key in stage 2. The challenger computes $C = \mathcal{E}^{IBE}(M_b, \mathit{ID}^*, \mathsf{params})$, for a random hidden bit $b \stackrel{R}{\leftarrow} \{0,1\}$, which is sent to $\mathcal{A}$.
3. *In the guess stage, $\mathcal{A}$ asks new queries but is restricted not to issue a key extraction request on the identity ID^* nor to submit C to the decryption oracle for the identity ID^*. Eventually, $\mathcal{A}$ outputs a bit b' and wins if $b' = b$.*

$\mathcal{A}$'s advantage is defined as $Adv(\mathcal{A}) := 2 \times Pr[b' = b] - 1$.